

Міністерство світи і науки України
Дніпровський національний університет імені Олеся Гончара
фізико-технічний факультет
кафедра радіоелектронної автоматики

В.Б.Мазуренко

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ПРИЙНЯТТЯ РІШЕНЬ
Конспект лекцій

Дніпро

2020

Наведено конспект лекцій з курсу «Сучасні інформаційні технології прийняття рішень», який розроблено у відповідності до освітньо-професійних програм першого рівня вищої освіти «Автоматизація та комп'ютерно-інтегровані технології» та «Кібербезпека». Для студентів фізико-технічного факультету ДНУ, що навчаються за спеціальностями 151 «Автоматизація та комп'ютерно-інтегровані технології» та 125 «Кібербезпека» на першому рівні вищої освіти.

Укладач: доцент кафедри радіоелектронної автоматики фізико-технічного факультету Дніпровського національного університету ім. Олеся Гончара Мазуренко Валерій Борисович.

Лекція № 1

Тема: Введення в інформаційні технології**Зміст**

Визначення технології.....	2
Інформаційні технології	3
Інструментарій та основні напрямки розвитку інформаційних технологій	4
Інформаційна технологія та інформаційна система	4
Узагальнена схема технологічного процесу обробки інформації	5
Комп'ютерно-інтегровані технології.....	6
Контрольні питання за темою.....	10
Рівень модуля	10
Рівень курсу	10

Визначення технології

Сучасне суспільство – це суспільство технологій. Всі предмети, вироби, що нас оточують, засоби і системи комунікації, сервіси та послуги, якими користується сучасна людина створені шляхом застосування тих чи інших технологій.

Технологія в загальному випадку – це сукупність методів обробки, виготовлення, зміни стану, властивостей, форми сировини, матеріалу або напівфабрикату в процесі виробництва.

Всі відомі технології можна класифікувати за сферою їх застосування, наприклад:

- Сільськогосподарські технології
- Виробничі технології
- Технології наукових досліджень
- Військові технології
- Транспортні технології
- Інформаційні технології
- Телекомунікаційні технології
- Соціальні технології
- Політтехнології
- Технології в сфері медицини: лікування, профілактика, оздоровлення та ін.

Наведені приклади – це узагальнення, яке вказує на сукупність, цілий набір окремих технологій. Так, наприклад, промислові технології включають в себе наступні розділи, які в свою чергу також є узагальненням:

- Технологія металів
- Хімічні технології
- Машинобудівні технології
- Технологія будівництва та ін.

Серед промислових технологій можна виділити групу під загальною назвою "Технології неруйнівного контролю", в яку входять технології радіохвильового, акустичного, оптичного та інших видів НК.

Якщо продовжувати заглиблюватися, то прийдемо зрештою до конкретної технології, яка реалізується під час виробництва конкретного виду продукції, наприклад: "Технологія контролю зварних з'єднань трубопроводів радіографічним методом".

Загальна кількість застосовуваних у всіх сферах людської діяльності конкретних технологій, мабуть, невідома.

Часто технології розрізняють за фізичними (або іншим) явищами, на основі яких вони функціонують, наприклад:

- Технології, пов'язані з електрикою

- Акустичні технології

Існує ще одна класифікація технологій: за ступенем залучення в технологічний процес праці людини. Роль людини при виробництві продукції різна. Найбільш простою класифікацією за ступенем залучення праці людини можна вважати наступне поділ технологій:

- ручні (використовується тільки праця людини);
- механізовані (механізація важкої фізичної праці)
- автоматизовані (часткова участь людини, найчастіше для маніпулювання технікою);
- автоматичні (без будь-якої участі людини під час виробництва).

Засоби масової інформації виділяють таке поняття, як "високі технології" – це технології, які визначають напрямок науково-технічного прогресу сучасного суспільства, які підвищують, підіймають вгору технічний рівень цивілізації, та й самі ці технології перебувають на вершині найпередовіших наукових знань. Високими ці технології називають ще й тому, що не тільки створення, але і застосування цих технологій вимагає залучення фахівців високої кваліфікації, вимагає наявності високоточного, спеціалізованого обладнання, а також вимагає високих фінансових витрат. В силу зазначених факторів високими технологіями володіють лише високорозвинені держави, з високим економічним і науковим потенціалом. До високих технологій, зокрема, відносять:

- Космічні технології
- Нанотехнології
- Біотехнології та ін.

Інформаційні технології

Серед усіх видів технологій окремо виділяються інформаційні технології (ІТ). Що це таке? Нагадаємо, що технологія – це сукупність методів обробки, виготовлення, зміни стану, властивостей, форми сировини, матеріалу або напівфабрикату в процесі виробництва. Конкретна технологія передбачає наявність трьох основних складових елементів: матеріал, засоби обробки матеріалу, методи обробки матеріалу. Коли сировиною і результатом обробки є *інформація*, а засобами обробки є комп'ютери, то сукупність методів (способів) для збору, накопичення, зберігання, пошуку, обробки і видачі інформації споживачеві – це вже і є *інформаційна технологія*.

Інформаційна технологія (ІТ) – сукупність засобів і методів збору, обробки і передачі даних (первинної інформації) для отримання інформації нової якості про стан об'єкта, процесу або явища (інформаційного продукту).

Мета інформаційної технології – представлення інформації для її аналізу людиною і прийняття на її основі рішення з виконання певної дії.

Практичне застосування методів і засобів обробки даних може бути різним, тому доцільно виділити глобальну базові і конкретні інформаційні технології

Глобальна інформаційна технологія включає моделі, методи і засоби, що формують і дозволяють використовувати інформаційні ресурси суспільства.

Базова інформаційна технологія призначена для певної області застосування (виробництво, наукові дослідження, навчання і т.д.).

Конкретні інформаційні технології реалізують обробку даних при вирішенні функціональних завдань користувачів (наприклад, задачі обліку, планування, аналізу).

Інструментарій та основні напрямки розвитку інформаційних технологій

Технічними засобами інформаційних технологій, або інакше кажучи *інструментарієм інформаційних технологій*, є апаратне, програмне і математичне забезпечення.

Інформаційні технології в даний час розвиваються за такими основними напрямками:

- активізація ролі фахівців з управління (непрофесіоналів у галузі обчислювальної техніки) в підготовці і розв'язанні задач управління;
- вдосконалення систем інтелектуального інтерфейсу кінцевих користувачів різних рівнів;
- об'єднання інформаційно-обчислювальних ресурсів за допомогою обчислювальних мереж різних рівнів (від локальних обчислюваних мереж, які об'єднують користувачів в рамках одного підрозділу організації до глобальних);
- розробка комплексних заходів забезпечення захисту інформації (технічних, організаційних, програмних, правових і т.п.) від несанкціонованого доступу.

Інформаційна технологія та інформаційна система

Інформаційна технологія тісно пов'язана з інформаційними системами, які є для неї основним середовищем. На перший погляд може здатися, що визначення інформаційної технології і системи дуже схожі між собою.

Інформаційна технологія є процесом, що складається з чітко регламентованих правил виконання операцій, дій, етапів різного ступеня складності над даними, що зберігаються в комп'ютерах. Основна мета

інформаційної технології: в результаті цілеспрямованих дій з переробки первинної інформації необхідно отримати потрібну для користувача інформацію.

Інформаційна система являє собою людино-комп'ютерну систему обробки інформації. Інформаційна система є середовищем, складовими елементами якої є комп'ютери, комп'ютерні мережі, програмні продукти, бази даних, люди, різного роду технічні і програмні засоби, зв'язку і т.д. Основна мета інформаційної системи – організація зберігання та передачі інформації.

Реалізація функцій інформаційної системи неможлива без знання орієнтованої на неї інформаційної технології. Інформаційна технологія може існувати і поза сферою інформаційної системи.

Приклад. Інформаційна технологія роботи в середовищі текстового процесора Microsoft Word, який не є інформаційною системою.

Таким чином, інформаційна технологія є більш ємним поняттям, яке відбиває сучасне уявлення про процеси перетворення інформації в інформаційному суспільстві.

Узагальнена схема технологічного процесу обробки інформації

Під час виробництва інформаційного продукту відповідно до поставлених завдань вхідний інформаційний ресурс піддається певній послідовності різних перетворень. Динаміка цих перетворень відображається в інформаційних процесах. Тобто, інформаційний процес – це процес перетворення інформації. В результаті інформація може змінити і зміст, і форму подання.

Керуючі впливи формуються на основі накопиченої і функціонуючої в системі управління інформації, а також надходять по каналах прямого і зворотного зв'язку відомостей із зовнішнього середовища.

Фази перетворення інформації в інформаційній технології досить численні. Однак якщо провести структурування технології, узагальнена схема технологічного процесу обробки інформації може бути представлена схемою, показаної на рис. 1.

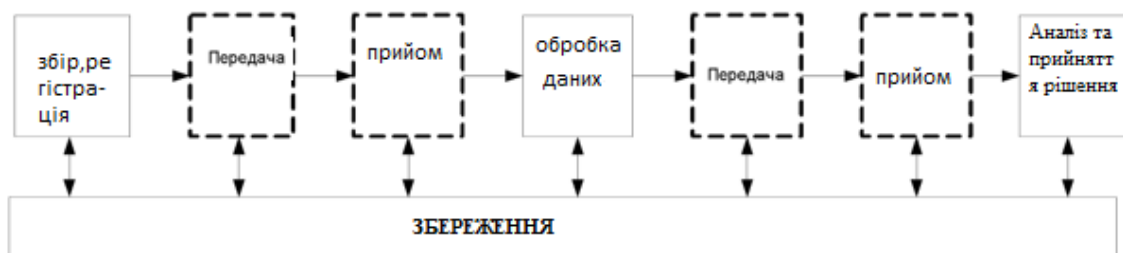


Рис. 1 – Узагальнена схема технологічного процесу обробки інформації

Комп'ютерно-інтегровані технології

Застосовувані технології постійно вдосконалюються в напрямку зменшення частки ручної праці, підвищення продуктивності та якості.

Поява комп'ютерів дозволила не тільки механізувати технологічні процеси, а й перейти до автоматизованих і навіть автоматичних технологій управління механізмами, які бере на себе комп'ютер. Комп'ютер, а точніше комп'ютерна техніка, стає складовою частиною технологічного процесу, інакше кажучи, інтегрується в нього. Багато технологій вже в принципі не реалізуються без комп'ютерних систем. До таких технологій належить більшість телекомунікаційних технологій, технології роботи з відео і фото зображеннями, технології управління космічною і ракетною технікою, енергетичними установками, системами розподілу електроенергії і, звичайно, інформаційні технології. Однак, багато технологій, як і раніше, не вимагають застосування комп'ютерної техніки, а деякі ще й досі залишаються ручними, наприклад, технології художніх промислів.

Таким чином, під комп'ютерно-інтегрованою технологією можна було б розуміти будь-яку технологію, яка реалізується із застосуванням комп'ютерної техніки. Однак, якщо користуватися цим визначенням, виникає суперечність. Виходить, що якщо виконується прасування тканини звичайною електричною праскою, то це – технологія прасування тканини. Якщо в прасці використовується мікропроцесор для управління нагрівом, то це – комп'ютерно-інтегрована технологія прасування?

Для того, щоб розібратися з цим питанням, слід в першу чергу з'ясувати, для вирішення яких завдань застосовується комп'ютерна техніка в технологічному процесі? Насправді, цих завдань зовсім небагато. Це:

- 1) отримання даних про об'єкт, тобто вимірювання;
- 2) вироблення керуючого впливу.

В автоматичних і автоматизованих технологіях – це все, інших завдань немає. Таким чином, комп'ютер має справу не з технологією виготовлення, лікування або прасування, а тільки з технологіями вимірювання і керування, без яких дана конкретна технологія виготовлення, лікування або прасування не може бути реалізована.

Узагальнюючи вищесказане, можна дати наступне визначення комп'ютерно-інтегрованих технологій.

Комп'ютерно-інтегрованими технологіями (KIT) називаються технології вимірювання та керування технічними об'єктами, які реалізуються із застосуванням комп'ютерної техніки.

Примітка: Комп'ютерно-інтегровані технології для вирішення великої частки своїх завдань використовують інформаційні технології, проте ІТ не є частиною КІТ, також як і КІТ не є частиною ІТ.

Повертаючись до наведеного вище прикладу можна говорити не про комп'ютерно-інтегрованої технології прасування, а про комп'ютерно-інтегрованої технології управління нагріванням праски. Термінологічно і лексично стосовно будь-якої технології, устаткування, агрегату правильно говорити про КІТ тільки стосовно управління, тобто: комп'ютерно-інтегрована технологія управління автомобілем, КІТ управління ткацьким верстатом і таке інше. При цьому мається на увазі, що дана КІТ включає і вимірювання і контроль, оскільки управління без отримання даних про об'єкт, тобто без вимірювання, та без виконання контролю, неможливе.

Комп'ютерно-інтегровані технології реалізуються у вигляді керуючих і вимірювальних систем. Можна сказати, що системи, в яких реалізуються комп'ютерно-інтегровані технології, являють собою наступне покоління вимірювальних систем, а також автоматизованих систем управління технологічними процесами (АСУ ТП). Тобто – це ті ж самі системи, але побудовані на основі комп'ютерної техніки.

Класична архітектура керуючої системи втілена в моносистемі (рис.2) – зосередженої системі, яка забезпечує управління зосередженим об'єктом управління. Зосередженими в даному випадку можна називати системи, в яких відстань від датчиків і приводів до комп'ютера обчислюється метрами або десятками метрів.

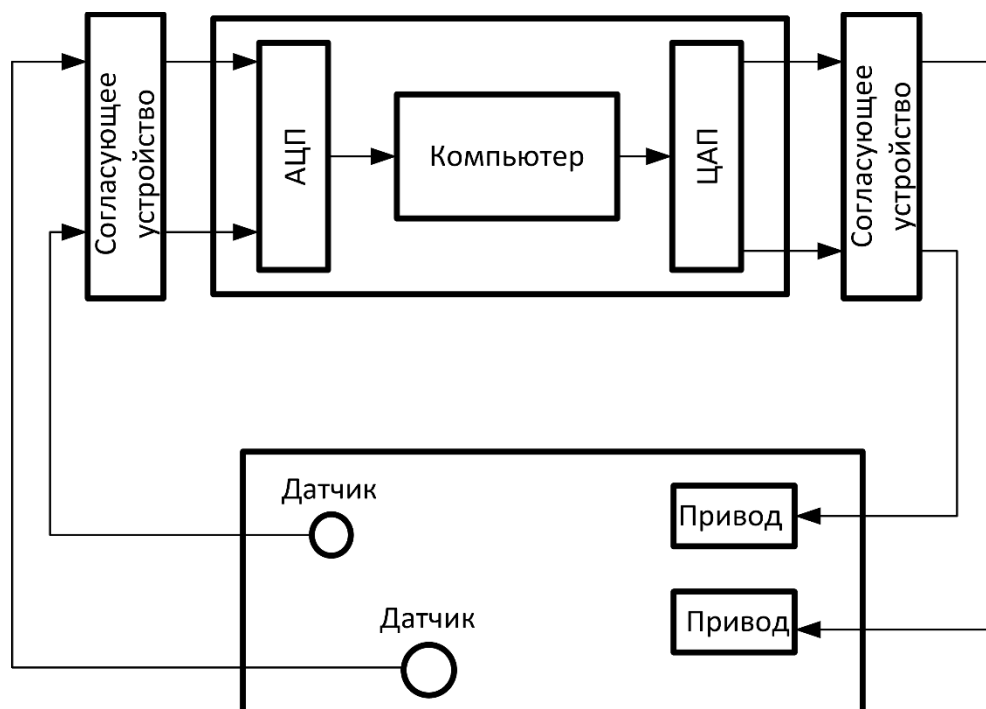


Рис. 2 Зосереджена система

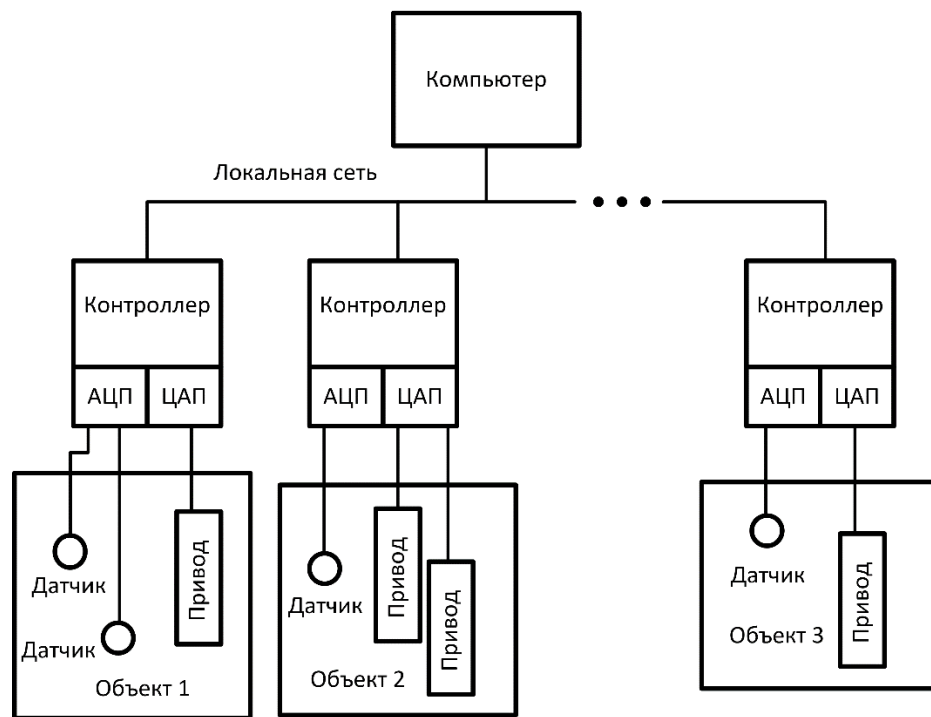


Рис.3 Розподілена система

Згодом почали розвиватися розподілені системи (рис. 3), які забезпечують вимірювання і управління на сукупності об'єктів. В цьому випадку відстані зазвичай складають сотні метрів і кілометри, а в деяких випадках – десятки кілометрів.

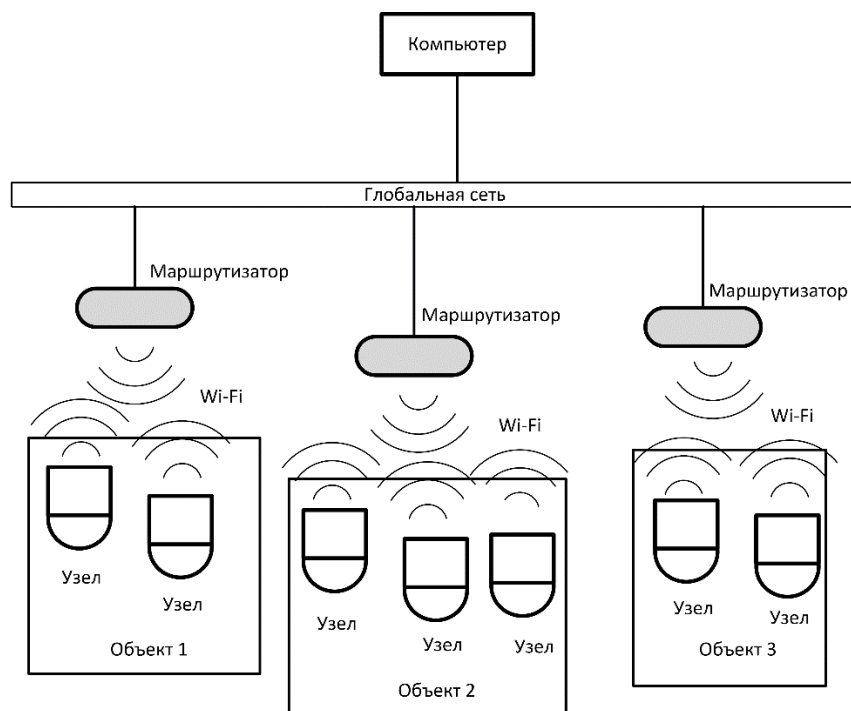


Рис. 4 Глобально розподілена система

Клас розв'язуваних за допомогою комп'ютерно-інтегрованих технологій задач постійно розширюється. Останнім часом все більш широке застосування

знаходять глобально розподілені системи – системи в яких об'єкти контролю знаходяться на значних відстанях – відстанях, які вже не обмежені розмірами міст, країн, або, навіть, континентів. Такі системи будуються за схемою, представленою на наступному рисунку і в основному, є вимірювальними.

Контрольні питання за темою

Рівень модуля

1. Дайте визначення поняття "технологія".
2. Наведіть приклад класифікації технологій за сферою їх застосування.
3. Наведіть приклади технологій, які розрізняються за видом фізичних явищ, що покладені в основу їх функціонування?
4. Яким чином класифікуються технології за ступенем залучення людської праці?
5. Які технології називають високими?
6. Чому, на вашу думку, застосовується термін високі технології?
7. Які технології можна віднести до високих технологій?
8. Які технології називають інформаційними?
9. Дайте визначення поняттю «Інформаційні технології».
10. В чому різниця між глобальними, базовими та конкретними інформаційними технологіями?
11. Що називають «Інструментарієм інформаційних технологій»?
12. Які основні напрямки розвитку інформаційних технологій в наш час?
13. Що таке «інформаційна система»?
14. Чим відрізняються інформаційна система і інформаційна технологія?
15. Як виглядає узагальнена схема технологічного процесу обробки інформації?
16. Для вирішення яких завдань застосовується комп'ютерна техніка в технологічному процесі?
17. Дайте визначення поняттю "комп'ютерно-інтегровані технології".
18. Намалюйте функціональну схему зосередженої керуючої системи.
19. Намалюйте функціональну схему розподіленої керуючої системи.
20. Намалюйте функціональну схему глобально розподіленої вимірювальної системи.

Рівень курсу

1. Визначення поняття «Інформаційна технологія».
2. Інструментарій та основні напрямки розвитку інформаційних технологій
3. Інформаційні системи та їх співвідношення з інформаційними технологіями.
4. Узагальнена схема технологічного процесу обробки інформації
5. Визначення поняття «Комп'ютерно-інтегрована технологія».
6. Архітектура зосередженої комп'ютерно-інтегрованої системи.
7. Архітектура розподіленої комп'ютерно-інтегрованої системи.

8. Архітектура глобально розподіленої комп'ютерно-інтегрованої системи.

Лекція № 2

Тема: Загальна побудова комп'ютера**Зміст**

Машина фон Неймана	2
Двійкова система числення.....	3
Організація комп'ютерних систем.....	4
Процесор	5
Устрій центрального процесора	5
Контрольні питання по темі.....	8
Рівень модуля	8
Рівень курсу	8

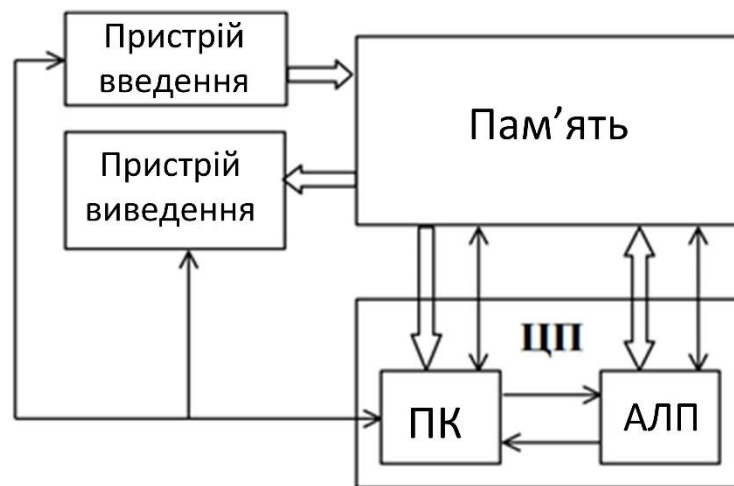
Джерела:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

<https://rutracker.org/forum/viewtopic.php?t=4956359>

Машина фон Неймана

У 1946 році Д. фон Нейман, Г. Голдстайн і А. Беркс в своїй спільній статті виклали принципи побудови і функціонування електронно-обчислювальної машини (ЕОМ). Запропонована побудова комп'ютера отримала назву "машина фон Неймана". Принципи Неймана реалізовані в усіх використовуваних в даний час ЕОМ. Сучасні комп'ютери, за своєю суттю, – це машини фон Неймана.



Мал. 2.1 Машина фон Неймана

Принципи фон Неймана:

1. Використання двійкової системи числення в обчислювальних машинах. Перевага перед десятковою системою числення полягає в тому, що пристрої можна робити досить простими, арифметичні і логічні операції в двійковій системі числення також виконуються досить просто.

2. Програмне управління ЕОМ. Робота ЕОМ контролюється програмою, що складається з набору команд. Команди виконуються послідовно одна за одною. Зі створенням машини з програмою, що зберігається в її пам'яті, було покладено початок тому, що ми сьогодні називаємо програмуванням.

3. Пам'ять комп'ютера використовується не тільки для зберігання даних, але і програм. Команди програми і дані кодуються в двійковій системі числення, тобто їх спосіб запису однаковий. Тому в певних ситуаціях над командами можна виконувати ті ж дії, що і над даними.

4. Комірки пам'яті ЕОМ мають адреси, які послідовно пронумеровані. У будь-який момент можна звернутися до будь-якої комірки пам'яті за її адресом. Цей принцип відкрив можливість використовувати змінні в програмуванні.

5. Можливість умовного переходу в процесі виконання програми.

Незважаючи на те, що команди виконуються послідовно, в програмах можна реалізувати можливість переходу до будь-якої ділянки коду.

Двійкова система числення

Двійкова система числення – позиційна система числення з основою 2. Завдяки безпосередньої реалізації в цифрових електронних схемах на логічних вентилях, двійкова система використовується практично у всіх сучасних комп'ютерах і інших обчислювальних електронних пристроях.

У ЕОМ застосовується двійкова система числення, тобто всі числа в комп'ютері представлені за допомогою нулів і одиниць. Комп'ютер може обробляти інформацію, представлену тільки в цифровій формі. Для перетворення числової, текстової, графічної, звукової інформації в цифрову необхідно застосувати кодування. **Кодування** – це перетворення даних одного типу в дані іншого типу. У ЕОМ застосовується система двійкового кодування, заснована на представленні даних послідовністю двох знаків: 1 і 0, які називаються двійковими цифрами (binary digit - скорочено bit). Двійкова система числення – позиційна система числення з основою 2. Завдяки простій реалізації в цифрових електронних схемах на логічних вентилях, двійкова система використовується практично у всіх сучасних комп'ютерах і інших обчислювальних електронних пристроях.

Таким чином, одиницею інформації в комп'ютері є один біт, тобто двійковий розряд, який може приймати значення 0 або 1. Вісім послідовних біт складають байт. В одному байті можна закодувати значення одного символу з 256 можливих ($256 = 2$ в ступні 8). Більшою одиницею інформації є кілобайт (Кбайт), що дорівнює 1024 байтам ($1024 = 2$ в ступні 10). Ще більші одиниці вимірювання даних: мегабайт, гігабайт, терабайт (1 Мбайт = 1024 Кбайт; 1 Гбайт = 1024 Мбайт; 1 Тбайт = 1024 Гбайт).

Цілі числа кодуються двійковим кодом досить просто (шляхом ділення числа на два). Негативні цілі числа кодуються в так званому додатковому коді. Спеціальний код застосовується для кодування дійсних чисел, дробових чисел, які представляються в комп'ютері у вигляді "чисел з плаваючою точкою". Однак і ці числа записуються у вигляді сукупності нулів і одиниць – в двійковому коді. Для кодування нечислової інформації використовується наступний алгоритм: всі можливі значення кодуємої інформації нумеруються і ці номери кодуються за допомогою двійкового коду. Наприклад, для представлення текстової інформації використовується таблиця нумерації символів або таблиця кодування символів, в якій кожному символу відповідає ціле число (порядковий номер). Вісім двійкових розрядів можуть закодувати 256 різних символів.

У будь-якому випадку, в пам'яті комп'ютера, в процесорі, в шинах є тільки сигнали або стани 0 та 1, інших немає. Всі дані представлені в двійковому коді. Всі команди на управління комп'ютером також складаються з нулів і одиниць, команди – це теж послідовність двійкових чисел. Неможливо зрозуміти, що знаходиться в даній комірці пам'яті. Наприклад, записано 10011101. Що це: просте число, частина дійсного числа або команда на управління процесором? Все залежить від того, як комп'ютер інтерпретує ці дані: як просте число, як частина дійсного числа, або як команду. Для того, щоб правильно інтерпретувати, необхідно заздалегідь знати, який тип інформації знаходиться в даній комірці пам'яті.

Організація комп'ютерних систем

Цифровий комп'ютер складається з пов'язаних між собою процесорів, модулів пам'яті і пристроїв введення-виведення.

На рис. 2.2 показана структура звичайного комп'ютера з шинною організацією. Центральний процесор – це мозок комп'ютера. Його завдання – виконувати програми, що знаходяться в основній пам'яті. Для цього він викликає команди з пам'яті, визначає їх тип, а потім виконує їх одну за одною. Компоненти з'єднані шиною, що представляє собою набір паралельно пов'язаних проводів для передачі адрес, даних і керуючих сигналів. Шини можуть бути зовнішніми (зв'язують процесор з пам'яттю і пристроями введення-виведення) і внутрішніми. Сучасний комп'ютер використовує кілька шин.

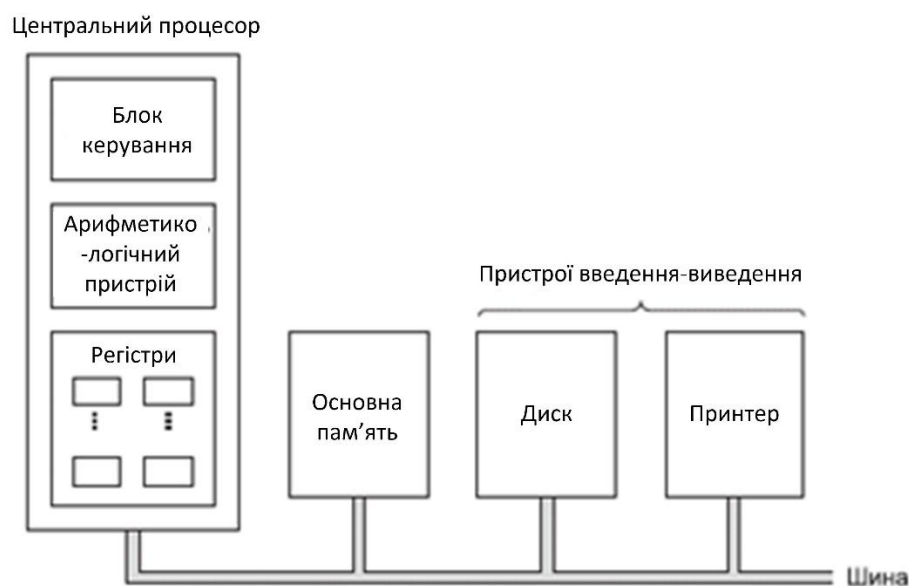


Рис. 2.2 Структура комп'ютера з шинної організацією

Процесор

Процесор складається з декількох частин. Блок управління відповідає за виклик команд з пам'яті і визначення їх типу. Арифметико-логічний пристрій виконує арифметичні операції (наприклад, складання) і логічні операції (наприклад, логічне «І»).

У середині центрального процесора знаходиться швидка пам'ять невеликого обсягу для зберігання проміжних результатів і деяких команд управління. Ця пам'ять складається з декількох регістрів даних, кожен з яких виконує певну функцію. Зазвичай розмір всіх регістрів однаковий. Кожен регістр містить одне число в діапазоні, верхня межа якого залежить від розміру регістра. Операції читання і записи з регістрами виконуються дуже швидко, оскільки вони знаходяться всередині центрального процесора.

Найважливіший регістр – лічильник команд або показчик команд, який вказує, яку команду потрібно виконувати наступної. Ще є регістр команд, в якому знаходиться виконувана в даний момент команда. У більшості комп'ютерів є і інші регістри, одні з них багатофункціональні, інші служать лише як певний меті. Треті регістри використовуються операційною системою для управління комп'ютером.

Устрій центрального процесора

Внутрішній устрій тракту даних типового фон-неймановського процесора ілюструє рис. 2.3. Тракт даних складається з регістрів даних (зазвичай від 1 до 32), арифметико-логічного пристрою (АЛП) і декількох сполучних шин. У самому АЛП є вхідні і вихідні регістри. Регістри процесора і регістри АЛП – це різні регістри. Вміст регістрів даних процесора (для прикладу на малюнку – це регістри №3 і №4) надходить у вхідні регістри АЛП, які на рис. 2.3 позначені буквами А і В. АЛП безпосередньо працює тільки зі своїми вхідними і вихідними регістрами. У вхідних регістрах АЛП знаходяться вхідні дані АЛП, поки АЛП виконує обчислення.

АЛП виконує додавання, віднімання та інші прості операції над вхідними даними і поміщає результат у вихідний регістр. Зміст цього вихідного регістра може записуватися знову в один з регістрів процесора (на малюнку це регістр №1 процесора) або зберігатися в пам'яті, якщо це необхідно. На рис. 2.3 представлена операція додавання, але АЛП може виконувати і інші операції.

Більшість команд можна розділити на дві групи: команди типу регістр - пам'ять і типу регістр-регістр. Команди першого типу викликають слова з пам'яті, поміщають їх в регістри процесора, де вони використовуються в якості вхідних даних АЛП. Слова – це такі елементи даних, порції даних, які переміщуються між пам'яттю і регістрами. Розмір слова зазвичай відповідає

розрядності регістру даних. Так, у 16-розрядних мікропроцесорів слово має довжину 16 біт, а у 32-розрядних мікропроцесорів – 32 біта. Інші команди цього типу поміщають дані з регістрів назад в пам'ять.

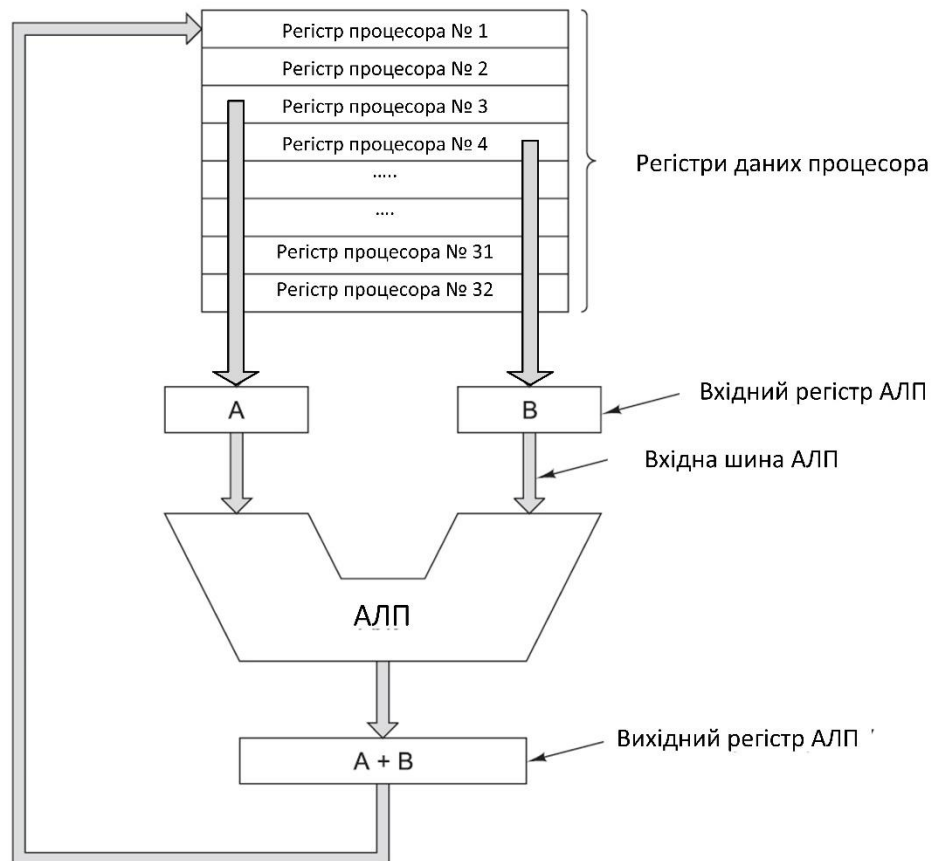


Рис. 2.3 Функціональна схема процесора

Команди другого типу викликають два операнда з регістрів, поміщають їх у вхідні регістри АЛП, виконують над ними якусь арифметичну або логічну операцію і переносять результат назад в один з регістрів процесора. Цей процес називається циклом тракту даних. В якійсь мірі він визначає, що може робити машина. Сучасні комп'ютери оснащуються кількома АЛП, які працюють паралельно і спеціалізуються на різних функціях. Чим швидше відбувається цикл тракту даних, тим швидше комп'ютер працює.

Виконання команд

Центральний процесор виконує кожну команду за кілька кроків. Під час виконання однієї команди він робить таку послідовність дій:

1. Викликає наступну команду з пам'яті і переносить її в регістр команд.
2. Змінює положення лічильника команд, який після цього вказує на наступну команду.
3. Визначає тип викликаної команди.

4. Якщо команда використовує слово з пам'яті, визначає, де знаходиться це слово.
5. Переносить слово, якщо це необхідно, з пам'яті в регістр центрального процесора.
6. Виконує команду.
7. Переходить до кроку 1, щоб почати виконання наступної команди.

Така послідовність кроків (вибірка – декодування – виконання) є основою роботи всіх комп'ютерів.

Контрольні питання по темі

Рівень модуля

Рівень курсу

1. Машина фон Неймана.
2. Двійкова система числення.
3. Організація комп'ютерних систем.
4. Устрій центрального процесора.

Лекція № 3

Тема: Загальна побудова комп'ютера. Частина 2**Зміст**

Основна пам'ять.....	2
Біт.....	2
Адреси пам'яті	2
Шини	3
Загальне уявлення про комп'ютерні шини	3
Фізична структура шини	4
Параметри шини.....	6
Шина PCI	6
Шина PCI Express	7
Відеопам'ять.....	8
Контрольні питання по темі.....	10
Рівень модуля	10
Рівень курсу	10

Джерела:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

<https://rutracker.org/forum/viewtopic.php?t=4956359>

Основна пам'ять

Пам'ять – це той компонент комп'ютера, в якому зберігаються програми і дані. Без пам'яті, звідки процесори зчитують і куди записують інформацію, не було б сучасних цифрових комп'ютерів.

Біт

Основною одиницею зберігання даних в пам'яті є двійковий розряд, який називається бітом. Біт може містити 0 або 1. Це – найменша одиниця пам'яті.

Адреси пам'яті

Пам'ять складається з комірок, кожна з яких може зберігати деяку порцію інформації. Кожна комірка має номер, який називається адресою. Виконувана комп'ютером програма може посилатися на певну комірку за її адресою. Якщо пам'ять містить n комірок, вони будуть мати адреси від 0 до $n-1$. Кожна комірка пам'яті складається з певного числа розрядів. Кожен розряд зберігає один біт інформації. Всі комірки пам'яті одного комп'ютера містять однакове число розрядів і відповідно можуть зберігати однакову кількість бітів. Якщо комірка складається з k розрядів, вона може зберігати рівно k біт інформації. У такій комірці може бути записана будь-яка з 2^k комбінацій нулів і одиниць. На рис. 1 показано різні варіанти організації 96-розрядної пам'яті, тобто пам'яті, яка повинна містити 96 біт інформації. Відзначимо, що сусідні комірки по визначенню мають послідовні адреси.

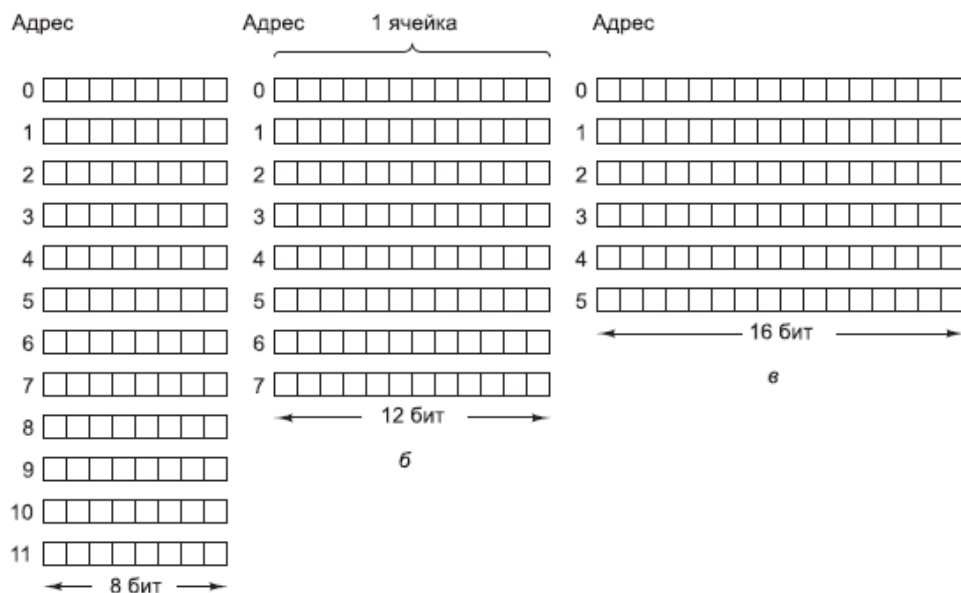


Рис. 1. Три варіанти організації 96-розрядної пам'яті

У комп'ютерах використовується двійкова система числення, адреси пам'яті також виражаються в двійкових числах. Якщо адреса складається з m біт, то максимальне число адресованих комірок складе 2^m . Наприклад, адреса для звернення до пам'яті, зображеної на рис. 1а повинна складатися принаймні

з 4 біт, щоб виражати всі числа від 0 до 11. В випадку влаштування пам'яті, як показано на рис. 1б і 1в достатньо 3-розрядної адреси. Число бітів в адресі визначає максимальну кількість адресованих комірок пам'яті і не залежить від числа бітів в комірці. 12-розрядні адреси потрібні як для пам'яті з 2^{12} комірок по 8 біт кожна, так й для пам'яті з 2^{12} комірок по 64 біт кожна.

Комірка – мінімальна одиниця пам'яті, яка адресується. В останні роки практично всі виробники випускають комп'ютери з 8-розрядними комірками, які називаються байтами. Байти групуються в слова. У комп'ютері з 32-розрядними словами на кожне слово доводиться 4 байт, а в комп'ютері з 64-розрядними словами - 8 байт. Така одиниця, як **слово**, необхідна, оскільки більшість команд виконують операції над цілими словами (наприклад, складають два слова). Таким чином, 32-розрядна машина містить 32-розрядні регістри і використовує команди виключно для маніпуляцій з 32-розрядними словами, тоді як 64-розрядна машина має 64-розрядні регістри і використовує команди для переміщення, додавання, віднімання та інших операцій виключно над 64-розрядними словами.

Шини

Як зазначалося, комп'ютерна система складається з трьох основних компонентів: центрального процесора, пам'яті та пристроїв введення-виведення (дисків, принтерів, сканерів, модемів). Обмін даними між цими компонентами проводиться за допомогою шин.

Загальне уявлення про комп'ютерні шини

Логічну структуру звичайного персонального комп'ютера ілюструє рис. 2.

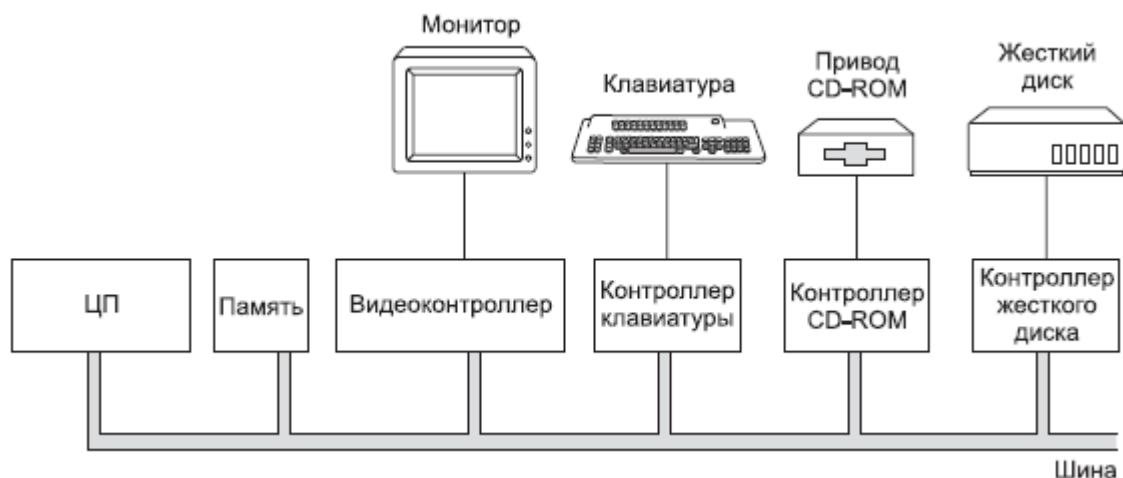


Рис. 2. Логічна структура звичайного персонального комп'ютера

У даного комп'ютера є одна шина для з'єднання центрального процесора, пам'яті і пристроїв введення-виведення; проте більшість систем мають дві і більше шин. Кожен пристрій введення-виведення складається з двох частин:

одна об'єднує більшу частину електроніки і називається контролером, а інша являє собою сам пристрій введення-виведення, наприклад, жорсткий диск. Контролер зазвичай розташовується на платі, яка вставляється у вільний роз'єм системної шини. Контролер зв'язується з самим пристроєм кабелем.

Контролер керує своїм пристроєм введення-виведення і одночасно регулює доступ до шини. Наприклад, якщо програма запитує дані з диска, вона посилає команду контролеру диска, який потім відправляє диску команду пошуку і інші команди. Після знаходження відповідної доріжки і сектора диск починає передавати контролеру дані у вигляді потоку бітів. Завдання контролера полягає в тому, щоб розбити потік бітів на фрагменти та записувати в пам'ять комп'ютера кожен такий фрагмент в міру накопичення для нього бітів. Окремий фрагмент зазвичай являє собою одне або декілька слів. Якщо контролер зчитує дані з пам'яті комп'ютера, або записує їх в пам'ять без участі центрального процесора, то кажуть, що здійснюється прямий доступ до пам'яті (Direct Memory Access, DMA). Коли передача даних закінчується, контролер видає *переривання*, змушуючи центральний процесор призупинити роботу поточної програми і почати виконувати особливу процедуру. Ця процедура називається програмою обробки переривань і потрібна вона для того, щоб перевірити, чи немає помилок, а у разі їх виявлення провести необхідні дії і повідомити операційній системі, що процес введення-виведення завершено. Коли програма обробки переривання завершується, процесор відновлює роботу програми, яка була припинена в момент переривання.

Шина використовується для передачі команд і даних не тільки контролерами введення-виведення, а й самим процесором. Що відбувається, якщо процесор і контролер введення-виведення хочуть отримати доступ до шини одночасно? В цьому випадку особлива мікросхема, яка називається арбітром шини, вирішує, чия черга перша. Зазвичай перевага віддається пристроям введення-виведення, оскільки роботу дисків та інших рухомих пристроїв не можна переривати, оскільки це може призвести до втрати даних. Коли ні один пристрій введення-виведення не функціонує, центральний процесор може повністю розпоряджатися шиною для взаємодії з пам'яттю. Однак якщо працює який-небудь пристрій введення-виведення, цей пристрій буде запитувати доступ до шини і отримувати його кожного разу, коли йому це потрібно. Цей процес, який пригальмовує роботу комп'ютера, називається захопленням циклу пам'яті (cycle stealing).

Фізична структура шини

Фізично шина – це кілька провідників, що з'єднують кілька пристроїв. Шини можна розділити на категорії відповідно до виконуваних функцій. Вони

можуть бути внутрішніми по відношенню до процесора і служити для передачі даних в АЛП і з АЛП, а можуть бути зовнішніми по відношенню до процесора і пов'язувати процесор з пам'яттю або пристроями введення-виведення. Кожен тип шини має певні властивості і до кожного з них пред'являються певні вимоги.

Перші персональні комп'ютери мали одну зовнішню шину, яка називалася системною. Вона складалася з декількох мідних проводів (від 50 до 100), які вбудовувалися в материнську плату. На материнській платі на однакових відстанях один від одного знаходилися роз'єми для мікросхем пам'яті і пристроїв введення-виведення. Сучасні персональні комп'ютери зазвичай містять спеціальну шину між центральним процесором і пам'яттю і, як мінімум, ще одну шину для пристроїв введення-виведення. На рис. 3 зображена система з одною шиною пам'яті і однієї шиною вводу-виводу.

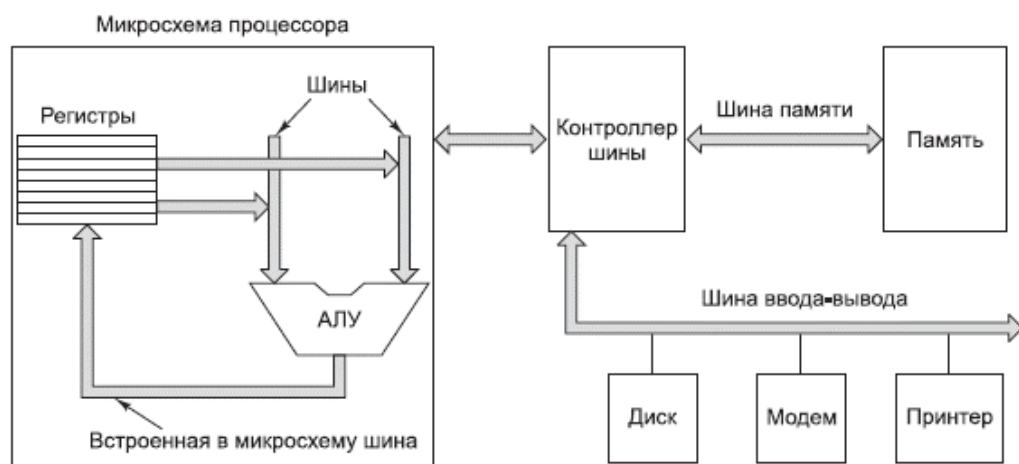


Рис. 3. Комп'ютерна система з декількома шинами

У літературі шини зазвичай зображуються у вигляді жирних стрілок, як показано на цьому малюнку. Різниця між жирною стрілкою і нежирною стрілкою, через яку проходить коротка діагональна лінія із зазначенням числа бітів, – невелика. Коли тип всіх бітів однаковий, наприклад, всі – адресні або всі – інформаційні, малюється звичайна стрілка. Коли зображають адресні лінії, лінії даних і управління всі разом, то використовується жирна стрілка.

Хоч розробники процесорів можуть використовувати будь-який тип шини, проте мають бути сформульовані чіткі правила про те, як працює шина; і всі пристрої, пов'язані з шиною, повинні підкорятися цим правилам, щоб плати, які випускаються сторонніми виробниками, підходили до системної шини. Ці правила називаються **протоколом** шини. Крім того, повинні існувати певні технічні вимоги, щоб плати від сторонніх виробників підходили до напрямних друкованих плат і мали роз'єми, механічно відповідні до материнської плати, а також з точки зору напруг, синхронізації і т. д.

Параметри шини

Ширина (кількість адресних ліній) шини – найочевидніший параметр. Чим більше адресних ліній містить шина, тим до більшого обсягу пам'яті може звертатися процесор. Якщо шина містить n адресних ліній, тоді процесор може використовувати її для звернення до 2^n різних комірок пам'яті. Для пам'яті великої місткості необхідно багато адресних ліній. Проблема полягає в тому, що для широких шин потрібно більше проводів, ніж для вузьких. Вони займають більше фізичного простору (наприклад, на материнській платі) і для них потрібні роз'єми більшого розміру. Всі ці фактори роблять шину коштовною.

З розвитком комп'ютерної техніки в комп'ютерах збільшується не тільки число адресних ліній, але і число інформаційних ліній, хоча це відбувається з іншої причини. Можна збільшити пропускну здатність шини двома способами: скоротити час циклу шини (зробити більшу кількість передач в секунду), або збільшити ширину шини даних (тобто збільшити кількість бітів, що передаються за цикл). Можна підвищити швидкість роботи шини, але зробити це досить складно, оскільки сигнали на різних лініях передаються з різною швидкістю (це явище називається расфазировкой шини). Чим швидше працює шина, тим більше расфазировка.

Шина PCI

Peripheral Component Interconnect (PCI) є стандартом для карт розширень персональних комп'ютерів. Intel розробила цю технологію в 1993 році для процесора Pentium. За допомогою цієї шини з'єднується процесор з пам'яттю і іншими периферійними пристроями.

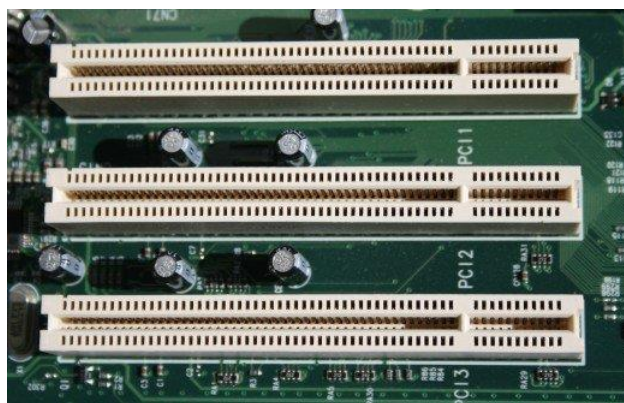


Рис. 4. Зовнішній вигляд роз'ємів PCI

PCI підтримує передачу 32 і 64 розрядних даних, кількість переданих даних дорівнює розрядності процесора, 32 бітний процесор буде використовувати 32 бітну шину, а 64 бітний – 64 бітну. Працює шина на частоті 33 МГц. Більшість PCI карт працюють на напрузі 5 вольт.

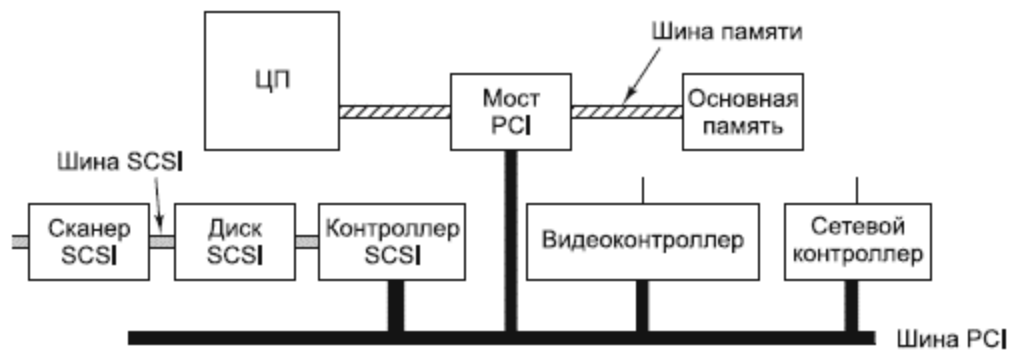


Рис. 5. Персональний комп'ютер з шиною PCI.

Існує багато різних конфігурацій шини PCI. Найбільш типова з них показана на рис. 5. У такій конфігурації центральний процесор взаємодіє з контролером пам'яті по виділеному високошвидкісному з'єднанню. Таким чином, контролер з'єднується з пам'яттю безпосередньо, тобто передача даних між центральним процесором і пам'яттю відбувається не через шину PCI. Інші периферійні пристрої під'єднуються прямо до шини PCI. Машина такого типу зазвичай містить 2 або 3 порожніх роз'єму PCI, щоб покупці мали можливість підключати карти PCI для нових периферійних пристроїв).

Шина PCI Express

Подальший розвиток в напрямку підвищення швидкості призвело до того, що PCI була замінена шиною **PCI Express** (скорочено **PCIe**). Багато сучасних комп'ютерів підтримують обидві шини, завдяки чому користувачі можуть підключати нові, швидкі пристрої до шини PCIe, а старі, більш повільні – до шини PCI.

Якщо шина PCI представляла собою оновлену версію старої шини ISA з більш високою швидкістю і розрядністю паралельно передаваних даних, PCIe являє собою кардинальну зміну в порівнянні з шиною PCI. Власне, це взагалі не шина, а однорангова мережа, що використовує разрядно-послідовні лінії і комутацію пакетів. У неї більше від Інтернету, ніж від традиційних шин. Архітектура PCIe зображена на рис. 6.

З'єднання між пристроями є послідовними, тобто мають розрядність в один біт замість 8, 16, 32 або 64 біт. Хоча, здавалося би, 64-розрядний з'єднання має вищу пропускну спроможність, на практиці відмінності в часі поширення 64-розрядної інформації, звані расфазировкой, змушують використовувати відносно низькі швидкості передачі даних. За послідовного з'єднання дані передаються на значно більш високій швидкості, що більш ніж компенсує втрату паралелізму.

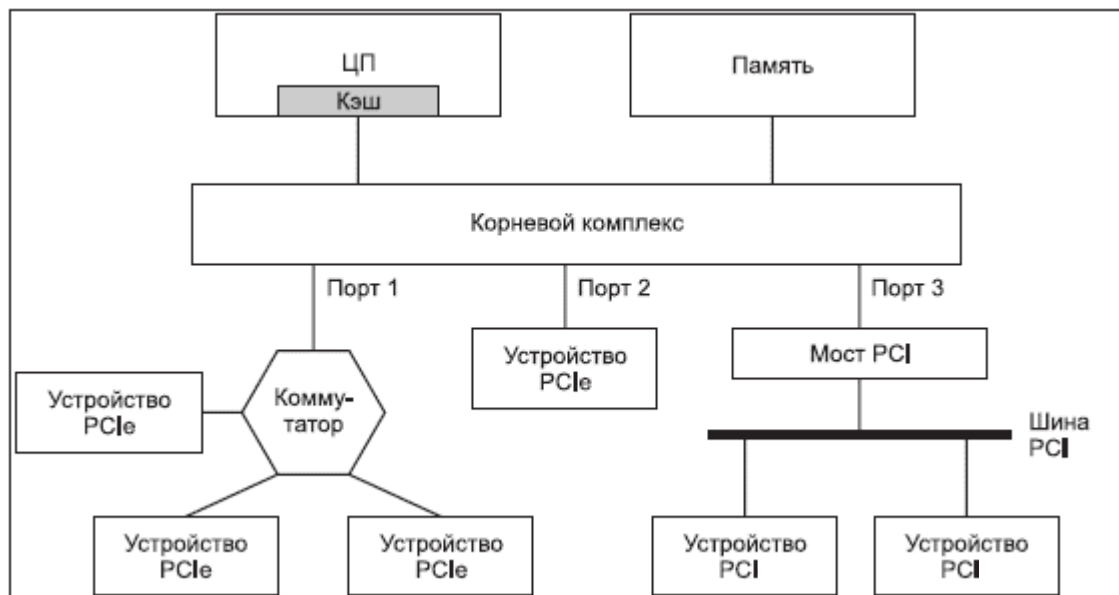


Рис. 6. Архітектура системи PCIe з трьома портами PCI

Шини PCI працюють на максимальній тактовій частоті 66 МГц. Під час передачі 64 біт за такт швидкість передачі даних складає 528 Мбайт/с. За тактовою частотою 8 Гбіт/с, навіть в разі послідовної передачі, швидкість передачі по шині PCIe становить 1 Гбайт/с. Крім того, обмін даними між пристроєм і кореневим комплексом або комутатором не обмежується однією провідний парою. Пристрій може мати до 32 провідних пар, які називаються трактами (lanes) або доріжками. Тракти працюють несинхронно, тому расфазировки в даному випадку не існує, оскільки немає паралельної прередачі даних. На більшості материнських плат є 16-трактовий роз'єм для графічної карти, що для PCIe 3.0 забезпечує пропускну здатність в 16 Гбайт/с – приблизно в 30 разів більше, ніж у графічних карт PCI. Така пропускну здатність необхідна для додатків, вимоги яких постійно зростають – наприклад, для тривимірної графіки.

Відеопам'ять

Оновлення картинки на екранах моніторів виконується від 60 до 100 разів на секунду; для цього використовується відеопам'ять, розміщена на платі контролера дисплея. Відеопам'ять містить одну або кілька бітових карт, що представляють собою виведене на екран зображення. Якщо, скажімо, на екрані вміщується 1920x1080 елементів зображення (пікселів), значить, в відеопам'яті міститься 1920x1080 значень, по одному на кожен піксель. З метою швидкого перемикання з одного зображення на інше в пам'яті може розміщуватися кілька таких карт.

В сучасних дисплеях кожен піксель представлений 3-байтним значенням RGB, яке визначає інтенсивність червоного (Red), зеленого (Green) і синього (Blue) компонентів зображення (потужні професійні монітори

використовують 10 і більше біт на колір). Як відомо, будь-який колір можна представити шляхом лінійної суперпозиції трьох згаданих базових квітів.

Якщо в відеопам'яті зберігається інформація про 1920x1080 пікселів, та ще й до того ж на кожен з них виділяється по 3 байта, загальний обсяг цих даних складає близько 6,2 Мбайт; тому на будь-які маніпуляції з таким зображенням йде досить багато процесорного часу. Для виведення растрових (тобто сформованих на основі бітових карт) зображень потрібна велика пропускна здатність. Наприклад, для відтворення одного кадру повнокольорових мультимедійних даних в повноекранному форматі на дисплеї розміром 1920x1080 необхідно скопіювати в відеопам'ять 6,2 Мбайт. Якщо врахувати, що повноцінний відеофільм виводиться зі швидкістю 25 кадрів в секунду, загальна швидкість передачі даних повинна складати 155 Мбайт/с. Таку пропускну здатність не здатна забезпечити навіть первісна версія шини PCI (132 Мбайт/с), але шина PCIe легко справляється з нею.

Контрольні питання по темі

Рівень модуля

Рівень курсу

1. Адреси пам'яті комп'ютера.
2. Загальне уявлення про комп'ютерні шини.
3. Фізична структура шини комп'ютера.
4. Параметри шини.
5. Шина PCI.
6. Шина PCI Express.
7. Відеопам'ять комп'ютера.

Лекція № 4

Тема: Технологія цифрового супутникового радіомовлення. Частина перша.

Зміст

Інформаційні технології широкого мовлення.....	2
Загальний опис системи цифрового супутникового радіомовлення...	3
Передача інформації в системах аналогового мовлення	4
Призначення модуляції в процесі передачі інформації	6
Природна необхідність модуляції.....	6
Частотний поділ каналів.....	6
Процес модуляції та демодуляції	8
Електромагнітний спектр	10
Прийнятий поділ діапазонів радіохвиль.....	11
Використання діапазонів радіохвиль.....	11
Контрольні питання по темі.....	13
Рівень модуля	13
Рівень курсу	13

Джерела:

1. Ріхтер С. Г. Цифрове радіомовлення. – 2-е изд., Стереотип. М. : Горяча лінія - Телеком, 2015. - 352 с. : іл.
2. Савельев И.В. Курс общей физики: Учеб. пособие. В 3-х томах. Том 2. Электричество и магнетизм. Волны. Оптика. – М.: Наука, 1987 - 445 с.

Інформаційні технології широкого мовлення

В цій та наступній лекції ми розглянемо одну з найсучасніших інформаційних технологій, яка відноситься до інформаційних технологій широкого мовлення, а саме – технологію цифрового супутникового радіомовлення, яка поєднує в собі як цифрові технології так й космічні технології. В теперішній час в світі існує дві системи цифрового радіомовлення: *Sirius XM Radio* в Північній Америці та *WorldSpace* яка покриває територію Європи, Азії, Африки та Південної Америки. В Україні та в пострадянських державах цифрове супутникове радіомовлення поки що відсутнє.

Ми будемо розглядати технологію цифрового супутникового радіомовлення на прикладі системи XM Radio, яка першою в світі почала надавати послуги супутникового радіомовлення. Інші системи використовують таку ж саму технологію. Слід відзначити, що до інформаційних технологій широкого мовлення також відноситься супутникове телебачення, яке в цілому побудовано за тими ж самими принципами, що й супутникове радіо.

Широке мовлення, або англійською **broadcasting**, – це є розповсюдження певної інформації одночасно багатьом клієнтам; клієнти не можуть впливати на зміст отримуваної інформації. До широкого мовлення, в першу чергу відносяться радіо та телебачення. В цьому разі клієнтами є радіослухачі, або телеглядачі. Вони можуть обирати канал отримання інформації, проте зміст інформації, що передається каналом, є незмінним та однаковим для всіх клієнтів.

Широке мовлення активно застосовується в цифрових технологіях, та в першу чергу, в мережевому устаткуванні. Зокрема мережі Wi-Fi побудовані саме на цій технології, яку ми й будемо подальше розглядати. Під час вивчення супутникового цифрового радіо ми познайомимся з основними процесами, які протікають в інформаційних системах, та визначимо терміни, які застосовуються до цих процесів, зокрема:

- ✓ широке мовлення,
- ✓ модуляція та демодуляція сигналів,
- ✓ частотний поділ каналів,
- ✓ аналого-цифрове перетворення,
- ✓ паралельне та послідовне кодування,
- ✓ мультимплексна передача даних, та інше.

Знання зазначеної термінології та розуміння процесів інформаційних технологій є необхідною складовою професійних знань спеціалістів з кібербезпеки та комп'ютерно-інтегрованих систем.



Загальний опис системи цифрового супутникового радіомовлення

Для функціонування цифрового супутникового радіомовлення XM Radio використовуються два супутники, які мають назви Rock и Roll та обертаються навколо Землі, знаходячись на так званій *геостаціонарній орбіті*. Геостаціонарна орбіта характеризується тим, що супутники, які на ній знаходяться, мають період обертання навколо Землі рівно 24 години, тобто вони обертаються з тою самою кутовою швидкістю, що й Земля. Завдяки цьому супутник завжди знаходиться над одною точкою поверхні Землі. Відстань від поверхні Землі до супутника, який знаходиться на геостаціонарній орбіті становить приблизно 36 тис. км.



Радіосигнал від наземної станції направляється на супутник, який транслює його на всі наземні радіоприймачі. Радіоприймачі встановлюються в авто, або в домівках, та отримують надійний цифровий сигнал зі супутника. Супутники Rock и Roll забезпечують **покриття** радіосигналом всієї території США та частково Канади. Якщо на якійсь території є покриття, то це означає, що радіоприймачі, які пристосовані для отримання цього сигналу і знаходяться на даній території, гарантовано отримують сигнал такої якості, що дозволяє клієнту отримувати послугу – слухати

радіопередачі. Проте це має бути спеціальний радіоприймач, в даному випадку – цифровий радіоприймач системи XM Radio і ніякий інший.

Послуги XM Radio платні. Система транслює близько 100 каналів (які можна називати «радіостанціями» лише умовно) різної тематики: новинні, музикальні, розважальні, розмовні та таке інше. Ці канали – однакові на всій території США. Вони надійно, без перешкод, приймаються в будь-якій точці країни, мають високу якість звукового сигналу, якісні та цікаві за своїм контентом радіопередачі, а їх передавання не потребує ніякої наземної інфраструктури, що дуже зручно для країн, які мають великі площі. Через зазначені переваги цей сервіс набув величезної популярності в США та Канаді

і він має значний комерційний успіх. Лише в великих містах через наявність хмарочосів, які закривають пряму видимість на супутник, використовують ретранслятори, які знаходяться на висотних спорудах, отримують сигнал від супутника та повторно його випромінюють через антени, які встановлені в зонах, де не можна напряду прийняти сигнал від супутника.

Далі ми розглянемо технічні аспекти побудови цифрового супутникового радіомовлення. Проте перед тим, як розглядати, як функціонує цифрове радіо, необхідно подивитися як працює звичайне радіо, яке ми тепер часто називаємо аналоговим радіо на відміну від цифрового.

Передача інформації в системах аналогового мовлення

Радіомовлення, або ефірне радіомовлення, – технологія передачі звукової інформації необмеженій кількості слухачів шляхом розповсюдження електромагнітних хвиль у відкритому просторі, званому в цьому випадку радіоефіром. Радіомовлення характеризується передачею сигналу за принципом «від одного – до багатьох», тобто більш ніж одному слухачеві, як правило – за заздалегідь відомим розкладом.

В звичайному радіо не використовуються цифрові данні, ніяка інформація не оцифровується, і тому ця технологія не є інформаційною, проте вона лежить в основі будь-якого передавання інформаційних сигналів через ефір. В радіо використовуються аналогові, тобто безперевні сигнали від джерела мовлення, як то, наприклад, мікрофон, або програвач музикальних вінілових платівок.

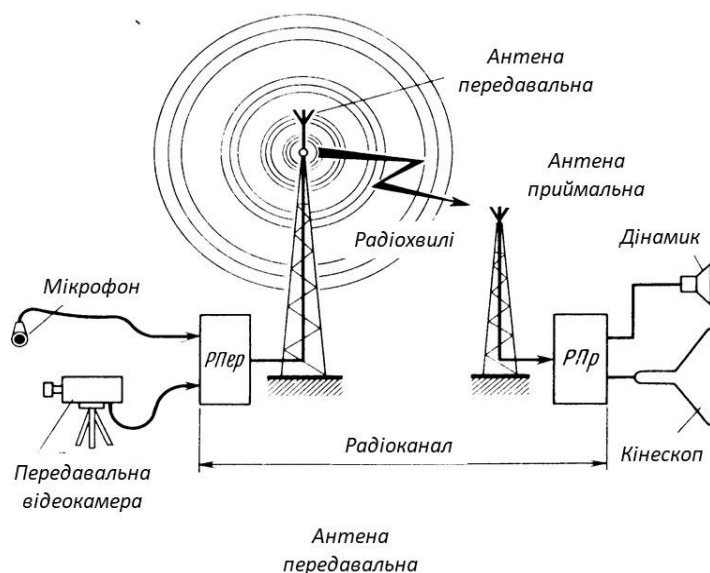


Рис. 1. Аналогове радіомовлення та телебачення

Мовлення в радіоефірі (рис. 1) здійснюється за допомогою радіопередавачів тієї чи іншої потужності, які передають інформацію на тій чи іншій частоті електромагнітного випромінювання. Радіопередавач з супутнім

обладнанням (студії, канали зв'язку і живлення, антена на щоглі або вищці) називається радіостанцією. Прийом передач на стороні радіослухача виконується за допомогою радіоприймачів.

Частота є головною характеристикою радіомовної станції. У перші десятиліття розвитку радіомовлення, для позначення характеристики опорних коливань використовували довжину хвилі випромінювання, відповідно – шкали радіоприймачів були проградуйовані в метрах. В наш час опорні коливання позначають частотою і, відповідно, шкали радіоприймачів градуують в кГц, МГц і ГГц.

Радіопередавач (рис. 2) складається з наступних частин:

- джерело звукового сигналу низької частоти (наприклад, мікрофон),
- підсилювач звукового сигналу низької частоти (ПНЧ),
- генератор синусоїдальних коливань високої частоти (опорної хвилі),
- модулюючий пристрій (модулятор), що змінює параметри випромінюваної хвилі (амплітуду, частоту, фазу або кілька параметрів одночасно) відповідно до сигналу низької частоти, який потрібно передати,
- підсилювач високої частоти (ПВЧ), що збільшує потужність вихідного сигналу, який направляється до антени,
- антена, що забезпечує випромінювання сигналу.

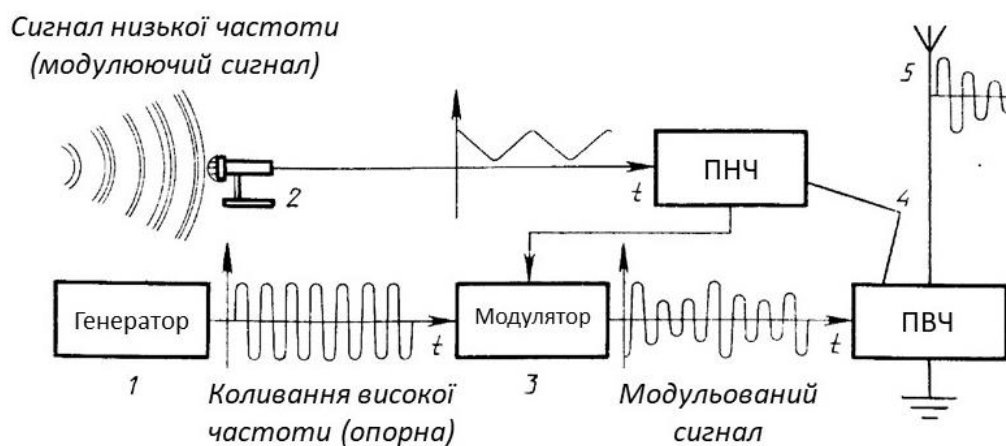


Рис. 2. Радіопередавач

Радіоприймач (рис.3) складається з наступних частин:

- антена разом із вхідним колом, яка приймає сигнал,
- підсилювач високої частоти (ПВЧ), який збільшує потужність вхідного сигналу,
- демодулятор, який виділяє інформаційний сигнал низької частоти,

- підсилювач звукового сигналу низької частоти (ПНЧ),
- динамік, який відтворює звуковий сигнал.

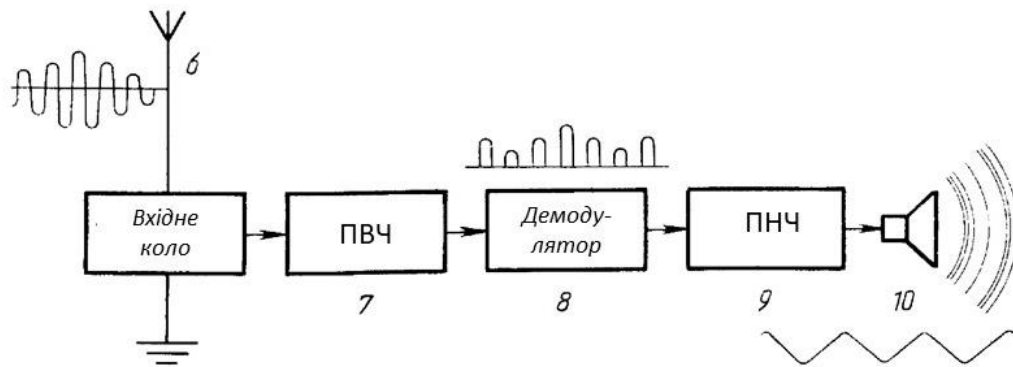


Рис. 3. Радіоприймач

Призначення модуляції в процесі передачі інформації

Для чого потрібна модуляція?

Природна необхідність модуляції

Модуляція потрібна тому, що згідно з законами фізики потужність випромінювання антени радіопередавача (а відповідно й потужність сигналу, що приймається антеною радіоприймача) пропорційна квадрату частоти електричного коливання. З практичної точки зору починати радіомовлення можна десь від частот 150-200 кГц. Більш низькі частоти майже не передаються в ефір. Найбільш популярні частоти радіостанцій знаходяться в діапазоні частот біля 100 МГц. Далі ми розглянемо ці діапазони.

Частотний поділ каналів

По друге, модуляція потрібна для того, щоб організувати так званий **частотний поділ каналів**. Кожна радіостанція – це окремий канал, який працює на якійсь своїй строго визначеній частоті, тобто кожна радіостанція використовує свою строго визначену частоту опорного коливання. Ось, наприклад частоти деяких відомих радіостанцій в нашому городі:

Частота МГц		Назва радіостанції
87.50		UA:Українське радіо Дніпро
88.10		UA:Радіо Культура
88.50		Радіо Relax
89.70		Ретро FM
90.10		Радіо Мелодія FM

90.50		Радіо РОКС
90.90		Авторадіо
91.40		Радіо НВ
100.50		Радіо Люкс FM
106.80		Kiss FM
107.30		Информатор FM
107.70		Радіо Максимум

Якщо частота радіостанції відома, то за допомогою коливального контуру (рис.4) можна виділити сигнал, який має частоту коливання тільки з того діапазону частот, який знаходиться близько до опорної частоти радіостанції, яку ми бажаємо слухати (рис. 5), а всі інші сигнали, які мають інші частоти – пригасити (щодо коливальних контурів зверніться до курсу «Електротехніка», який вам було прочитано у попередньому семестрі). Цей процес називається **фільтрацією** (рис. 6) сигналів і він буде розглянутий в курсі «Радіотехнічні кола та сигнали», який вам буде прочитано на наступних семестрах.

Таким чином ми можемо отримувати інформацію тільки з того каналу, який нам потрібен. Це й є **частотне розділення каналів**, або **частотний поділ каналів**.

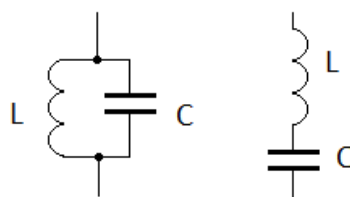


Рис. 4. Паралельний та послідовний коливальні контури

В старих радіоприймачах (рис. 5) настройка коливального контуру на частоту станції проводилась вручну шляхом повільного змінення користувачем ємності конденсатора в коливальному контурі а також завдяки перемиканню діапазонів шляхом натискання на клавіши, що призводить до підключення до контуру тої чи іншої котушки індуктивності з відповідними параметрами



Рис. 5. Шкала радіоприймача 60-80-х років

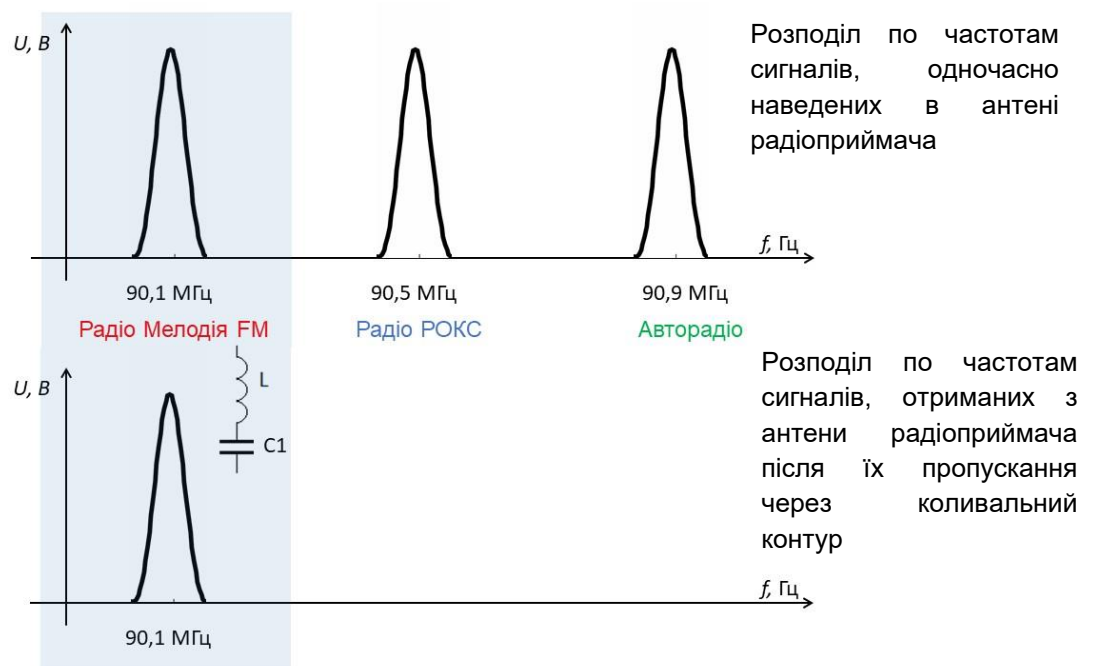


Рис. 6. Виділення сигналу радіостанції за допомогою коливального контуру шляхом налаштування його параметрів (L або C , або те й інше разом) таким чином, щоб через контур могли проходити сигнали тільки того діапазону частот, на якій працює станція.

Процес модуляції та демодуляції

На наступному рисунку (рис. 7) показано, як проходить процес модуляції сигналу. Тут показано тільки амплітудну модуляцію, хоча існують ще й інші види модуляції, як то: частотна та фазова модуляція. На рисунку послідовно зверху вниз зображено:

- високочастотне *опорне* колювання,
- звуковий сигнал $U_{зв}$, який є низькочастотним, це є *модулюючий* сигнал,
- результуючий сигнал (добуток модулюючого та опорного сигналу), це є *модульований* сигнал.

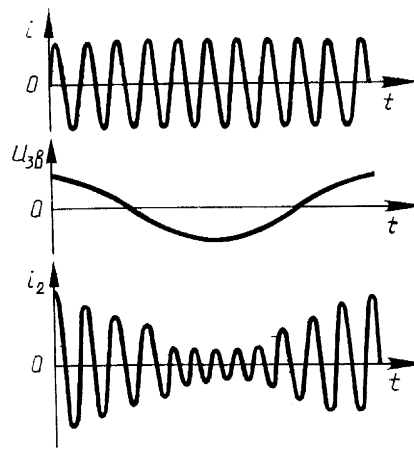


Рис. Амплітудна модуляція

На наступному рисунку (рис. 8) показано, як проходить демодуляція сигналу в радіоприймачі (рис. 3), тобто виділення інформаційного (в нашому випадку – звукового сигналу) за допомогою діодного детектора та фільтру низької частоти. Демодулювати сигнал можна й іншими способами, про які ви дізнаєтесь в курсах, які вам будуть прочитані.

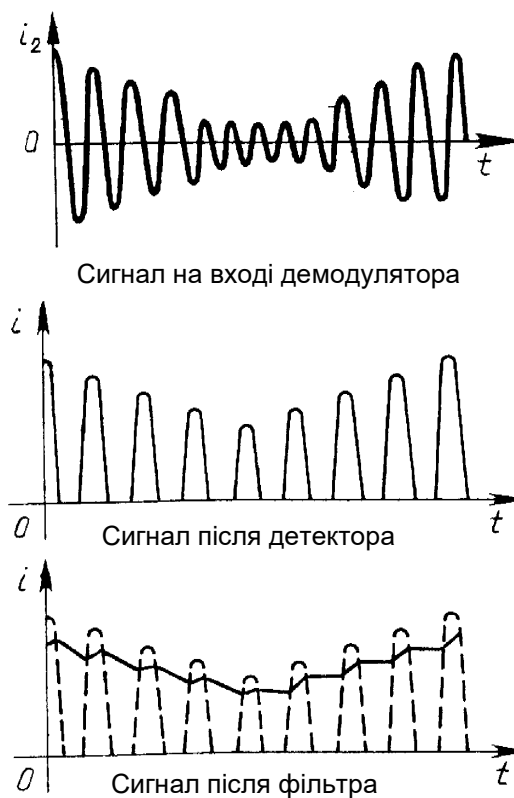


Рис. 8. Демодуляція амплітудно модульованого сигналу за допомогою діодного детектора та фільтру низької частоти

Таким чином, суть *модуляції* полягає в наступному. Формується деяке коливання (найчастіше гармонійне), зване *опорним коливанням* або просто *опорне*, і один з параметрів цього коливання змінюється в часі пропорційно до

вихідного сигналу. Первинний інформаційний сигнал називають *модулюючим*, а результуюче коливання зі змінними в часі параметрами – *модульованим* сигналом. Зворотний процес, а саме виділення модулюючого сигналу з модульованого коливання, називається *демодуляцією* або *детектуванням*.

Електромагнітний спектр

З курсу фізики відомо, що радіохвилі – це є ніщо інше, як електромагнітні хвилі, які ми зустрічаємо в природі в декількох різних видах. Залежно від частоти (або довжини хвилі в вакуумі), а також способу випромінювання та реєстрації, розрізняють кілька видів електромагнітних хвиль: радіохвилі, оптичне випромінювання (у тому числі видиме світло), рентгенівське випромінювання та гамма-випромінювання.

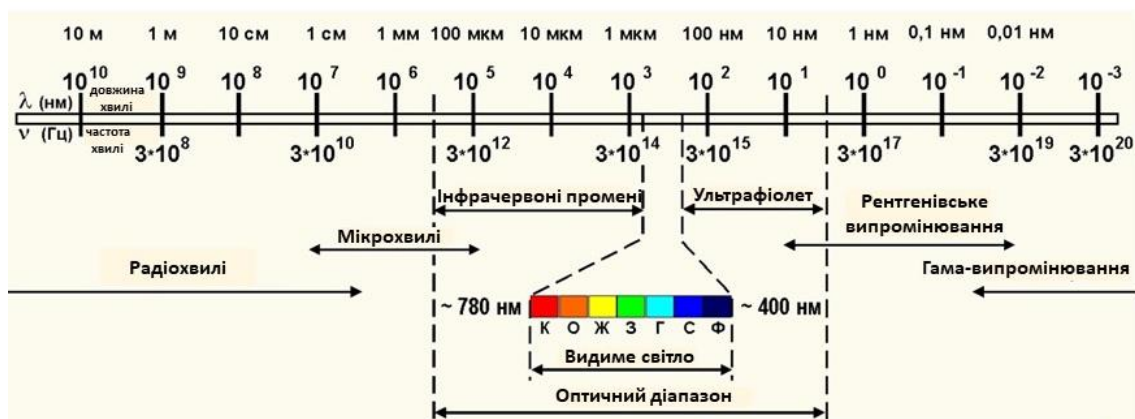


Рис. 1 Діапазони спектру електромагнітних хвиль

Будь-яка радіохвиля характеризується довжиною хвилі (або частотою), інтенсивністю (або амплітудою), фазою, поляризацією та типом хвилі.

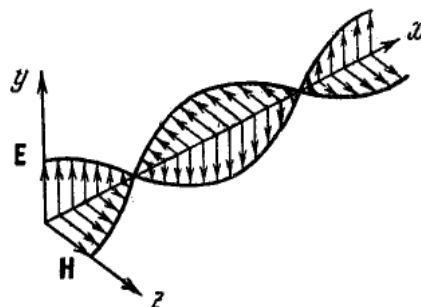


Рис. Електромагнітна хвиля у просторі

Між періодом T , частотою f , довжиною хвилі λ , та швидкістю її розповсюдження c (швидкістю світла) існують наступні співвідношення:

$$f = \frac{1}{T}, \quad T = \frac{1}{f}, \quad \lambda f = c, \quad \lambda = \frac{c}{f}, \quad f = \frac{c}{\lambda}$$

Прийнятий поділ діапазонів радіохвиль

У зв'язку з особливостями поширення та генерації весь діапазон довжин хвиль і частот радіохвиль прийнято ділити на дев'ять діапазонів, наведених в таблиці 1.

Таблиця 1

<i>Діапазон</i>	<i>Довжина хвилі</i>	<i>Частота</i>
Наддовгі хвилі	100 – 10 км	3 – 30 кГц
Довгі хвилі (LF)	10 – 1 км	30 – 300 кГц
Середні хвилі (MF)	1000 – 100 м	0,3– 3 МГц
Короткі хвилі (HF)	100 – 10 м	3 – 30 МГц
Ультракоткі хвилі:		
– метрові (VHF)	10 – 1 м	30 – 300 МГц
– дециметрові (UHF)	10 – 1 дм	0,3 – 3 ГГц
– сантиметрові (SHF)	10 – 1 см	3 – 30 ГГц
– міліметрові	10 – 1 мм	30 – 300 ГГц
– субміліметрові	1 – 0,1 мм	0,3– 3 ТГц

Відповідно до міжнародних стандартів кожному мікрохвильовому субдіапазону назначено певну латинську літеру, або сполучення декількох літер:

UHF – от 0.3 до 1 ГГц

L – от 1 до 2 ГГц

S – от 2 до 4 ГГц

C – от 4 до 8 ГГц

X – от 8 до 12 ГГц

Ku – от 12 до 18 ГГц

K – от 18 до 27 ГГц

Ka – от 27 до 40 ГГц

V – от 40 до 75 ГГц

W – от 75 до 110 ГГц

mm – от 110 до 300 ГГц

Використання діапазонів радіохвиль

В наступній таблиці наведені деякі приклади використання діапазонів радіохвиль для мовлення та інших потреб.

Таблиця 2

<i>Назва</i>	<i>Полоса частот</i>	<i>Довжина хвилі</i>
Діапазон середніх хвиль (MW)	530–1610 кГц	565,65–186,21 м
Діапазон коротких хвиль	5,9–26,1 МГц	50,8–11,49 м
Цивільний діапазон	26,965–27,405 МГц	11,118–10,940 м

Телевізійні канали: з 1 по 5	48–100 МГц	6,25–3,00 м
Кабельне телебачення	100–174 МГц	
Телевізійні канали: з 6 по 12	174–230 МГц	1,72–1,30 м
Кабельне телебачення	230–470 МГц	
Телевізійні канали: з 21 по 39	470–622 МГц	6,38–4,82 дм
Діапазон ультракоротких хвиль мовлення	62–108 МГц	1 м
ISM-діапазон	2–4 ГГц	15–7,5 см
Діапазони військових частот	29.50–31.75 МГц	
Діапазони частот цивільної авіації	108–136 МГц	

Діапазон ISM (ISM - Industrial, Scientific, Medical) є тією частиною радіочастотного спектру загального призначення, яка може бути використана без ліцензування. Єдина вимога для пристроїв в ISM-діапазоні – це відповідність нормам, які встановлюються регулюючими органами даної країни для цієї частини частотного спектра.

Контрольні питання по темі

Рівень модуля

Рівень курсу

1. Інформаційні технології широкого мовлення.
2. Загальний опис системи цифрового супутникового радіомовлення.
3. Передача інформації в системах аналогового мовлення.
4. Призначення модуляції в процесі передачі інформації скрізь радіоефір.
5. Частотний поділ каналів.
6. Процес модуляції та демодуляції.
7. Електромагнітний спектр.
8. Прийнятий поділ діапазонів радіохвиль.
9. Використання діапазонів радіохвиль.

Лекція № 5

Тема: Технологія цифрового супутникового радіомовлення. Частина 2**Зміст**

Передавання інформації в системах цифрового мовлення.....	2
Аналого-цифрове та цифро-аналогове перетворення	3
Аналогові, дискретні та цифрові сигнали	3
Дискретизація та квантування	4
Теорема Котелникова-Найквіста. Вибір частоти дискретизації.	6
Контрольні питання по темі.....	8
Рівень модуля	8
Рівень курсу	8

Джерела:

1. Ріхтер С. Г. Цифрове радіомовлення. – 2-е изд., Стереотип. М .: Горяча лінія - Телеком, 2015. - 352 с .: іл.
2. Сергиенко А.Б. Цифровая обработка сигналов. – 3-е изд. — СПб.: БХВ-Петербург, 2011. — 768 с.: ил.

Передавання інформації в системах цифрового мовлення

Головною відмінністю цифрового мовлення від аналогового – це те, що інформація, яка передається каналами цифрового радіо, або телебачення є саме цифровою: інформація набуває форми даних, а до інформації у формі даних вже можуть бути застосовані комп'ютерні методи збору, накопичення, зберігання, пошуку, обробки і видачі інформації споживачеві. Таким чином, цифрове мовлення (радіо, або телевізійне) – це вже є інформаційна технологія, на відміну від аналогового мовлення, яке не може буде віднесене до інформаційних технологій.

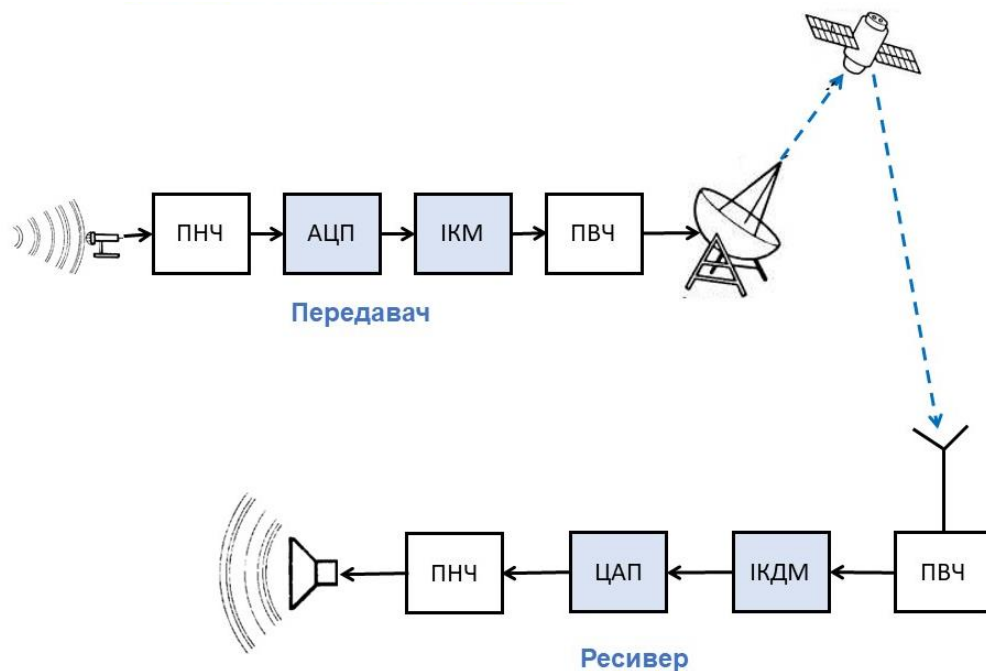


Рис.1 Передавання та приймання сигналів цифрового радіо

Всі сигнали, які сприймає, або створює природа й людина зокрема – вони аналогові, тобто безперервні у часі. Але комп'ютер може працювати виключно з цифровими даними, які є дискретними, тобто визначеними тільки в окремі моменти часу, вони не є безперервними. Людина не може сприймати цифровий сигнал, наприклад цифровий звук, а комп'ютер не може працювати з аналоговим сигналом. Тому обов'язковою складовою системи цифрового мовлення має бути пристрій, який спочатку перетворює аналоговий сигнал (в цьому випадку – електричний сигнал від мікрофону, який відповідає звуку) у цифрову форму, та ще один пристрій, який потім виконує зворотну задачу – перетворення сигналу з цифрової форми в аналогову (знову в електричний сигнал, який динамік перетворює на звуковий сигнал). Перший пристрій називається «аналого-цифровий перетворювач», скорочено АЦП. Другий пристрій «цифро-аналоговий перетворювач» – ЦАП.

Ці два пристрої: АЦП та ЦАП, – входять у склад будь якої інформаційної або керуючої системи, коли ці системи пов’язані з фізичними об’єктами, або з природними джерелами сигналів. Аналого-цифрове та цифро-аналогове перетворення завжди відбувається в електричній формі. Перетворюються виключно електричні сигнали. Тобто спочатку фізична величина, як то звук наприклад, має перетворитися в електричну, а вже потім – перетворення в цифровий код. В зворотному напрямку – так само: спочатку перетворення «цифра – аналог» для електричних сигналів, а вже потім електричний сигнал перетворюється в фізичний, наприклад – акустичний сигнал.

Як відбувається аналого-цифрове та цифро-аналогове перетворення ми розглянемо надалі.

Надалі ми ще також ознайомимся з іншим пристроєм, який позначено на рисунку 1 як «ІКМ» – це імпульсно-кодовий модулятор. Ви вже знаєте, що для того щоби бути переданим в ефір, низькочастотний сигнал має модулюватися. І це не залежить від того, яка форма сигналу – аналогова, чи цифрова: будь-який низькочастотний сигнал має модулюватися, бо інакше не буде випромінювання в ефір (точніше – воно буде дуже, дуже слабим). В цифровому радіо модуляція проходить так само, як і в аналоговому радіо, але є особливості, пов’язані з тим, що сигнал, який передається, є цифровим, тобто – це код. В цьому випадку модуляції підвергається не аналоговий сигнал, а кодовий сигнал. І кажуть не просто про модуляцію, а про імпульсно-кодову модуляцію.

Природньо, що в зворотному напрямку, на стороні радіоприймача (або ресивера – саме так прийнято називати радіоприймач в цифровому радіо та телебаченні), треба зробити демодуляцію сигналу з імпульсно-ковою модуляцією. Для цього, звичайно, потрібен ІКМ-демодулятор, який також зазначено на рисунку 1 як ІКДМ.

Про імпульсно-кодову модуляцію ми поговоримо в наступній лекції.

Аналого-цифрове та цифро-аналогове перетворення

Аналогові, дискретні та цифрові сигнали

Сигнали можуть бути безперервними і **дискретними**. У зв'язку з цим сигнали можна розділити на наступні класи:

- а) довільні за величиною і безперервні за часом (рис. 2, а);
- б) довільні за величиною і дискретні за часом (рис. 2, б);
- в) квантовані за величиною і безперервні за часом (рис. 2, в);
- г) квантовані за величиною і дискретні за часом (рис. 2, г).

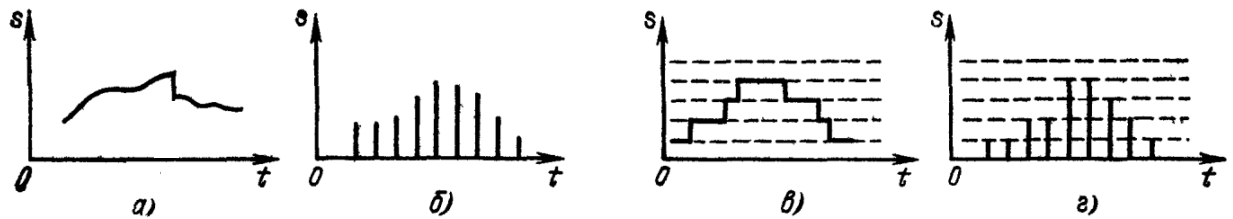


Рис. 2. Сигнали довільні за величиною і за часом (а), довільні за величиною і дискретні за часом (б), квантовані за величиною і безперервні за часом (в), квантовані за величиною і дискретні за часом (г)

Сигнали першого класу (рис. 2а) називають **аналоговими** або безперервними, оскільки вони задаються по осі часу на незліченній множині точок. Такі множини називають континуальними. По осі ординат сигнали можуть приймати будь-яке значення в певному інтервалі.

На рис. 2б представлено сигнал, заданий на дискретних значеннях часу t (на зліченій множині точок); величина ж сигналу в цих точках може приймати будь-яке значення в певному інтервалі по осі ординат (як і на рис. 2а). Таким чином, термін **дискретний** характеризує не самий сигнал, а спосіб задавання його на часовій осі. Дискретні сигнали можуть створюватися безпосередньо джерелом інформації (наприклад, дискретними датчиками в системах керування або телеметрії), або утворюватися в результаті дискретизації континуальних сигналів.

Сигнал на рис. 2в заданий на всій часовій осі, проте його величина може приймати лише дискретні значення. У подібних випадках говорять про сигнал, **квантованим** за рівнем.

Квантування використовують при поданні сигналів в цифровій формі за допомогою цифрового кодування, оскільки рівні можна пронумерувати числами з кінцевим числом розрядів. Тому дискретний за часом і квантований за рівнем сигнал (рис. 2г) називається **цифровим**.

Таким чином, можна розрізняти: аналогові (рис. 2а), дискретні (рис. 2б), квантовані (рис. 1.2в) та цифрові (рис. 1.2г) сигнали.

Дискретизація та квантування

Усякий сигнал, який існує в природі, або техніці) є безперервною функцією часу, всі сигнали від початку – аналогові. А от вже послідовність чисел, що представляє сигнал при цифровій обробці, є дискретним рядом, і він, природньо, не може повністю відповідати аналоговому сигналу. Числа, що складають послідовність, є значеннями сигналу в окремі (дискретні) моменти часу, і вони називаються **відліками** сигналу. Як правило, відліки беруться через рівні проміжки часу T , які зветься періодом дискретизації (або

інтервалом, кроком дискретизації). Величина, обернена до періоду дискретизації, називається частотою дискретизації: $f_d = 1/T$.

Ясно, що подання сигналу набором дискретних відліків призводить до втрати інформації, так як ми нічого не знаємо про поведінку сигналу в проміжках між відліками. Однак, як буде показано в далі, можна підібрати таку частоту дискретизації, за якої втрата інформації не відбувається, завдяки чому аналогові сигнали можуть бути точно відновлені за значеннями своїх дискретних відліків.

Процес перетворення аналогового сигналу в послідовність відліків називається **дискретизацією за часом** (*sampling*), а результат такого перетворення – дискретним сигналом.

При обробці сигналу в обчислювальних пристроях його відліки представляються у вигляді двійкових чисел, що мають обмежене число розрядів. Внаслідок цього відліки можуть приймати значення лише з певної кінцевої множини і, отже, при поданні сигналу неминуче відбувається його округлення. Процес перетворення відліків сигналу в числа називається **квантуванням за рівнем** (*quantization*).

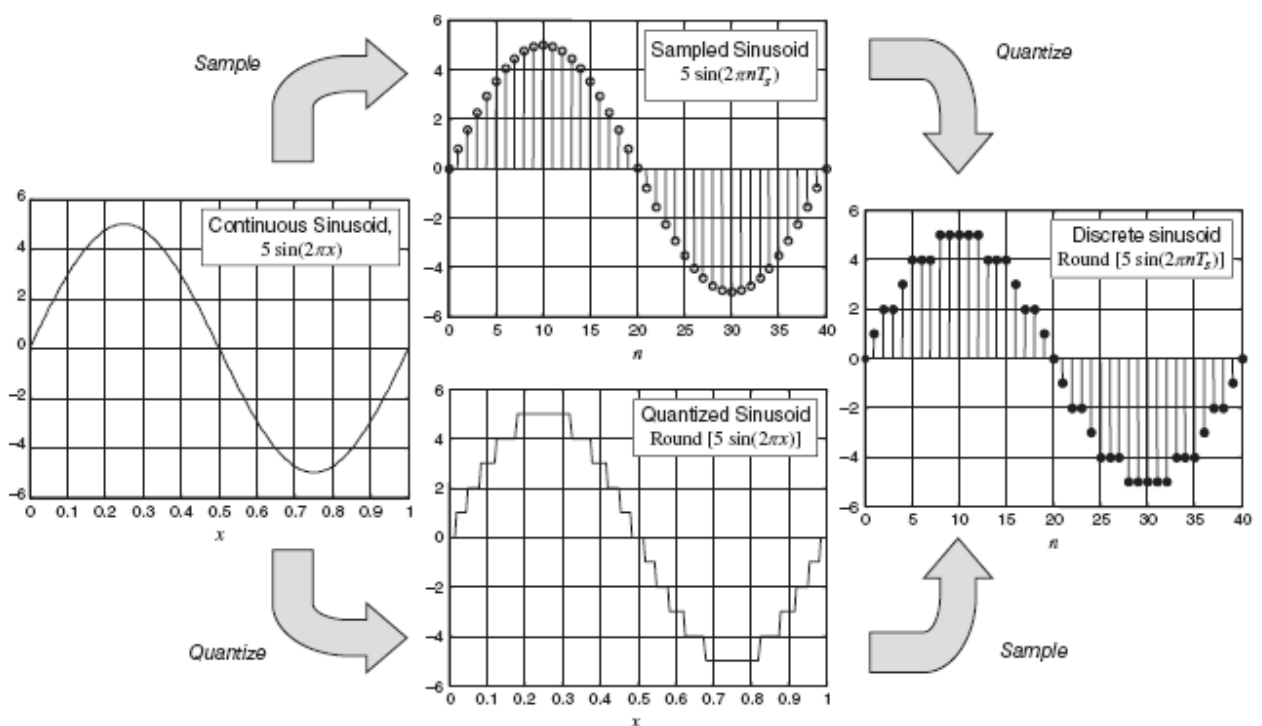


Рис 3. Процеси дискретизації за часом та квантування за рівнем

Нагадаємо ще раз, що сигнал, дискретний у часі, але не квантований за рівнем, називається просто **дискретним** сигналом. Сигнал, дискретний у часі і квантований за рівнем, називають **цифровим** сигналом. Різницю між аналоговими, дискретними і цифровими сигналами ще раз ілюструє рис. 4.

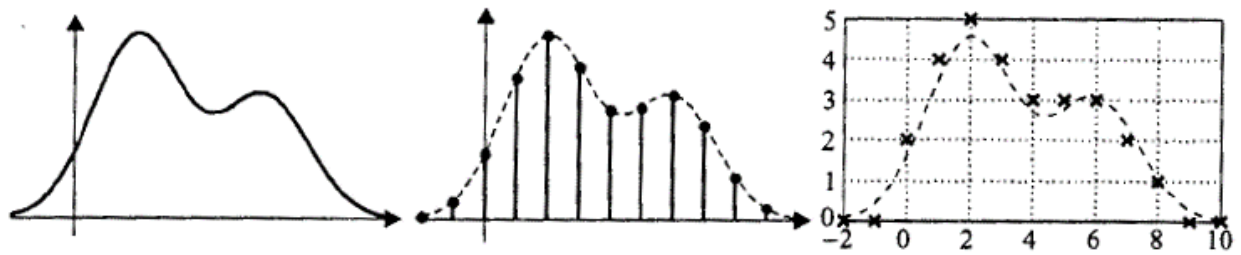


Рис. 4 - Аналоговий (зліва), дискретний (в центрі) і цифровий (справа) сигнали

Обчислювальні пристрої, призначені для обробки сигналів, можуть оперувати тільки цифровими сигналами.

Теорема Котелникова-Найквіста. Вибір частоти дискретизації.

Теорема Котелникова-Найквіста формулюється наступним чином:

Будь-який сигнал, який не містить складових з частотами вище деякого значення $f_{\text{с}}$, можна без втрат інформації представити своїми дискретними відліками, взятими з частотою дискретизації $f_{\text{д}}$, якщо виконується співвідношення:

$$f_{\text{д}} > 2 f_{\text{с}}$$

Ця теорема свого часу математично доказана незалежно один від одного двома вченими: Найквістом та Котелниковим. Що нам дає ця теорема та як її використовувати? А використовують її в інформаційних технологіях дуже й дуже часто.

Покажемо це на прикладах.

Приклад 1. Людина може чути звуки, частота яких – не вище 20 кГц. Тобто ніякі звуки, частота яких вище 20 кГц людина не чує. Це є верхня частота сигналу: $f_{\text{с}} = 20$ кГц. Тепер, якщо покластися на теорему Котелникова-Найквіста та провести дискретизацію звукового сигналу з частотою $f_{\text{д}}$, яка буде вдвічі вищою ніж $f_{\text{с}}$, тобто вищою за 40 кГц, то такий сигнал можна потім абсолютно точно відтворити – отримати такий самий аналоговий сигнал, точно той звук, який ми перед цим дискретизували. Людина почує абсолютно такий самий звук, як і раніше. Ніякої різниці ніхто і ніщо не зможе знайти. Таким чином, завдяки теоремі Котелникова-Найквіста ми знаємо, що найякісніша передача звука відбувається, якщо вибрати частоту дискретизації більшою, ніж 40 кГц. Саме тому під час оцифрування звуку найчастіше використовують такі стандартні значення як 44,1 кГц та 48 кГц.

Приклад 2. Людська мова містить у собі звуки з частотами від 80 Гц до $f_6 = 4$ кГц.

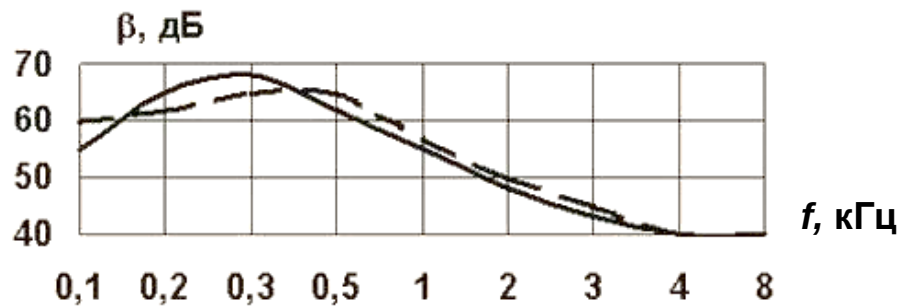


Рис. 5. Енергетичний спектр людської мови: суцільна лінія – українська мова, штрихова лінія – англійська мова

Це означає, що для того, щоби дискретизувати мовну інформацію (голос людини, розмову, виступ та таке інше) таким чином, щоб потім можна було її повністю відтворити, достатньо застосувати частоту дискретизації $f_o = 2f_6 = 2 \cdot 4 = 8$ кГц, або більше.

Теорема Котелникова-Найквіста вказує, яким має бути мінімальне значення частоти дискретизації. І в цьому полягає найцінніше значення цієї теореми для світу інформаційних технологій. Проте постає питання, а наскільки більше від мінімального значення слід вибирати частоту дискретизації? Зрозуміло, що чим вище частота дискретизації, тим вищою є якість відновлення сигналу. Але з іншого боку, чим вона вище, тим більшою стає кількість відліків, тим більше треба пам'яті, щоб зберегти ці відліки, тим більшою має бути пропускна здатність каналів передачі інформації, тим більша розрядність АЦП і ЦАП має бути й тим дорожчими вони стають. Тобто з практичної точки зору немає ніякого сенсу завищувати частоту дискретизації. Саме через це стандартні значення частоти дискретизації звукового сигналу – 44,1 кГц (Audio CD) та 48 кГц (MP3), тобто вони ненабагато вищі за мінімально потрібну частоту в 40 кГц. Більш того, якщо сам звуковий сигнал не дуже якісний, коли в ньому відсутні верхні частоти, то з метою зменшення навантаження на канали передавання даних і, відповідно, вартості цього передавання, використовують частоти дискретизації значно нижчі, ніж 40 кГц. Так для передавання мовної інформації в телефонії використовують частоту дискретизації 8 кГц (а іноді навіть 6 кГц), а для звичайних радіопередач можна застосовувати частоту дискретизації 22050 Гц.

Контрольні питання по темі

Рівень модуля

Рівень курсу

1. Передавання інформації в системах цифрового мовлення.
2. Аналогові, дискретні та цифрові сигнали.
3. Дискретизація та квантування сигналу.
4. Теорема Котелникова-Найквіста. Вибір частоти дискретизації.

Лекція № 6

Тема: Технологія цифрового супутникового радіомовлення. Частина 3**Зміст**

Поняття динамічного діапазону	2
Визначення необхідної кількості рівнів квантування в АЦП	3
Динамічний діапазон звукового сигналу	4
Динамічний діапазон АЦП	4
Необхідний динамічний діапазон АЦП для якісного оцифровування цифрового сигналу	6
Паралельне та послідовне передавання даних	6
Модуляція та маніпуляція сигналів	9
Двійкова фазова маніпуляція	10
Квадратурна фазова маніпуляція	11
Мультиплексне передавання цифрових даних	12
Мультиплексування з ортогональним частотним поділом каналів...	12
Контрольні питання по темі	15
Рівень модуля	15
Рівень курсу	15

Джерела:

1. Ріхтер С. Г. Цифрове радіомовлення. – 2-е изд., Стереотип. М.: Горячая лінія - Телеком, 2015. - 352 с.: іл.
2. Сергиенко А.Б. Цифровая обработка сигналов. – 3-е изд. — СПб.: БХВ-Петербург, 2011. — 768 с.: ил.

Поняття динамічного діапазону

У попередній лекції ми розібралися з тим, яким чином вибирається частота дискретизації коли проводиться аналого-цифрове перетворення. Залишається питання: яким чином слід обирати кількість рівнів квантування? Щоб відповісти на це питання розглянемо таке поняття, як динамічний діапазон.

Самим простим визначенням динамічного діапазону є таке: **динамічний діапазон** – це відношення найбільшого значення певної фізичної величини до найменшого значення цієї фізичної величини за певних умов вимірювання. Для вимірювання динамічного діапазону використовується одиниця вимірювання децибел, яка є десятою частиною одиниці, що має назву Бел. Бели та децибели використовують для вимірювання відношення потужностей $\frac{P_1}{P_0}$ за допомогою десяткових логарифмів. Якщо ми говоримо про електричні сигнали, то тоді відношення потужності сигналу P_1 до деякого базового значення P_0 виражене в Белах визначається таким чином:

$$D = \lg \frac{P_1}{P_0} \quad (1)$$

А це ж саме співвідношення, виражене через децибели, природньо, має вигляд:

$$D = 10 \lg \frac{P_1}{P_0} \quad (2)$$

бо 1 Бел дорівнює 10-ти децибелам, як вже було вказано раніше.

Повертаємось до динамічного діапазону. Отже, якщо певна величина (в нашому випадку ми розглядаємо сигнали, зокрема – електричні) під час спостереження мала найбільшу та найменшу виміряну потужність P_{max} та P_{min} відповідно, то тоді кажуть, що динамічний діапазон змінення величини (сигналу) в децибелах складає

$$D = 10 \lg \frac{P_{max}}{P_{min}} \text{ дБ} \quad (3)$$

Якщо ж ми будемо порівнювати не потужності, а амплітуди, інтенсивності, середні значення та таке інше, наприклад, виміряні значення електричних струмів, тоді динамічний діапазон розраховується за формулою:

$$D = 20 \lg \left(\frac{I_{max}}{I_{min}} \right) \text{ дБ} \quad (4)$$

Як бачите, формула (4) така сама, як й (3), проте в ній замість коефіцієнту 10 стоїть коефіцієнт 20, хоча й там й там результат в децибелах. Чому це так? Пам'ятаємо: Бели використовуються для порівняння **потужностей**, а потужність будь якої фізичної величини пов'язана зі значенням фізичної величини (миттєвим значенням, амплітудою, інтенсивністю тощо) виключно квадратичною залежністю. Завжди: $P \sim I^2$. А раз так, то маємо наступне:

$$D = \lg \frac{P_1}{P_0} = \lg \frac{I_1^2}{I_0^2} = \lg \left(\frac{I_1}{I_0} \right)^2 = 2 \lg \left(\frac{I_1}{I_0} \right) \quad (5)$$

Це якщо вимірювати в Белах. Якщо ж вимірювати в децибелах, то отримуємо формулу (4).

Таким чином, якщо в децибелах порівнюємо безпосередньо потужності, то коефіцієнт 10, якщо значення – коефіцієнт 20!

Приклади порівняння потужності

Таблиця 1

P_1/P_0	Бели	Децибели
0.001	-3	-30
0.01	-2	-20
0.1	-1	-10
1	0	0
10	1	10
100	2	20
1000	3	30

Приклади порівняння значень

Таблиця 2

I_1/I_0	Бели	Децибели
0.001	-6	-60
0.01	-4	-40
0.1	-2	-20
1	0	0
10	2	20
100	4	40
1000	6	60

Визначення необхідної кількості рівнів квантування в АЦП

Тепер, коли ми знайомі з поняттям динамічного діапазону, ми можемо вирішити поставлену задачу: скільки треба мати рівнів квантування, аби перевести аналоговий сигнал в цифрову форму? Відповідь така: квантування

сигналу за рівнем має відбуватися в тих самих межах, в яких змінюється сигнал, який ми оцифровуємо. Більш строго: динамічний діапазон квантування має бути не меншим, ніж динамічний діапазон самого сигналу. Перейдемо до прикладу з цифрової обробки звукових сигналів.

Динамічний діапазон звукового сигналу

Інтенсивність, або гучність звуку характеризується акустичним тиском. Найтихіший звук, який може почути людина має інтенсивність (акустичний тиск) – 20 мкПа. Тобто $I_{min}=20$ мкПа. Інтенсивність (гучність) інших звуків у співвідношенні до цієї величини можна виразити у децибелах:

Інтенсивності звукових сигналів

Таблиця 3

Вид звукового сигналу	Рівень
➤ дуже тиха мова (шепіт)	5 - 10 дБ,
➤ тиха мова	30 - 40 дБ,
➤ мова на середньому рівні гучності	50 - 60 дБ,
➤ гучна мова	60 - 70 дБ,
➤ крик	70 - 80 дБ,
➤ фортисимо великого симфонічного оркестру	90 дБ,
➤ виття сирени «швидкої допомоги»	100 дБ,
➤ шум реактивного двигуна на відстані 5 м	120 дБ.

Таким чином, для того, аби забезпечити якісне відтворення звуку будь якого музикального твору достатньо під час перетворення аналогового сигналу в цифровий код забезпечити динамічний діапазон 90 дБ, оскільки звук симфонічного оркестру змінюється саме в цих межах.

Динамічний діапазон АЦП

Кількість рівнів квантування в аналого-цифровому перетворювачі відповідає розрядності АЦП. Розрядність АЦП – це кількість розрядів у вихідному слові даних АЦП, тобто кількість розрядів двійкового числа, яке АЦП формує на своєму виході. Наприклад, на рис. 1 представлено статичну характеристика АЦП, тобто графік, який показує, як пов'язані вхідний та вихідний сигнал АЦП.

Вхідним сигналом для АЦП є аналоговий сигнал, який ми оцифровуємо. Вихідним сигналом є двійковий код номеру того рівня квантування, до якого було заокруглено значення поточного відліку вхідного сигналу. Цей номер подається у вигляді n -розрядного слова. Тут n – це і є розрядність АЦП. Як вже було вказано кількість розрядів визначає кількість рівнів квантування в АЦП. Наприклад, на рис. 1 ми бачимо статичну характеристику 3-розрядного АЦП. Таке АЦП має 8 рівнів ($2^3=8$) квантування, оскільки в 3-розрядному слові, що складається з 3-х біт, можна записати тільки вісім чисел: від 0 до 7.

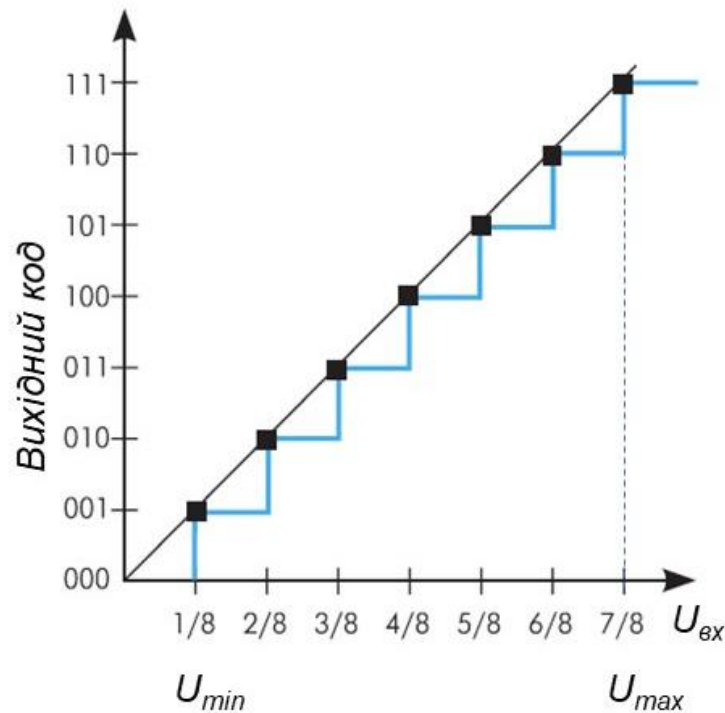


Рис.1 Статична характеристика АЦП

4-х розрядне АЦП має 16 рівнів квантування ($2^4=16$).

8-и розрядне АЦП має 256 рівнів квантування ($2^8=256$).

10-ти розрядне АЦП має 1024 рівнів квантування ($2^{10}=1024$) і так далі.

Рівні квантування в АЦП розташовані рівномірно, на однаковій відстані один від одного. Ця відстань називається кроком квантування. Зрозуміло, що найвищий рівень квантування в 2^n разів більше, ніж найменший рівень, який є кроком квантування. Найвищий рівень квантування відповідає найбільшому значенню напруги вхідного сигналу U_{max} , яке ще може бути перетворено у код. Найнижчий рівень квантування відповідає найменшому значенню напруги вхідного сигналу U_{min} , тобто кроку квантування (рис.1). Оскільки код найменшого рівня дорівнює одиниці, а найбільшого 2^n-1 , то відношення найбільшого рівня квантування в АЦП до найменшого приблизно дорівнює 2^n . Звідси ми легко знаходимо динамічний діапазон АЦП:

$$D_{АЦП} = 20 \lg \frac{U_{max}}{U_{min}} = 20 \lg N = 20 \lg (2^n) \quad (6)$$

Якщо підставити в цю формулу різні значення розрядності, то отримаємо динамічний діапазон АЦП відповідної розрядності:

*Динамічний діапазон АЦП різної розрядності**Таблиця 4*

Розрядність	Динамічний діапазон
3	18 дБ
8	48 дБ
10	60 дБ
12	72 дБ
16	96 дБ
24	144 дБ

Необхідний динамічний діапазон АЦП для якісного оцифровування цифрового сигналу

Наведена таблиця нам ясно показує, що для того, щоб якісно витворювати будь які музикальні твори, тобто для того, щоб забезпечувати квантування з динамічним діапазоном не менш ніж 90 дБ, достатньо вибрати 16-розрядне АЦП, яке має динамічний діапазон 96 дБ. До речі, стандарт Audio CD передбачає саме таку розрядність АЦП для запису цифрового звукового сигналу.

Проте слід зазначити наступне. Через те, що під час аналого-цифрового перетворення відбувається закруглення значень відліків, то ідеально відтворити первинний аналоговий сигнал за його цифровими значеннями принципово неможливо. Отриманий після цифро-аналогового перетворення сигнал завжди буде відрізнятися від первинного аналогового сигналу як мінімум на величину кроку квантування. Це – так звана похибка квантування. Вона неминуче присутня у вихідному сигналі АЦП. Що стосується звукових сигналів, то використання 16-розрядних АЦП робить похибку квантування настільки малою, що слух людини її зовсім не помічає.

Паралельне та послідовне передавання даних

Отже, цифровий сигнал отримано – ми перевели аналоговий звуковий сигнал у цифрову форму завдяки АЦП. Тепер цифровий сигнал має бути передано надалі аж до ресивера цифрового супутникового радіо. Яким чином це відбувається?

Існують різні способи передавання цифрових даних. Сигнал може передаватися або паралельно (рис. 2 зліва), або послідовно (рис. 2 справа).

За **паралельного передавання даних** всі розряди одного слова даних передаються одночасно, або як кажуть – синхронно, проте кожний розряд цього слова передається по окремому електричному проводу. Саме так передаються дані в паралельних шинах комп'ютера (див. лекції 2, 3). Коли ми розглядали це питання, то ми казали про негативний вплив так званої

«розфазіровки», коли електричні імпульси, які відповідають значенням розрядів (бітам) одного й того ж слова, доходять до приймача в різний час – одні раніше, а інші з запізненням. Розфазіровка неминує виникати під час паралельного передавання даних, а через її наявність швидкість паралельного передавання має свої обмеження.

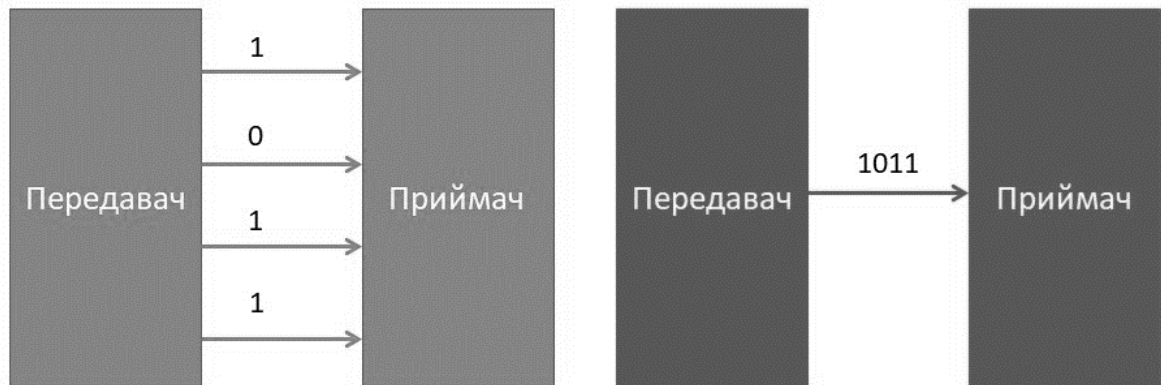


Рис. 2. Паралельне (зліва) та послідовне (справа) передавання цифрових даних

Для **послідовного передавання цифрових даних** використовують лише одну пару проводів. Один з проводів має нульовий потенціал, відносно якого на другому проводі передавач виставляє той чи інший потенціал, який відповідає нулю, або одиниці. Цей сигнал («0» або «1») виставляється лише на короткий проміжок часу – один такт передавання і відповідає значенню лише одного розряду слова, що передається. Таким чином, біти одного слова передаються в часі один за одним (у вигляді імпульсів) – тобто, послідовно.

Наприклад, в стандартному послідовному порту введення-виведення UART (*universal asynchronous receiver/transmitter*) одиниця передається високим потенціалом, а нуль – низьким (рис.3). В іншому стандарті – стандарті послідовного інтерфейсу RS-232 передбачено, що одиниця передається позитивною напругою +12В, а нуль передається негативною напругою –12В. Але й там й там біти 8-розрядного слова у вигляді імпульсів передаються один за одним, послідовно – від наймолодшого розряду D0 до найстаршого D7. Передача слова починається службовими сигналами – старт-бітами, які показують приймачу, що почалося передавання даних. Передавання завершується так само службовими сигналами, на цей раз це – стоп-біти.

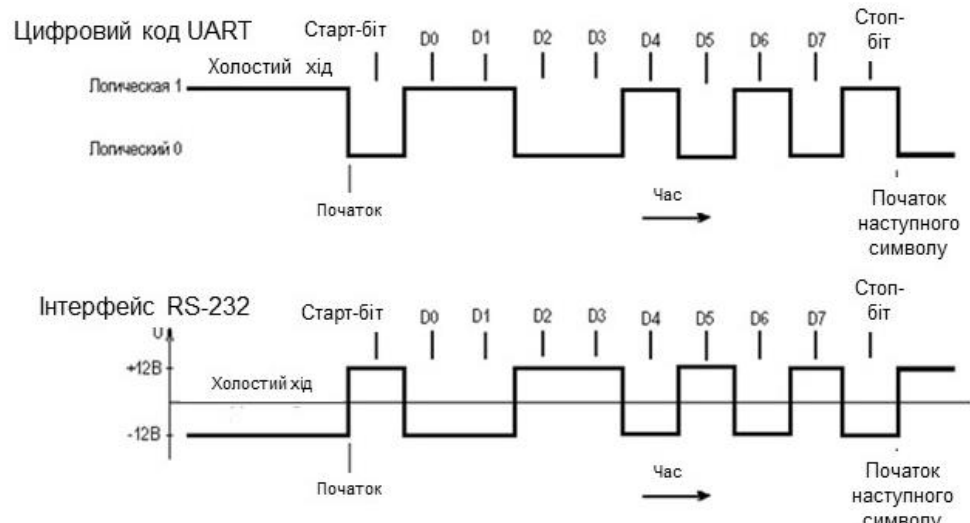


Рис. 3 Електричні сигнали під час послідовного передавання цифрових даних однорозрядним послідовним кодом

В розглянутому нами випадку послідовного передавання даних за один такт передається лише один біт інформації. Проте за один такт можна передавати й більше інформації, наприклад, – два біти зразу. Наступний рисунок показує, як це можна зробити.

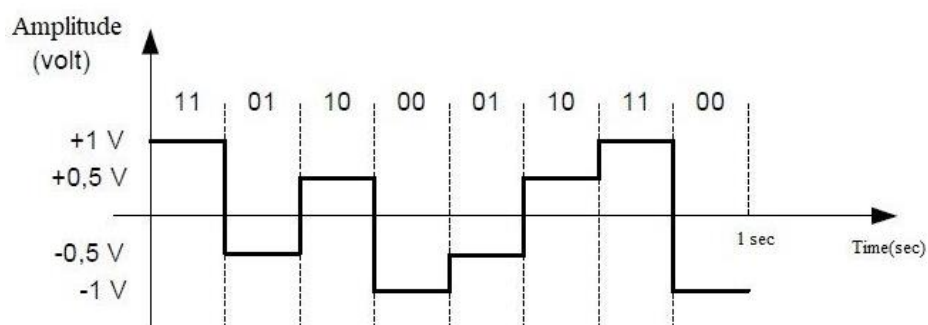


Рис. 4 Дворозрядний послідовний код

На рисунку видно, що для послідовного передавання даних використовується не два рівні електричного потенціалу, а відразу чотири: -1В, -0.5В, +0.5В, +1В. Тобто, в електричній лінії розрізняють чотири різних стани. А якщо кожному стану приписати якесь числове значення, то таким чином ми зможемо передавати за один такт не одне з двох можливих значень (0 або 1), а одне з чотирьох. Такими значеннями, вочевидь є числа 0 (двійковий код 00), 1 (двійковий код 01), 2 (двійковий код 10) та 3 (двійковий код 11). В результаті пропускна спроможність лінії збільшується вдвічі.

Існує та використовується на практиці й трирозрядний послідовний код з 8-ма різними амплітудами сигналу ($2^3=8$), й чотирирозрядний послідовний код з 16-ма різними амплітудами сигналу ($2^4=16$). Проте з підвищенням розрядності коду надійність передавання знижується, оскільки розпізнати

якою саме є амплітуда сигналу, який поступив на вхід приймача (тобто, який код прийшов?), тим складніше, чим більше існує ймовірних варіантів.

Модуляція та маніпуляція сигналів

Ми отримали цифровий код сигналу. Тепер його треба передати на супутник, потім від супутника на ресивер, що знаходиться на Землі. Між цими пристроями немає електричних проводів. Проте існує радіоефір, через який й відбувається передача даних. Передавання через радіоефір схоже на передавання через електричну лінію – пару проводів; це передавання також виконується послідовним кодом, це – послідовне передавання. Але, як відомо низькочастотний сигнал передати через ефір неможливо (лекція 4). Для того аби потужність випромінювання радіохвиль стала достатньою для передавання сигналу на потрібну відстань, необхідно збільшити частоту електричного сигналу, яким є послідовний цифровий код. Це робиться шляхом модуляції інформаційного сигналу, яку ми вже розглядали в лекції 4. В разі модуляції цифрових сигналів замість терміну модуляція використовують термін маніпуляція. Тобто **маніпуляція** – це модуляція цифрових сигналів.

Як ми вже розглядали в лекції 4 для модуляції інформаційного сигналу використовують гармонійний сигнал високої частоти – так зване опорне коливання. Гармонійний сигнал (рис. 5), як відомо, має всього три параметри: амплітуда I_m , частота f (величина зворотна до періоду T) та початкова фаза φ .

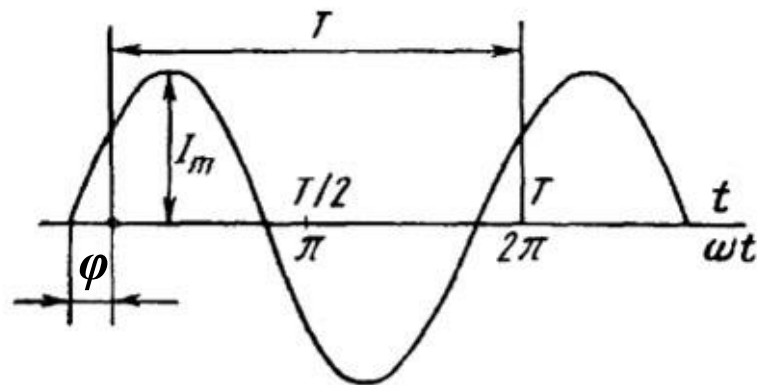


Рис. 5. Гармонійне коливання

Маніпулювання однорозрядного цифрового коду проводиться наступним чином: у відповідність до значення біту «0» або «1» ставиться або амплітуда, або частота, або початкова фаза опорного коливання (рис. 6).

Якщо відповідно до значення біта змінюється амплітуда опорного коливання, то це – **амплітудна маніпуляція** (рис. 6 зверху). Якщо відповідно до значення біта змінюється частота опорного коливання, то це – **частотна маніпуляція** (рис. 6 всередині). Якщо відповідно до значення біта змінюється

початкова фаза опорного коливання, то це – **фазова маніпуляція** (рис. 6 знизу).

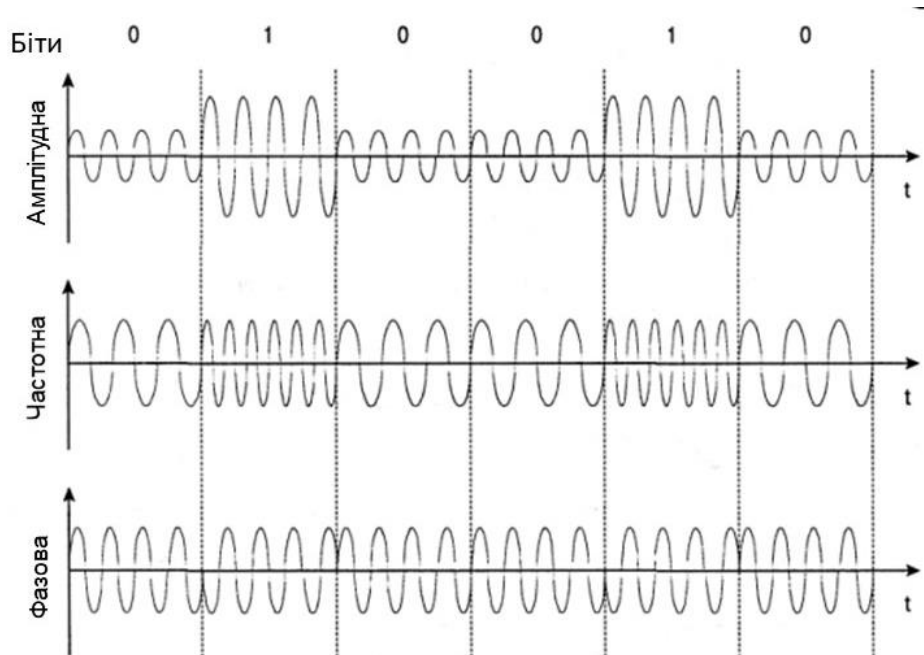


Рис. 6 Амплітудна, частотна та фазова маніпуляція цифрового сигналу з послідовним однорозрядним кодом

Двійкова фазова маніпуляція

Якщо цифровий сигнал представлено у однорозрядному послідовному коді (рис. 3), тоді маніпуляція відбувається таким чином (рис.7):

- коли значення біта, який передається, дорівнює «0», то початкова фаза модульованого приймає значення $90^0 (+\pi/2)$
- коли значення біта, який передається, дорівнює «1», то початкова фаза модульованого приймає значення $-90^0 (-\pi/2)$

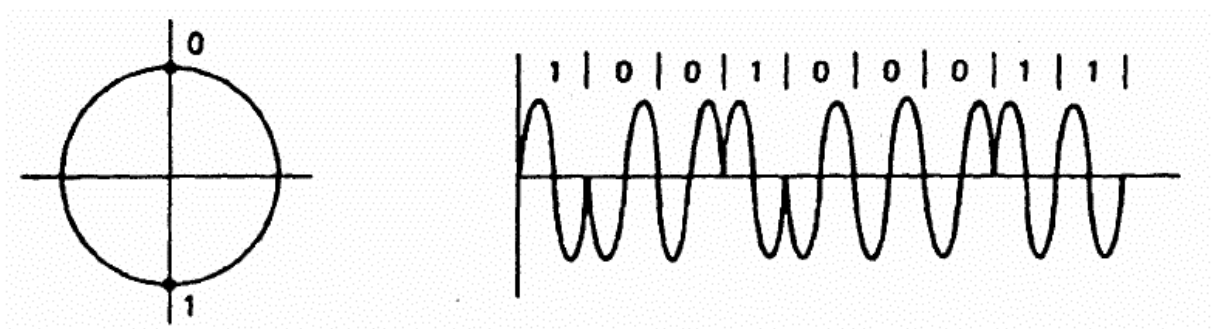


Рис 7. Фазова маніпуляція цифрового сигналу з послідовним однорозрядним кодом

Такий спосіб маніпуляції має назву **двійкова фазова маніпуляція**. В зарубіжній технічній літературі для нього використовують назву *Binary phase-shift keying (BPSK)*.

Фазова маніпуляція є найбільш надійним способом передавання послідовного цифрового коду: ймовірність помилки в розпізнаванні «0» або «1» у порівнянні з іншими видами маніпуляції у фазовій маніпуляції є найнижчою. Саме тому фазова маніпуляція найчастіше застосовується для передавання цифрової інформації скрізь радіоефір. У системі цифрового супутникового радіо ХМ Radio використовується якраз фазова маніпуляція, проте замість двійкової використовують квадратурну фазову маніпуляцію, яку буде розглянуто в наступному розділі. Процес маніпуляції виконує спеціальний пристрій – імпульсно-кодовий модулятор ІКМ (див. рис 1 лекції 5). Зворотну функцію отримання цифрового сигналу з фазо-маніпульованого сигналу виконує імпульсно-кодовий модулятор ІКДМ, який встановлено в ресивері.

Квадратурна фазова маніпуляція

Якщо цифровий сигнал представлено у дворозрядному послідовному коді (рис. 4), тоді для фазової маніпуляції використовують чотири значення початкової фази (рис.8):

- коли значення двійкового коду, який передається, дорівнює «00», то початкова фаза модульованого приймає значення 135° ($+3\pi/4$)
- коли значення двійкового коду, який передається, дорівнює «01», то початкова фаза модульованого приймає значення 45° ($+\pi/4$)
- коли значення двійкового коду, який передається, дорівнює «10», то початкова фаза модульованого приймає значення -135° ($-3\pi/4$)
- коли значення біта, який передається, дорівнює «11», то початкова фаза модульованого приймає значення -45° ($-\pi/4$)

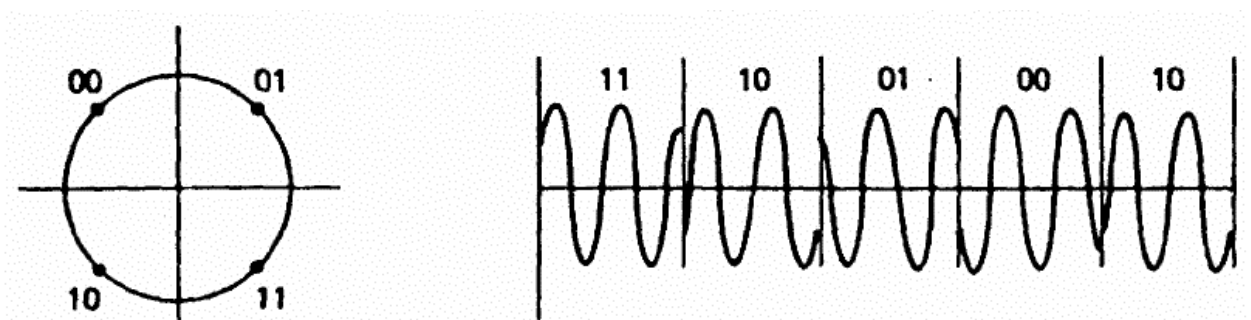


Рис. 8 Квадратурна фазова маніпуляція

Такий спосіб маніпуляції має назву «**квадратурна фазова маніпуляція**» й він буде вивчатися на старших курсах. Англійська назва цього методу *Quadrature phase-shift keying (QPSK)*.

Мультиплексне передавання цифрових даних

Можна значно підвищити швидкість передавання цифрових даних, якщо об'єднати послідовний та паралельний (рис.3) способи передавання в одне ціле. Таке об'єднання зображено на рис. 9 і носить назву **мультиплексне передавання даних**. Мультиплексне передавання побудовано таким чином, що по окремим каналам дані передаються послідовно, але таких каналів декілька. Передавання даних по кожному з цих каналів відбувається незалежно один від одного, проте дані передаються блоками. Наприклад, йде передача слова, яке складається з двох байт 1011110000101010. Це й є блок. Розбиваємо цей блок на чотири частини (пакети) по 4 біти кожен 1011, 1100, 0010 та 1010. Тепер по чотирьом паралельним каналам послідовним кодом передаємо ці пакети. Скрізь один канал – один пакет. На стороні приймача з отриманих чотирьох пакетів знову складаємо передане слово 1011110000101010.



Рис. 9 Мультиплексне передавання даних

В процесі мультиплексного передавання відсутня розфазіровка, тобто немає того чинника, який обмежує швидкість паралельного передавання. Проте дані передаються блоками, і доки до приймача не дійшли всі частини (пакети) попереднього блока, кожна з яких передається окремо по одній з паралельних ліній, передавання нового блоку даних не розпочнеться.

Мультиплексування з ортогональним частотним поділом каналів

Мультиплексне передавання даних використовується в системі цифрового супутникового радіо XM Radio для передавання цифрових даних скрізь радіоефір. Природньо, виникає таке питання: відомо, що передавання скрізь радіоефір відбувається послідовним кодом, тобто маємо лише одну лінію послідовного передавання даних; тоді де ж взяти інші лінії, необхідні для мультиплексування? Все дуже просто і відповідь ми вже маємо з

попередніх лекцій. Задля організації паралельних ліній використовується частотний поділ каналів, з яким ми познайомилися в лекції 4.

В окремому діапазоні частот формують набір опорних частот (рис. 10) – це як набір окремих радіостанцій (див. рис. 6 лекції 3), на яких інформація передається незалежно одна від одної. Цей набір формують на основі строгих математичних співвідношень: опорні частоти знаходяться на однаковій відстані друг від друга, а ця відстань Δf пов'язана з тривалістю імпульсів та блоків, які передаються. Ці частоти, якщо вони вибрані правильно, мають властивість ортогональності, тому й називаються ортогональними. А частотний поділ каналів в цьому випадку є **ортогональним частотним поділом каналів**. В англійській мові цей спосіб називають **OFDM** – *Orthogonal Frequency-Division Multiplexing*. Що таке ортогональність, та як її використовують – про це ви дізнаєтесь в інших курсах.

По кожному з цих частотних каналів дані передаються послідовним кодом з використанням фазової модуляції. Блок даних на стороні передавача розділяється на окремі пакети, кожний з яких передається по своєму частотному каналу (на своїй опорній частоті – одній з зазначеного набору). Всі пакети одного блоку передаються паралельно, а потім всі вони збираються разом на стороні приймача, після чого відновлюється весь блок в цілому. Якщо блок відновився успішно, починається передавання наступного блоку.

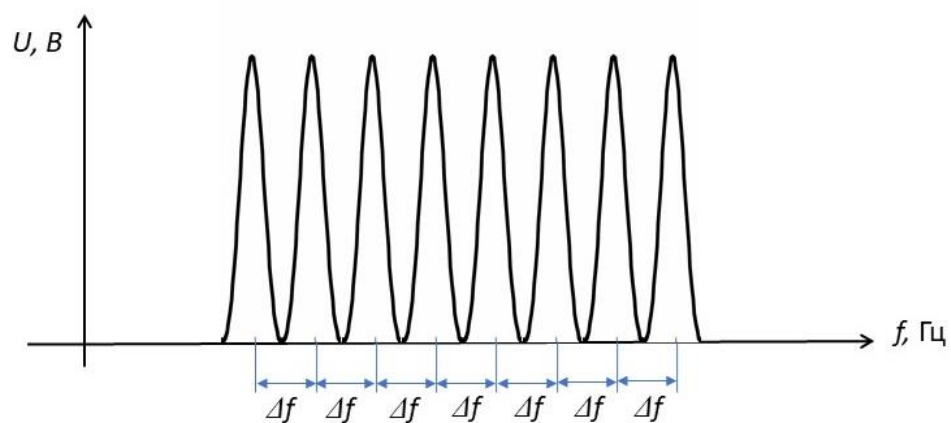


Рис. 10 Набір ортогональних частот

Резюмуючи усе вище сказане відзначимо наступне. В системі цифрового супутникового радіо XM Radio використовують мультиплексування з ортогональним частотним поділом каналів **OFDM**. По кожному з ортогональних каналів передавання даних проводиться з використанням квадратурної фазової маніпуляції **QPSK**. Таким чином, цифрове супутникове радіо XM Radio використовує кодування, яке позначається як **OFDM-QPSK**.

Кодування **OFDM-QPSK** широко використовується в багатьох інших інформаційних технологіях, зокрема на основі цього кодування побудовані мережі Wi-Fi та цифрове супутникове телебачення. Саме через це ми й розглянули цей вид кодування, а знайомство з ним допоможе нам у вивченні інших інформаційних технологій.

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Поняття динамічного діапазону.
2. Визначення необхідної кількості рівнів квантування. Динамічний діапазон АЦП.
3. Паралельне та послідовне передавання даних.
4. Модуляція та маніпуляція сигналів.
5. Двійкова фазова маніпуляція.
6. Квадратурна фазова маніпуляція.
7. Мультиплексне передавання цифрових даних.
8. Мультиплексування з ортогональним частотним поділом каналів.

Лекція № 7

Тема: Комп'ютерні мережі. Частина перша**Зміст**

Введення в комп'ютерні мережі: визначення, класифікація та призначення	2
Лінії і канали передавання даних в мережах	3
Основні визначення	3
Провідні лінії зв'язку	3
Кабельні лінії зв'язку	3
Бездротові (радіоканали наземного і супутникового зв'язку) канали передавання даних	5
Радіоканали передавання даних для локальних мереж	6
Способи передавання даних	6
Методи передавання даних на фізичному рівні	6
Режими передавання та обміну даних	7
Методи передавання на канальному рівні	7
Відкриті системи і модель OSI	8
Мережеві протоколи, інтерфейси, стеки протоколів	8
Еталонна модель мережевої взаємодії OSI з семи рівнів	10
Контрольні питання за темою	12
Рівень модуля	12
Рівень курсу	12

Джерела:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

<https://rutracker.org/forum/viewtopic.php?t=4956359>

Введення в комп'ютерні мережі: визначення, класифікація та призначення

Комп'ютерні мережі – це сукупність вузлів, об'єднаних каналами передавання даних. Комп'ютерні мережі також називають інформаційно-обчислювальними мережами, обчислювальними мережами або мережами передавання даних. Вузлом в мережі є комп'ютер, наприклад персональний комп'ютер, сервер, роутер, смартфон, контролер – тобто будь-який пристрій, який під'єднано до мережі. Залежно від відстані між комп'ютерами розрізняють наступні типи мереж:

- локальні мережі LAN (Local Area Networks) – комп'ютерні мережі, розташовані в межах невеликої обмеженої території (будівлі або в сусідніх будинках);
- територіальні комп'ютерні мережі, які охоплюють значний географічний простір. До територіальних мереж можна віднести міські, регіональні та національні мережі.
- глобальні мережі (WAN - Wide Area Network).

В цій лекції ми розглянемо локальні комп'ютерні мережі, основне призначення яких – забезпечити доступ до поділюваних або мережевих ресурсів, даних і програм. Ці ресурси є загальними, вони використовуються спільно. Розглянемо їх.

Локальні мережі забезпечують:

1. Розподіл даних (Data Sharing). Дані зберігаються на центральному сервері і можуть бути доступні на робочих станціях, тому на кожному робочому місці не треба мати накопичувачі для зберігання однієї і тієї ж інформації.
2. Розподіл інформаційних і технічних ресурсів (Resource Sharing):
 - диски та інші зовнішні пристрої пам'яті;
 - каталоги (папки) і файли;
 - принтери, мережеві пристрої та таке інше.
3. Розподіл програм (Software Sharing). Всі користувачі мережі можуть спільно мати доступ до програм, які централізовано встановлюються в мережі.
4. Обмін повідомленнями так інші види комунікації: чати, телефонія, електронна пошта тощо.

Локальні комп'ютерні мережі, в залежності від способів взаємодії комп'ютерів в них, можна розділити на централізовані і однорангові мережі. Централізовані локальні мережі будуються на основі архітектури "клієнт-сервер", яка передбачає виділення в мережі "серверів" і "клієнтів". Однорангові мережі засновані на рівноправній (peer-to-peer, точка-точка) моделі взаємодії комп'ютерів, в якій кожен комп'ютер може бути як сервером, так і клієнтом.

Лінії і канали передавання даних в мережах

Основні визначення

Для побудови комп'ютерних мереж застосовуються лінії зв'язку, які використовують різні фізичні середовища. В якості фізичного середовища в комунікаціях використовуються: метали (в основному мідь), надпрозоре скло (кварц) або пластик та радіоефір.

Лінії зв'язку або лінії передавання даних – це проміжна апаратура і фізичне середовище, по якій передаються інформаційні сигнали (дані). В одній лінії зв'язку можна утворити кілька **каналів зв'язку** (віртуальних або логічних каналів), наприклад, шляхом частотного або часового поділу каналів.

Канал передавання даних – це вся сукупність засобів двостороннього обміну даними, які включають в себе лінії зв'язку та апаратуру передавання (прийому) даних. Канали передавання даних пов'язують між собою джерела інформації та приймачі інформації.

Залежно від фізичного середовища лінії зв'язку можна розділити на:

- провідні лінії зв'язку без ізолюючих і екрануючих опліток;
- кабельні лінії зв'язку, де для передавання сигналів використовуються такі лінії зв'язку як кабелі "вита пара", коаксіальні кабелі або оптоволоконні кабелі;
- бездротові (радіоканали наземного та супутникового зв'язку) лінії зв'язку, що використовують для передавання сигналів електромагнітні хвилі, які розповсюджуються по радіоефіру.

Провідні лінії зв'язку

Провідні (повітряні) лінії зв'язку використовуються для передавання телефонних і телеграфних сигналів, а також для передавання цифрових даних.

По провідних лініях зв'язку можуть бути організовані аналогові і цифрові канали передавання даних. Швидкість передавання по провідних лініях "простої старої телефонної лінії" (POST - Primitive Old Telephone System) є дуже низькою. Крім того, до недоліків цих ліній відносяться мала перешкодозахищеність і можливість простого несанкціонованого підключення до мережі.

Кабельні лінії зв'язку

Кабельні лінії зв'язку мають досить складну структуру. Кабель складається з провідників, укладених в кілька шарів ізоляції. У комп'ютерних мережах використовуються три типи кабелів.

Вита пара (twisted pair) – кабель зв'язку, який представляє собою виту пару мідних проводів (або кілька пар проводів), укладених в екрановану оболонку. Пари проводів скручуються між собою з метою зменшення

наведень. Вита пара є досить перешкодостійкою. Існує два типи цього кабелю: неекранована кручена пара і екранована кручена пара.

Характерним для цього кабелю є простота монтажу. Даний кабель є найдешевшим і найпоширенішим видом зв'язку, який знайшов широке застосування в локальних мережах з архітектурою Ethernet, побудованих по топології типу "зірка". Кабель підключається до мережевих пристроїв за допомогою з'єднувача RJ45.

Кабель використовується для передавання даних на швидкості 10 Мбіт/с, 100 Мбіт/с і 1 Гбіт/с. Вита пара зазвичай використовується для зв'язку на відстань не більше кількох сотень метрів. До недоліків кабелю "вита пара" можна віднести можливість простого несанкціонованого підключення до мережі.

Коаксіальний кабель (coaxial cable) – це кабель з центральним мідним проводом, який оточений шаром ізолюючого матеріалу для того, щоб відокремити центральний провідник від зовнішнього провідного екрану (мідної обплетення або шар алюмінієвої фольги). Зовнішній провідний екран кабелю покривається ізоляцією.

Існує два типи коаксіального кабелю: тонкий коаксіальний кабель діаметром 5 мм і товстий коаксіальний кабель діаметром 10 мм. У товстого коаксіального кабелю загасання менше, ніж у тонкого. Вартість коаксіального кабелю перевищує номінальну вартість виті пари і виконання монтажу мережі складніше, ніж виті парою. Коаксіальний кабель застосовується, наприклад, в локальних мережах з архітектурою Ethernet, побудованих по топології типу "загальна шина".

Коаксіальний кабель є більш перешкодозахищеним, ніж вита пара і знижує власне випромінювання. Пропускна здатність – 50-100 Мбіт/с. Допустима довжина лінії зв'язку – кілька кілометрів. Несанкціоноване підключення до коаксіального кабелю складніше, ніж до виті пари.

Кабельні оптоволоконні канали зв'язку. Оптоволоконний кабель (fiber optic) – це оптичне волокно на кремнієвій або пластмасовій основі, укладене в матеріал з низьким коефіцієнтом заломлення світла, який закритий зовнішньою оболонкою.

Оптичне волокно передає сигнали тільки в одному напрямку, тому кабель складається з двох волокон. На передавальному кінці оптоволоконного кабелю потрібно перетворення електричного сигналу в світловий, а на приймальному кінці зворотне перетворення.

Основна перевага цього типу кабелю – надзвичайно високий рівень перешкодозахищеності і відсутність випромінювання. Несанкціоноване підключення дуже складне. Швидкість передавання даних обчислюється гігабайтами в секунду. Основні недоліки оптоволоконного кабелю – це

складність його монтажу, невелика механічна міцність і чутливість до іонізуючого випромінювання.

Бездротові (радіоканали наземного і супутникового зв'язку) канали передавання даних

Радіоканали наземного (радіорелейного і стільникового) і супутникового зв'язку утворюються за допомогою передавача і приймача радіохвиль і відносяться до технології бездротового передавання даних.

Радіорелейні канали передавання даних або «радіорелейні канали зв'язку» складаються з послідовності станцій, які є ретрансляторами. Зв'язок здійснюється в межах прямої видимості, дальності між сусідніми станціями – до 50 км. Цифрові радіорелейні лінії зв'язку (ЦРРС) застосовуються в якості регіональних і місцевих систем зв'язку і передавання даних, а також для зв'язку між базовими станціями стільникового зв'язку.

Супутникові канали передавання даних. В супутникових системах використовуються антени надвисокого діапазону частот для прийому радіосигналів від наземних станцій і ретрансляції цих сигналів назад на наземні станції. У супутникових мережах використовуються три основні типи супутників, які знаходяться на геостаціонарних орбітах, середніх або низьких орбітах. Супутники запускаються, як правило, групами. Рознесені один від одного вони можуть забезпечити охоплення практично всієї поверхні Землі. Робота супутникового каналу передавання даних представлена на рисунку.

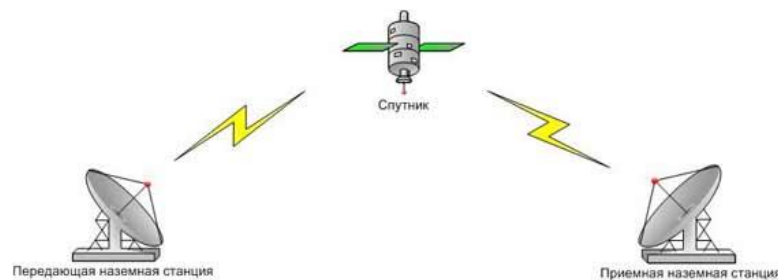


Рис. 1. Супутникові канали передавання даних

Доцільніше використовувати супутниковий зв'язок для організації каналу зв'язку між станціями, розташованими на дуже великих відстанях, а також для забезпечення можливості обслуговування абонентів в самих важкодоступних точках. Пропускна здатність висока – кілька десятків Мбіт/с.

Стільникові канали передавання даних. Радіоканали стільникового зв'язку будуються за тими ж самими принципами, що і стільникові телефонні мережі. Стільниковий зв'язок – це бездротова телекомунікаційна система, що складається з мережі наземних базових приймально-передавальних станцій і стільникового комутатора (або центру комутації мобільного зв'язку).

Базові станції підключаються до центру комутації, який забезпечує зв'язок, як між базовими станціями, так і з іншими телефонними мережами і з

глобальною мережею Інтернет. За виконуваними функціями центр комутації є дуже схожим до звичайної автоматичної телефонної станції провідного зв'язку.

Радіоканали передавання даних для локальних мереж

Стандартом безпроводного зв'язку для локальних мереж є технологія Wi-Fi. Wi-Fi забезпечує підключення в двох режимах: точка-точка (для підключення двох комп'ютерів) і інфраструктурне з'єднання (для підключення декількох вузлів до однієї точки доступу). Швидкість обміну даними – до 300 Мбіт/с за інфраструктурного з'єднання.

Радіоканали передавання даних Bluetooth – це технологія передавання даних на короткі відстані (не більше 10 м) і може бути використана для створення домашніх мереж. Швидкість передавання даних не перевищує 2,5 Мбіт/с.

Способи передавання даних

Методи передавання даних на фізичному рівні

Пересилання даних в комп'ютерних мережах від одного комп'ютера до іншого здійснюється послідовно, біт за бітом. Фізично біти даних передаються по каналах передавання даних у вигляді аналогових або цифрових сигналів. Сукупність засобів (ліній зв'язку, апаратури передавання і прийому даних), що служить для передавання даних в мережах, називається каналом передавання даних.

Залежно від форми переданої інформації канали передавання даних можна розділити на аналогові (безперервні) та цифрові (дискретні).

Оскільки апаратура передавання і прийому даних працює з даними в дискретному вигляді (тобто одиницям і нулям даних відповідають дискретні електричні сигнали), то задля їх передавання скрізь аналогові канали потрібне перетворення дискретних даних в аналогові (модуляція).

Під час прийому таких аналогових даних необхідно зворотне перетворення – демодуляція. Модуляція/демодуляція – процеси перетворення цифрової інформації в аналогові сигнали і навпаки (див. лекцію 4). З використанням модуляції інформація представляється синусоїдальним сигналом тієї частоти, яку добре пропускає канал передавання даних.

До способів модуляції відносяться:

- ✓ амплітудна модуляція;
- ✓ частотна модуляція;
- ✓ фазова модуляція.

Модуляцію та маніпуляцію цифрових сигналів ми детально розглянули у попередніх лекціях (лекції 4 та 6).

Для передавання дискретних сигналів скрізь цифровий канал без застосування модуляції використовується кодування:

- потенційне,
- імпульсне.

В потенційному способі одиниці відповідає один рівень напруги, а нулю – інший. В імпульсному кодуванні для представлення цифр використовуються імпульси однієї, або різної полярності. Потенційне або імпульсне кодування застосовується на каналах високої якості, а модуляція на основі синусоїдальних сигналів – переважно в тих випадках, коли канал сильно спотворює сигнали.

Зазвичай модуляція використовується під час передавання даних скрізь аналогові телефонні канали зв'язку, які були розроблені для передавання голосу в аналоговій формі і тому погано підходять для передавання імпульсів.

Режими передавання та обміну даних

Залежно від способів синхронізації канали передавання даних можна розділити на **синхронні** і **асинхронні**. Синхронізація полягає в тому, що передавальний вузол передає приймальному вузлу спеціальний сигнал, який вказує, коли треба починати прийом даних. Синхронне передавання даних вимагає додаткової лінії зв'язку для передавання синхронізуючих імпульсів. Передавання бітів передавальною станцією і їх прийом приймаючої станцією здійснюється в моменти появи синхроімпульсів.

За асинхронного передавання даних додаткової лінії зв'язку не потрібно. У цьому випадку передавання даних здійснюється блоками фіксованої довжини (байтами). Синхронізація здійснюється додатковими бітами (старт-бітами і стоп-бітами), які передаються перед переданим байтом та після нього.

Під час обміну даними між вузлами застосовуються три режими функціонування каналу:

- **симплексне** (односпрямоване) передавання (телебачення, радіо);
- **напівдуплексне** (прийом/передавання інформації здійснюється по черзі);
- **дуплексне** (двонаправлене), кожен вузол одночасно передає і приймає дані (наприклад, переговори по телефону).

Методи передавання на каналному рівні

Перш ніж послати дані в комп'ютерну мережу, вузол-відправник розбиває дані на невеликі блоки, звані пакетами даних. На вузлі-одержувачі пакети накопичуються і шикуються в належному порядку для відновлення початкового вигляду.

У складі будь-якого пакета повинна бути наступна інформація:

- 1) Дані або інформація, призначена для передавання по мережі.

2) Адреса, що вказує місце призначення пакета. Кожен вузол мережі має адресу. Крім того, адресу має і додаток (програма або процес), який має отримати дані. Адреса програми необхідна для того, аби ідентифікувати, якому саме з додатків належить пакет.

3) Керуючі коди – це інформація, яка визначає обсяг і тип пакета. Керуючі коди включають в себе також коди перевірки помилок і іншу інформацію.

Існує три принципово різні схеми комутації в комп'ютерних мережах:

- ✓ комутація каналів;
- ✓ комутація повідомлень;
- ✓ комутація пакетів.

За комутації каналів встановлюється постійне з'єднання між передавальною та приймаючою стороною у вигляді безперервного фізичного каналу з послідовно з'єднаних окремих ліній зв'язку для прямого передавання даних між вузлами. Потім повідомлення передається по створеному каналу.

Комутація повідомлень – це процес пересилки даних, що включає прийом, зберігання, вибір вихідного напрямку і подальшу передачу блоків повідомлень (без розбивки на пакети). За комутації повідомлень блоки повідомлень передаються послідовно від одного проміжного вузла до іншого з тимчасової буферизацією їх на пристроях пам'яті кожного вузла, доки не досягнуть адресата. До цього ж нова передача може початися тільки після того, як весь блок буде прийнятий. Помилка під час передавання спричинить нове повторне передавання всього блоку.

Передавання пакетів здійснюється аналогічно передаванню повідомлень, але оскільки розмір пакету значно менший ніж блок повідомлення, то досягається швидка його обробка проміжним комунікаційним обладнанням. Тому канал передавання даних зайнятий лише під час передавання пакету, а по його завершенню звільняється для передавання інших пакетів. Шлюзи та маршрутизатори, приймають пакети від вузлів-відправників та на підставі адресної інформації передають їх один одному, а в кінцевому рахунку – станції призначення. Даний вид передавання даних є стандартом для мережі Інтернет.

Відкриті системи і модель OSI

Мережеві протоколи, інтерфейси, стеки протоколів

Комп'ютерні мережі, як правило, складаються з різного обладнання різних виробників. Тому без прийняття всіма виробниками загальноприйнятих правил побудови комп'ютерів та мережевого устаткування забезпечити нормальне функціонування мереж було б неможливо.

Тобто, для забезпечення нормальної взаємодії цього обладнання потрібен єдиний уніфікований стандарт, який визначав би алгоритм передавання інформації в комп'ютерних мережах. В сучасних комп'ютерних мережах роль такого стандарту виконують мережеві протоколи.

У зв'язку з тим, що описати єдиним протоколом всю взаємодію пристроїв в мережі не представляється можливим, то доцільно розділити процес мережевої взаємодії на ряд концептуальних рівнів (модулів) і визначити функції для кожного модуля окремо, а також порядок їх взаємодії, застосувавши метод декомпозиції.

Використовуючи метод декомпозиції та багаторівневий підхід набір модулів (кожен з яких вирішує окреме специфічне завдання) впорядковують за рівнями, і це створює ієрархію. В результаті цього процес мережевої взаємодії можна представити у вигляді ієрархічно організованого набору модулів.

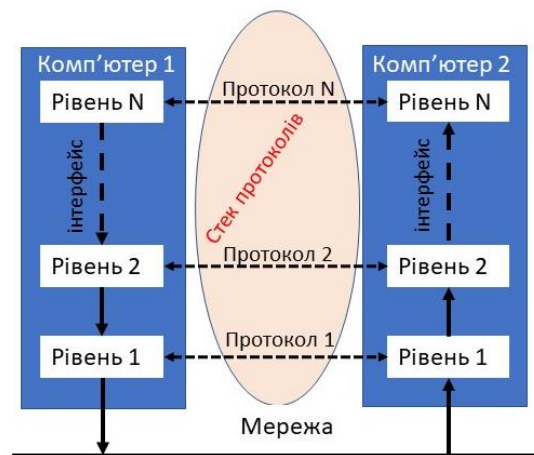


Рис. 2. Мережеві рівні

Багаторівневе представлення засобів мережевої взаємодії має свою специфіку, пов'язану з тим, що в процесі обміну повідомленнями беруть участь дві сторони, тобто необхідно організувати узгоджену роботу двох ієрархій, які працюють на різних комп'ютерах.

Обидва учасники мережевого обміну повинні прийняти багато угод. Угоди повинні бути прийнятні для всіх рівнів, починаючи від найнижчого рівня передачі бітів до найвищого, що реалізує сервіс для користувача. Декомпозиція передбачає чітке визначення функції кожного рівня і інтерфейсів між рівнями.

Взаємодія однойменних функціональних рівнів по горизонталі здійснюється за допомогою протоколу. Протоколом називається набір правил і методів взаємодії однойменних функціональних рівнів об'єктів мережевого обміну.

Взаємодії функціональних рівнів по вертикалі здійснюється через інтерфейси. Інтерфейс визначає набір функцій, які нижчий рівень надає вищому рівневі.

Таким чином, механізм передавання будь-якого пакета інформації скрізь мережу від клієнтської програми, яка працює на першому комп'ютері, до клієнтській програмі, що працює на другому комп'ютері, можна умовно представити у вигляді послідовної пересилки цього пакета зверху вниз від верхнього рівня, що забезпечує взаємодію з призначеним для користувача додатком, до нижнього рівня, який організовує інтерфейс з мережею, трансляцію пакета на комп'ютері №2 і зворотню передачу до верхнього рівня вже на комп'ютері №2.

Комунікаційні протоколи можуть бути реалізовані як програмно, так і апаратно. Протоколи нижніх рівнів часто реалізуються комбінацією програмних і апаратних засобів, а протоколи верхніх рівнів, як правило, – чисто програмними засобами. Протоколи реалізуються не тільки комп'ютерами, але і іншими мережевими пристроями – концентраторами, мостами, комутаторами, маршрутизаторами та т.і. Залежно від типу пристроїв в ньому повинні бути вбудовані засоби, що реалізують той чи інший набір протоколів.

Ієрархічно організований набір протоколів, достатній для організації взаємодії вузлів в мережі, називається стеком комунікаційних протоколів. В мережі Інтернет базовим набором протоколів є стек протоколів TCP / IP.

Еталонна модель мережевої взаємодії OSI з семи рівнів

Еталонна модель OSI (Open System Interconnection - OSI), розроблена в 1984 році Міжнародною організацією зі стандартизації (International Organization of Standardization - ISO), є визначальним документом концепції розробки відкритих стандартів для організації з'єднання систем. Відкрита система – це система, доступна для взаємодії з іншими системами відповідно до прийнятих стандартів. Еталонна модель "Взаємодія відкритих систем" була розроблена з метою спрощення взаємодії пристроїв в мережах.

Еталонна модель являє собою рекомендації (розробникам мереж і протоколів) для побудови стандартів сумісних онлайнових програмних продуктів, і служить базою для виробників під час розробки сумісного мережевого обладнання. Рекомендації стандарту повинні бути реалізовані як в апаратурі, так і в програмних засобах комп'ютерних мереж.

Еталонна модель OSI визначає сім рівнів взаємодії систем в мережах з комунікацією пакетів, дає їм стандартні імена і вказує, які функції повинен виконувати кожен рівень. Кожен рівень функціонує незалежно від вищих та нижчих рівнів. Кожен рівень може спілкуватися з безпосередньо сусіднім

рівнем, проте він повністю ізольований від прямого звернення до наступних рівнів.

Еталонна модель OSI описує тільки системні засоби взаємодії, що реалізуються операційною системою, системними утилітами, системними апаратними засобами.

Відповідно до семирівневої еталонної моделі мережева система представляється прикладними процесами і процесами взаємодії абонентів. Останні розбиваються на сім функціональних рівнів:

- 1)прикладний,
- 2)представницький (рівень представлення даних),
- 3)сеансовий,
- 4)транспортний,
- 5)мережевий,
- 6)канальний
- 7)фізичний.

Під час запуску на комп'ютері будь-якої програми, для функціонування якої потрібен діалог з мережею, ця програма викликає відповідний протокол прикладного рівня мережевого програмного забезпечення. Прикладний рівень формує повідомлення стандартного формату. Зазвичай повідомлення складається з заголовка і поля даних.

Заголовок містить службову інформацію, яку необхідно передати скрізь мережу прикладному рівню комп'ютера-адресата, щоб повідомити йому, яку роботу треба виконати. Поле даних містить інформацію, необхідну для виконання цієї роботи.

Після формування повідомлення прикладний рівень направляє його вниз представницькому рівню. Представницький рівень, на підставі інформації, отриманої із заголовка прикладного рівня, виконує необхідні дії і додає до повідомлення власну службову інформацію – заголовок представницького рівня, в якому містяться вказівки для представницького рівня машини-адресата.

Отримане повідомлення відправляється вниз сеансовому рівню, який додає свій заголовок і т.д. до фізичного рівня, який передає сформований повідомлення по лініях зв'язку. До цього моменту повідомлення має заголовки всіх рівнів.

Коли повідомлення надходить в комп'ютер-адресат, воно приймається фізичним рівнем і послідовно передається вгору з рівня на рівень. Причому кожен рівень аналізує і обробляє заголовок свого рівня, виконуючи відповідні даному рівню функції, а потім видаляє цей заголовок і передає повідомлення рівню.

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Визначення, класифікація та призначення комп'ютерних мереж.
2. Лінії і канали передавання даних в мережах: основні визначення.
3. Кабельні лінії зв'язку комп'ютерних мереж.
4. Бездротові канали передавання даних.
5. Методи передавання даних на фізичному рівні в комп'ютерних мережах.
6. Режимы передавання та обміну даних.
7. Методи передавання на канальному рівні в комп'ютерних мережах.
8. Мережеві протоколи, інтерфейси та стеки протоколів.
9. Еталонна модель мережевої взаємодії OSI.

Лекція № 8

Тема: Мережі. Частина друга**Зміст**

Мережеві топології	2
Шинна топологія.....	2
Топологія типу "зірка"	3
Топологія "кільце"	4
Топологія Token Ring	5
Методи доступу і протоколи передачі даних в локальних мережах ...	6
Методи доступу до середовища передачі даних	7
Виявлення зіткнень.....	7
Передавання маркера в локальних мережах	8
Введення в глобальні мережі	8
Глобальна мережа Інтернет	8
Структура та основні принципи побудови мережі Інтернет	9
Способи доступу до глобальної мережі Інтернет.....	11
Адресація в глобальній мережі Інтернет	13
Уніфіковані локатори ресурсів	15
World Wide Web	15
Контрольні питання за темою.....	17
Рівень модуля	17
Рівень курсу	17

Джерела:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.

<https://rutracker.org/forum/viewtopic.php?t=4956359>

Мережеві топології

Всі комп'ютери в локальній мережі з'єднані лініями зв'язку. Геометричне розташування ліній зв'язку щодо вузлів мережі і фізичне підключення вузлів до мережі називається фізичною топологією. Залежно від топології розрізняють мережі: шинної, кільцевої, зоряної, ієрархічної та довільної структури.

Розрізняють фізичну і логічну топологію. Логічна і фізична топології мережі незалежні один від одного. Фізична топологія – це геометрія побудови мережі, а логічна топологія визначає напрямки потоків даних між вузлами мережі і способи передачі даних.

В теперішній час в локальних мережах використовуються наступні фізичні топології:

- ✓ фізична "шина" (bus);
- ✓ фізична "зірка" (star);
- ✓ фізичне "кільце" (ring);
- ✓ фізична "зірка" і логічне "кільце" (Token Ring).

Шинна топологія

Мережі з шинної топологією використовують лінійний моноканал (коаксіальний кабель) передачі даних, на кінцях якого встановлюються термінатори. Кожен комп'ютер підключається до коаксіального кабелю з допомогою Т-роз'єму (Т-коннектор). Дані від передавального вузла мережі передаються по шині в обидві сторони, відбиваючись від кінцевих термінаторів. Термінатори запобігають віддзеркалення сигналів, тобто використовуються для гасіння сигналів, які досягають кінців каналу передачі даних.

Таким чином, інформація надходить на всі вузли, але приймається тільки тим вузлом, якому вона призначається. У топології «логічна шина» середовище передачі даних використовуються спільно і одночасно всіма ПК мережі, а сигнали від ПК поширюються одночасно у всіх напрямках по середовищу передачі. Оскільки передача сигналів в топології фізична шина є широкомовною, тобто сигнали поширюються одночасно у всіх напрямках, то логічна топологія даної локальної мережі є логічною шиною.

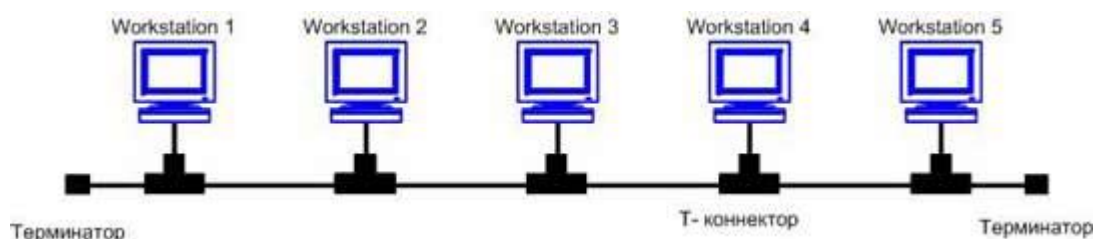


Рис. 3. Шинна топологія

Дана топологія застосовується в локальних мережах з архітектурою Ethernet (класи 10Base-5 і 10Base-2 для товстого і тонкого коаксіального кабелю відповідно).

Переваги мереж шинної топології:

- ✓ відмова одного з вузлів не впливає на роботу мережі в цілому;
- ✓ мережу легко налаштовувати і конфігурувати;
- ✓ мережу стійка до несправностей окремих вузлів.

Недоліки мереж шинної топології:

- ✓ розрив кабелю може вплинути на роботу всієї мережі;
- ✓ обмежена довжина кабелю і кількість робочих станцій;
- ✓ важко знайти дефекти з'єднань.

Топологія типу "зірка"

У мережі побудованій по топології типу "зірка" кожна робоча станція з'єднується кабелем (витою парою) до концентратора або хабу (hub). Концентратор забезпечує паралельне з'єднання ПК і, таким чином, всі комп'ютери, підключені до мережі, можуть спілкуватися один з одним.

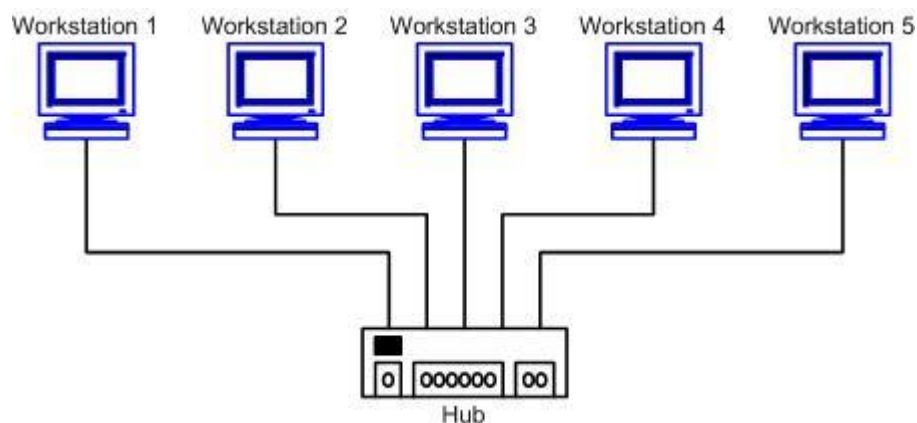


Рис. 4. Топологія "зірка"

Дані від передавальної станції мережі передаються через хаб по всіх лініях зв'язку всім ПК. Інформація надходить на всі робочі станції, але приймається тільки тими станціями, яким вона призначається. Оскільки передача сигналів в топології фізична зірка є широкомовною, тобто сигнали від ПК поширюються одночасно у всіх напрямках, то логічна топологія даної локальної мережі є логічною шиною.

Дана топологія застосовується в локальних мережах з архітектурою 10Base-T Ethernet.

Переваги мереж топології зірка:

- ✓ легко підключити новий ПК;
- ✓ є можливість централізованого управління;

- ✓ мережу стійка до відмови окремих ПК і до розривів з'єднання окремих ПК.

Недоліки мереж топології зірка:

- ✓ відмова хаба впливає на роботу всієї мережі;
- ✓ великі витрати кабелю.

Топологія "кільце"

В мережі з топологією кільце все вузли з'єднані каналами зв'язку в нерозривне кільце (необов'язково окружність), по якому передаються дані. Вихід одного ПК з'єднується з входом іншого ПК. Почавши рух з однієї точки, дані, в кінцевому рахунку, потрапляють на його початок. Дані в колі завжди рухаються в одному і тому ж напрямку.

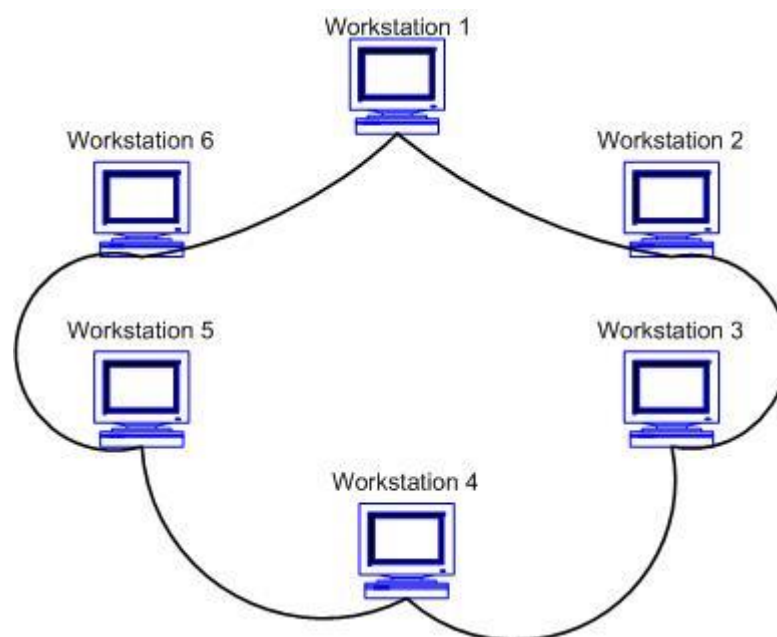


Рис. 5. Топологія "кільце"

Приймаюча робоча станція розпізнає і отримує тільки її адресоване повідомлення. У мережі з топологією типу фізичне кільце використовується маркерний доступ, який надає станції право на використання кільця у визначеному порядку. Логічна топологія даної мережі – логічне кільце. Дану мережу дуже легко створювати і налаштовувати.

До основного недоліку мереж топології кільце є те, що пошкодження лінії зв'язку в одному місці або відмова ПК призводить до непрацездатності всієї мережі.

Як правило, в чистому вигляді топологія "кільце" не застосовується через свою ненадійність, тому на практиці застосовуються різні модифікації кільцевої топології.

Топологія Token Ring

Ця топологія заснована на топології "фізична кільце з підключенням типу зірка". У даній топології все робочі станції підключаються до центрального концентратора (Token Ring) як в топології фізична зірка. Центральний концентратор – це інтелектуальний пристрій, яке за допомогою перемичок забезпечує послідовне з'єднання виходу однієї станції зі входом іншої станції.

Іншими словами, за допомогою концентратора кожна станція з'єднується тільки з двома іншими станціями (попередньою та подальшою станціями). Таким чином, робочі станції пов'язані петлею кабелю, по якій пакети даних передаються від однієї станції до іншої, і кожна станція ретранслює ці послані пакети. У кожній робочій станції є для цього приймально-передавальний пристрій, який дозволяє управляти проходженням даних в мережі. Фізично така мережа побудована за типом топології "зірка".

Концентратор створює первинне (основне) і резервне кільця. Якщо в основному кільці відбудеться обрив, то його можна обійти, скориставшись резервним кільцем, так як використовується чотирижильний кабель. Відмова станції або обрив лінії зв'язку робочої станції не тягне за собою відмову мережі як в топології кільце, тому що концентратор відключає несправну станцію і замикає кільце передачі даних.

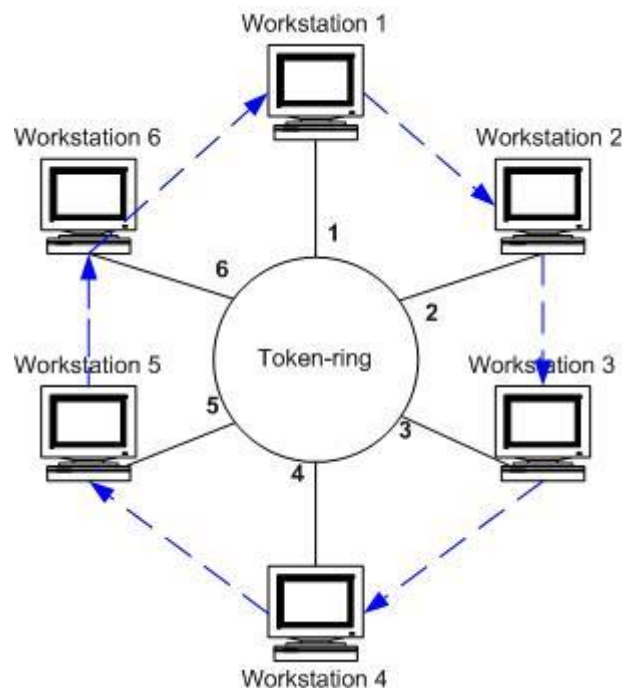


Рис. 6. Топологія Token Ring

В архітектурі Token Ring маркер передається від вузла до вузла по логічному кільцю, створеному центральним концентратором. Таке маркерне передавання здійснюється у фіксованому напрямку (напрямок руху маркера і

пакетів даних представлено на малюнку стрілками синього кольору). Станція, що володіє маркером, може відправити дані іншій станції.

Для передачі даних робочі станції повинні спочатку дочекатися приходу вільного маркера. У маркері міститься адреса станції, яка послала цей маркер, а також адреса тієї станції, якій він призначається. Після цього відправник передає маркер наступній в мережі станції для того, щоб і та могла відправити свої дані.

Один з вузлів мережі (зазвичай для цього використовується файл-сервер) створює маркер, який відправляється в кільце мережі. Такий вузол виступає в якості активного монітора, який стежить за тим, щоб маркер ні загублений або зруйнований.

Переваги мереж з топологією Token Ring:

- ✓ топологія забезпечує рівний доступ до всіх робочих станцій;
- ✓ висока надійність, оскільки мережа стійка до несправності окремих станцій і до розривів з'єднання окремих станцій.

Недоліки мереж топології Token Ring: велика витрата кабелю і відповідно дорога розводка ліній зв'язку.

Методи доступу і протоколи передачі даних в локальних мережах

В різних мережах застосовуються різні мережеві протоколи (протоколи передачі даних) для обміну даними між робочими станціями.

У 1980 році в Міжнародному інституті інженерів з електротехніки та електроніки (Institute of Electronics Engineers – IEEE) був організований комітет №802 зі стандартизації локальних мереж. Комітет 802 розробив сімейство стандартів IEEE802.х, які містять рекомендації з проектування нижніх рівнів локальних мереж.

Стандарти сімейства IEEE802.х охоплюють тільки два нижніх рівні семирівневої моделі OSI – фізичний та канальний, оскільки саме ці рівні найбільшою мірою відображають специфіку локальних мереж. Старші ж рівні, починаючи з мережного, в значній мірі мають загальні риси, як для локальних, так і глобальних мереж.

До найбільш поширених методів доступу відносяться: Ethernet, ArcNet і Token Ring, які реалізовані відповідно в стандартах IEEE802.3, IEEE802.4 і IEEE802.5. Крім того, для локальних мереж, що працюють на оптичному волокні, американським інститутом по стандартизації ANSI був розроблений стандарт FDDI, що забезпечує швидкість передачі даних 100 Мбіт / с.

У цих стандартах канальний рівень поділяється на два підрівні, які називаються рівнями:

- управління логічним каналом (LCC - Logical Link Control);

- управління доступом до середовища (MAC - Media Access Control).

Рівень управління доступом до середовища передачі даних (MAC) з'явився з тої причини, що в локальних мережах використовується колективне середовище передачі даних. У сучасних локальних мережах набули поширення кілька протоколів рівня MAC, що реалізують різні алгоритми доступу до середовища. Ці протоколи повністю визначають специфіку таких технологій локальних мереж, як Ethernet, Fast Ethernet, Gigabit Ethernet, Token Ring, FDDI.

Після того, як доступ до середовища отриманий, нею може скористатися більш високий канальний рівень – рівень LCC, організуючи передачу логічних одиниць даних, кадрів інформації, з різним рівнем якості транспортних послуг.

Методи доступу до середовища передачі даних

В локальних мережах, що використовують середовище передачі даних (наприклад, локальні мережі з топологією шина і фізична зірка), актуальним є доступ робочих станцій до цього середовища через те що, якщо два ПК починають одночасно передавати дані, то в мережі відбувається зіткнення.

Для того щоб уникнути цих зіткнень необхідно мати спеціальний механізм, здатний вирішити цю проблему. Шинний арбітраж – це механізм покликаний вирішити проблему зіткнень. Він встановлює правила, за якими робочі станції визначають, коли середовище вільне, і чи можна передавати дані.

Існують два методи шинного арбітражу в локальних мережах:

- виявлення зіткнень;
- передача маркера.

Виявлення зіткнень

Коли в локальних мережах працює метод виявлення зіткнень, комп'ютер спочатку слухає, а потім передає. Якщо комп'ютер чує, що передачу веде хтось інший, він повинен зачекати закінчення передачі даних і потім зробити повторну спробу.

У тій ситуації, якщо два комп'ютери, передають в один і той же час, система виявлення зіткнень вимагає, щоб передавальний комп'ютер продовжував прослуховувати канал і, виявивши на ньому чужі дані, припиняв передачу, намагаючись відновити її через невеликий (випадковий) проміжок часу.

Прослуховування каналу до передачі називається "прослуховування опорної" (carrier sense), а прослуховування під час передачі – виявлення зіткнень (collision detection). Комп'ютер, що діє таким чином, використовує

метод, який має назву "виявлення зіткнень з прослуховуванням опорної", скорочено CSMA/CD.

Передавання маркера в локальних мережах

Системи з передаванням маркера працюють інакше. Для того, аби передати дані, комп'ютер спочатку повинен отримати дозвіл. Це означає, що він повинен "зловити" циркулюючий в мережі пакет даних спеціального виду, званий маркером. Маркер переміщається по замкнутому колу, минаючи по черзі кожен мережевий комп'ютер.

Кожен раз, коли комп'ютер повинен послати повідомлення, він ловить і тримає маркер у себе. Як тільки передача закінчилася, він посилає новий маркер в подорож далі по мережі. Такий підхід дає гарантію, що будь-який комп'ютер рано чи пізно отримає право спіймати і утримувати маркер до того часу, поки його власна передача не закінчиться.

Введення в глобальні мережі

Глобальні обчислювальні мережі – це комп'ютерні мережі, які об'єднують локальні мережі та окремі комп'ютери, віддалені один від одного на великі відстані. Насьогодні всеосяжною глобальною мережею стала мережа Інтернет. Спроби створення інших мереж залишилися в минулому. Тому, замість глобальних обчислювальних мереж фактично можна говорити тільки про одну – мережу Інтернет.

Глобальна мережа Інтернет

Інтернет – це безліч комп'ютерів (хостів) і різних мереж, об'єднаних мережею на базі протоколів зв'язку TCP/IP. Комп'ютери, підключені до мережі Інтернет, можуть мати будь-які апаратні та програмні платформи, але при цьому всьому вони повинні підтримувати стек протоколів (сімейство протоколів) зв'язку TCP/IP. Єдиного власника і центру управління мережею Інтернет не існує.

Інтернет почав своє існування з мережі ARPANet в 1969 році. Ця комп'ютерна мережа із застосуванням технології комутації пакетів була створена в США за завданням військового відомства США як високонадійна мережа передавання даних. У 1983 році ARPANet розділилася на дві мережі, одна – MILNET стала частиною оборонної мережі передавання даних США, інша була використана для з'єднання академічних та дослідницьких центрів, яка поступово розвивалася і в 1990 році трансформувалася в Інтернет.

Компоненти мережі Інтернет об'єднуються в загальну ієрархію.

В даний час в інтернеті існує досить велика кількість служб, що забезпечують роботу широкого спектру ресурсів. Найбільш відомими серед них є:

- служба DNS, або система доменних імен, що забезпечує можливість використання для адресації вузлів мережі мнемонічних імен замість числових адрес;
- електронна пошта (E-mail), що забезпечує можливість обміну повідомленнями одну людину з одним або декількома абонентами;
- служба IRC, призначена для підтримки текстового спілкування в реальному часі (chat);
- телеконференції, або групи новин (Usenet), що забезпечують можливість колективного обміну повідомленнями;
- служба FTP – система файлових архівів, забезпечує зберігання та пересилання файлів різних типів;
- служба Telnet, призначена для управління віддаленими комп'ютерами в термінальному режимі;
- World Wide Web (WWW, W3, «Всесвітня павутина») – гіпертекстова (гіпермедіа) система, призначена для інтеграції різних мережевих ресурсів в єдиний інформаційний простір;
- потокове мультимедіа.

Список послуг, що надаються безперервно зростає.

Структура та основні принципи побудови мережі Інтернет

Internet – всесвітня інформаційна комп'ютерна мережа, що є об'єднанням безлічі регіональних комп'ютерних мереж і комп'ютерів, які обмінюють один з одним інформацією по каналах громадських телекомунікацій (виділеним телефонним аналоговим і цифровим лініям, оптичним каналам зв'язку і радіоканалами, в тому числі супутникових лініям зв'язку).

Інформація в Internet зберігається на серверах. Сервери мають свої адреси і управляються спеціалізованими програмами. Вони дозволяють пересилати пошту і файли, проводити пошук в базах даних і виконувати інші завдання.

Обмін інформацією між серверами мережі виконується по високошвидкісних каналах зв'язку (виділеним телефонним лініям, оптоволоконним і супутниковим каналам зв'язку). Доступ окремих користувачів до інформаційних ресурсів Internet зазвичай здійснюється через провайдера або корпоративну мережу.

Провайдер – постачальник мережевих послуг, – особа або організація, яка надає послуги з підключення до комп'ютерних мереж. В якості провайдера

виступає деяка організація, що має модемний пул для з'єднання з клієнтами і виходу у всесвітню мережу.

Основними осередками глобальної мережі є локальні комп'ютерні мережі. Якщо деяка локальна мережа безпосередньо підключена до глобальної, то і кожна робоча станція цієї мережі може бути підключена до Інтернету.

Існують також комп'ютери, які безпосередньо підключені до глобальної мережі. Вони називаються хост - комп'ютерами (host - господар). Хост - це будь-який комп'ютер, який є постійною частиною Internet, тобто з'єднаний за допомогою Internet-протоколу з іншим хостом, який в свою чергу, з'єднаний з іншим, і так далі.

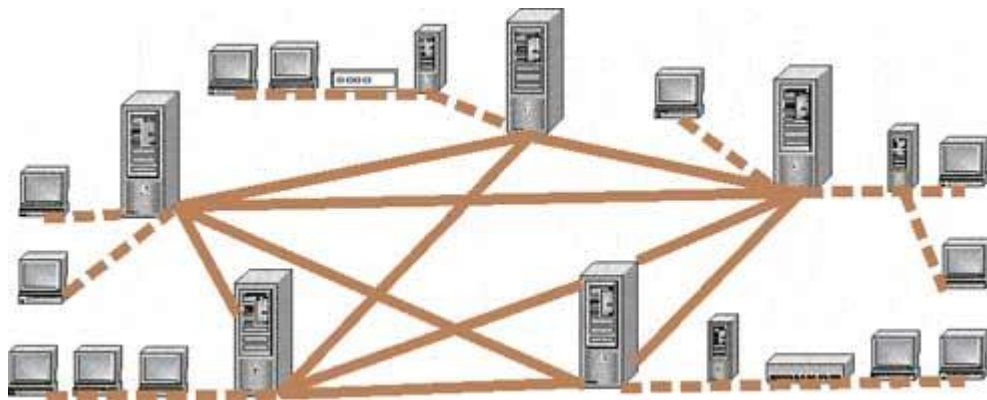


Рис. 1. Структура глобальної мережі Internet

Для під'єднання ліній зв'язку до комп'ютерів використовуються спеціальні електронні пристрої, які називаються мережевими платами, мережними адаптерами, модемами та т.і.

Практично всі послуги Internet побудовані за принципом клієнт-сервер. Вся інформація в Інтернет зберігається на серверах. Обмін інформацією між серверами здійснюється через високошвидкісні канали зв'язку або «магістралі». Сервери, об'єднані високошвидкісними магістралями, становлять базову частину мережі Інтернет.

Окремі користувачі підключаються до мережі через комп'ютери місцевих постачальників послуг Інтернету, Internet-провайдерів (Internet Service Provider – ISP), які мають постійне підключення до Інтернет. Регіональний провайдер, підключається до більш великому провайдеру національного масштабу, що має вузли в різних містах країни. Мережі національних провайдерів об'єднуються в мережі транснаціональних провайдерів або провайдерів першого рівня. Об'єднані мережі провайдерів першого рівня становлять глобальну мережу Internet.

Передача інформації в Інтернет забезпечується завдяки тому, що кожен комп'ютер в мережі має унікальну адресу (IP-адреса), а мережеві протоколи

забезпечують взаємодію різнотипних комп'ютерів, що працюють під управлінням різних операційних систем.

В основному в Інтернет використовується сімейство мережевих протоколів (стек) TCP/IP. На канальному і фізичному рівні стек TCP/IP підтримує технологію Ethernet, FDDI і інші технології. Основою сімейства протоколів TCP/IP є мережевий рівень, представлений протоколом IP, а також різними протоколами маршрутизації. Цей рівень забезпечує переміщення пакетів в мережі і управляє їх маршрутизацією. Розмір пакета, параметри передачі, контроль цілісності здійснюється на транспортному рівні TCP.

Прикладний рівень об'єднує всі служби, які система надає користувачеві. До основних прикладних протоколів відносяться: протокол віддаленого доступу telnet, протокол передачі файлів FTP, протокол передачі гіпертексту HTTP, протоколи електронної пошти: SMTP, POP, IMAP, MIME.

Найбільш поширені в інтернеті протоколи, згруповані приблизно у відповідності до моделі OSI, представлено в таблиці:

Рівень OSI	Протоколи, приблизно відповідають рівню OSI
Прикладний	BGP, DNS, FTP, HTTP, HTTPS, IMAP, LDAP, POP3, SNMP, SMTP, SSH, Telnet, XMPP (Jabber)
Представницький та сеансовий	SSL, TLS
Транспортний	TCP, UDP
Мережевий	EIGRP, ICMP, IGMP, IP, IS-IS, OSPF, RIP
Канальний	Arcnet, ATM, Ethernet, Framerelay, HDLC, PPP, L2TP, SLIP, Token ring

Способи доступу до глобальної мережі Інтернет

Відомі такі способи доступу в Інтернет:

1. Dial-Up (коли комп'ютер користувача підключається до сервера провайдера, використовуючи стаціонарну телефонну лінію зв'язку) – комутований доступ по аналоговій телефонній мережі швидкість передачі даних до 56 Кбіт / с.

2. ISDN – комутований доступ через цифрову телефонну мережу. Головна особливість використання ISDN – це більш висока швидкість передачі інформації, в порівнянні з Dial-Up доступом. Швидкість передачі даних складає 64 Кбіт/с за використання одного і 128 Кбіт/с, за використання двох каналів зв'язку.

3. DSL (Digital Subscriber Line) – сімейство цифрових абонентських ліній, призначених для організації доступу по аналоговій телефонній мережі, використовуючи кабельний модем. Ця технологія (ADSL, VDSL, HDSL, ISDL, SDSL, SHDSL, RADSL під загальною назвою xDSL) забезпечує

високошвидкісне з'єднання до 50 Мбіт/с (фактична швидкість до 2 Мбіт/с). Основною перевагою технологій xDSL є можливість значно збільшити швидкість передачі даних по телефонних дротах без модернізації абонентської телефонної лінії. Користувач отримує доступ в мережу Інтернет із збереженням звичайної роботи телефонного зв'язку.

4. Доступ в Інтернет по виділених лініях (аналогових та цифрових). Доступ по виділеній лінії – це такий спосіб підключення до Інтернет, коли комп'ютер користувача з'єднаний з сервером провайдера за допомогою кабелю (витої пари) і це з'єднання є постійним, тобто некомутованими, і в цьому головна відмінність від звичайного телефонного зв'язку. Швидкість передачі даних до 100 Мбіт/с.

5. Доступ в Інтернет по локальній мережі (Fast Ethernet). Підключення здійснюється за допомогою мережевої карти зі швидкістю передачі даних до 1 Гбіт/с.

6. Доступ в Інтернет по локальній мережі з підключенням через оптичний кабель зі швидкістю передачі даних від 1 Гбіт/с для кінцевого користувача.

7. Супутниковий доступ в Інтернет або супутниковий Інтернет (DirecPC, Europe Online). Супутниковий доступ в Інтернет буває двох видів – асиметричний і симетричний:

- обмін даними комп'ютера користувача з супутником двосторонній;
- запити від користувача передаються на сервер супутникового оператора через будь-яке доступне наземне підключення, а сервер передає дані користувачеві з супутника. Максимальна швидкість прийому даних до 52,5 Мбіт/с (реальна середня швидкість до 3 Мбіт/с).

8. Доступ в Інтернет з використанням каналів кабельної телевізійної мережі ("soax at a home"), швидкість прийому даних від 2 до 56 Мб/сек. В теперішній час відомо дві архітектури передачі даних – це симетрична і асиметрична архітектури.

9. Бездротові технології:

- WiFi;
- WiMax;
- RadioEthernet;
- MMDS;
- LMDS
- мобільний GPRS - Інтернет;
- мобільний CDMA - Internet.

Адресація в глобальній мережі Інтернет

Адреси в Інтернеті можуть бути представлені як послідовністю цифр, так і ім'ям, побудованим за певними правилами. Комп'ютери під час пересилання інформації використовують цифрові адреси, а користувачі в роботі з Інтернетом використовують в основному імена. Таким чином, адреси комп'ютерів мають подвійну систему кодування:

- IP-адреса.
- DNS-адреса (доменна адреса).

Основним протоколом мережі Інтернет є мережевий протокол TCP/IP. Кожен комп'ютер в мережі TCP/IP (підключений до мережі Інтернет) має свою унікальну IP-адресу або IP-номер. IP-адреси, представлені в цифровому вигляді (IP-номера), складаються з чотирьох байтів, тобто з 32-розрядної двійкового числа, яке розділяється на чотири блоки по 8 біт (октети). IP-адресу записують у вигляді чотирьох десяткових чисел, розділених точками. Наприклад, 195.82.54.17. Кожне число не повинно перевищувати двохсот п'ятдесяти шести. IP-адреса включає дві частини: номер мережі і номер вузла (комп'ютера) в мережі. Такий спосіб нумерації дозволяє мати в мережі більше чотирьох мільярдів комп'ютерів. Якщо окремий комп'ютер (хост-комп'ютер) або мережа є складовою частиною мережі Інтернет, то IP-адреса присвоюється спеціальним підрозділом Інтернету. Для окремого комп'ютера або локальної мережі, які вперше підключаються до мережі Інтернет, спеціальна організація, що займається адмініструванням доменних імен, привласнює IP-номери. Розподілом IP адрес займається організація ICANN (Internet Corporation for Assigned Names and Numbers), а в Європі розподілом IP адрес між регіональними провайдерами займається RIPE NCC (Réseaux IP Européens Network Coordination Centre), який знаходиться в Нідерландах. Адреси комп'ютерів всередині мережі визначають адміністратори мережі.

Спочатку в мережі Internet застосовувалися IP-номери, але коли кількість комп'ютерів в мережі стало більшою за 1000, то був прийнятий метод зв'язку імен і IP-номерів, який називається «сервер імені домена» (Domain Name Server, DNS).

Перетворення DNS-адреси в IP-адресу здійснює сервер імені домена DNS (Domain Name Server). Сервер DNS підтримує список імен локальних мереж і комп'ютерів і відповідних їм IP - номерів.

Доменну адресу (DNS-адреса) складається з декількох доменів (буквено-числових позначень), які відокремлюються один від одного крапкою. Доменна адреса побудована на основі ієрархічної класифікації, тобто доменна адреса включає в себе кілька рівнів доменів, наприклад: gorod.dp.ua.

Домен верхнього рівня розташовується в імені правіше, а домен нижнього рівня – лівіше. Користувач мережі Інтернет працює не з IP-

адресами, а тільки з доменними адресами. Кожен рівень в такій системі називається доменом. Типове ім'я домена складається з декількох частин, розташованих в певному порядку і між якими ставлять крапку. Домени відокремлюються один від одного крапками, наприклад: `www.lessons.info` або `ууу.zzz.ua`.

В Інтернеті доменна система імен використовує принцип послідовних уточнень, так само, як і в звичайних поштових адресах – країна, місто, вулиця і будинок, куди слід доставити лист.

Домен верхнього рівня розташовується в імені правіше, а домен нижнього рівня - лівіше. У нашому прикладі домени верхнього рівня `info` і `ua` вказують на те, що мова йде про приналежність сайту `www.lessons.info` до тематичного домену верхнього рівня `info`, а сайту `ууу.zzz.ua` до української (`ua`) частини Інтернету. Але в Україні безліч користувачів Інтернету, і наступний рівень визначає організацію, якій належить дана адреса. У нашому випадку це компанія `zzz`.

Інтернет-адреса цієї компанії - `zzz.ua`. Всі комп'ютери, підключені до Інтернету в цій компанії, об'єднуються в групу, що має таку адресу. Ім'я окремого комп'ютера або мережі кожна компанія вибирає для себе самостійно, а потім реєструє його в тій організації Інтернет, яка забезпечує підключення.

Це ім'я в межах домена верхнього рівня має бути унікальним. Далі йде ім'я хосту `ууу`, таким чином, повне ім'я домену третього рівня: `ууу.zzz.ua`. В імені може бути будь-яке число доменів, але найчастіше використовуються імена з кількістю доменів від трьох до п'яти.

Доменна система адрес гарантує, що у всьому Інтернеті більше не знайдеться іншого комп'ютера з такою самою адресою. Для доменів нижніх рівнів можна використовувати будь-які адреси, але для доменів самого верхнього рівня існує угода.

В системі адрес Інтернету прийняті домени, представлені географічними регіонами. Вони мають ім'я, що складається з двох літер, наприклад:

- Україна – `ua`;
- Франція – `fr`;
- Канада – `ca`;
- США – `us`;
- Росія – `ru`.

Існують і домени, розділені за тематичними ознаками, наприклад:

- Навчальні заклади – `edu`.
- Урядові установи – `gov`.
- Комерційні організації – `com`.

Останнім часом додано нових зон, зокрема: `biz`, `info`, `in`, `.cn` і так далі.

Реєстрація домену здійснюється в обраній користувачем зоні ua, ru, com, net, info і так далі. Залежно від призначення сайту вибирається його зона реєстрації. Для реєстрації сайту бажано вибрати домен другого рівня, наприклад, lessons.info, хоча можна працювати і з доменом третього рівня. Домен другого рівня реєструється у реєстратора - організації що займається адмініструванням доменних імен. Домен третього рівня отримується, як правило, разом з хостингом у хостингової компанії. Ім'я сайту вибирають виходячи з роду діяльності, назви компанії або прізвища власника сайту.

Уніфіковані локатори ресурсів

Для посилань на конкретний ресурс в Internet використовуються не самі доменні імена (які замінюють лише IP-адресу), а уніфіковані локатори ресурсів, звані URL (Universal Resource Locator). URL – це адреса будь-якого ресурсу (документа, файлу) в Internet. Вона вказує, за допомогою якого протоколу слід звертатися до ресурсу, яку програму слід запустити на сервері і до якого конкретного файлу слід звернутися на сервері. Загальний вигляд URL: протокол: // хост-комп'ютер / ім'я файлу.

World Wide Web

В даний час велика частина трафіку в Інтернет припадає на службу World Wide Web (всесвітня павутина). Дана служба використовує протокол HTTP.

Принцип роботи сервісу WWW був розроблений фізиками Тімом Бернес-Лі і Робертом Кайо в європейському дослідницькому центрі CERN (Женева) в 1989 році. В даний час Web-служба Інтернет містить мільйони сторінок інформації з різними видами документів.

Сутність системи World Wide Web (WWW) полягає в застосуванні гіпертекстової моделі до інформаційних ресурсів, розподілених у глобальній мережі. WWW – один з видів сервісу Інтернет. WWW надає можливість роботи з документами, в яких об'єднані текст, графічні зображення, звуки, анімація, що значно полегшує сприйняття інформації. Гіпертекстові документи (Web-сторінки) створюються за допомогою спеціальної мови розмітки гіпертексту HTML (Hyper Text Markup Language).

Система WWW працює за принципом клієнт-сервер. Клієнт – це інтерпретатор HTML, спеціальна програма перегляду, звана WWW-браузер (WWW-browser). WWW-браузер – це прикладна програма, яка взаємодіє з системою WWW, отримує витребувані документи, інтерпретує дані і відображає зміст документів на екрані. Програма-клієнт забезпечує доступ практично до всіх інформаційних ресурсів Інтернет, які зберігаються на серверах.

HTTP (англ. HyperText Transfer Protocol - «протокол передачі гіпертексту») – протокол прикладного рівня передачі даних (спочатку – у вигляді гіпертекстових документів в форматі «HTML», зараз використовується для передачі довільних даних, зокрема – мультимедійні дані). Основою HTTP є технологія «клієнт-сервер», яка передбачає наявність:

- Споживачів (клієнтів), які ініціюють з'єднання і надсилають запит;
- Постачальників (серверів), які очікують з'єднання для отримання запиту, виконують необхідні дії та повертають назад повідомлення з результатом.

Основним об'єктом маніпуляції в HTTP є ресурс, на який вказує URI (Uniform Resource Identifier) в запиті клієнта. Зазвичай такими ресурсами є файли, що зберігаються на сервері, але ними можуть бути логічні об'єкти або щось абстрактне. Особливістю протоколу HTTP є можливість вказати в запиті та відповіді спосіб представлення одного і того ж ресурсу за різними параметрами: формату, кодуванні, мови і таке інше. (Зокрема, для цього використовується HTTP-заголовок).

Для роботи з системою WWW необхідно встановити на своєму комп'ютері одну з програм перегляду Web-сторінок, наприклад, Internet Explorer, Mozilla Firefox, MyIE Web Browser, Opera і т.д. Більшість браузерів надають доступ до інших серверів Інтернету: до FTP-серверів, Gopher-серверів і серверів телеконференцій UseNet.

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Мережеві топології.
2. Шинна топологія локальної мережі.
3. Топологія локальної мережі типу "зірка".
4. Топологія локальної мережі "кільце".
5. Топологія локальної мережі Token Ring.
6. Шинний арбітраж в локальних мережах за методом виявлення зіткнень.
7. Шинний арбітраж в локальних мережах за методом передавання маркера.
8. Структура та основні принципи побудови мережі Інтернет.
9. Адресація в глобальній мережі Інтернет.
10. Структура та основні принципи побудови World Wide Web.

Лекція № 9

Тема: Стільниковий зв'язок за стандартом GSM. Частина перша**Зміст**

Введення в стільникові мережі.....	2
Абоненти і оператори стільникового зв'язку та їх ідентифікація.....	2
Принципи забезпечення мережевого покриття	2
Інфраструктура стільникових мереж.....	4
Взаємодія між операторами зв'язку.....	5
Um-інтерфейс (GSM Air Interface) - початок	5
Частотні діапазони.....	5
Контрольні питання за темою.....	8
Рівень модуля	8
Рівень курсу.....	8

Джерела:

1. Берлин А. Н. Цифровые сотовые системы связи – М.: Эко-Трендз, 2007 – 296с з іл.

Введення в стільникові мережі

В даній лекції ми розглянемо, як побудовані стільникові мережі, та як реалізується мобільний зв'язок засобами цих мереж. Стільниковий зв'язок працює за стандартом **GSM** (Глобальна система мобільного зв'язку, французькою – **Groupe Spécial Mobile**).

Абоненти і оператори стільникового зв'язку та їх ідентифікація

За аналогією з інтернет-провайдерами, послуги стільникового зв'язку надають певні компанії, які найчастіше називаються «операторами». Кожен з них пропонує свій спектр послуг, а також встановлює свої тарифні плани. Найчастіше оператори використовують власне обладнання для побудови основної інфраструктури мережі.

Абонентами мережі в загальному випадку є мобільні станції (**MS** – Mobile Station). В якості **MS** виступає будь-який пристрій, який використовується для під'єднання до стільникової мережі. Найпоширенішим пристроєм, звичайно, є мобільний телефон, або смартфон. Кожний мобільний пристрій має унікальний номер **IMEI** (англ. International Mobile Equipment Identity — міжнародний ідентифікатор мобільного обладнання) – серійний номер мобільного пристрою (п'ятнадцятизначне число, тобто 14 цифр коду плюс 15-та контрольна цифра), який є унікальним для кожного мобільного телефону. Цей номер встановлюється заводом-виробником при виготовленні апарату та служить для точної і повної ідентифікації телефону в мережах форматів GSM. Він автоматично передається апаратом у мережу оператора при підключенні.

Ідентифікація операторів здійснюється за кодом країни (**MCC** – Mobile Country Code) та унікальним кодом мережі всередині країни (**MNC** – Mobile Network Code).

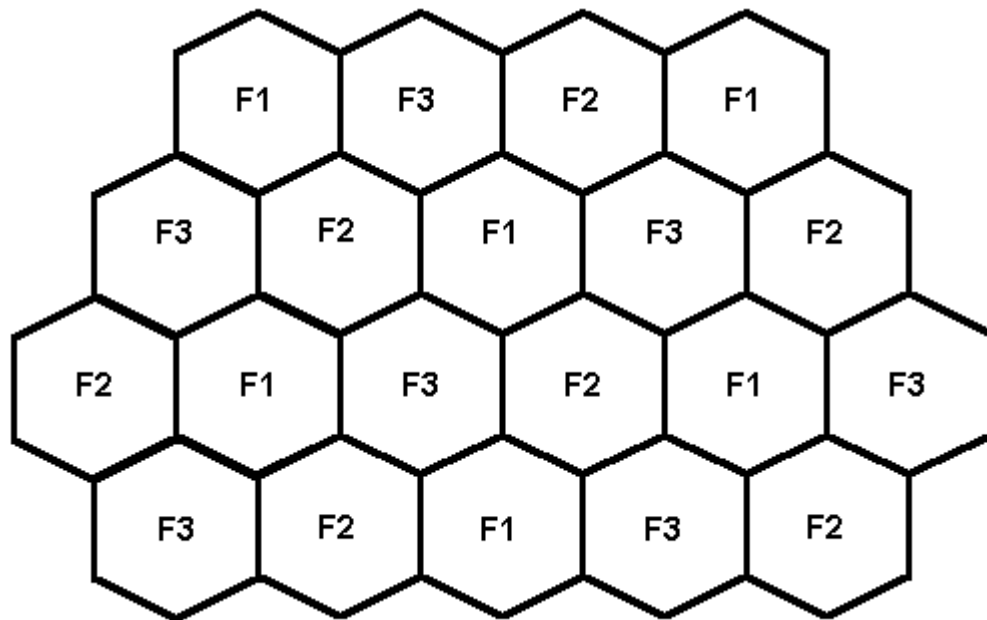
Ідентифікація абонентів здійснюється не за звичним для нас телефонним номером, а за міжнародним ідентифікатором абонента – **IMSI** (International Mobile Subscriber Identity), який записано в SIM-карті абонента, а також в базі даних оператора. Телефонні номери «прив'язуються» до певного **IMSI**, завдяки чому абонент може змінити оператора, зберігши свій номер телефону.

Принципи забезпечення мережевого покриття

Покриття певної місцевості стільниковим зв'язком забезпечується за рахунок розподілу приймально-передавальних пристроїв по її площі. Їх можна бачити на різних будівлях, на вежах і навіть на окремих щоглах. Найчастіше вони являють собою кілька спрямованих антен білого кольору, а також невеличке приміщення, куди тягнуться проводи. В термінології **GSM** такі комплекси називаються базовими станціями (**BTS** – Base Transceiver Station) і

можуть складатися з декількох приймально-передавальних пристроїв – трансиверів (**TRX** – Transmitter/Receiver).

Ключова особливість стільникового зв'язку полягає в тому, що загальна зона покриття ділиться на клітинки – «стільники», що визначаються зонами покриття окремих базових станцій (**BTS**). Через це й виникла назва «стільниковий зв'язок». Кожна базова станція покриває один або кілька секторів, а також має один або кілька приймачів в кожному секторі, кожен з яких випромінює сигнал на своїй частоті. Простіше кажучи, стільник – це одна з клітинок покриття, що має свій унікальний ідентифікатор, званий **CI (Cell ID)**. Стільники можна класифікувати за приблизним радіусом покритої території: макростільники (до 35 км, іноді до 70 км), звичайний стільник (до 5 км), мікростільник (до 1 км), пікостільник (до 300 метрів) і фемтостільник (частіше зустрічаються всередині приміщень, покривають десятки метрів).



F1, F2, F3 – частоти базових станцій

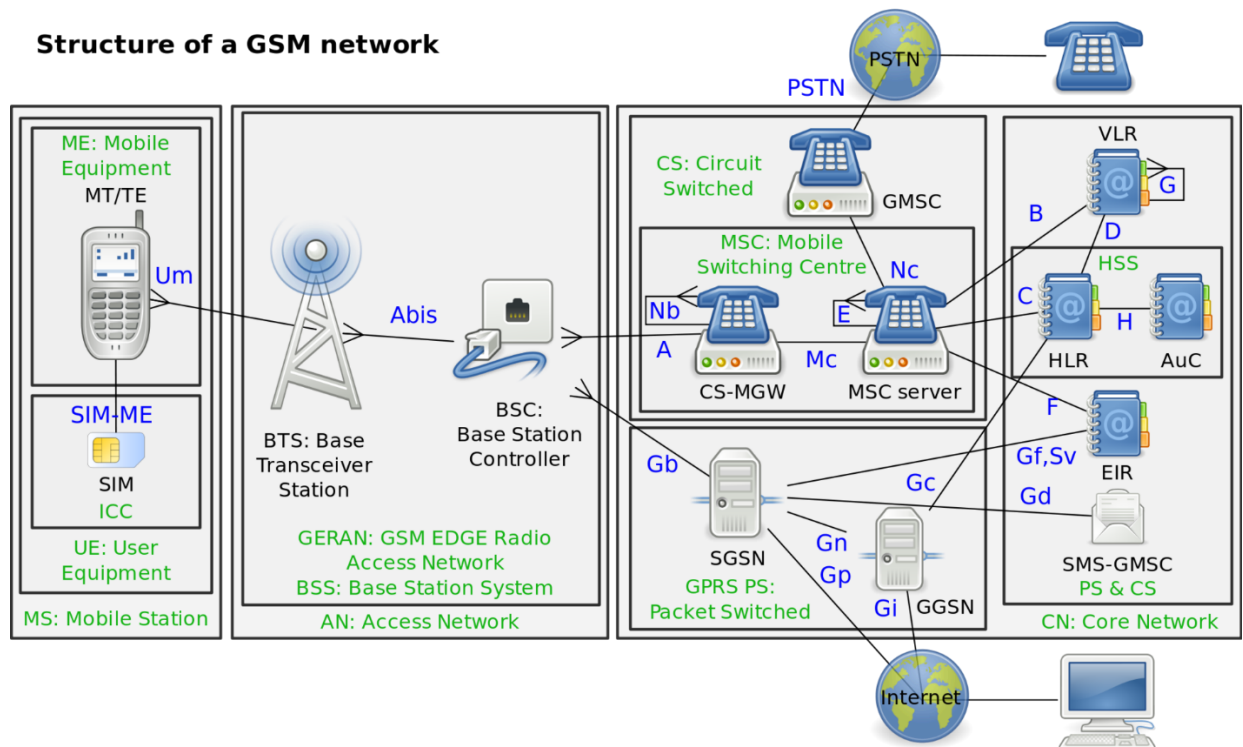
Базові станції, розташовані поблизу, працюють в різних частотних діапазонах, завдяки чому стільники різних операторів можуть частково або майже повністю накладатися один на одного. Сукупність базових станцій, що працюють спільно, називається зоною розташування – **LAC (Location Area Code)**. Всі базові станції обов'язково передають в ефір свої ідентифікаційні дані, такі як MCC, MNC, Cell ID, а також LAC, завдяки чому, мобільні телефони підключається тільки до **BTS** свого оператора. Крім цього, мобільні телефони з певним інтервалом повідомляють мережу про своє поточне місцезнаходження, тобто LAC. Дана процедура називається **Location Update**.

Інфраструктура стільникових мереж

Базові станції не можуть існувати самі по собі, тому, перебуваючи в певному LAC, вони підключаються до контролера базових станцій – **BSC** (Base Station Controller). Контролери, в свою чергу, виконують балансування навантаження, а також беруть активну участь в процесі обміну трафіку між мережею і своїми «підлеглими». В межах одної мережі оператори мають кілька контролерів базових станцій, які підключаються до центру комутації (**MSC** - Mobile Switching Center).

MSC утворює ядро мережевої інфраструктури (Core Network), до якого входять такі основні компоненти:

- **HLR (Home Location Register)** – база даних, що містить персональні дані кожного абонента, включаючи телефонний номер, тарифний план, список підключених послуг, а також інформацію про використання абонентом SIM-карту.
- **VLR (Visitor Location Register)** – тимчасова база даних абонентів, які знаходяться в зоні дії певного центру мобільної комутації. Кожна базова станція в мережі приписана до певного VLR, так що абонент не може бути присутнім в декількох VLR одночасно.
- **AuC (Authentication Center)** – центр аутентифікації абонентів, що виконує перевірку справжності кожної SIM-карти, яка підключається до мережі.
- **SMSC (SMS Center)** – центр обміну короткими текстовими повідомленнями, що займається їх зберіганням і маршрутизацією.
- **GMSC (Gateway MSC)** – шлюз, що надає доступ до мереж провідних міських телефонів. Завдяки цьому елементу можливі дзвінки між абонентами стільникових і міських телефонних мереж.
- **SGSN (Serving GPRS Support Node)** – вузол обслуговування абонентів **GPRS** (General Packet Radio Service), який забезпечує пакетне передавання даних. Сервіс використовує незайняту голосовим зв'язком смугу частот для передавання інформації. Використовується в мобільних пристроях для передачі MMS, WAP-серфінгу та повноцінного з'єднання з Інтернет.



Крім цього, всередині інфраструктури мережі існує рахункова система, де зберігається наш грошовий баланс, списується плата за користування послугами, а також обробляються різні платіжні операції.

Оператор може приєднувати і інші підсистеми до ядра мережі на свій розсуд.

Взаємодія між операторами зв'язку

Мережі різних операторів взаємодіють між собою, завдяки чому абонент оператора А може зателефонувати абоненту оператора В. Називається ця мережа **ЗКС-7** (загальноканальна сигналізація) – англійська аббревіатура **SS7** (Signalling System No.7). Вона працює або на базі спеціальних провідних/безпроводних комунікаційних мереж, або поверх Інтернету. SS7 надає набір протоколів для взаємодії різних операторів. Роумінг теж працює завдяки даній мережі.

Um-інтерфейс (початок)

Частотні діапазони

Будь-яке устаткування в стільникових мережах взаємодіє за допомогою певних інтерфейсів. Обмін даними між базовою станцією і абонентом здійснюється через **Um-інтерфейс** (GSM Air Interface), який є радіоінтерфейсом, отже обмін даними відбувається в процесі приймання/передавання радіохвиль.

Мережі GSM працюють в діапазоні UHF. Відповідно до стандарту 3GPP TS 45.005, в ефірі їм виділено 14 діапазонів, проте в різних країнах використовуються різні діапазони. Розглянемо найбільш поширені:

Характеристики	GSM-850	P-GSM-900	E-GSM-900	DCS-1800	PCS - 1900
Uplink , МГц	824.2 - 849.2	890.0 - 915.0	880.0 - 915.0	1710.2 - 1784.8	1850.2 - 1909.8
Downlink , МГц	869.2 - 893.8	935.0 - 960.0	925.0 - 960.0	1805.2 - 1879.8	1930.2 - 1989.8
ARFCN	128 - 251	1 - 124	975 - 1023, 0 - 124	512 - 885	512 - 810

У мережах GSM абсолютний номер каналу (англ. Absolute radio-frequency channel number, **ARFCN**) являє собою пару частот, що використовуються операторами стільникового зв'язку для прийому і передачі інформації. Список номерів (кодів) каналів визначено в специфікації GSM.

В даній таблиці представлені розрахункові дані частот для прийому і передачі певних номерів ARFCN для різних опорних частот.

Діапазон	Позначення	ARFCN	Частота Uplink, f_{UL}	Частота Downlink, f_{DL}
GSM 500	GSM 450	259-293	$450.6 + 0.2 (n-259)$	$f_{UL}(n)+10$
	GSM 480	306-340	$479.0 + 0.2 (n-306)$	$f_{UL}(n)+10$
GSM 700	GSM 750	438-511	$747.2 + 0.2 (n-438)$	$f_{UL}(n) + 30$
GSM 850	GSM 850	128-251	$824.2 + 0.2 (n-128)$	$f_{UL}(n) + 45$
GSM 900	P-GSM	1-124	$890.0 + 0.2 n$	$f_{UL}(n) + 45$
	E-GSM	0-124 975-1023	$890.0 + 0.2 n$ $890.0 + 0.2 (n-1024)$	$f_{UL}(n) + 45$
	GSM-R	0-124 955-1023	$890.0 + 0.2 n$ $890.0 + 0.2 (n-1024)$	$f_{UL}(n) + 45$
GSM 1800	DCS 1800	512-885	$1710.2 + 0.2 (n-512)$	$f_{UL}(n) + 95$
GSM 1900	PCS 1900	512-810	$1850.2 + 0.2 (n-512)$	$f_{UL}(n) + 80$

P-GSM-900, E-GSM-900 та DCS-1800 використовуються переважно в країнах Європи і Азії. Діапазони GSM-850 і PCS-1900 використовуються в США, Канаді, окремих країнах Латинської Америки і Африки. В Україні операторами стільникового зв'язку використовуються два діапазони частот: P-GSM-900 та DCS-1800.

Кожен виділений під стільникову мережу діапазон ділиться на певну кількість відрізків (зазвичай по 200 КГц), частина з яких називається **Downlink** – тут дані в ефір передають тільки базові станції (BTS), частина – **Uplink**, де віщають тільки телефони (MS). Пари таких відрізків, де один належить Downlink, а інший Uplink, утворюють радіочастотні канали з певними номерами. Ці номери називаються звані **ARFCN** (Absolute radio-frequency channel number). Іншими словами, телефон не може приймати і передавати дані на одній частоті, натомість під час передавання він переключається на частоти Uplink, а під час прийому – на Downlink. Пара частот Uplink та Downlink має один (загальний на дві частоти) номер ARFCN.

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Абоненти і оператори стільникового зв'язку та їх ідентифікація.
2. Принципи забезпечення покриття стільникової мережі.
3. Інфраструктура стільникових мереж.
4. Частотні діапазони мереж стільникового зв'язку.

Лекція № 10

Тема: Стільниковий зв'язок за стандартом GSM**Зміст**

Um-інтерфейс (продовження).....	2
Фізичні канали, поділ множинного доступу.....	2
Логічні канали.....	4
Пакет даних	6
Види пакетів	7
Основні принципи взаємодії MS і BTS	8
Handover.....	10
Кодування розмови.....	10
Безпека і конфіденційність	11
Основні вектори атак.....	11
Ідентифікація абонентів	11
Аутентифікація	12
Шифрування трафіку.....	13
Контрольні питання за темою.....	14
Рівень модуля	14
Рівень курсу	14

Джерела:

1. Берлин А. Н. Цифровые сотовые системы связи – М.: Эко-Трендз, 2007 – 296с з іл.

Um-інтерфейс (продовження)

Фізичні канали, поділ множинного доступу

Множинний доступ ми розглядали у попередній лекції. Це стосувалося доступу багатьох комп'ютерів (множини комп'ютерів) до одного загального ресурсу – мережевої шини. Треба вирішити задачу, яка полягає в тому, щоб всі комп'ютери мали рівні можливості у доступі до мережевої шини задля того, щоб почати передавання даних; а під час передавання необхідно, аби не виникало конфліктів (колізій) через одночасне передавання пакетів даних різними комп'ютерами. Ця задача в локальних мережах з логічною топологією «шина» вирішується шляхом використання методу доступу CSMA/CD ("виявлення зіткнень з прослуховуванням опорної"). Метод CSMA/CD розподіляє час використання шини.

Така ж сама задача стоїть і в мережах стільникового зв'язку, оскільки необхідно забезпечити одночасний безконфліктний обмін даними між однією базовою станцією і багатьма мобільними пристроями.

Потреба в поділі множинного доступу виникла давно і застосовується як в провідних комунікаціях (I2C, USB, Ethernet), так і в бездротових. У стільникових мережах найчастіше використовуються технології **FDMA** (Frequency Division Multiple Access) – частотний поділ, **TDMA** (Time Division Multiple Access) – розподіл за часом та **CDMA** (Code Division Multiple Access) – кодовий розподіл. В мережах другого покоління – GSM використовуються разом перші дві технології: **FDMA** та **TDMA**. CDMA є основою сучасних стільникових мереж, які перевершують GSM як в плані безпеки, так і максимальної швидкості передачі даних. Цю технологію ми наразі не розглядаємо, оскільки для її засвоєння необхідні додаткові знання з кодування та теорії ймовірності.

Для радіосистем існує два основних ресурси – частота і час. Поділ множинного доступу по частотах, коли кожному приймачу і передавача виділяється певна частота, називається **FDMA**. Цей засіб, тобто частотний поділ каналів ми вже вивчали в попередніх лекціях.

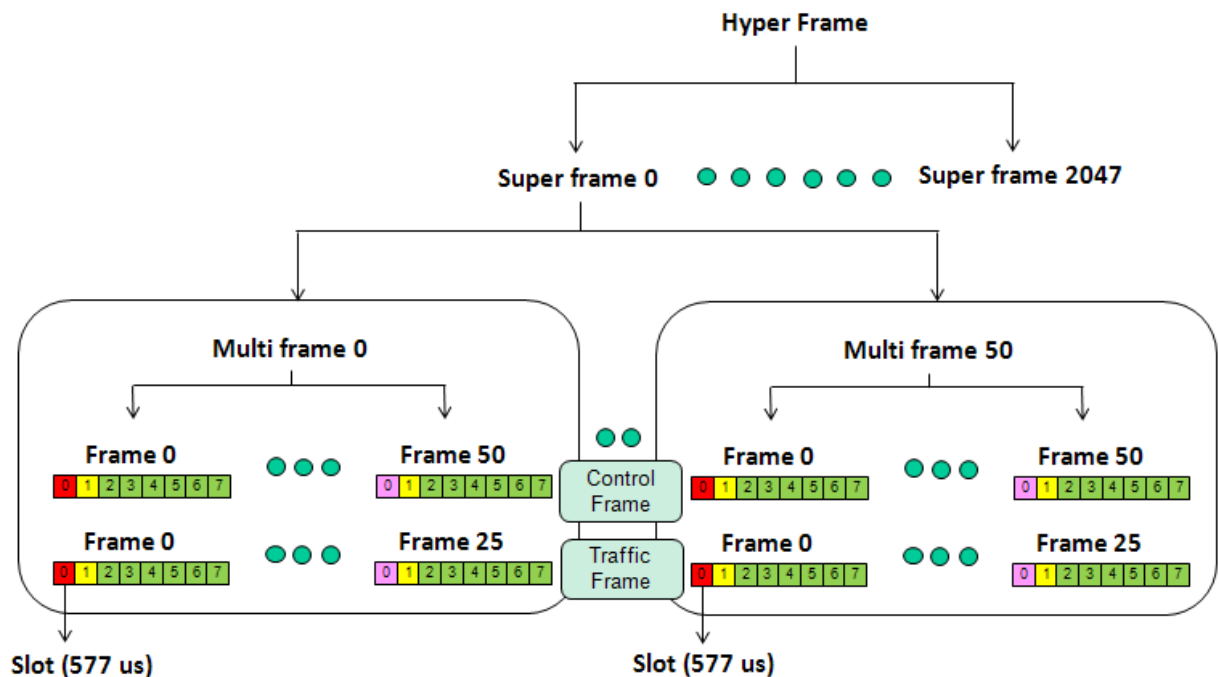
Поділ за часом, коли кожній парі приймач-передавач виділяється весь спектр або більша його частина на виділений відрізок часу, називають **TDMA**.

Частотний поділ множинного доступу FDMA

Принцип даного методу полягає в тому, що доступний частотний спектр поділяється між приймачами і передавачами на рівні або нерівні частотні смуги, частина з яких виділяється під Downlink (трафік від BTS до MS), частина під Uplink (трафік від MS до BTS). Про це ми вже говорили.

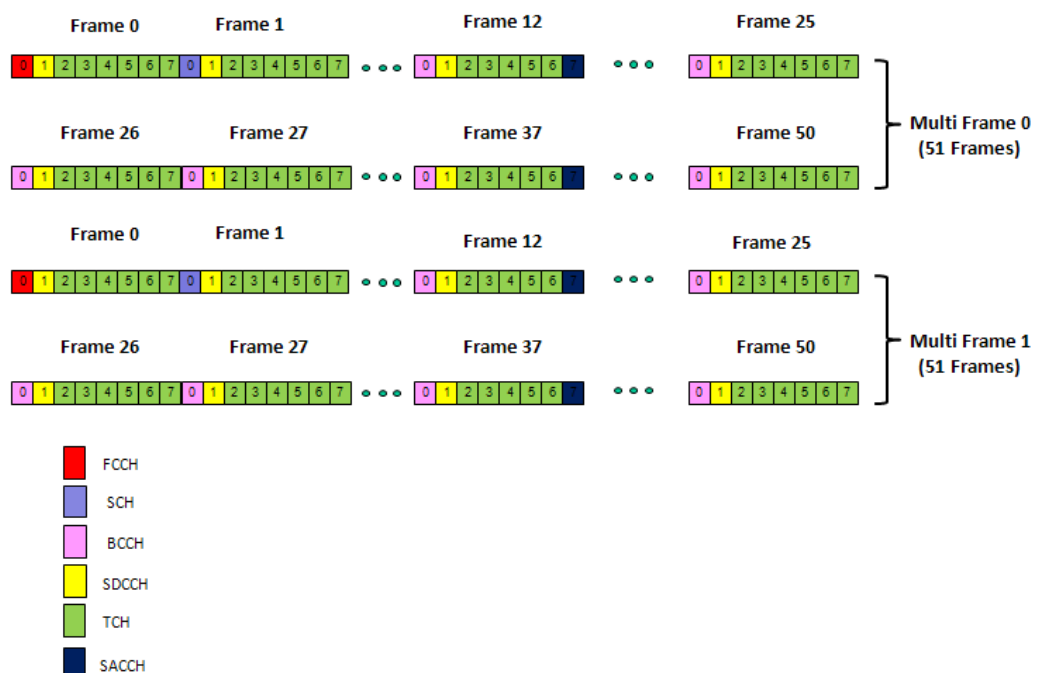
Розподіл множинного доступу за часом TDMA

Разом з частотним поділом (FDMA), в GSM застосовується метод поділу за часом – TDMA. Згідно TDMA, весь потік даних ділиться на фрейми (кадри), а фрейми в свою чергу діляться на кілька таймслотов, які розподіляються між приймально-передавальними пристроями. Отже, телефон може виконувати обмін інформацією з мережею тільки в певні, виділені для нього проміжки часу.

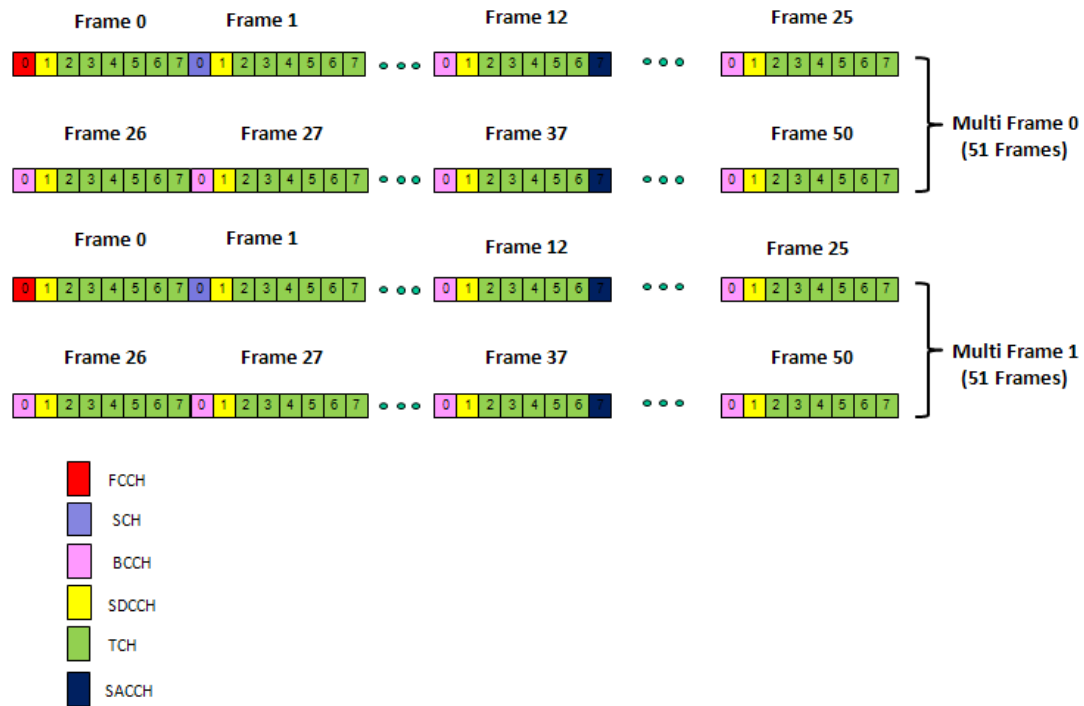


Фрейми об'єднуються в мультифрейми, які бувають двох видів:

Мультифрейм управління – Control Multiframe (містить 51 кадр)



Мультифрейм даних – Traffic Multiframe (містить 26 кадрів)



Мультифрейми утворюють суперфрейми, а вже суперфрейми утворюють гіперфрейми.

В результаті, фізичний канал між приймачем і передавачем визначається частотою, виділеними фреймами і номерами таймслотів в них. Зазвичай базові станції використовують один або кілька каналів ARFCN, один з яких використовується для ідентифікації присутності BTS в ефірі. Перший таймслот (індекс 0) фреймів цього каналу використовується в якості базового службового каналу (**base-control channel** або **beacon**-канал, тобто «маяк»). Залишок каналів ARFCN розподіляється оператором зв'язку на свій розсуд між службовими каналами та каналами даних.

Логічні канали

На основі фізичних каналів формуються логічні. **Um**-інтерфейс має на увазі обмін як користувальницької інформацією, так і службовою. Згідно зі специфікацією GSM, кожному виду інформації відповідає спеціальний вид логічних каналів, що реалізуються за допомогою фізичних каналів:

- канали даних (**TCH** – Traffic Channel),
- канали службової інформації (**CCH** – Control Channel).

Канали даних

Канали даних поділяться на два основних види: **TCH/F** – Full rate канал з максимальною швидкістю до 22,8 Кбіт/с і **TCH/H** – Half rate канал з максимальною швидкістю до 11,4 Кбіт/с. Дані види каналів можуть бути

використані для передавання мови (TCH/FS, TCH/HS) та призначених для користувача даних (TCH/F9.6, TCH/F4.8, TCH/H4.8, TCH/F2.4, TCH/H2.4), наприклад, SMS.

Канали службової інформації

Канали службової інформації діляться на:

Широкомовні (BCH - Broadcast Channels)

- ✓ **FCCH - Frequency Correction Channel (канал корекції частоти).** Надає інформацію, необхідну мобільному телефону для корекції частоти.
- ✓ **SCCH - Synchronization Channel (канал синхронізації).** Надає мобільному телефону інформацію, необхідну для TDMA-синхронізації з базовою станцією (BTS), а також її ідентифікаційні дані станції.
- ✓ **BCCH - Broadcast Control Channel (широкомовний канал службової інформації).** Передає основну інформацію про базову станцію, таку як спосіб організації службових каналів, кількість блоків, зарезервованих для повідомлень надання доступу, а також кількість мультіфреймов (об'ємом по 51 TDMA-кадру) між Paging-запитами.

Канали загального призначення (CCCH - Common Control Channels)

- ✓ **PCCH - Paging Channel.** Paging – це свого роду ring мобільного телефону, що дозволяє визначити його доступність в певній зоні покриття. Даний канал призначений саме для цього.
- ✓ **RACH - Random Access Channel (канал довільного доступу).** Використовується мобільними телефонами для запиту власного службового каналу SDCCCH. Виключно Uplink-канал.
- ✓ **AGCH - Access Grant Channel (канал повідомлень про надання доступу).** На цьому каналі базові станції відповідають на RACH-запити мобільних телефонів, виділяючи SDCCCH, або відразу TCH.

Власні канали (DCCH - Dedicated Control Channels)

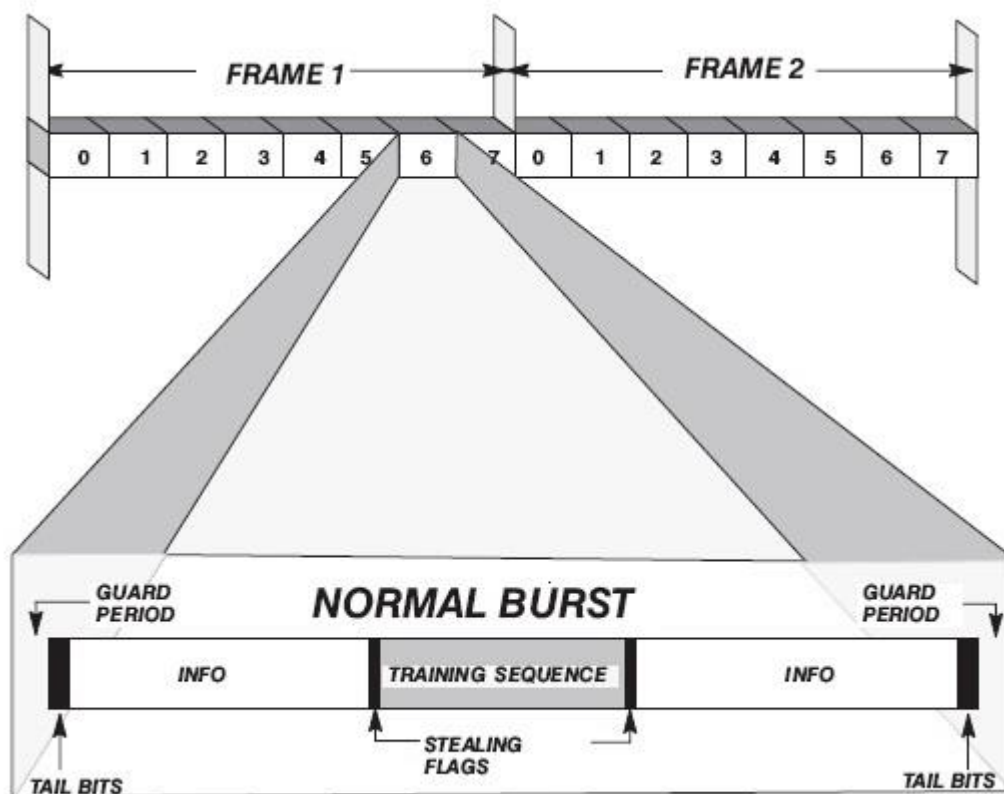
Власні канали, так само як і TCH, виділяються певним мобільних телефонів. Існує кілька підвидів:

- ✓ **SDCCCH - Stand-alone Dedicated Control Channel (Автономний виділений канал управління).** Даний канал використовується для аутентифікації мобільного телефону, обміну ключами шифрування, процедури поновлення розташування (location update), а також для здійснення голосових викликів і обміну SMS-повідомленнями.

- ✓ **SACCH - Slow Associated Control Channel (повільний зв'язаний канал управління).** Використовується під час розмови, або коли вже задіяний канал SDCCH. З його допомогою BTS передає телефону періодичні інструкції про зміну таймінгів і потужності сигналу. У зворотний бік йдуть дані про рівень сигналу (RSSI), як TCH, а також рівень сигналу найближчих базових станцій (BTS Measurements).
- ✓ **FACCH - Fast Associated Control Channel (швидкий зв'язаний канал управління).** Даний канал надається разом з TCH і дозволяє передавати термінові повідомлення, наприклад, під час переходу від однієї базової станції до іншої (**Handover**).

Пакет даних

Дані в ефірі передаються у вигляді послідовностей бітів званої як **пакет даних** або «burst». Пакет даних займає один таймслот. Схематичне представлення пакету виглядає таким чином:



Guard Period (інтервал захисту)

Щоб уникнути виникнення інтерференції (тобто накладення двох пакетів один на одного), тривалість пакету завжди менше тривалості таймслота на певне значення ($0,577 - 0,546 = 0,031$ мс), зване «Guard Period», що перекладається як інтервал захисту. Даний період є свого роду запас часу для компенсації можливих затримок у часі під час передавання сигналу.

Tail Bits (крайні біти)

Дані маркери визначають початок і кінець пакету.

Info (інформація)

Корисне навантаження пакету, наприклад, дані абонентів, телефона розмова або службові данні. Поділена на дві частини.

Stealing Flags (флаг крадіжки)

Ці два біта встановлюються в одиницю, коли обидві частини даних пакету каналу TCH передані по каналу FACH. Один переданий біт замість двох означає, що тільки одна частина пакету передана по FACH.

Training Sequence

Ця частина burst використовується приймачем для визначення фізичних характеристик каналу між телефоном і базовою станцією.

Види пакетів

Кожному логічному каналу відповідають певні види пакетів:

Normal Burst (нормальний пакет)

Послідовності цього типу реалізують канали трафіку (TCH) між мережею та абонентами, а також всі види каналів управління (SCH): CCCH, BCCH і DCCH.

Frequency Correction Burst (пакет корекції частоти)

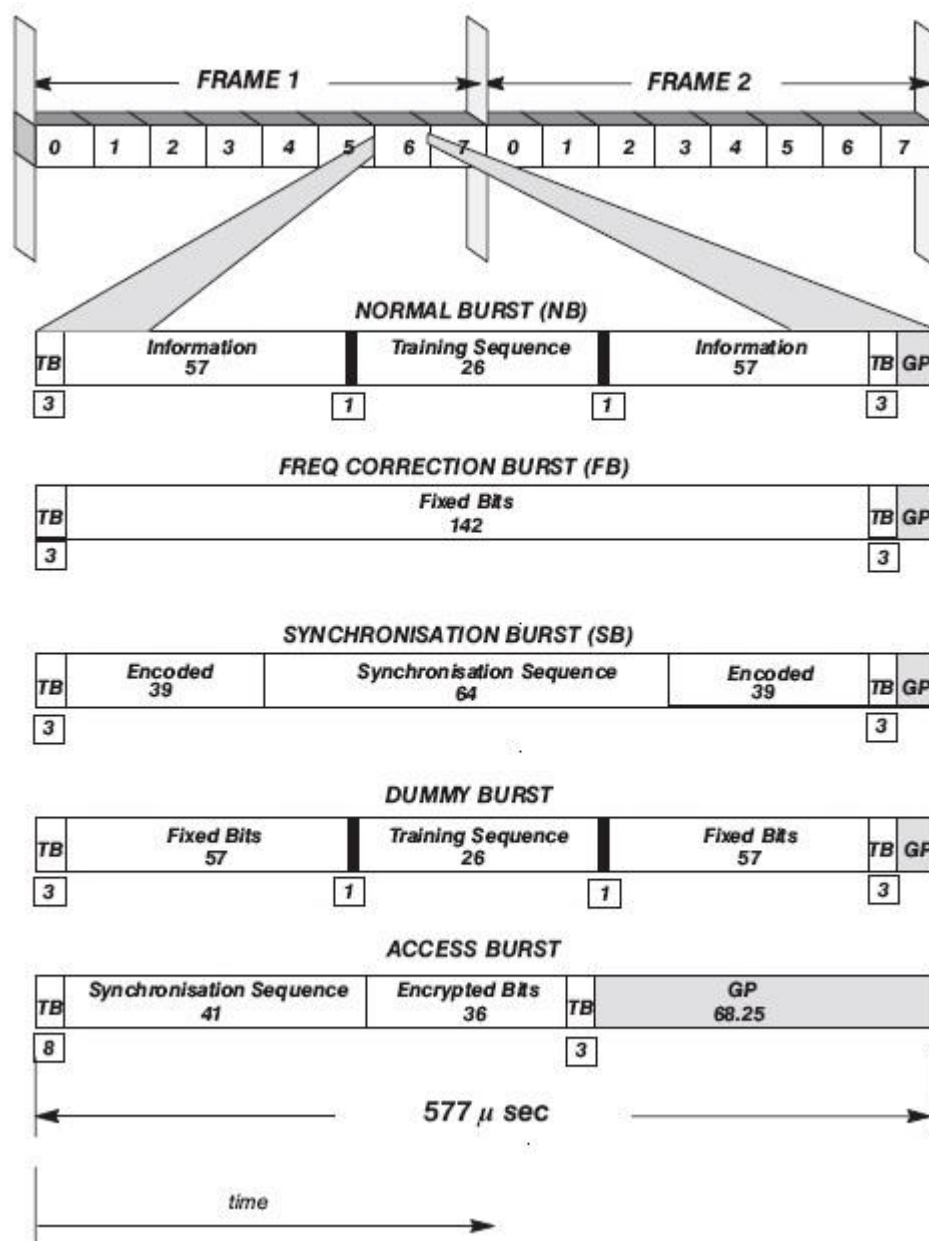
Назва говорить сама за себе. Реалізує односторонній downlink-канал FACH, що дозволяє мобільним телефонам більш точно налаштовуватися на частоту BTS.

Synchronization Burst (пакет синхронізації)

Пакет даного типу, так само як і Frequency Correction Burst, реалізує downlink-канал, тільки вже SCH, який призначений для ідентифікації присутності базових станцій в ефірі. Кожен такий пакет передається на повній потужності, а також містить інформацію про BTS, необхідну для синхронізації з нею: частота кадрів, ідентифікаційні дані (BSIC), та інші.

Dummy Burst (фальшивий пакет)

Фіктивний пакет, що передається базовою станцією для заповнення невикористовуваних таймслотів. Справа в тому, що якщо на каналі немає ніякої активності, потужність сигналу поточного ARFCN буде значно менше. В цьому випадку мобільному телефону може здатися, що він далеко від базової станції. Щоб цього уникнути, BTS заповнює невикористовувані таймслоти безглуздим трафіком.



Access Burst (пакет доступу)

При встановленні з'єднання з BTS мобільний телефон надсилає запит на виділення каналу SDCCH на каналі RACH. Базова станція, отримавши такий пакету, призначає абоненту його тайминги системи FDMA і відповідає на каналі AGCH, після чого мобільний телефон може отримувати і відправляти Normal Bursts. Варто відзначити збільшену тривалість Guard time, так як спочатку ні телефону, ні базової станції не відома інформація про тимчасові затримки. У разі, якщо RACH-запит не потрапив в таймслот, мобільний телефон через псевдовипадковий проміжок часу посилає його знову.

Основні принципи взаємодії MS і BTS

Що відбувається при включенні мобільного телефону? Найчастіше, навіть якщо телефон вимкнений зі вставленої батареєю, він продовжує працювати. У цей час працює невелика програма, яка називається

«завантажувачем». Завантажувач очікує натискання клавіші живлення, запускає процес зарядки при підключенні зарядного пристрою, а іноді і будильник. Все залежить від конкретної моделі телефону. Як тільки-но натискається кнопка включення, починається процес завантаження операційної системи, яка спочатку перевіряє наявність SIM-карти, а потім запускає сканування ефіру в пошуках мережі оператора. Навіть якщо SIM-карти немає, телефон все рівно підключається до найближчої базової станції, надаючи можливість екстреного виклику. Якщо SIM-карта на місці, виконується запит Location Update, що повідомляє мережі про поточний LAC абонента. Потім, базова станція запитує IMEI телефону і IMSI SIM-карти, щоб ідентифікувати абонента (Identity Request). Якщо наданий IMEI відрізняється від того, з яким абонент підключався раніше, оператор може вислати настройки інтернету. До речі, саме на цьому побудовано метод пошуку вкраденого або загубленого телефону. Потім виконується авторизація, після чого телефон може знаходитися в одному з двох станів:

- **IDLE** – «режим простою». Ваш телефон не передає ніяких даних мережі, прослуховуючи CCCH.
- **DEDICATED** («назначений») – між мережею та телефоном встановлено активне з'єднання, протягом якого телефон періодично передає мережі інформацію про якість сигналу, а також обмінюється даними користувача.

Тепер докладніше зупинимося на процесі підключення до мережі. Кожна базова станція обов'язково має широкомовний канал CCCH, який розташовується на нульовому таймслота певного ARFCN. У процесі сканування ефіру телефон послідовно перемикає частоту тюнера, вимірюючи потужність сигналу. Як тільки BTS з найбільш сильним сигналом буде знайдена, телефон перемикається на її канал синхронізації (SCH). Потім, отримавши перший Synchronization Burst, телефон визначає порядок проходження таймслотів, а також ідентифікаційні дані **BSIC**, які складаються з NCC (Network Color Code) і BCC (Base station Color Code). Список дозволених і заборонених для підключення ідентифікаторів зберігається на SIM-карті.

Як тільки телефон знаходить дозволений BSCH, надсилається RACH-запит, базова станція виділяє певний фізичний канал, виконує аутентифікацію абонента, а також реєструє його перебування в VLR і HLR. Після цього телефон знаходиться в режимі IDLE. При вхідному дзвінку або SMS-повідомленні, всі базові станції поточного LAC починають розсилати Paging Requests, щоб повідомити абонента про будь-яку подію. Якщо телефон його «почув», він відповідає, мережа висилає пакет Immediate Assignment, що описує виділені абоненту ресурси (частота, номер таймслота і т.д.). Дуже схоже на Ping в

Інтернеті. З цього моменту телефон перебуває в режимі DEDICATED до моменту розриву з'єднання.

У разі, якщо абонент сам виступає в ролі ініціатора з'єднання, йому необхідно спочатку вислати запит CM Service Request, а потім дочекатися Immediate Assignment від мережі.

Handover

Handover (американський варіант – handoff) – в стільникового зв'язку процес передачі абонента від однієї базової станції до іншої під час телефонної розмови або сесії передачі даних. Даний процес відбувається, коли абонент залишає зону дії однієї базової станції і входить в зону дії іншої. Також handover може виконуватися в разі, якщо поточна базова станція перевантажена, або її фізичні канали занадто зашумленими.

Handover буває двох типів:

Жорсткий handover («break-before-make»). У цьому випадку з'єднання з поточної BTS переривається, після чого створюється з'єднання з новою. З недоліків можна виділити ймовірність короткочасного розриву сесії даних, або непередбаченого завершення виклику.

М'який handover («make-before-break»). В цьому випадку телефон, не розриваючи з'єднання з поточної BTS, встановлює з'єднання з однією або декількома іншими, після чого передає сесію нової BTS і розриває з'єднання з попередньою. Недоліком даного методу є більш висока ціна компонентів телефону, що дозволяють підтримувати з'єднання відразу з декількома базовими станціями.

Кодування розмови

Як було сказано, розмова абонентів передається на каналі TCH, який буває двох видів: Full Rate (FR) і Half Rate (HR). Для кодування аудіопотоку в мережах мобільного зв'язку GSM (і не тільки) застосовуються такі стандарти:

- **GSM-FR (Full Rate, 13 Kbit/c)** – перший цифровий стандарт кодування розмови, що забезпечує досить низьку якість звуку в порівнянні з сучасними стандартами. Не дивлячись на існування більш сучасних кодеків, GSM-FR досі має дуже широке застосування.
- **GSM-HR (Half Rate, 5,6 Kbit / c)** – кодек, який використовується телефонами в режимі енергозбереження. Займає половину пропускної здатності Full Rate каналу. Економія заряду акумулятора може зайняти до 30%.
- **GSM-EFR (Enhanced Full Rate, 12,2 Kbit/c)** – алгоритм стиснення, розроблений компанією Nokia і університетом Шербрук, що є продовженням розвитку алгоритму GSM-FR.

Забезпечує гарну якість зв'язку, проте споживання електроенергії при його використанні збільшується приблизно на 5% відносно GSM-FR.

- **AMR (Adaptive multi rate)** – є алгоритмом адаптивного кодування зі змінною швидкістю. Має широке застосування в мережах GSM, забезпечуючи високу ємність мережі одночасно з високою якістю звуку. Швидкість кодування/декодування вибирається залежно від навколишніх умов і завантаження мережі.

Безпека і конфіденційність

Основні вектори атак

Оскільки Um-інтерфейс є радіоінтерфейсом, увесь його трафік «видно» будь-якого бажаному, що знаходиться в радіусі дії BTS. До того ж аналізувати дані, що передаються через радіоефір, можна навіть не виходячи з дому, використовуючи спеціальне обладнання (наприклад, старий мобільний телефон, підтримуваний проектом і звичайний комп'ютер).

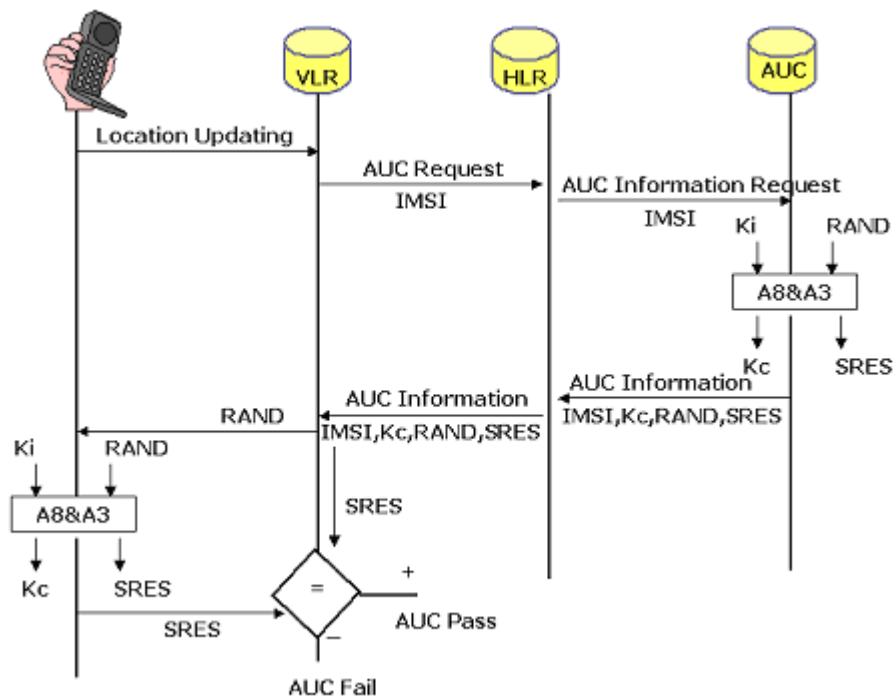
Виділяють два види атаки: пасивна та активна. У першому випадку атакуючий ніяк не взаємодіє ні з мережею, ні з абонентом – має місце виключно прийом та обробка інформації. Виявити таку атаку майже неможливо, проте й перспектив в неї не так багато, як у активної. Активна атака має на увазі взаємодію атакуючого з атакованим абонентом і/або мережею.

Можна виділити найбільш небезпечні види атак, до яких схильні абоненти стільникових мереж:

- ✓ Сніффінг – прослуховування
- ✓ Витік персональних даних, СМС і голосових дзвінків
- ✓ Витік даних про місцезнаходження
- ✓ Спуфінг (FakeBTS або IMSI Catcher) – маскування
- ✓ Віддалене захоплення SIM-карти, виконання шкідливого коду (RCE)
- ✓ Відмова в обслуговуванні (DoS)

Ідентифікація абонентів

Як було зазначено раніше, ідентифікація абонентів виконується за IMSI, який записано в SIM-карті абонента і HLR оператора. Ідентифікація мобільних телефонів виконується за серійним номером – IMEI. Однак, після аутентифікації ні IMSI, ні IMEI у відкритому вигляді через радіоефір не передаються. Після процедури Location Update абоненту присвоюється тимчасовий ідентифікатор – **TMSI** (Temporary Mobile Subscriber Identity), і подальша взаємодія здійснюється саме з його допомогою.



Способи атаки

В ідеалі, TMSI абонента відомий тільки мобільному телефону і мережі. Однак, існують способи обходу даного захисту. Якщо циклічно дзвонити абоненту або відправляти SMS-повідомлення, спостерігаючи за каналом PCN і виконуючи кореляцію, можна з певною точністю виділити TMSI атакованого абонента.

Крім того, маючи доступ до мережі межоператорної взаємодії SS7, за номером телефону можна дізнатися IMSI і LAC його власника. Проблема в тому, що в мережі SS7 всі оператори «довіряють» один одному, тим самим знижуючи рівень конфіденційності даних своїх абонентів.

Аутентифікація

Для захисту від спуфінга, мережа виконує аутентифікацію абонента перед тим, як почати його обслуговування. Крім IMSI, в SIM-карті зберігається випадково згенерована послідовність, яка називається K_i , яку вона повертає тільки в хешірованого вигляді. Також K_i зберігається в HLR оператора і ніколи не передається у відкритому вигляді. В цілому, процес аутентифікації заснований на принципі чотиристороннього рукостискання:

1. Абонент виконує Location Update Request, потім надає IMSI.
2. Мережа надсилає псевдовипадкове значення RAND.
3. SIM-карта телефону хешірує K_i і RAND за алгоритмом A3:
 $A3(RAND, K_i) = SRES$
4. Мережа теж хешірує K_i і RAND за алгоритмом A3.
5. Якщо значення SRES з боку абонента збіглося з обчисленим на стороні мережі, значить, абонент пройшов аутентифікацію.

Способи атаки

Перебір Кі, маючи значення RAND і SRAND, може зайняти досить багато часу. Крім того, оператори можуть використовувати свої алгоритми хешування. У мережі досить мало інформації про спроби перебору. Однак, не всі SIM-карти ідеально захищені. Деяким дослідникам вдалося отримати прямий доступ до файлової системи SIM-карти, а потім витягти Кі.

Шифрування трафіку

Згідно зі специфікацією, існує три алгоритму шифрування для користувача трафіку:

A5/0 – формальне позначення відсутності шифрування, так само як OPEN в WiFi-мережах. Передавання даних без шифрування практично не буває, проте, згідно з джерелами в Сирії та Південної Кореї використовується A5/0.

A5/1 – найпоширеніший алгоритм шифрування. Злам цього алгоритму неодноразово демонструвався на різних конференціях, проте усі мережі використовують саме цей алгоритм. Для розшифровки трафіку достатньо мати 2 Тб дискового простору, звичайний персональний комп'ютер з Linux і програмою Kraken на борту.

A5/2 – алгоритм шифрування з навмисне ослабленою захистом. Практично не використовується.

A5/3 – на даний момент найстійкіший алгоритм шифрування, розроблений ще в 2002 році. В інтернеті можна знайти відомості про деякі теоретично можливих вразливості, однак на практиці його злом ще ніхто не демонстрував. В GSM-мережах не використовується, хоча всі сучасні телефони його підтримують. Цей алгоритм використовують в мережах третього покоління.

Способи атаки

Маючи обладнання для сніффінга і комп'ютер з 2 Тб дискової пам'яті та програму Kraken, можна досить швидко (кілька секунд) знаходити сесійні ключі шифрування A5/1, а потім розшифровувати чий завгодно трафік. Німецький криптолог Карстен Нол (Karsten Nohl) в 2009 році продемонстрував спосіб злому A5/1. А через кілька років Карстен і Сільвіан Мюно продемонстрували перехоплення і спосіб дешифрування телефонної розмови з допомогою кількох старих телефонів Motorola (проект OsmocomBB).

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Розподіл множинного доступу за часом в стільникових мережах GSM (TDMA).
2. Логічні канали стільникових мереж GSM.
3. Пакети даних стільникових мереж GSM.
4. Безпека і конфіденційність в стільникових мережах GSM.

Лекція № 11

Тема: Технологія «клієнт-сервер». Основні поняття з баз даних**Зміст**

Введення в клієнт-серверні технології. Основні поняття.....	2
Види клієнт-серверних технологій	2
Клієнтський застосунок.....	4
Клієнт-серверна архітектура.....	5
Контрольні питання за темою.....	7
Рівень модуля	7
Рівень курсу	7

Джерела:

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
2. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. — К., 2014. — 832 с.

Введення в клієнт-серверні технології. Основні поняття.

Як правило, комп'ютери і програми, що входять до складу інформаційної системи, не є рівноправними. Деякі з них володіють ресурсами (файлова система, процесор, принтер, база даних та т.і.), інші мають можливість звертатися до цих ресурсів. Комп'ютер (або програму), що керує ресурсом, називають сервером цього ресурсу (файл-сервер, сервер бази даних, обчислювальний сервер ...). Клієнт – це апаратний або програмний компонент обчислювальної системи, який посилає запити серверу і отримує від сервера запитувану інформацію. Сервер – апаратний або програмний компонент обчислювальної системи, що виконує сервісні (обслуговуючі) функції за запитом клієнта, надаючи йому доступ до певних ресурсів або послуг. Клієнт і сервер будь-якого ресурсу можуть знаходитися як на одному комп'ютері, так і на різних комп'ютерах, пов'язаних мережею.

Архітектура "клієнт-сервер" визначає загальні принципи організації взаємодії в системі, де є *сервери* – вузли-постачальники деяких специфічних функцій (сервісів) і *клієнти* – споживачі цих функцій.

Основна ідея архітектури «клієнт-сервер» полягає в поділі мережевого застосунку (*англ. application*) на кілька компонентів, кожен з яких реалізує специфічний набір сервісів. Компоненти такого застосунку можуть виконуватися на різних комп'ютерах, виконуючи серверні і/або клієнтські функції. Це дозволяє підвищити надійність, безпеку і продуктивність мережесов'язаних програм і мережі в цілому.

Практичні реалізації такої архітектури називаються **клієнт-серверними** технологіями. Кожна технологія визначає власні або використовує наявні правила взаємодії між клієнтом і сервером, які називаються *протоколом обміну (протоколом взаємодії)*.

Архітектура «клієнт-сервер» широко застосовується в вимірювальних і керуючих системах. Можна сказати, що технологія «клієнт-сервер» є частиною комп'ютерно-інтегрованих технологій. Датчики комп'ютерно-інтегрованих систем є джерелом вимірювальної інформації, і вони відправляють виміряні дані в програми збору та обробки даних. Тобто, датчики надають сервіс з надання вимірювальних даних. Вони фактично є серверами. А програма збору даних в даному випадку є клієнтом.

Види клієнт-серверних технологій

Архітектура клієнт-сервер застосовується в великій кількості мережесов'язаних технологій (практично – в усіх), що використовуються для доступу до різних мережесов'язаних сервісів. Коротко розглянемо деякі типи таких сервісів (і серверів). Це лише кілька типів зі всієї великої кількості клієнт-серверних технологій, що використовуються як в локальних, так і в глобальних мережах.

Web-сервери

Спочатку ці сервери представляли доступ до гіпертекстових документів по протоколу HTTP (Hyper Text Transfer Protocol). Зараз підтримують розширені можливості, зокрема роботу з бінарними файлами (зображення, мультимедіа тощо).

Сервери застосунків

Призначені для централізованого вирішення прикладних завдань в деякій предметній області. Для цього користувачі мають право запускати серверні програми на виконання. Використання серверів застосунків дозволяє знизити вимоги до конфігурації клієнтів і спрощує загальне управління мережею.

Сервери баз даних

Сервери баз даних використовуються для обробки запитів користувачів на мові SQL. При цьому система керування базами даних (СКБД) знаходиться на сервері, до якого і підключаються клієнтські програми.

Файл-сервери

Файл-сервер зберігає інформацію у вигляді файлів і представляє користувачам доступ до неї. Як правило файл-сервер забезпечує і певний рівень захисту від несанкціонованого доступу.

Поштові сервери

Представляють послуги з відправлення та одержання електронних поштових повідомлень.

OPC-сервери

OPC-сервери (*Open Platform Communications server*) призначені для управління об'єктами автоматизації і технологічними процесами. Вони реалізують інтерфейс OPC – це набір специфікацій стандартів, розроблений спеціально для цілей автоматизації.

Мережеві служби

Однією з мережевих служб є DNS-сервер, роботу якого було розглянуто в попередніх лекціях.

Інформаційні служби

Прикладом інформаційної служби є сервер точного часу.

Служби обміну повідомленнями

Відомими службами, які використовують свої сервера є Skype, Viber, WhatsApp, Telegram та інші.

Сервери віддаленого доступу

Сервери віддаленого доступу через відповідну клієнтську програму забезпечують користувача аналогом локального терміналу (текстового або графічного) для роботи на віддаленій системі. Для цього використовуються зокрема сервери telnet, SSH.

Ігрові сервери

Забезпечують виконання загального сценарію для багатоклієнтських ігор.

Сервер охоронної системи

Сервер охоронної системи забезпечує безперервний контроль повідомлень від охоронної системи та направлення сигналу «Тривога» клієнту.

Сервер друку

Забезпечує комп'ютерам, під'єднаним до мережі, доступ до загального принтера.

Клієнтський застосунок

Щоб отримати доступ до тих чи інших мережесервісів використовуються програми-клієнти, можливості яких характеризуються поняттям «товщини». Воно визначає конфігурацію устаткування і програмного забезпечення, що є у клієнта. Розглянемо можливі граничні значення:

«Тонкий» клієнт

Цей термін визначає клієнта, обчислювальних ресурсів якого достатньо лише для запуску необхідного мережевого застосунку через web-інтерфейс. Інтерфейс такого застосунку формується засобами статичного HTML (виконання JavaScript не передбачено), вся прикладна логіка виконується на сервері.

Для роботи тонкого клієнта досить лише забезпечити можливість запуску web-браузера, у вікні якого і здійснюються всі дії. З цієї причини web-браузер часто називають "універсальним клієнтом".

«Товстий» клієнт

Таким є робоча станція або персональний комп'ютер, що працюють під управлінням власної операційної системи (ОС) і мають необхідний набір програмного забезпечення. До мережесерверів «товсті» клієнти звертаються в основному за додатковими послугами (наприклад, доступ до web-серверу або корпоративної бази даних). Так само під «товстим» клієнтом

мається на увазі і клієнтський мережевий застосунок, запущений під керуванням локальної ОС. Такий застосунок поєднує компонент подання даних (графічний користувацький інтерфейс ОС) і прикладної компонент (обчислювальні потужності комп'ютера клієнта).

Клієнт-серверна архітектура

Дволанкова архітектура

У будь-якій мережі, побудованій на сучасних мережевих технологіях, присутні елементи клієнт-серверної взаємодії, найчастіше на основі дволанкової архітектури. Дволанковою вона називається через необхідність розподілу трьох базових компонентів: 1) представлення даних, 2) прикладний компонент, 3) управління ресурсами – між двома вузлами (клієнтом і сервером).

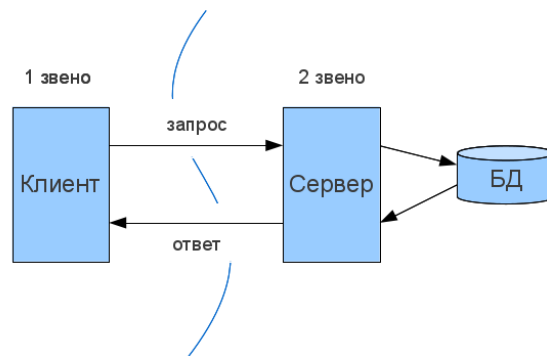


Рис.1. Дволанкова клієнт-серверна архітектура

Дволанкова архітектура використовується в клієнт-серверних системах, де сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, використовуючи для цього лише власні ресурси. Тобто сервер не викликає сторонні мережеві програми та не звертається до сторонніх ресурсів для виконання будь-якої частини запиту (рис. 1)

Триланкова архітектура

У мережевих технологіях все більше і більше використовують розподілені обчислення. Вони реалізуються на основі моделі сервера застосунків, де мережевий застосунок розділено на дві і більше частини, кожна з яких може виконуватися на окремому комп'ютері. Виділені частини застосунку взаємодіють один з одним, обмінюючись повідомленнями в заздалегідь узгодженому форматі. В цьому випадку дволанкова клієнт-серверна архітектура стає триланковою.

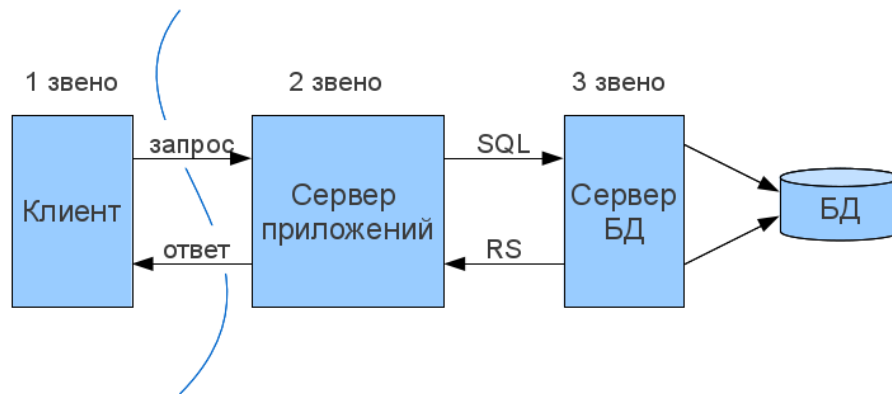


Рис.2. Триланкова клієнт-серверна архітектура

Як правило, третьою ланкою в триланковій архітектурі стає сервер застосунків, тобто компоненти розподіляються наступним чином (рис. 2):

1. Представлення даних – на стороні клієнта.
2. Прикладний компонент – на виділеному сервері застосунків.
3. Управління ресурсами – на сервері БД, який і представляє запитовані дані.

Багатоланкова архітектура

Триланкова архітектура може бути розширена до багатоланкової шляхом виділення додаткових серверів, кожен з яких представлятиме власні сервіси і може користуватися послугами інших серверів різного рівня. Абстрактний приклад багатоланкової моделі наведено на 3.

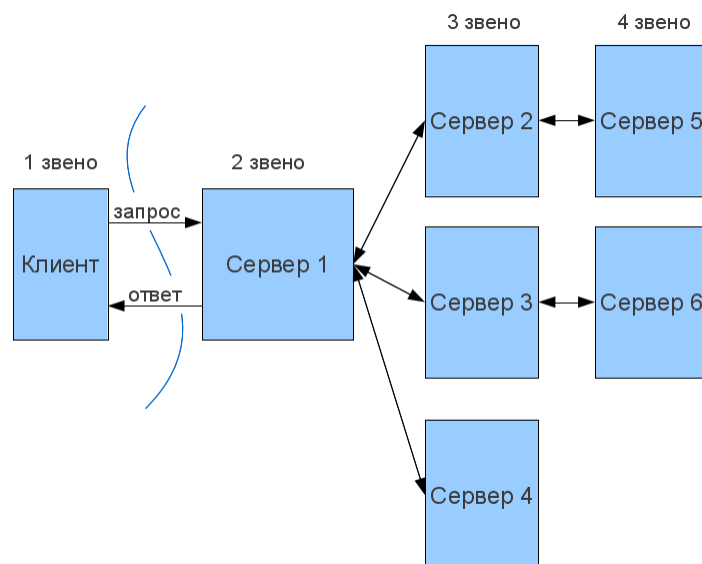


Рис.3. Багатоланкова клієнт-серверна архітектура

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Основні поняття клієнт-серверних технологій.
2. Види клієнт-серверних технологій.
3. Клієнт-серверна архітектура.

Лекція № 12

Тема: Базы даних**Зміст**

Поняття баз даних	2
Структура бази даних	3
Ієрархічна структура бази даних.....	3
Мережева структура бази даних	4
Реляційна структура бази даних	4
Об'єктно-орієнтовані і гібридні бази даних	4
Реляційні бази даних.....	5
Структурована мова запитів SQL.....	8
Простий приклад застосування оператора SELECT	10
Контрольні питання за темою.....	13
Рівень модуля	13
Рівень курсу.....	13

Джерела:

1. Волк В.К. Базы данных. Проектирование, программирование, управление и администрирование : учебник / В.К.Волк.— СП : Лань, 2020.— 244 с.

Поняття баз даних

Можливо, ви ще не знаєте, що входить в поняття бази даних, але те, що ви ними постійно користуєтеся абсолютно точно. Кожен раз, коли ви щось шукаєте в пошуковику, ви використовуєте базу даних. Коли ви вводите свої логін і пароль для входу на який-небудь сервіс, вони порівнюються зі значеннями, які зберігаються в базі даних цього сервісу.

Незважаючи на те, що ми постійно використовуємо бази даних, для багатьох залишається незрозумілим, що ж це таке насправді. І пов'язано це частково з тим, що одні й ті ж терміни, що відносяться до баз даних, використовуються людьми для визначення абсолютно різних речей.

Давайте розберемося з термінами і поняттями баз даних:

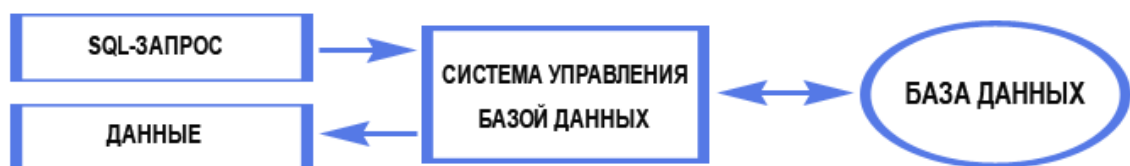
База даних (англ. database) – набір відомостей, що зберігаються певним впорядкованим способом. Можна порівняти базу даних з шафою, в якій зберігаються документи. Іншими словами, база даних - це сховище даних. Самі по собі бази даних не представляли б інтересу, якби не було систем управління базами даних (СКБД).

Якщо більш строго, то база даних – це сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування (за стандартом ISO/IEC 2382:2015). В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки.

Система управління базами даних - це сукупність мовних і програмних засобів, яка здійснює доступ до даних, дозволяє їх створювати, змінювати і видаляти, забезпечує безпеку даних і т.д. Загалом СУБД - це система, що дозволяє створювати бази даних і маніпулювати даними з них. А здійснює цей доступ до даних СУБД за допомогою спеціальної мови - SQL.

SQL - мова структурованих запитів, основним завданням якого є надання простого способу зчитування і запису інформації в базу даних.

Отже, найпростіша схема роботи з базою даних виглядає приблизно так:



За характером використання СУБД ділять на СУБД для одного користувача (призначені для створення і використання БД на персональному комп'ютері) і розраховані на багато користувачів (призначені для роботи з єдиною БД декількох комп'ютерів, об'єднаних в локальній мережі). Взагалі поділ за характером використання можна представити наступною схемою:



Не вдаючись далі в подробиці, відзначимо, що на сьогоднішній день число використовуваних СУБД обчислюється десятками. Найбільш відомі однопользовательские СУБД - Microsoft Visual FoxPro і Access, розраховані на багато користувачів - MS SQL Server, Oracle та MySQL.

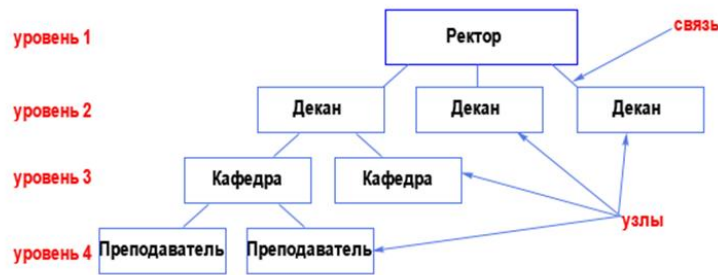
У визначенні бази даних йдеться, що це відомості, які впорядковані певним чином. А як, власне, вони впорядковані? Про це і піде мова в наступному розділі.

Структура бази даних

Створюючи базу даних, ми прагнемо впорядкувати інформацію за різними ознаками для того, щоб потім отримувати від неї необхідні нам дані в будь-якому поєднанні. Зробити це можливо, тільки якщо дані структуровані. Структурування - це набір угод про способи представлення даних. Зрозуміло, що структурувати інформацію можна по-різному. Залежно від структури розрізняють ієрархічну, мережеву, реляційну, об'єктно-орієнтовану і гібридну модель баз даних. Найпопулярнішою на сьогоднішній день є реляційна структура, тому про інших згадаємо лише побіжно.

Ієрархічна структура бази даних

Це деревоподібна структура представлення інформації. Її особливість в тому, що кожен вузол на більш низькому рівні має зв'язок тільки з одним вузлом на більш високому рівні. Подивимось, наприклад, на фрагмент ієрархічної структури бази даних "Університет":



Зі структури зрозуміло, що на одній кафедрі може працювати кілька викладачів. Такий зв'язок називається "один до багатьох" (одна кафедра - багато викладачів). Але якщо ми спробуємо додати в цю структуру групи студентів, то нам знадобиться зв'язок "багато до багатьох":



(один викладач може працювати з багатьма групами, а одна група може вчитися у багатьох викладачів), а такого зв'язку в ієрархічній структурі бути не може (тому що зв'язок може бути тільки з одним вузлом на більш високому рівні). Це основний недолік такої структури бази даних.

Мережева структура бази даних

По суті, це розширення ієрархічної структури. Все те ж саме, але існує зв'язок "багато до багатьох". Мережева структура бази даних дозволяє нам додати групи в наш приклад. Недоліком мережевої моделі є складність розробки серйозних програм.

Реляційна структура бази даних

Всі дані представлені у вигляді простих таблиць, розбитих на рядки і стовпці, на перетині яких розташовані дані. Детально про це ми будемо говорити в наступному розділі.

Об'єктно-орієнтовані і гібридні бази даних

В об'єктно-орієнтованих базах даних дані зберігаються у вигляді об'єктів, що дуже зручно. Але на сьогоднішній день такі БД ще не поширені, тому що поступаються в продуктивності реляційним.

Гібридні БД поєднують в собі можливості реляційних та об'єктно-орієнтованих, тому їх часто називають об'єктно-реляційними. Прикладом такої СУБД є Oracle, починаючи з восьмої версії.

Безсумнівно, такі БД будуть розвиватися в майбутньому, але поки першість залишається за реляційними структурами.

Реляційні бази даних

Реляційні бази даних складаються з таблиць. Кожна таблиця складається з стовпців (їх називають полями або атрибутами) та рядків (їх називають записами або кортежами). Таблиці в реляційних базах даних мають ряд властивостей. Основними є такі:

- ✓ В таблиці не може бути двох однакових рядків.
- ✓ Стовпці розташовуються в певному порядку, який створюється під час створення таблиці. В таблиці може не бути жодного рядка, але обов'язково повинен бути хоча б один стовець.
- ✓ У кожного стовпця є унікальне ім'я (в межах таблиці), і всі значення в одному стовпці мають один тип (число, текст, дата ...).
- ✓ На перетині кожного стовпця та рядку може перебувати тільки *атомарне* значення (одне значення, яка не складається з групи значень). Таблиці, що задовольняють цій умові, називають нормалізованими.

В теорії баз даних атомарні дані (значення) - це атрибути, які зберігають єдине значення і не є ні списком, ані множиною значень. Іншими словами, це такі дані, поділ яких на складові призводить до втрати їх сенсу з точки зору розв'язуваної задачі. Наприклад, якщо атрибут «Ціна» містить значення 15, то спроба розділити його на 1 і 5 призведе до повної нісенітничі. Дані, які не є атомарними, називаються складовими.

Це стане зрозуміліше на прикладі. Припустимо, ми захотіли створити базу даних для форуму. У форумі є зареєстровані користувачі, які створюють теми і залишають повідомлення в цих темах. Ця інформація і повинна зберігатися в базі даних.

Теоретично (на папері) ми можемо все це розташувати в одній таблиці, наприклад, так:

<i>Ім'я</i>	<i>E-mail</i>	<i>Пароль</i>	<i>Створені теми</i>	<i>Створені повідомлення</i>

Але це суперечить властивості атомарності (одне значення в одній комірці), тому що в стовпчиках *Теми* і *Повідомлення* у нас передбачається необмежена кількість значень. Отже, нашу таблицю треба розбити на три: Користувачі, Теми і Повідомлення.

Користувачі

<i>Ім'я</i>	<i>e-mail</i>	<i>Пароль</i>

Теми

<i>Назва</i>	<i>Автор</i>

Повідомлення

<i>Текст</i>	<i>Автор</i>

Наша таблиця **Користувачі** задовольняє всім умовам. А ось таблиці **Теми** і **Повідомлення** – ні. Адже в таблиці не може бути двох однакових рядків, а де гарантія, що один користувач не залишить два однакових повідомлення, наприклад:

Повідомлення

<i>Текст</i>	<i>Автор</i>
Думаю що треба зробити таким чином...	Кирило
Згоден	Василь
А ще можна зробити так...	Семен
Згоден	Василь

Крім того, ми знаємо, що кожне повідомлення обов'язково відноситься до будь-якої теми. А як це можна дізнатися з наших таблиць? Ніяк. Для вирішення цих проблем, в реляційних базах даних існують ключі.

Первинний ключ (скорочено РК - Primary Key) – стовпець, значення якого у всіх рядках різні. Первинні ключі можуть бути логічними (природними) і сурогатними (штучними). Так, для нашої таблиці **Користувачі** первинним ключем може стати стовпець **e-mail** (адже теоретично не може бути двох користувачів з однаковим e-mail). На практиці краще використовувати сурогатні ключі, тому що їх застосування дозволяє абстрагувати ключі від реальних даних. Крім того, первинні ключі міняти не можна, а що якщо у користувача зміниться e-mail?

Сурогатний ключ являє собою додаткове поле в базі даних. Як правило, це порядковий номер запису (хоча ви можете задавати їх на свій розсуд, контролюючи, щоб вони були унікальні). Давайте внесемо поля первинних ключів в наші таблиці:

Користувачі

<i>id користувача</i>	<i>Ім'я</i>	<i>e-mail</i>	<i>Пароль</i>
1	Кирило	kirilo@gmail.com	Gh345fgh
2	Василь	vasil@yahoo.com	As3bh7
3	Семен	semen@i.ua	gk4bb6

Теми

<i>id теми</i>	<i>Назва</i>	<i>Автор</i>
1	Про рибалку	Кирило
2	Велосипеди	Василь
3	Нічні клуби	Семен
4	Про рибалку	Василь

Повідомлення

<i>id повідомлення</i>	<i>Текст</i>	<i>Автор</i>
1	Думаю що треба зробити таким чином...	Кирило
2	Згоден	Василь
3	А ще можна зробити так...	Семен
4	Згоден	Василь

Тепер кожен запис в наших таблицях – унікальний. Нам залишилося встановити відповідність між темами і повідомленнями в них. Робиться це також за допомогою первинних ключів. У таблицю повідомлення ми додамо ще одне поле:

Повідомлення

<i>id повідомлення</i>	<i>Текст</i>	<i>Автор</i>	<i>id теми</i>
1	Думаю що треба зробити таким чином...	Кирило	1
2	Згоден	Василь	4
3	А ще можна зробити так...	Семен	1
4	Згоден	Василь	1

Тепер зрозуміло, що повідомлення з *id* = 2 належить темі "Про рибалку" (*id* теми = 4), створеної Василем, а решта повідомлення належать темі "Про рибалку" (*id* теми = 1), створеної Кирилом. Таке поле називається зовнішній ключ (скорочено FK – Foreign Key). Кожне значення цього поля відповідає якомусь первинному ключу з таблиці "Теми". Так встановлюється однозначна відповідність між повідомленнями і темами, до яких вони належать.

Останній нюанс. Припустимо, у нас додався новий користувач, і його теж звать Василь:

Користувачі

<i>id користувача</i>	<i>Ім'я</i>	<i>e-mail</i>	<i>Пароль</i>
1	Кирило	kirilo@gmail.com	Gh345fgh
2	Василь	vasil@yahoo.com	As3bh7
3	Семен	semen@i.ua	gk4bb6
4	Василь	vasil@gmail.com	Tr6t9y00

Як ми дізнаємося, який саме Василь залишив повідомлення? Щоб все працювало добре, необхідно зміст поля *Автор* в таблицях "Теми" і "Повідомлення" замінити зовнішніми ключами:

Повідомлення

<i>id повідомлення</i>	<i>Текст</i>	<i>Автор</i>	<i>Id теми</i>
1	Думаю що треба зробити таким чином...	1	1
2	Згоден	2	4
3	А ще можна зробити так...	3	1
4	Згоден	2	1

Теми

<i>id теми</i>	<i>Назва</i>	<i>Автор</i>
1	Про рибалку	1
2	Велосипеди	2
3	Нічні клуби	3
4	Про рибалку	1
5	До кого звернутися	4

Наша база даних готова. Схематично її можна представити таким чином:



В нашій маленькій базі даних всього три таблички, а якби їх було 10 або 100? Зрозуміло, що відразу неможливо уявити всі таблиці, поля та зв'язки, які нам можуть знадобитися. Саме тому проектування бази даних завжди починається з її концептуальної моделі.

Структурована мова запитів SQL

Structured Query Language (структурована мова запитів) або SQL – мова керування базами даних для реляційних баз даних. SQL спочатку створювався IBM. Вона була прийнята як стандарт американським Національним інститутом стандартів (ANSI) в 1986 і ISO в 1987.

Як випливає з назви, мову програмування SQL призначено для конкретних, обмежених цілей – запитів даних, що містяться в реляційній базі даних. Як такий, він являє собою набір інструкцій мови програмування для створення вибірок даних, а не процедурну мову, таку як C або BASIC, які призначені для вирішення набагато ширшого кола проблем. Основний підхід полягає в тому, що в команди процедурних мов програмування дозволяється вбудовувати запити SQL і таким чином взаємодіяти з базою даних.

SQL складається з чотирьох окремих частин (груп):

1. Перша – це Data Manipulation Language або DML (мова керування даними). DML є підмножина мови SQL, яка використовується для запиту до баз даних, додавання, оновлення та видалення даних.

- SELECT є однією з найбільш часто використовуваних команд; вона вибирає дані, що задовольняють заданим умовам
- INSERT використовується для додавання рядків для існуючої таблиці.
- UPDATE використовується для зміни значень даних в існуючій рядку таблиці.
- DELETE визначає рядки, які будуть видалені з таблиці.

Друга група – це Data Definition Language або DDL (мова визначення даних). DDL дозволяє користувачеві визначати нові таблиці і пов'язані з ними елементи. Основними командами DDL є команди "створювати" і "видаляти" об'єкти.

CREATE визначає об'єкти (наприклад, таблиці), які будуть створені в базі даних.

DROP визначає, які існуючі об'єкти будуть видалені з бази даних.

Деякі системи баз даних також підтримують команду ALTER, яка дозволяє користувачеві змінювати існуючий об'єкт. Наприклад, можна додати стовпці в існуючу таблицю.

3. Третьою групою ключових слів SQL є Data Control Language або DCL (мова контролю даних). DCL відповідає за права доступу до даних і дозволяє користувачеві контролювати, хто має доступ, щоб переглядати або маніпулювати даними в базі даних. Є два основних ключових слова:

GRANT – дозволяє користувачеві виконувати операції;

REVOKE – видаляє або обмежує можливість користувачеві виконувати операції.

4. Мова управління транзакціями (TCL) використовується для контролю обробки транзакцій в БД. Зазвичай оператори TCL виконують такі операції як **commit** для підтвердження змін, зроблених в ході транзакції, **rollback** для їх скасування та **savepoint** для розбиття транзакції на кілька менших частин.

Простий приклад застосування оператора SELECT

Синтаксис оператора SELECT

```
SELECT column_list  
FROM table_name  
[WHERE умова]  
[GROUP BY умова]  
[HAVING умова]  
[ORDER BY умова]
```

SELECT Ключове слово, яке повідомляє базі даних про те, що оператор є запитом. Всі запити починаються з цього слова, за ним йде пробіл.

Column_list Список стовпців таблиці, які вибираються за запитом. Стовпці, які не вказані в операторі, не будуть включені в результат. Якщо необхідно вивести дані всіх стовпців, можна використовувати скорочений запис: зірочка (*) означає повний список стовпців.

FROM table_name Ключове слово, яке повинно бути присутнім в кожному запиті. Після нього через пробіл вказується ім'я таблиці, яка є джерелом даних.

Код в дужках не є обов'язковим для оператора SELECT. Він необхідний для більш точного визначення запиту.

Також необхідно сказати, що SQL код є регістронезалежним. Це означає, що запис SELECT можна написати як select. СУБД не відрізняє ці два записи, однак рекомендується всі оператори SQL писати прописними літерами, щоб їх було легко відрізнити від іншого коду.

Приклади розглянемо на таблиці Salespeople (продавці).

Таблиця виглядає таким чином:

snum	sname	city	comm
1001	Peel	London	0,12
1002	Serres	San Jose	0,13
1003	Axelrod	New York	0,1
1004	Motika	London	0,11
1007	Rifkin	Barcelona	0,15

Стовпці таблиці Salespeople:

snum	Номер продавця
sname	Ім'я продавця
city	Місто
comm	Комісійні продавця, в десятковій формі

Приклад використання оператора SELECT

1. Необхідно вивести список продавців, і відобразити їх імена (sname)

```
SELECT sname  
FROM Salespeople
```

Результат:

sname
Peel
Serres
Axelrod
Motika
Rifkin

В даному запиті, після оператора SELECT йде ім'я стовпця, яке необхідно відобразити. Після ключового слова FROM вказується ім'я таблиці.

2. Необхідно вивести список продавців, і відобразити їх імена і місто (sname і city)

```
SELECT sname , city  
FROM Salespeople
```

Результат:

sname	city
Peel	London
Serres	San Jose
Axelrod	New York
Motika	London
Rifkin	Barcelona

Тут після оператора SELECT перераховуються стовпці, які необхідно вивести. Імена стовпців пишуться через кому.

3. Необхідно вивести всю таблицю

Для цього можна використовувати різний синтаксис написання запитів. Перераховуємо кожен стовпець після оператора SELECT:

```
SELECT snum , sname , city , comm  
FROM Salespeople
```

Можна отримати той самий результат, якщо використати скорочену форму:

```
SELECT * FROM Salespeople
```

Результат:

snum	sname	city	comm
1001	Peel	London	0,12
1002	Serres	San Jose	0,13
1003	Axelrod	New York	0,1
1004	Motika	London	0,11
1007	Rifkin	Barcelona	0,15

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Поняття баз даних.
2. Структура бази даних.
3. Реляційні бази даних.
4. Структурована мова запитів SQL.

Лекція № 13

Тема: Комп'ютерно-інтегровані системи вимірювання, контролю та керування

Зміст

Основні поняття	2
Вимірювання	2
Контроль	2
Керування	3
Приклади систем автоматичного керування і контролю	4
Система наповнення бака	4
Автоматичні двері.....	4
Безпілотний автомобіль	6
Система керування промисловим об'єктом	7
Архітектура автоматизованих систем керування і контролю на основі комп'ютерно-інтегрованих технологій	7
Контрольні питання за темою.....	10
Рівень модуля	10
Рівень курсу	10

Основні поняття

Вимірювання

Отримання інформації про кількісні характеристики властивостей об'єктів і явищ навколишнього світу експериментальним шляхом називається вимірюванням. На відміну від кількісної інформації, одержуваної теоретичним шляхом, тобто за допомогою обчислень і розрахунків, така інформація називається вимірювальною.

Отримана тим чи іншим способом кількісна інформація про властивості об'єктів і явищ навколишнього світу перетворюється, передається і представляється у наочній формі в інформаційно-вимірювальних системах або інших пристроях відображення і реєстрації інформації.

Контроль

Технічний контроль – перевірка відповідності об'єкта встановленим технічним вимогам

Сутність будь-якого контролю зводиться до здійснення двох основних етапів:

1. Отримання інформації про фактичний стан певного об'єкта, про ознаки і показники його властивостей. Цю інформацію можна назвати первинною.

2. Зіставлення первинної інформації із заздалегідь встановленими вимогами, нормами, критеріями, тобто виявлення відповідності або невідповідності фактичних даних необхідним (очікуваним). Інформацію про неузгодженість (розбіжність) фактичних і необхідних даних можна називати вторинною.

Об'єктом, дані про стан і (або) властивості якого підлягають порівнянню з встановленими вимогами, може бути продукція або процес.

У ряді випадків неможливо розрізнити в часі межу між першим і другим етапами контролю. У таких випадках перший етап може бути виражений нечітко або може практично не спостерігатися. Характерним прикладом є контроль розміру калібром, що зводиться до операції зіставлення фактичного і гранично допустимого значень розміру.

Надалі вторинна інформація використовується для вироблення відповідних керуючих впливів на об'єкт, який підлягав контролю. У цьому сенсі будь-який контроль завжди є активним. В зв'язку з цим необхідно відзначити, що будь-який контроль, завжди в якийсь мірі має бути профілактичним, оскільки вторинна інформація може використовуватися для вдосконалення розробки, виробництва і експлуатації продукції, для підвищення її якості та т.і.

Однак, прийняття рішень на основі аналізу вторинної інформації та вироблення відповідних керуючих впливів – це вже не є частиною контролю. Це вже є наступний етап – етап керування. Він заснований на результатах контролю, який є невід'ємною і важливою частиною будь якого керування. При технічному контролі первинна інформація зіставляється з технічними вимогами, записаними в нормативній документації, або з параметрами контрольного зразка, або з даними, зафіксованими за допомогою калібру та т.і.

Керування

З давніх часів людина хотіла використовувати предмети і сили природи в своїх цілях, тобто керувати ними. Безліч завдань керування в сучасному світі пов'язано з технічними системами – автомобілями, кораблями, літаками, верстатами. Наприклад, потрібно підтримувати заданий курс корабля, висоту літака, частоту обертання двигуна, температуру в холодильнику або в печі. Якщо ці завдання вирішуються без участі людини, кажуть про автоматичне керування.

Керування – діяльність суб'єкта зі зміни об'єкта для досягнення певної мети.

Для керування необхідно виконувати наступні три дії:

1. Отримувати інформацію щодо стану об'єкта, тобто виконувати *вимірювання*. Якщо ми хочемо змінити об'єкт, то ми в першу чергу повинні знати яким він є зараз, в даний момент. Тобто знати поточний стан об'єкта: які його параметри, його властивості, а якщо точніше – мати чисельні значення фізичних величин, які характеризують даний об'єкт.

2. Приймати рішення про необхідність зміни стану об'єкта, вибрати спосіб впливу на об'єкт і визначати характеристики необхідного керуючого впливу. Рішення про необхідність зміни стану об'єкта приймається в тому випадку, якщо поточний стан об'єкта з певних причин нас не влаштовує. Таким чином, ми виконуємо *контроль*: ми порівнюємо відомий нам виміряний стан об'єкта з бажаним, тобто тим станом, який відповідає заданій нами мети.

3. Впливати на об'єкт.

Для виконання зазначених функцій до складу будь-якої системи керування повинні входити:

- ✓ датчики,
- ✓ керуюча система (або вирішуючий пристрій),
- ✓ виконавчі пристрої (або привод).

Приклади систем автоматичного керування і контролю

Система наповнення бака

Розглянемо найпростішу систему автоматичного наповнення резервуара з водою (рис. 1). Завдання системи керування – підтримувати в заданих межах (від мінімального рівня - **min** до максимального - **max**) кількість води в ємності.

До складу системи входять датчики – два датчика рівня і виконавчий пристрій – насос з електродвигуном. Роль вирішуючого пристрою грає спеціальне реле.

Робота реле заснована на кондуктометричному методі виявлення наявності рідини, який використовує електропровідність рідин і появу струму між електродами. Реле мають перемикаючі контакти, і це дозволяє організувати режим наповнення або зливу. Принцип роботи реле заснований на вимірюванні опору рідини, що знаходиться між двома зануреними датчиками. Якщо виміряний опір виявляється менш величини порогу спрацьовування, тоді стан контактів реле змінюється.

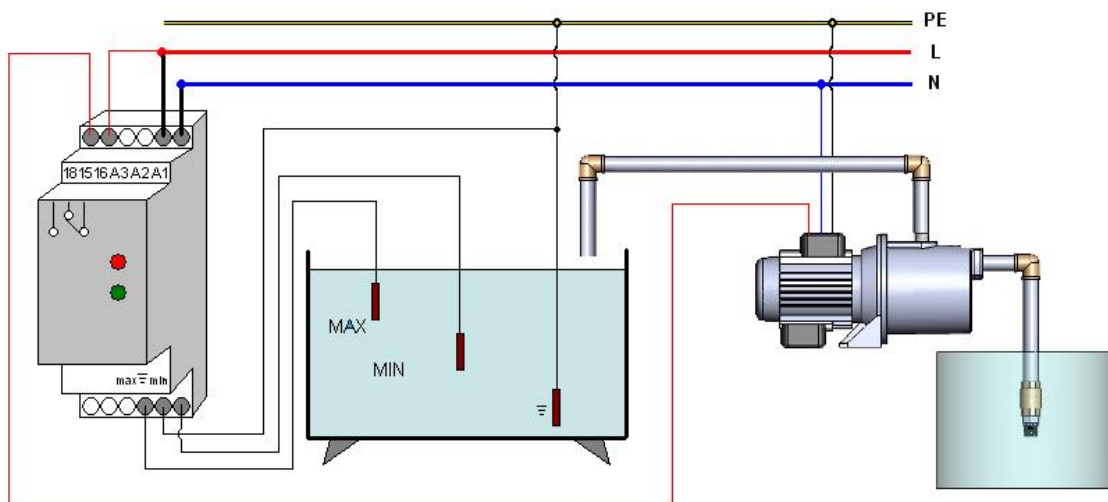


Рис. 1 – Система автоматичного наповнення резервуара з водою

Автоматичні двері

В основу роботи всіх дверей, які автоматично відкриваються та закриваються, закладено принцип переміщення полотна за допомогою електричного мотора, який через спеціальні механізми передає енергію свого обертання до дверного полотна, змушуючи його переміщатися в потрібному напрямку. Щоб забезпечити автоматичне відкривання та закривання дверей, ці конструкції обладнуються датчиками, що реагують на рух. Виявивши присутність людини на підході до дверей, вони подають команду керуючому електронному пристрою, який в свою чергу запускає електромотор й тим самим приводить в рух систему відкривання дверей. Якщо датчик протягом

певного часу не уловлює рухів, то електронний пристрій самостійно формує команду на закриття дверного полотна.



Рис. 2 – Виконавчий пристрій автоматичних дверей.

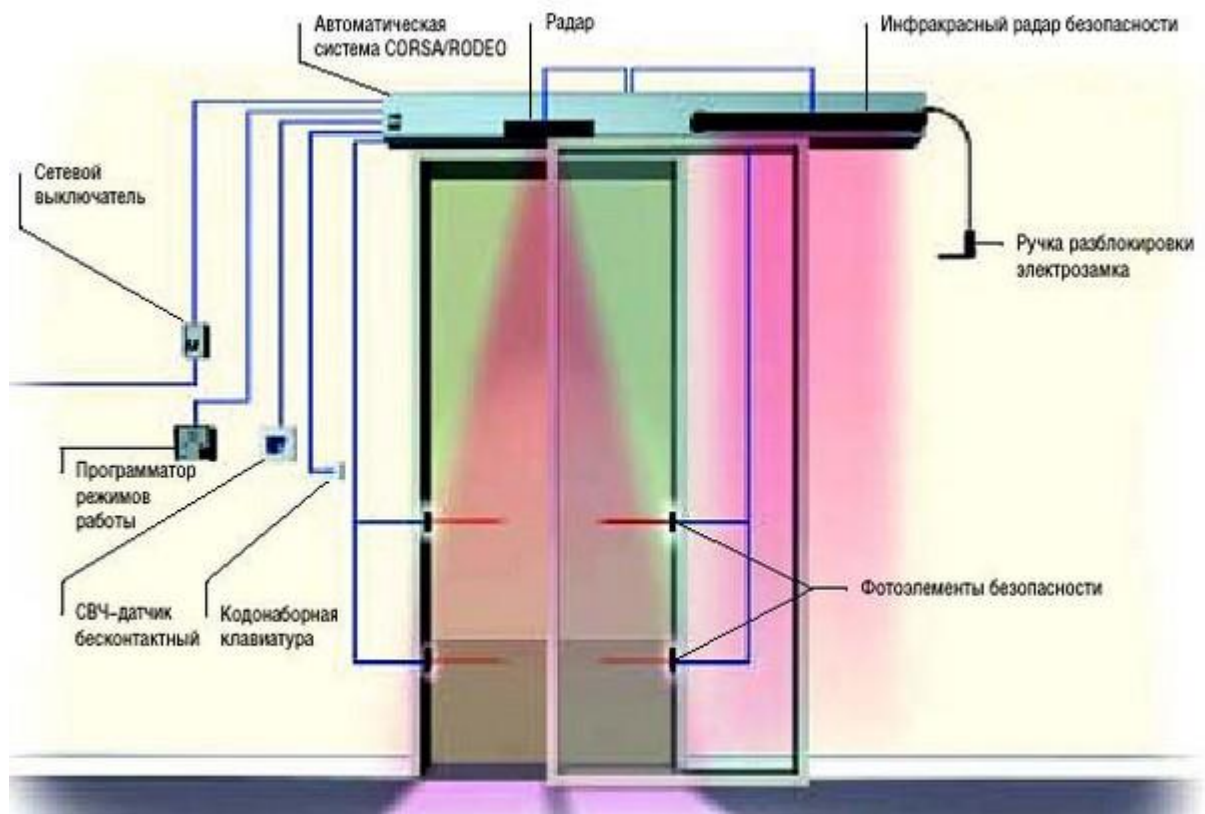


Рис. 3 – Схема розташування елементів автоматичних дверей

Автоматичні двері стандартної комплектації мають кілька режимів роботи, за дотриманням яких стежить все той же електронний контролер. Залежно від пори року або умов експлуатації дверей, їх можна перевести в один з п'яти режимів роботи.

- Стандартний або основний режим, при якому двері повноцінно забезпечують прохід людини з обох сторін.
- Зимовий режим роботи – електроніка не дає полотнам відкриватися до крайнього положення, а пригальмовує їх на півдорозі, тим самим знижуючи викид теплого повітря на вулицю.
- Режим «закрито».
- Режим «повністю відкрито» – літній режим роботи.
- Односторонній режим роботи.

Крім усього іншого, практично всі автоматичні двері мають функції налаштування. За допомогою налаштувань встановлюється час знаходження дверей у відкритому стані; відстань до людини, коли двері мають відкриватися; швидкість руху дверного полотна; введення гальмування дверей в останній момент закривання – в цілому, все необхідне, щоб забезпечити комфортну експлуатацію автоматичних дверей. Подібні двері мають систему безпеки: як правило, їх обладнають так званим світловим бар'єром, фотоелемент якого реєструє перешкоди на шляху закривання стулок. Вловлюючи їх, він подає команду на відкривання дверних полотен. Людину не стисне між стулками, як це інколи буває в тролейбусах або міських автобусах, обладнаних складеними дверима.

Безпілотний автомобіль



Рис. 4 – Принцип роботи безпілотного автомобіля Toyota Prius в версії Google.

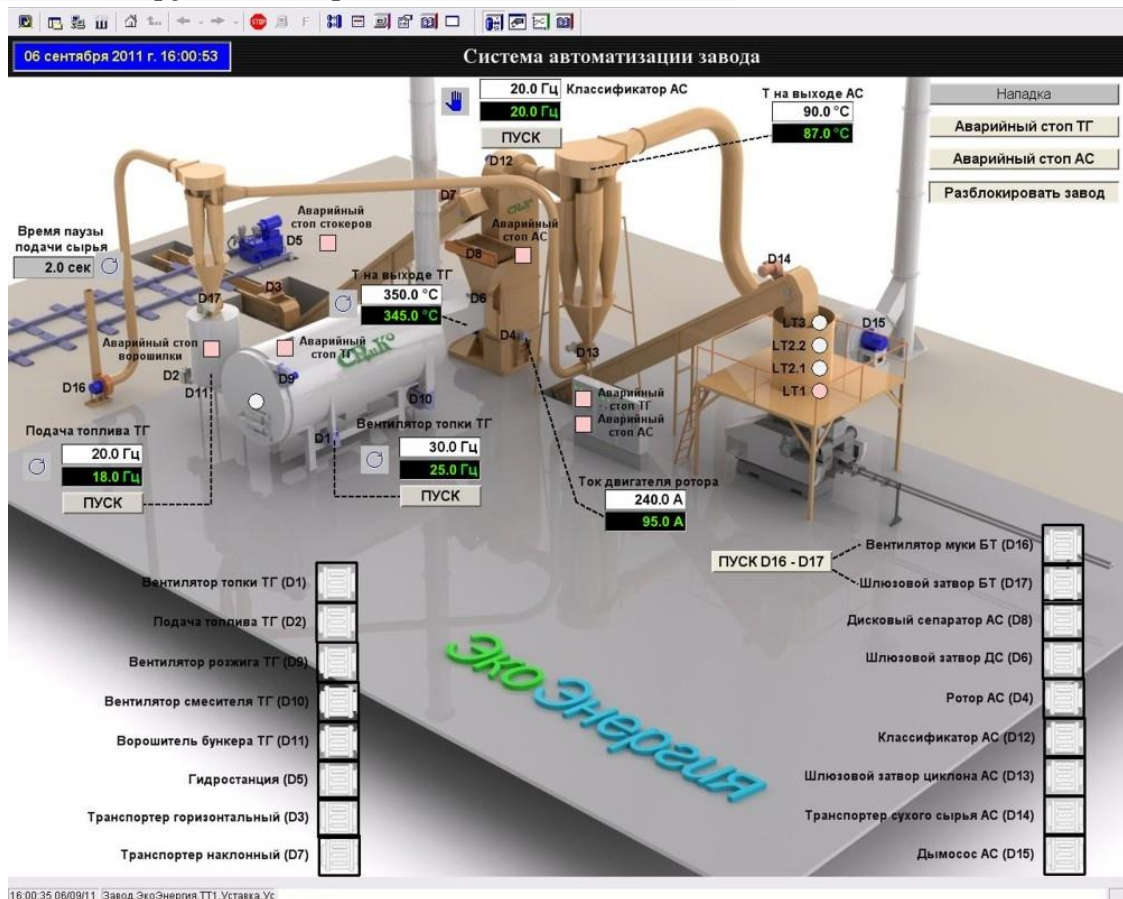
Система керування промисловим об'єктом

Рис. 5 – Система керування промисловим об'єктом

Архітектура автоматизованих систем керування і контролю на основі комп'ютерно-інтегрованих технологій

Сучасні автоматизовані системи керування та контролю будуються на основі комп'ютерно-інтегрованих технологій. Можна сказати, що системи, в яких реалізуються комп'ютерно-інтегровані технології, являють собою наступне покоління автоматичних і автоматизованих вимірювальних систем, систем керування і контролю, а також АСУ ТП – автоматизованих систем керування технологічними процесами.

Класична архітектура керуючої системи втілена в моносистемі (рис.6) – зосередженій системі, яка забезпечує керування зосередженим об'єктом. Зосередженими в даному випадку можна називати системи, в яких відстань від датчиків і приводів до комп'ютера обчислюється метрами або десятками метрів.

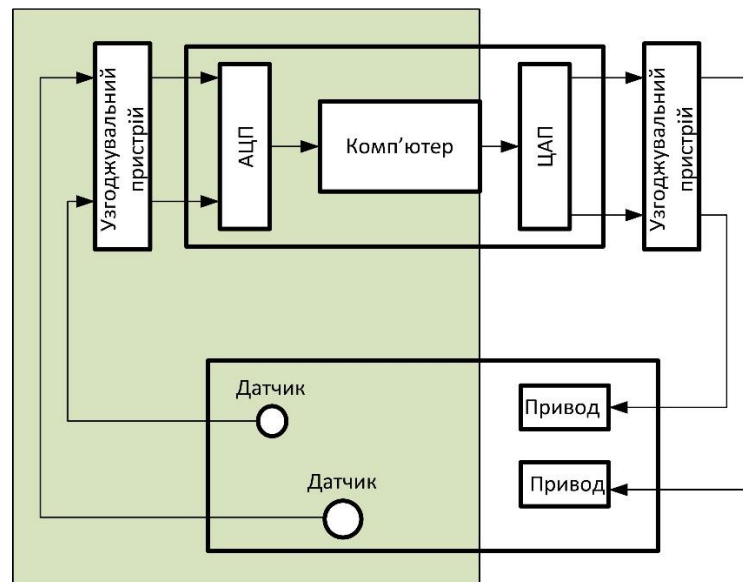


Рис. 6 – Зосереджена система

Якщо розглядати тільки завдання контролю, без завдання керування, то комп'ютерно-інтегрована система контролю зображена на малюнку 2 на тлі зеленого прямокутника. Комп'ютер в даному випадку не формує керуючий сигнал, а надає тільки вторинну інформацію, тобто результат контролю, наприклад «Норма» або «Ненорма». Інформація подається у вигляді, необхідному для її подальшого використання. Найбільш поширений варіант – видача результату контролю на екран в текстовому або графічному вигляді для її сприйняття оператором.

Згодом стали розвиватися розподілені системи (рис. 7), які забезпечують вимірювання і керування на сукупності об'єктів. В цьому випадку відстані зазвичай складають сотні метрів і кілометри, а в деяких випадках – десятки кілометрів.

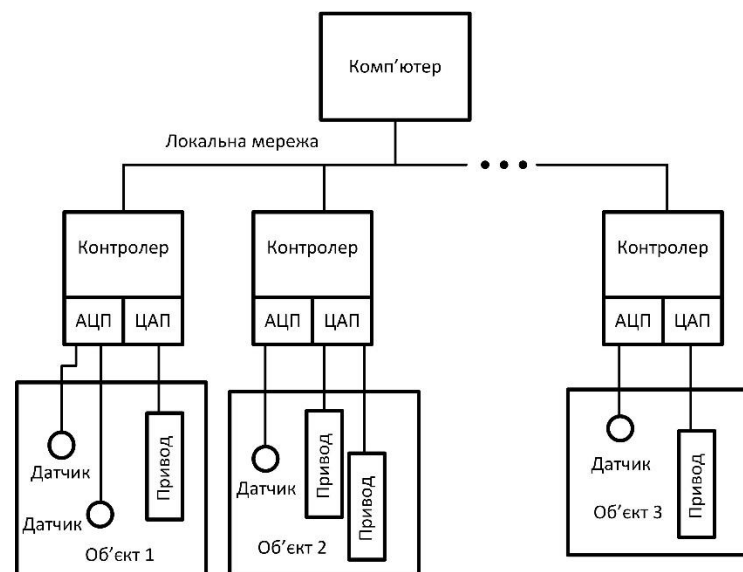


Рис.7 – Розподілена система

Клас завдань, які вирішуються за допомогою комп'ютерно-інтегрованих технологій, постійно розширюється. Останнім часом все більш широке застосування знаходять глобально розподілені системи – системи в яких об'єкти контролю знаходяться друг від друга на значних відстанях, які практично вже не обмежені. Такі системи будуються за схемою, представленою на наступному рисунку. Такі системи, в основному, є вимірювальними.

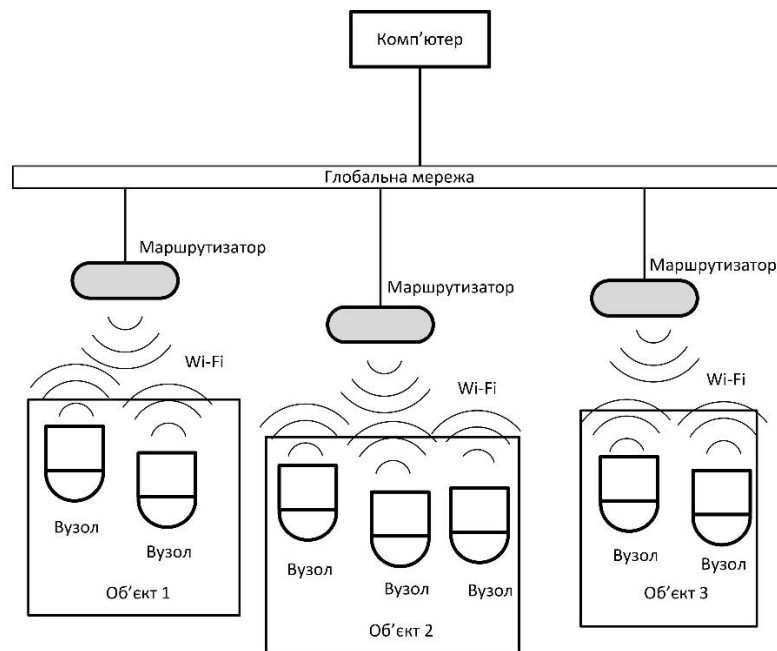


Рис. 8 – Глобально розподілена система

Як можна бачити, системи, що вирішують задачі автоматизації використовуючи комп'ютерно-інтегровані технології, в основному будуються із застосуванням таких апаратних засобів:

- комп'ютер і/або контролер,
- аналого-цифровий перетворювач (АЦП),
- цифро-аналоговий перетворювач (ЦАП),
- датчики (або "вузли"),
- виконавчі пристрої (або приводи),
- комп'ютерна мережа.

Контрольні питання за темою

Рівень модуля

1. Що таке вимірювання?
2. Що таке контроль?
3. Що таке керування?
4. У чому відмінність вимірювання і контролю?
5. Як пов'язаний контроль з вимірюванням та керуванням?
6. Під час проведення технічного контролю яка інформація називається первинною?
7. Під час проведення технічного контролю яка інформація називається вторинною?
8. Які дії необхідні для здійснення керування?
9. Які складові частини повинна включати в себе будь-яка система управління?
10. Опишіть роботу найпростішої системи наповнення резервуара водою.
11. Які датчики входять до складу автоматичних дверей?
12. Що є виконавчим механізмом автоматичних дверей?
13. Що є вирішальним пристроєм автоматичних дверей?
14. Подайте рисунком функціональну схему зосередженої керуючої системи.
15. Подайте рисунком функціональну схему розподіленої керуючої системи.
16. Подайте рисунком функціональну схему глобально розподіленої вимірювальної системи.
17. Назвіть основні апаратні засоби комп'ютеризованих систем керування і контролю.

Рівень курсу

1. Основні поняття: вимірювання, контроль, керування та їх зміст.
2. Архітектура зосередженої комп'ютерно-інтегрованої системи.
3. Архітектура розподіленої комп'ютерно-інтегрованої системи.
4. Архітектура глобально розподіленої комп'ютерно-інтегрованої системи.

Лекція № 14

Тема: Операційні системи. Частина перша**Зміст**

Структура програмного забезпечення	2
Системне ПО	2
Прикладне ПО	4
Інструментальне ПЗ	4
Операційна система, її призначення і склад	5
Функції операційної системи.....	7
Абстрактні ресурси.....	7
Операційна система як менеджер ресурсів.....	8
Контрольні питання за темою.....	11
Рівень модуля	11
Рівень курсу.....	11

Джерела:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.
2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёмченко и др. — К., 2014. — 832 с.

Структура програмного забезпечення

Сукупність програм, призначена для вирішення завдань на комп'ютері, називається програмним забезпеченням. Програмне забезпечення (ПО), можна умовно розділити на три категорії:

1. **Системне** (програми загального користування), що виконують різні допоміжні функції, наприклад створення копій використовуваної інформації, видачу довідкової інформації про комп'ютер, перевірку працездатності пристроїв комп'ютера і т.д.

2. **Прикладне**, що забезпечує виконання завдань, необхідних користувачеві: обробка інформаційних масивів, математичне моделювання, редагування текстових документів, створення і редагування зображень, і т.д.

3. **Інструментальне** (системи програмування), що забезпечує розробку нових програм для комп'ютера на мові програмування.

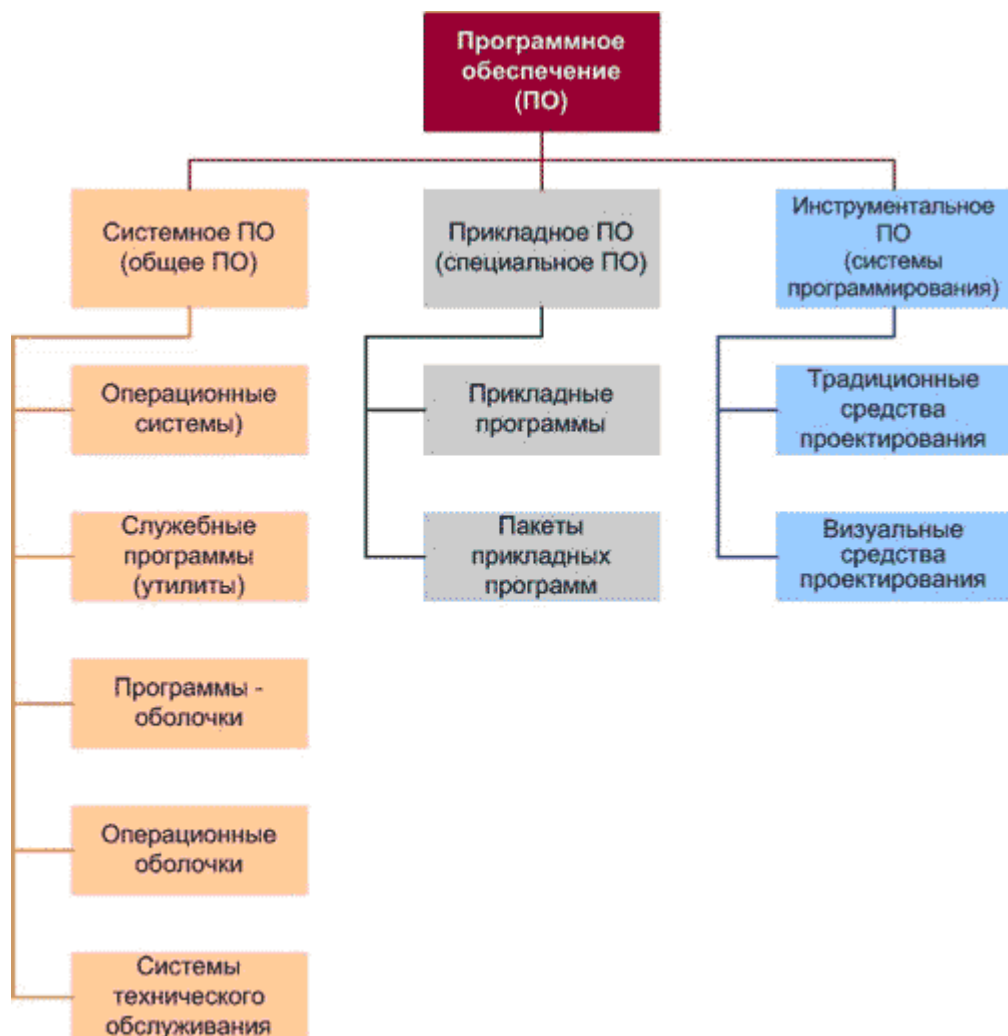


Рис. 1. Класифікація програмного забезпечення

Системне ПО

Системне програмне забезпечення – це комплекс програм, які забезпечують управління компонентами комп'ютерної системи, такими як

процесор, оперативна пам'ять, пристрої введення-виведення, мережеве обладнання, виступаючи в якості «міжшарового інтерфейсу», з одного боку якого апаратура, а з іншого – застосунки користувача. На відміну від прикладного програмного забезпечення, системне не вирішує конкретні практичні завдання, а лише забезпечує роботу інших програм, надаючи їм сервісні функції, які абстрагують деталі апаратної і програмної реалізації обчислювальної системи, управляє апаратними ресурсами обчислювальної системи.

До системного ПЗ належать:

- операційні системи;
- драйвери (програми, призначені для управління портами периферійних пристроїв);
- утиліти.

Утиліта – це допоміжна комп'ютерна програма в складі загального програмного забезпечення для виконання спеціалізованих типових задач, пов'язаних з роботою обладнання і операційної системи. Утиліти надають доступ до можливостей (параметри, налаштування, конфігурація), недоступним без їх застосування, або роблять процес зміни деяких параметрів простіше (автоматизують його).

Утиліти можуть входити до складу операційних систем, йти в комплекті зі спеціалізованим обладнанням або поширюватися окремо. До утиліт, зокрема, відносяться:

- утиліти керування процесами;
- диспетчери файлів або файлові менеджери;
- архіватори;
- засоби динамічного стиснення даних;
- засоби перегляду і відтворення;
- засоби контролю і діагностики, які дозволяють перевірити конфігурацію комп'ютера та перевірити працездатність пристроїв комп'ютера, перш за все жорстких дисків;
- утиліти відновлення після збоїв;
- оптимізатор диску – тип утиліти для оптимізації розміщення файлів на дисковому накопичувачі, наприклад, шляхом дефрагментації диска;
- деінсталятор;
- засоби забезпечення комп'ютерної безпеки (резервне копіювання, антивірусне ПЗ).

Необхідно відзначити, що частина утиліт входить до складу операційної системи, а інша частина функціонує автономно. Велика частина загального (системного) ПО входить до складу ОС. Частина загального ПО входить до

складу самого комп'ютера, зокрема, частина програм ОС і контролюючих тестів записана в постійному запам'ятовуючому пристрої системної плати. Частина загального ПО відноситься до автономних програм і поставляється окремо.

Прикладне ПО

Прикладні програми можуть використовуватися або автономно, або в складі програмних комплексів і пакетів.

Пакети прикладних програм – це система програм, які за сферою застосування діляться на проблемно-орієнтовані, пакети загального призначення і інтегровані пакети. До прикладного ПО, наприклад, відносяться:

- комплект офісних застосунків MS OFFICE,
- бухгалтерські системи і системи управління підприємствами,
- CAD-системи (системи автоматизованого проектування),
- браузері,
- графічні редактори,
- редактори HTML або Web - редактори,
- системи виконання математичних розрахунків і моделювання,
- експертні системи та таке інше

Інструментальне ПЗ

Інструментальне ПЗ або системи програмування - це системи для автоматизації розробки нових програм мовами програмування.

У найзагальнішому випадку для створення програми обраною мовою потрібно мати наступні компоненти:

1. **Текстовий редактор** для створення файлу з вихідним текстом програми.

2. **Компілятор** або інтерпретатор. Вихідний текст програми за допомогою програми-компілятора переводиться в проміжний об'єктний код. Вихідний текст великої програми складається з кількох *модулів* (файлів з вихідними текстами). Кожен модуль компілюється в окремий файл з об'єктним кодом, які потім треба об'єднати в одне ціле.

3. **Редактор зв'язків** або збирач або компоувальник, який виконує зв'язування об'єктних модулів і формує на виході працездатний застосунок – виконуваний модуль. Виконаний модуль – це закінчена програма в машинних кодах, яку можна запустити на будь-якому комп'ютері, де встановлена операційна система, для якої ця програма створювалася. Як правило, підсумковий файл має розширення EXE або COM.

Операційна система, її призначення і склад

Сучасний комп'ютер являє собою складну систему. Комп'ютер складається з одного або декількох процесорів, оперативної пам'яті, дисків, принтера, клавіатури, миші, дисплея, мережевих інтерфейсів та інших різноманітних пристроїв введення-виведення. Якщо кожному програмісту, що створює прикладну програму, потрібно буде розбиратися у всіх тонкощах роботи всіх цих пристроїв, то він не напише жодного рядка коду. Більш того, керування всіма цими компонентами і їх оптимальне використання являє собою дуже непросту задачу. З цієї причини комп'ютери оснащують спеціальним рівнем програмного забезпечення, який називається операційною системою, в чисе завдань входить управління призначеними для користувача програмами, а також всіма згаданими ресурсами.

Програми, з якими безпосередньо взаємодіють користувачі, зазвичай звані оболонкою, є двох типів: консоль, та графічний інтерфейс. Коли для спілкування з операційною системою використовують командний рядок, в якому вводяться команди в текстовому вигляді, то це – консоль. Графічний інтерфейс користувача (Graphical User Interface (GUI)) використовує значки, графічні об'єкти та таке інше, а керування проводиться переважно за допомогою маніпулятора «миша». Оболонка фактично не є частиною операційної системи, хоча задіюють цю систему в своїй роботі. Схематично основні компоненти програмного забезпечення в їх взаємодії представлено на рис. 2.

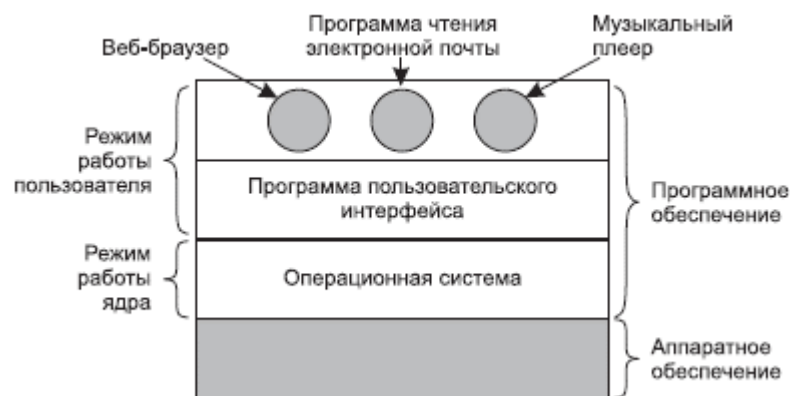


Рис. 2. Місце операційної системи в структурі програмного забезпечення

В нижній частині рисунку показано апаратне забезпечення. Воно складається з мікросхем, плат, дисків, клавіатури, монітора і інших фізичних об'єктів. Над апаратним забезпеченням знаходиться програмне забезпечення. Більшість комп'ютерів мають два режими роботи: режим ядра і режим користувача. Операційна система – найбільш фундаментальна частина програмного забезпечення, що працює в *режимі ядра* (цей режим називають

ще режимом супервізора). В цьому режимі вона має повний доступ до всього апаратного забезпечення і може задіяти будь-яку інструкцію, яку машина в змозі виконати. Вся інша частина програмного забезпечення працює в режимі користувача, в якому є лише підмножина інструкцій машини. Зокрема, програмам, що працюють в *режимі користувача*, заборонено використання інструкцій, які керують машиною або здійснюють операції введення-виведення (Input/Output - I/O).

Програми для користувача інтерфейсу - оболонка або GUI - знаходяться на найнижчому рівні програмного забезпечення, яке працює в режимі користувача. Вони дозволяють користувачеві запускати інші програми, такі як веб-браузер, програма читання електронної пошти або музичний плеєр. Ці програми також активно користуються операційною системою.

Місцезнаходження операційної системи показано на рис. 2. Вона працює безпосередньо з апаратним забезпеченням і є основою всього іншого програмного забезпечення. Важлива відмінність операційної системи від звичайного (що працює в режимі користувача) програмного забезпечення полягає в наступному: якщо користувач не задоволений конкретною програмою читання електронної пошти, то він може вибрати іншу програму або, якщо захоче, написати власну програму, але не може написати власний обробник переривань системного годинника, який є частиною операційної системи і захищений на апаратному рівні від будь-яких спроб внесення змін з боку користувача.

У багатьох системах також є програми, що працюють в режимі користувача, але тим не менш вони допомагають роботі операційної системи або виконують особливі функції. Наприклад, досить часто зустрічаються програми, які дозволяють користувачам змінювати їх паролі. Вони не є частиною операційної системи і не працюють в режимі ядра, але вони виконують важливу функцію і повинні бути особливим чином захищені. У деяких системах ця ідея доведена до крайньої форми, і ті області, які традиційно належали до операційної системи (наприклад, файлова система), працюють в просторі користувача. У таких системах важко провести чітку межу. Всі програми, що працюють в режимі ядра, безумовно, є частиною операційної системи, але деякі програми, що працюють поза цього режиму, можливо, також є її частиною або, принаймні, мають з нею тісний зв'язок.

Операційні системи відрізняються від програм користувача (тобто програм) не тільки місцем розташування. Їх особливість – досить великий обсяг, складна структура і тривалі терміни використання. Вихідний код основи операційної системи типу Linux або Windows займає близько 5 млн рядків. І це стосується тільки тієї частини, яка працює в режимі ядра. При включенні необхідних загальних бібліотек обсяг Windows перевищує 70 млн рядків коду,

не рахуючи основних прикладних програм (таких, як Windows Explorer, Windows Media Player і т.д.). Тому операційні системи живуть досить довго, – їх дуже важко створювати. Після створення однієї такої системи немає сенсу її викидати і приступати до створення нової. Тому операційні системи розвиваються протягом довгого періоду часу. Сімейство Windows 95/98/Me за своєю суттю представляло одну операційну систему, а сімейство Windows NT/2000/XP/Vista/Windows 7 – іншу, Windows 10 – ще іншу. Для користувача вони були схожі одна на одну, оскільки Microsoft подбала про те, щоб для користувача інтерфейс Windows 2000/XP/Vista/Windows 7 був дуже схожий на ту систему, якою він йшов на заміну, а найчастіше це була Windows 98. Іншим прикладом є операційна система UNIX, її варіанти і клони. Вона також розвивалася протягом багатьох років. Базуючись на вихідній системі вона існує в таких версіях, як System V, Solaris і FreeBSD. А ось Linux має нову програмну основу, хоча її модель дуже близька до UNIX і вона має високий ступінь сумісності з цією системою.

Функції операційної системи

Дати точне визначення операційної системи досить важко. Можна сказати, що це – програмне забезпечення, яке працює в режимі ядра. Операційні системи здійснюють дві дуже різні функції: надають прикладним програмістам (і прикладним програмам, природно) цілком зрозумілий абстрактний набір ресурсів замість неупорядкованого набору апаратного забезпечення, а також керують цими ресурсами. Розглянемо ці функції.

Абстрактні ресурси

Архітектура більшості комп'ютерів (система команд, організація пам'яті, введення-виведення даних та структура шин) на рівні машинної мови дуже примітивна і незручна для використання в програмах. Особливо це стосується систем введення-виведення. Тому обладнанням, наприклад жорстким диском, займається не програма користувача (для чого в кожному програмі користувача необхідно було б вводити спеціальні складні блоки програмних кодів), а та частина системного програмного забезпечення, яка називається драйвером диска і надає, не вдаючись у деталі, інтерфейс для читання і запису дискових блоків. Операційні системи містять безліч драйверів для управління самими різними пристроями введення-виведення.

Але для більшості застосунків занадто низьким є навіть цей рівень. Тому всі операційні системи надають ще один рівень абстракції для використання дисків – файли. Використовуючи цю абстракцію, програми можуть створювати, записувати і читати файли, не вникаючи в подробиці реальної роботи обладнання.

Ця абстракція є ключем до управління складністю. Хороша абстракція перетворює дуже складне завдання в два, вирішити які цілком під силу. Перша з цих задач полягає у визначенні та реалізації абстракцій, а друга – в використанні цих абстракцій для вирішення поточної проблеми. Одна з абстракцій, зрозуміла практично будь-якому користувачеві комп'ютера, – це вже згаданий раніше «файл». Він являє собою об'єм корисної інформації, скажімо, цифрову фотографію, або збережене повідомлення електронної пошти, або веб-сторінку. Працювати з фотографіями, повідомленнями електронної пошти та веб-сторінками набагато легше, ніж з особливостями SATA-дисків (або інших дискових пристроїв). Завдання операційної системи полягає в створенні гарної абстракції, а потім в реалізації абстрактних об'єктів, що створюються в рамках цієї абстракції, та в керуванні ними.

Існуючі процесори, блоки пам'яті, диски та інші компоненти являють собою дуже складні пристрої, що надають важкі, незручні, не схожі один на одного інтерфейси, які ще й з часом можуть змінюватися. Одна з головних задач операційної системи – приховати апаратне забезпечення і існуючі програми під незмінними абстракціями, створюваними замість них і пристосованими для нормальної роботи. Слід зазначити, що в дійсності «замовниками» операційних систем є прикладні програми (зрозуміло, не без допомоги прикладних програмістів). Саме вони безпосередньо працюють з операційною системою і її абстракціями. А кінцеві користувачі працюють з абстракціями, наданими призначеним для користувача інтерфейсом, – це або командний рядок оболонки, або графічний інтерфейс. Абстракції призначеного для користувача інтерфейсу можуть бути схожими на абстракції, що надаються операційною системою, але так буває не завжди. Щоб пояснити це положення, розглянемо звичайний робочий стіл Windows і командний рядок. І те й інше – програми, що працюють під управлінням операційної системи Windows і використовують надані цією системою абстракції, але вони пропонують користувача інтерфейси, які істотно відрізняються один від одного.

Операційна система як менеджер ресурсів

Уявлення про те, що операційна система головним чином надає абстракції для прикладних програм, – це погляд зверху вниз. Прихильники альтернативного погляду – знизу вгору, дотримуються тієї думки, що операційна система існує для управління всіма частинами складної системи. Сучасні комп'ютери складаються з процесорів, пам'яті, таймерів, дисків, маніпуляторів, мережевих інтерфейсів, принтерів і широкого спектру інших пристроїв. Прихильники погляду знизу вгору вважають, що завдання операційної системи полягає в забезпеченні впорядкованого і керованого

розподілу процесорів, пам'яті і пристроїв введення-виведення між різними програмами, які претендують на їх використання.

Сучасні операційні системи допускають одночасну роботу декількох програм. Уявіть собі, що буде, якщо всі три програми, що працюють на одному і тому ж самому комп'ютері, спробують роздрукувати свої вихідні дані одночасно на одному принтері. Перші кілька рядків роздруківки можуть бути від програми № 1, наступні кілька рядків – від програми № 2, потім кілька рядків від програми № 3 і т.д. В результаті вийде повний хаос. Операційна система покликана навести порядок в потенційно можливому хаосі за рахунок буферизації на диску всіх вихідних даних, призначених для принтера. Після того як одна програма закінчить свою роботу, операційна система зможе скопіювати її вихідні дані з файлу на диску, де вони були збережені, на принтер, а в той же самий час інша програма може продовжити генерацію даних, не помічаючи того, що вихідні дані фактично (до певного часу) не потрапляють на принтер.

Коли з комп'ютером (або з мережею) працюють кілька користувачів, потреба в управлінні та захисту пам'яті, пристроїв введення-виведення та інших ресурсів значно зростає, оскільки без цього користувачі будуть заважати один одному працювати. Крім цього, користувачам часто потрібно спільно використовувати не тільки апаратне забезпечення, а й інформацію (файли, бази даних і т. П.). Первинним завданням операційної системи є відстежування того, якою програмою який ресурс використовується, щоб задовольняти запити на використання ресурсів, нести відповідальність за їх використання і приймати рішення по конфліктуючим запитам від різних програм і користувачів.

Управління ресурсами включає в себе **мультиплексування** (розподіл) ресурсів двома різними способами: в часі і в просторі. Коли ресурс розділяється в часі, різні програми або користувачі використовують його по черзі: спочатку ресурс отримують в користування одні, потім інші і т.д. Наприклад, маючи в своєму розпорядженні лише один центральний процесор і декількома програмами, які прагнуть на ньому виконуватися, операційна система спочатку виділяє центральний процесор одній програмі, потім, після того як вона вже досить попрацювала, центральний процесор отримує в своє розпорядження інша програма, потім ще одна програма, і, нарешті, його знову отримує в своє розпорядження перша програма. Визначення того, як саме ресурс буде поділятися в часі – хто буде наступним споживачем і як довго, – це завдання операційної системи. Іншим прикладом мультиплексування в часі може послужити спільне використання принтера. Коли в черзі на друк до одного принтера знаходяться кілька завдань на друк, потрібно прийняти рішення, яке з них буде виконано наступним.

Іншим видом поділу ресурсів є просторове розділення. Замість почергової роботи кожен клієнт отримує якусь частину ресурсу. Наприклад, оперативна пам'ять зазвичай ділиться проміж кількох працюючих програм, так що всі вони одночасно можуть постійно перебувати в пам'яті (наприклад, використовуючи центральний процесор по черзі). За умови, якщо пам'яті достатньо для зберігання більш ніж однієї програми, ефективніше розмістити в пам'яті відразу кілька програм, ніж виділяти всю пам'ять одній програмі, особливо якщо їй потрібна лише невелика частина від загального простору. Зрозуміло, в цьому разі виникають проблеми рівного доступу, забезпечення безпеки і таке інше, і їх повинна вирішувати операційна система. Іншим ресурсом з поділом простору є жорсткий диск. У багатьох системах на одному і тому ж диску можуть одночасно зберігатися файли, що належать багатьом користувачам. Розподіл дискового простору і відстеження того, хто які дискові блоки використовує, – це типова задача операційної системи з керування ресурсами.

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Структура програмного забезпечення комп'ютерно-інтегрованих систем.
2. Системне програмне забезпечення, його призначення та склад.
3. Прикладне і інструментальне програмне забезпечення.
4. Операційна система, її призначення та склад.
5. Поняття абстрактних ресурсів.
6. Операційна система як менеджер ресурсів.

Лекція № 15

Тема: Операційні системи. Частина друга**Зміст**

Основні поняття і абстракції операційної системи	2
Процеси.....	2
Адресний простір.....	4
Файли	5
Введення-виведення даних	6
Безпека	6
Оболонка.....	7
Класи операційних систем	7
Операційні системи мейнфреймів	8
Серверні операційні системи.....	8
Багатопроцесорні операційні системи.....	8
Операційні системи персональних комп'ютерів	9
Операційні системи планшетів і смартфонів.....	9
Вбудовані операційні системи	9
Операційні системи сенсорних вузлів.....	10
Операційні системи реального часу	10
Операційні системи смарт-карт	11
Контрольні питання за темою.....	12
Рівень модуля	12
Рівень курсу.....	12

Джерела:

1. Таненбаум Э., Остин Т. Архитектура компьютера. 6-е изд. — СПб.: Питер, 2013. — 816 с.: ил.
2. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.
3. Бабак В.П. Теоретические основы информационно-измерительных систем: Учебник / В.П. Бабак, С.В. Бабак, В.С. Ерёменко и др. — К., 2014. — 832 с.

Основні поняття і абстракції операційної системи

Більшість операційних систем використовують певні основні поняття і абстракції, такі як процес, адресний простір і файл. Саме ці абстракції надають розуміння сутності операційних систем.

Процеси

Ключовим поняттям всіх операційних систем є процес. Процесом, по суті, є програма під час її виконання. З кожним процесом пов'язано його адресний простір – перелік адресів комірок пам'яті від нуля до деякого максимуму, звідки процес може зчитувати дані і куди може записувати їх. Адресний простір містить виконувану програму, дані цієї програми і її стек. Крім цього, з кожним процесом пов'язаний набір ресурсів, який зазвичай включає реєстри (в тому числі лічильник команд і показчик стека), список відкритих файлів, необроблені попередження, список пов'язаних процесів і всю іншу інформацію, необхідну в процесі роботи програми. Таким чином, процес – це контейнер, в якому міститься вся інформація, необхідна для роботи програми.

Для того щоб виробити інтуїтивне уявлення про процес, розглянемо систему, що працює в мультипрограмному режимі. Користувач може запустити програму редагування відео і вказати конвертування одногодинного відеофайлу в який-небудь певний формат (процес займе кілька годин), а потім переключитися на блукання по Інтернету. В цей час може запрацювати фоновий процес, який періодично «прокидається» для перевірки вхідних повідомлень електронної пошти. І у нас вже буде (як мінімум) три активних процеси: відеоредактор, веб-браузер і програма отримання (клієнт) електронної пошти. Періодично операційна система буде приймати рішення зупинити роботу одного процесу і запустити виконання іншого, можливо, через те, що перший вичерпав свою частку процесорного часу в попередню секунду або дві.

Якщо процес призупиняється таким чином, пізніше він повинен відновитися саме з того стану, в якому був зупинений. Це означає, що на період призупинення вся інформація про процес має бути явним чином десь збережена. Наприклад, в процесі можуть бути одночасно відкриті для читання кілька файлів. З кожним з цих файлів пов'язано показчик поточної позиції (тобто номер байту або запису, який має бути зчитаним наступним). Коли процес призупиняється, всі ці показчики повинні бути збережені, щоб виклик read, що виконується після відновлення процесу, приводив до читання потрібних даних. У багатьох операційних системах вся інформація про кожний процес, за винятком вмісту його власного адресного простору, зберігається в таблиці операційної системи, яка називається **таблицею**

процесів і являє собою масив (або пов'язаний список) структур, по одній на кожен з існуючих на даний момент процесів.

Таким чином, процес (в тому числі який був загальмований) складається з:

- ✓ власного адресного простору, яке зазвичай називають образом пам'яті,
- ✓ записів в таблиці процесів,
- ✓ вмісту реєстрів процесора на момент зупинки процесу,
- ✓ інша інформація, необхідна для подальшого відновлення процесу.

Системний виклик – це звернення прикладної програми (процесу) до операційної системи для виконання будь-якої операції.

Головними системними викликами, використовуваними для керування процесами, є виклики, пов'язані зі створенням і завершенням процесів. Розглянемо простий приклад. Процес, званий інтерпретатором команд, або оболонкою, зчитує команди з терміналу. Користувач тільки що набрав команду, що вимагає компіляції програми. Тепер оболонка повинна створити новий процес, який запускає компілятор. Коли цей процес завершить компіляцію, він зробить системний виклик для завершення власного існування.

Якщо процес здатний створювати кілька інших процесів (які в цьому разі називаються дочірніми процесами), а ці процеси в свою чергу можуть створювати власні дочірні процеси, то перед нами постає дерево процесів, подібне зображеному на рис. 1.

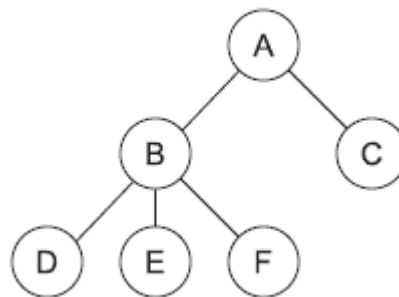


Рис. 1 – Дерево процесів. Процес А створив два дочірніх процеси: В і С.

Процес В створив три дочірні процеси: D, E і F

Пов'язані процеси, які спільно працюють над виконанням якогось завдання, часто потребують обміну даними один з одним та синхронізації своїх дій. Такий зв'язок називається взаємодією між процесами. Інші системні виклики, призначені для керування процесом, дозволяють запросити виділення додаткової пам'яті (або звільнення незадіяної), організувати

очікування завершення дочірнього процесу або завантаження будь-якої іншої програми поверх своєї.

Часом виникає потреба в передачі інформації запущеному процесу, який не перебуває у стані очікування цієї інформації. Можна привести в приклад процес, який обмінюється інформацією з іншим процесом, запущеним на іншому комп'ютері, і посилає віддаленому процесу повідомлення по мережі. Щоб застрахуватися від можливої втрати повідомлення або відповіді на нього, відправник може запросити власну операційну систему повідомити його після закінчення певного інтервалу часу, щоб він міг повторно відправити повідомлення, якщо не отримає підтвердження його отримання раніше. Після установки такого таймера програма може продовжити виконання іншої роботи.

Коли закінчиться заданий інтервал часу, операційна система посилає процесу сигнал тривоги. Цей сигнал змушує процес призупинити виконувану роботу, зберегти в стеку стан своїх регістрів і запустити спеціальну процедуру обробки сигналу тривоги, для того щоб, наприклад, заново передати ймовірно втрачене повідомлення. Коли обробник сигналу завершить свою роботу, запущений процес відновиться в тому самому стані, яке було до надходження сигналу. Сигнали є програмними аналогами апаратних переривань. Вони можуть генеруватися в різних ситуаціях, а не тільки після закінчення часу, встановленого в таймері. Багато апаратних переривань (наприклад, виконання неприпустимою команди або звернення за невірною адресою) також транслюються процесу, під час виконання якого сталася помилка.

Нагадаємо (лекція 3), що *переривання* (англ. Interrupt) – це сигнал від програмного або апаратного забезпечення, який повідомляє процесору про настання якої-небудь події, що вимагає негайної уваги. Переривання сповіщає процесор про настання високопріоритетної події, яка вимагає переривання поточного коду, що виконується процесором. Процесор відповідає припиненням своєї поточної активності, зберігає свій стан, і виконує функцію, звану оброблювач переривання (або програмою обробки переривання), яка реагує на подію і обслуговує її, після чого повертає управління в перерваний код.

Адресний простір

Кожен комп'ютер володіє певним обсягом оперативної пам'яті, використовуваної для зберігання виконуваних програм. У найпростіших операційних системах в пам'яті присутня тільки одна програма. Для запуску другої програми спочатку потрібно видалити першу, а потім на її місце завантажити в пам'ять іншу.

Більш складні операційні системи дозволяють одночасне перебування в пам'яті комп'ютера декількох програм. Щоб виключити взаємний негативний вплив програм одна на одну через неправильне звернення до пам'яті (одна програма звертається до ділянки оперативної пам'яті, де в цей час зберігаються дані або коди зовсім іншої програми або навіть самої операційної системи), що призведе до загального збою, необхідний захисний механізм. Цей механізм управляється операційною системою.

Крім необхідності захисту пам'яті, керування адресним простором процесів необхідно і з іншої причини. Зазвичай кожному процесу відводиться для використання деякий безперервний набір адрес, як правило, від нуля і до деякого максимуму. У найпростішому випадку максимальний обсяг адресного простору, що виділяється процесу, менше обсягу оперативної пам'яті. В цьому разі процес заповняє свій адресний простір і для його розміщення в оперативній пам'яті буде достатньо місця. На багатьох комп'ютерах використовується 32- або 64-розрядна адресація, що дозволяє мати адресний простір розміром 2^{32} або 2^{64} байт відповідно. Що станеться, якщо адресний простір процесу перевищує обсяг оперативної пам'яті, встановленої на комп'ютері, а процесу потрібно використовувати весь свій простір цілком? На перших комп'ютерах такий процес незмінно терпів крах. У наш час, на сучасних комп'ютерах, застосовується технологія віртуальної пам'яті, яка полягає в тому, що операційна система зберігає частину адресного простору в оперативній пам'яті, а частину – на диску, в разі потреби змінюючи їх фрагменти місцями. По суті, операційна система створює абстракцію адресного простору у вигляді набору адрес, на які може посилатися процес. Адресний простір відокремлено від фізичної пам'яті машини і він може бути як більше, так і менше неї. Управління адресними просторами і фізичною пам'яттю є важливою частиною роботи операційної системи.

Файли

Іншим ключовим поняттям, підтримуваним практично всіма операційними системами, є файлова система. Основна функція операційної системи – приховати специфіку дисків та інших пристроїв введення-виведення і надати програмісту зручну і зрозумілу абстрактну модель, що складається з незалежних від пристроїв файлів. Цілком очевидно, що для створення, видалення, читання і запису файлів знадобляться системні виклики. Перед тим, як файл буде готовий до читання, його має бути знайдено на диску і відкрито, а після зчитування – закрито. Для проведення цих операцій існують передбачені системні виклики.

Щоб надати місце для зберігання файлів, багато операційних систем використовують каталог як спосіб об'єднання файлів в групи. Для створення і

видалення каталогів потрібні системні виклики. Вони також потрібні для приміщення в каталог існуючого файлу і видалення його звідти. Елементами каталогу можуть бути або файли, або інші каталоги.

Кожен файл, що належить ієрархії каталогів, може бути позначено своїм повним ім'ям із зазначенням шляху до файлу, починаючи з вершини ієрархії – кореневого каталогу. Цей абсолютний шлях складається зі списку каталогів, які потрібно пройти від кореневого каталогу, щоб дістатися до файлу, де в якості роздільників компонентів служать символи косої риси (слеша).

У будь-який момент часу в кожного процесу є поточний робочий каталог, щодо якого розглядаються шляхи файлів, які не починаються з косої риси. Процес може змінити свій робочий каталог, скориставшись системним викликом, який визначає новий робочий каталог. Перед тим, як з файлом можна буде працювати в режимі запису або читання, він повинен бути відкритий. На цьому етапі відбувається також перевірка прав доступу. Якщо доступ дозволено, система повертає ціле число, зване дескриптором файлу, який використовується в подальших операціях. Якщо доступ заборонено, то повертається код помилки.

Введення-виведення даних

У всіх комп'ютерів є фізичні пристрої для отримання вхідної та виведення вихідної інформації. Існує маса різноманітних пристроїв введення-виведення: клавіатури, монітори, принтери і т.і., а також АЦП і ЦАП. Управління всіма цими пристроями покладається на операційну систему. Тому у кожній операційній системі для управління такими пристроями існує своя підсистема введення-виведення. Деякі програми введення-виведення не залежать від конкретного пристрою, тобто в рівній мірі підходять для застосування з багатьма або з усіма пристроями введення-виведення. Інша частина програм, наприклад драйвери пристроїв, призначена для певних пристроїв введення-виведення.

Безпека

Комп'ютери містять великий обсяг інформації, і часто користувачам потрібно захистити її, а також зберегти її конфіденційність. Можливо, це електронна пошта, бізнес-плани, податкові декларації і багато іншого. Керування безпекою системи також покладається на операційну систему: наприклад, вона повинна забезпечити доступ до файлів тільки користувачам, які мають на це право.

Щоб зрозуміти сам задум можливої організації роботи системи безпеки, звернемося в якості прикладу до системи UNIX. Файлам в UNIX присвоюється 9-розрядний двійковий код захисту. Цей код складається з трибітних полів. Одне поле – для власника файлу, друге – для представників групи, в яку

входить власник файлу (розділяє користувачів на групи системний адміністратор), третє – для всіх інших. У кожному полі є біт, який визначає доступ до читання, біт, що визначає доступ до запису, і біт, що визначає доступ до виконання. Ці три біта називаються rwx-бітами (read, write, execute). Наприклад, код захисту rwxr-x - x означає, що власнику доступні читання, запис або виконання файлу, іншим представникам його групи дозволяється читання або виконання файлу (але не запис), а всім іншим дозволено виконання файлу (але не читання або запис) . Дефіс (мінус) означає, що відповідний дозвіл відсутній.

Крім захисту файлів існує безліч інших аспектів безпеки. Один з них – це захист системи від небажаних вторгнень як за участю, так і без участі людей (наприклад, шляхом вірусних атак).

Оболонка

Операційна система являє собою програму, що виконує системні виклики. Не будучи частиною операційної системи, оболонка знайшла широке застосування як засіб доступу до багатьох її функцій. Вона також є основним інтерфейсом між користувачем, що сидить за своїм терміналом, і операційною системою, в випадку, коли не застосовується графічний користувальницький інтерфейс. Існує безліч оболонок. Оболонка запускається після входу в систему будь-якого користувача. У наші дні на більшості персональних комп'ютерів використовується графічний користувальницький інтерфейс (graphical user interface, GUI). По суті, графічний користувальницький інтерфейс – це просто програма (або сукупність програм), що працює поверх операційної системи на зразок оболонки. У системах Linux цей факт проявляється явно, оскільки у користувача є вибір принаймні з двох середовищ, що реалізують графічний користувальницький інтерфейс: Gnome і KDE. Або він може взагалі не вибирати жодну з них, скориставшись вікном терміналу. У Windows також є можливість замінити стандартний менеджер робочого столу (Windows Explorer) якоюсь іншою програмою шляхом внесення змін до деяких значення реєстру, хоча цією можливістю практично ніхто не користується.

Редактори, компілятори, асемблери, компоновщики, утиліти і інтерпретатори команд за визначенням не є частиною операційної системи. Вони відносяться до інструментального ПО.

Класи операційних систем

Історія операційних систем налічує вже понад півстоліття. За цей час було розроблено величезну кількість різноманітних операційних систем, але не всі вони отримали широку популярність. В даному розділі коротко, до відома, розглянемо класи операційних систем.

Операційні системи мейнфреймів

До вищої категорії відносяться операційні системи мейнфреймів (великих універсальних машин) – комп'ютерів, що займають цілі зали і їх до сьогодні ще можна зустріти у великих центрах обробки корпоративних даних. Такі комп'ютери відрізняються від персональних комп'ютерів обсягами введення-виведення даних. Мейнфрейми, які мають тисячі дисків і петабайт даних, – це вельми звичайне явище. Мейнфрейми також знаходять застосування в якості потужних веб-серверів, серверів великих інтернет-магазинів і серверів, що займаються міжкорпоративними транзакціями. Операційні системи мейнфреймів орієнтовані переважно на одночасну обробку безлічі завдань, більшість з яких вимагає колосальних обсягів введення-виведення даних. Робота в режимі поділу часу дає можливість великій кількості віддалених користувачів одночасно запускати на комп'ютері свої завдання, наприклад робити запити до великої бази даних. Всі ці функції тісно пов'язані один з одним, і часто операційні системи універсальних машин виконують їх в комплексі. Прикладом операційної системи універсальних машин може послужити OS/390 – спадкоємиця OS/360. Однак ці операційні системи поступово витісняються варіантами операційної системи UNIX, наприклад Linux.

Серверні операційні системи

Трохи нижче за рівнем стоять серверні операційні системи. Вони працюють на серверах, які представлені дуже потужними персональними комп'ютерами, робочими станціями або навіть універсальними машинами. Вони одночасно обслуговують по мережі безліч користувачів, забезпечуючи їм загальний доступ до апаратних та програмних ресурсів. Сервери можуть надавати послуги друку, зберігання файлів або веб-служб. Інтернет-провайдери для обслуговування своїх клієнтів зазвичай задіють відразу декілька серверних машин. Під час обслуговування веб-сайтів сервери зберігають веб-сторінки і обробляють запити, які до них надходять. Типовими представниками серверних операційних систем є Solaris, FreeBSD, Linux і Windows Server 201x.

Багатопроцесорні операційні системи

Зараз все ширше використовується об'єднання багатьох центральних процесорів в єдину систему, що дозволяє домогтися великої обчислювальної потужності. Залежно від того, як саме відбувається це об'єднання, а також які використовуються ресурси загального користування, ці системи називаються паралельними комп'ютерами, мультикомп'ютерами або багатопроцесорними системами. Їм потрібні спеціальні операційні системи, в якості яких часто застосовуються особливі версії серверних операційних систем, оснащені

спеціальними функціями зв'язку, сполучення і синхронізації. З появою багатоядерних процесорів для персональних комп'ютерів операційні системи навіть звичайних настільних комп'ютерів і ноутбуків стали працювати щонайменше з невеликою багатопроцесорною системою. Згодом число ядр буде тільки рости. На багатопроцесорних системах можуть працювати багато популярних операційних систем, включаючи Windows і Linux.

Операційні системи персональних комп'ютерів

До наступної категорії відносяться операційні системи персональних комп'ютерів. Всі їх сучасні представники підтримують багатозадачність. Досить часто вже в процесі завантаження десятки програм запускаються на одночасне виконання. Завданням операційних систем персональних комп'ютерів є якісна підтримка роботи окремого користувача. Вони широко використовуються для обробки текстів, створення електронних таблиць, ігор та доступу до Інтернету. Типовими прикладами можуть служити операційні системи Linux, FreeBSD, Windows 7, Windows 8, Windows 10 і OS X (macOS) компанії Apple.

Операційні системи планшетів і смартфонів

Продовжуючи рухатися по низхідній до все більш простих систем, ми дійшли до планшетів, смартфонів та інших кишенькових комп'ютерів. Ці комп'ютери, спочатку відомі як КПК, або PDA (Personal Digital Assistant - персональний цифровий секретар), являють собою невеликі комп'ютери, які під час роботи тримають в руці. Найвідомішими їх представниками є смартфони і планшети. На цьому ринку домінують операційні системи Android від Google і iOS від Apple, але у них є безліч конкурентів. Більшість таких пристроїв обладнано багатоядерними процесорами, GPS, камерами та іншими датчиками, достатнім обсягом пам'яті та складними операційними системами.

Вбудовані операційні системи

Вбудовані системи працюють на комп'ютерах, які керують різними пристроями. Оскільки на цих системах установка програм користувачем не передбачається, то їх зазвичай комп'ютерами не вважають. Прикладами пристроїв, де встановлюються вбудовані комп'ютери, є мікрохвильові печі, телевізори, автомобілі, музикальні центри, плеєри, звичайні телефони. В основному, вбудовані системи відрізняються тим, що на них ні за яких умов не працюватиме стороннє програмне забезпечення. У мікрохвильову піч неможливо завантажити новий застосунок, оскільки всі її програми записані в постійну пам'ять. Отже, відпадає необхідність в захисті застосунків один від

одного і операційну систему можна спростити. Найбільш популярними в цій області вважаються операційні системи Embedded Linux, QNX і VxWorks.

Операційні системи сенсорних вузлів

Мережі, складені з мініатюрних сенсорних вузлів, пов'язаних один з одним і з базовою станцією по бездротових каналах, розгортаються для різних цілей. Такі сенсорні мережі використовуються для захисту периметрів будівель, охорони державного кордону, виявлення загорянь в лісі, вимірювання температури і рівня опадів в цілях складання прогнозів погоди, збору інформації про переміщення противника на полі бою і багато чого іншого. Вузли такої мережі є мініатюрними комп'ютерами, що живляться від батареї і мають вбудовану радіосистему. Вони обмежені за потужністю і повинні працювати тривалий період часу в необслуговуваному режимі на відкритому повітрі, часто в складних кліматичних умовах. Мережа повинна бути досить надійною і допускати відмови окремих вузлів, що в міру втрати ємності батареєю живлення буде траплятися все частіше. Кожен сенсорний вузол є справжнім комп'ютером, оснащеним процесором, оперативною пам'яттю і постійним запам'ятовуючим пристроєм, а також одним або декількома датчиками. На ньому працює невелика, але справжня операційна система, зазвичай керована подіями, яка відгукується на зовнішні події або періодично проводить вимірювання за сигналами вбудованого годинника. Операційна система повинна бути невеликою за обсягом і нескладною, оскільки основними проблемами цих вузлів є мала ємність оперативної пам'яті і обмежений час роботи батареї. Так само, як і в вбудованих системах, всі програми є попередньо завантаженими, і користувачі не можуть запустити програму, завантажену з Інтернету, що значно спрощує всю конструкцію. Прикладом широко відомої операційної системи для сенсорних вузлів може послужити TinyOS.

Операційні системи реального часу

Ще один різновид операційних систем – це системи реального часу. Ці системи характеризуються тим, що для них ключовим параметром є час. Наприклад, в системах керування виробничими процесами комп'ютери, що працюють в режимі реального часу, повинні збирати відомості про процес і використовувати їх для керування верстатами на підприємстві. Досить часто вони повинні відповідати дуже жорстким вимогам по часу. Наприклад, коли автомобіль переміщається через складальний конвеєр, то в певні моменти часу повинні здійснюватися цілком конкретні операції. Якщо, наприклад, зварювальний робот приступить до зварювання з випередженням або запізненням, машина прийде в непридатність. Якщо операція повинна бути проведена точно в строк (або в певний період часу), то ми маємо справу з

системою **жорсткого реального часу**. Безліч подібних систем зустрічається в керуванні виробничими процесами, в авіаційно-космічному електронному обладнанні, у військовій та інших подібних областях застосування. Ці системи повинні надавати абсолютні гарантії того, що певні дії будуть здійснюватися точно в конкретний момент часу. Системами жорсткого реального часу є такі системи як QNX, RTOS, VxWorks.

Іншим різновидом подібних систем є система м'якого реального часу, в якій хоча і небажано, але цілком допустимо недотримання терміну будь-якої дії і це не завдає непоправної шкоди. До цієї категорії відносяться цифрові аудіо- або мультимедійні системи. Смартфони також є системами м'якого реального часу.

Оскільки до систем реального часу пред'являються дуже жорсткі вимоги, іноді операційні системи являють собою просту бібліотеку, пов'язану з прикладними програмами, де все тісно взаємопов'язано, і між частинами системи не існує ніякого захисту. Прикладом такої системи може послужити eCos. Категорії операційних систем для КПК, вбудованих систем і систем реального часу в значній мірі перекриваються один з одним за властивими їм ознаками. Практично всі вони мають принаймні деякі аспекти систем м'якого реального часу. Вбудовані системи і системи реального часу працюють тільки з тим програмним забезпеченням, яке вклали в них розробники цих систем; користувачі не можуть додати в цей арсенал власне програмне забезпечення, що істотно полегшує вирішення завдань захисту. КПК і вбудовані системи призначені для індивідуальних споживачів, а системи реального часу частіше використовуються в промисловому виробництві. Проте, не дивлячись на все це, у них є певна кількість спільних рис.

Операційні системи смарт-карт

Найменші операційні системи працюють на смарт-картах. Смарт-карта являє собою пристрій розміром з кредитну карту, яка має власний процесор. На операційні системи для них накладаються дуже жорсткі обмеження щодо необхідної обчислювальної потужності процесора і обсягу пам'яті. Деякі з смарт-карт отримують живлення через контакти зчитувального пристрою, в яке вони вставляються. Інші – безконтактні смарт-карти, – отримують живлення за рахунок ефекту індукції, що істотно обмежує їх можливості. Деякі з них здатні впоратися лише з однією функцією, наприклад з електронними платежами, але існують і багатофункціональні смарт-карти.

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Поняття "процес" в операційних системах.
2. Поняття "адресний простір" в операційних системах.
3. Поняття "файл" в операційних системах.
4. Оболонка операційної системи.
5. Безпека в операційній системі.
6. Класи операційних систем.
7. Операційні системи реального часу.

Лекція № 16

Тема: Хмарні технології**Зміст**

Хмарні обчислення	2
Моделі топології хмар	3
Приватна хмара	3
Публічна хмара	4
Гібридна хмара	4
Хмарні сервіси	4
Інфраструктура як послуга	6
Віртуальна машина	7
Системні віртуальні машини	8
Прикладні віртуальні машини	8
Гіпервізор	8
Гіпервізор як система	9
Типи гіпервізорів	10
Контрольні питання за темою	11
Рівень модуля	11
Рівень курсу	11

Джерела:

1. Таненбаум Э., Бос Х. Современные операционные системы. 4-е изд. — СПб.: Питер, 2015. — 1120 с.: ил.

Хмарні обчислення

Хмарні обчислення (англ. Cloud Computing) – модель забезпечення повсюдного та зручного доступу через мережу до спільного комплексу обчислювальних ресурсів (комунікаційних мереж, серверів, засобів збереження даних, прикладних програм та сервісів), які можуть налаштовуватися, і які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера.

Поняття «хмара» відноситься до інфраструктури обчислювальних служб, які зазвичай виділяються відповідно до запиту на них. Набір ресурсів (обчислень, мереж, сховищ і пов'язаних з ними програмних сервісів) може динамічно масштабуватися в бік збільшення або зменшення в залежності від середнього навантаження і якості обслуговування. Хмари, як правило, – це великі центри обробки даних, які надають клієнтам послуги, орієнтовані на зовнішнього споживача. Ці центри створюють ілюзію єдиного хмарного ресурсу, в той час як насправді може бути використано багато географічно розподілених ресурсів. Це дає користувачеві відчуття незалежності від місця розташування. Ресурси є еластичними (що означає масштабованість), а сервіси – це еквівалент плати за використання. Сервіси, які працюють в хмарі, відрізняються від традиційного програмного забезпечення своєю конструкцією та реалізацією. Хмарні застосунки можуть розроблятися і розвиватися швидше і в меншій мірі залежати від мінливості середовища. Таким чином, розгортання хмар відбувається дуже швидко.

Поняття «хмара» міцно увійшло в життя звичайних людей. Необхідність наявності локальних жорстких дисків та інших накопичувачів для зберігання інформації вже не є актуальною. Власники бізнесу все частіше звертаються до хмарних провайдерів для розгортання або міграції своїх ІТ-ресурсів. В цій лекції буде розглянуто, що таке хмарні сервіси, та якими вони бувають.

Хмарні сервіси – це платформи та програми, які "живуть" і працюють на серверах спеціальних компаній – хмарних провайдерів. Ці програми не потрібно встановлювати на комп'ютер, а доступ до них можна отримати з будь-якої точки світу. Dropbox, Google Drive, iCloud – всі ми давно користуємось цими та схожими продуктами, навіть не усвідомлюючи, що вони розміщені у хмарах. Немає сенсу тримати файли на власному комп'ютері, набагато простіше зберігати дані та працювати з ними на зручних та захищених хмарних платформах.

Схожа ситуація і з бізнесом. Хмарне сховище надається користувачу в необхідному обсязі, оплачується за фактом використання і позбавляє від необхідності купувати власну ІТ-інфраструктуру для зберігання даних і керувати нею. Це забезпечує гнучкість, швидку масштабованість та надійність. Наприклад, зберігати 1С в хмарному сховищі набагато безпечніше,

ніж на власному сервері, який знаходиться в звичайному офісі. Сервер може вийти з ладу, може бути атакований вірусами, або, взагалі, його може бути викрадено. В період масового розпродажу товару, власник інтернет-магазину зможе просто замовити більше ресурсів у хмарного провайдера і не перейматися, що сайт відмовить.

Хмарні сховища даних дозволяють розміщувати та зберігати великі об'єми інформації, на відміну від традиційних серверів та ПК.

Моделі топології хмар

У хмарному середовищі існують три різні моделі топології хмар, які зазвичай використовуються: приватна хмара (private cloud), хмара загального користування (public cloud) та гібридна хмара (Hybrid cloud). Незалежно від моделі, хмарні ресурси повинні забезпечувати динамічну масштабованість, швидкість розробки та розгортання, а також доступність з будь-якого місця. Сучасні корпоративні системи, як правило, використовують гібридну архітектуру: для забезпечення безпеки критично важливих застосунків і даних використовують приватні хмари, та використовують публічні хмари для підключення, простоти і швидкості розгортання.

- загального користування (**Public Cloud**) - хмарна інфраструктура, за якої серверні потужності обов'язково знаходяться на стороні провайдера хмарних послуг, який надає ресурси відразу кільком організаціям з одного хмари
- приватна хмара (**Private cloud**) – хмарна інфраструктура, надана в виключне користування одній організації
- гібридна хмара (**Hybrid cloud**) – одночасне використання приватної і публічної хмари

Приватна хмара

У приватній хмарі інфраструктура надана одній організації або корпорації. Немає концепції спільного використання ресурсів або об'єднання за межами власної інфраструктури власника. У приміщеннях спільне використання та розпорядження ресурсами є загальними. Приватна хмара існує за рядом причин, включаючи безпеку та перевіреність якості. Тобто, для гарантії, що інформація обробляється виключно системами, керованими клієнтом. Однак, щоб вважатися хмарою, повинні існувати деякі аспекти хмарних сервісів, такі як віртуалізація і балансування навантаження. Приватна хмара може бути локальною або може бути спеціалізованим обладнанням, що надаються провайдером виключно для використання цим клієнтом.

Публічна хмара

Публічна хмара – протилежна ситуація приватній. Тут інфраструктура надається на вимогу для великої кількості клієнтів та застосунків. Інфраструктура являє собою набір ресурсів, які будь-яка людина може використовувати в будь-який час в рамках своїх угод про рівень обслуговування. Перевага тут у тому, що хмарні центри обробки даних дозволяють забезпечити безпрецедентну масштабованість для багатьох клієнтів, які обмежені тільки тим, яку частину послуг вони хочуть придбати.

Гібридна хмара

Гібридна архітектурна модель являє собою поєднання приватних і публічних хмар. Такими комбінаціями можуть бути декілька публічних хмар, які використовуються одночасно або комбінація публічної та приватної хмарної інфраструктури.

Хмарні сервіси

Хмарні сервіси (cloud services) – це програми та платформи, які «живуть» та працюють на серверах хмарних провайдерів. Головна особливість хмарних сервісів полягає в тому, що створюючи акаунт на такій платформі, клієнт зможе отримувати доступ до власної інформації з будь-якого пристрою в будь-якій точці світу. Для цього потрібно мати логін та пароль. Використовувати ці сервіси не лише зручно, але й безпечно. Навіть якщо з телефоном або комп'ютером щось станеться, ваші дані не зникнуть.

Моделі надання хмарних сервісів

Існує більше десяти моделей надання cloud services. Хмарні оператори виділяють три найпоширеніші з них:

- **IaaS** (Infrastructure as a Service – Інфраструктура як Послуга) – надання замовнику в оренду обчислювальних ресурсів, у вигляді віртуальної інфраструктури.
- **PaaS** (Platform as a Service – Платформа як Послуга) – клієнт отримує повноцінну віртуальну платформу з різними інструментами та сервісами.
- **SaaS** (Software as a Service – Програмне забезпечення як Послуга) – клієнт отримує в своє розпорядження певні програмні продукти за допомогою мережі інтернет.

Як бачите, усюди є приставка as a Service. Це означає, що всі види хмар надаються за моделлю підписки, тобто ви використовуєте їх тільки тоді, коли в них є необхідність. Наступна діаграма демонструє різницю між різними видами хмарних сервісів.

Separation of Responsibilities

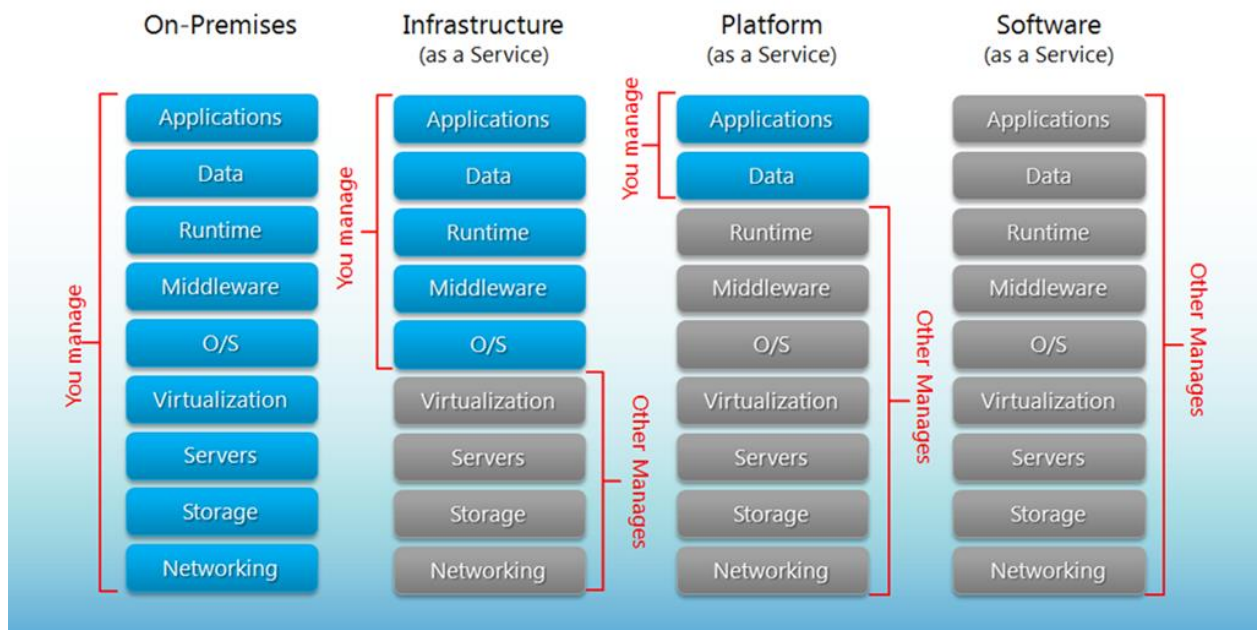


Рис. 1 – Розподіл відповідальності між користувачем та провайдером для різних хмарних сервісів

Отже, почнемо з типу хмарних технологій, представники якого найбільш відомі — Software as a Service. Даний вид передбачає надання готового рішення клієнту з мінімальною необхідністю налаштування. Тобто теоретично, підписуючись на такий сервіс, керувати ним може будь-який користувач з мінімальним залученням системного адміністратора або взагалі без нього. Найвідоміші представники такого сервісу у корпоративному середовищі – це Office 365.

Послуги типу PaaS розраховані в першу чергу на розробників. Вони являють собою набори готових компонентів для створення застосунків, а також засоби керування платформою. В даному випадку компонентами будуть сервіси даних, репозиторії, інструменти автоматизованої загрузки, середовища тестування і тому подібні сервіси. Приклади PaaS-сервісів – Google AppEngine, VMWare Pivotal Cloud Foundry, Red Hat's OpenShift, Heroku та ін.

І нарешті найближчий до системних адміністраторів тип сервісу – це IaaS. Інфраструктура як послуга за своїми об'єктами та характеристиками найбільш наближена до володіння власним "залізом" та віртуалізацією. У випадку з IaaS користувач отримує у своє розпорядження хмарні процесори, пам'ять, диски та мережі, з яких згодом створює сервери-маршрутизатори та налаштовує мережеву топологію таким чином, як йому це потрібно.

Приклади хмарних сервісів SaaS:

Google One – це хмарний застосунок для зберігання, редагування та синхронізації файлів, розроблений компанією Google. Можливості:

- ✓ Можна зберігати файли будь-якого формату: фото, відео, аудіо. У кожному акаунті Google безкоштовно доступно 15 ГБ пам'яті.
- ✓ Відсутня прив'язка до конкретної операційної системи.
- ✓ Файли на диску можна відкривати зі смартфона, планшета або комп'ютера.
- ✓ Щоб інші користувачі могли переглядати, редагувати і завантажувати файли користувача, потрібно лише відправити їм запрошення.

Microsoft Office 365 – це хмарний сервіс, який поєднує зручні сучасні інструменти для роботи, що поширюються на основі передплати. У пакет Microsoft Office 365 входить:

Електронна пошта бізнес-класу на сервері Exchange.

- ✓ Портал Sharepoint і публічний сайт-візитка з простим конструктором сторінок.
- ✓ Комунікатор Lync, за допомогою якого можна обмінюватися текстовими фразами, проводити відео- та аудіоконференції, здійснювати показ робочого столу.
- ✓ Доступ до застосунків останньої версії Microsoft Office.
- ✓ Доступ до застосунків Microsoft Office 2010, 2013 і 2016 корпоративному тарифному плані.
- ✓ Місце в OneDrive (1 терабайт на користувача).

Інфраструктура як послуга

В межах моделі обслуговування «Інфраструктура як послуга» IaaS споживачу надається можливість керувати засобами обробки та збереження, комунікаційними мережами, та іншими фундаментальними обчислювальними ресурсами, на базі яких споживач може розгортати та виконувати довільне програмне забезпечення, до складу якого можуть входити операційні системи та прикладні програми. Споживач не керує фізичною та віртуальною інфраструктурою, що лежить в основі хмари, проте він контролює операційні системи, системи збереження, встановлені програми. IaaS складається з трьох основних компонентів:

- Апаратні засоби (сервери, системи зберігання даних, клієнтські системи, мережеве обладнання)
- Операційні системи та системне ПЗ (засоби віртуалізації, автоматизації, основні засоби управління ресурсами)
- Зв'язуюче ПЗ (наприклад, для управління системами)

Для того, щоб зрозуміти, як працює цей сервіс, необхідно ознайомитись з поняттям «Віртуальна машина» та «Гіпервізор».

Віртуальна машина

Віртуальна машина – модель обчислювальної машини, створеної шляхом віртуалізації обчислювальних ресурсів: процесора, оперативної пам'яті, пристроїв зберігання та вводу і виводу інформації. Віртуальна машина, на відміну від програми емуляції конкретного пристрою, забезпечує повну емуляцію фізичної машини чи середовища виконання (для програми).

Раніше віртуальну машину визначали як «ефективну ізольовану копію реальної машини». Проте сучасні віртуальні машини можуть не мати прямого апаратного аналогу. Наприклад, в залежності від способу моделювання набору інструкцій віртуального центрального процесора, віртуальна машина може моделювати реальну або абстрактну обчислювальні машини. Під час моделювання (емуляції) реальної обчислювальної машини набір інструкцій процесора віртуальної машини збігається з набором інструкцій обраного для моделювання центрального процесора.

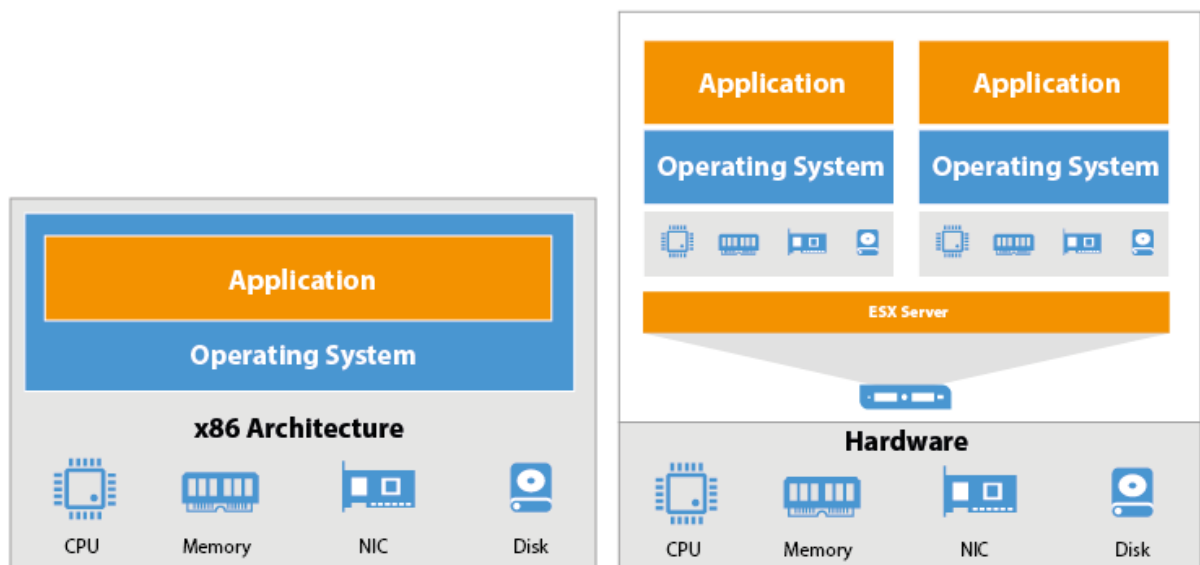


Рис. 2 Віртуальна машина зліва – прикладна, справа – системна

Віртуальні машини поділяються на 2 головні категорії, в залежності від їх використання та відповідності до реальної апаратури (рис. 2):

- системні (апаратні) віртуальні машини, що забезпечують повноцінну емуляцію всієї апаратної платформи і відповідно підтримують виконання операційної системи.
- прикладні віртуальні машини, які розроблені для виконання лише застосунків (прикладних програм), наприклад, віртуальна машина Java.

Системні віртуальні машини

Системні віртуальні машини дозволяють розподіл апаратних ресурсів фізичної машини між різними копіями віртуальних машин, на кожній з яких може бути встановлена своя операційна система. Пласт програмного забезпечення, що виконує віртуалізацію, називається гіпервізором. Гіпервізори поділяються на 2 типи: ті, що можуть виконуватися на «голій» апаратурі (1-й тип, або рідні (англ. native)), та ті, що виконуються в певній операційній системі (2-й тип, або хостові).

Основні переваги системних ВМ:

- різні операційні системи можуть співіснувати на одному комп'ютері, проте вони знаходяться в строгій ізоляції одна від одної
- ВМ можуть забезпечувати розширений набір машинних інструкцій, адже під час моделювання (емуляції) абстрактної обчислювальної машини набір інструкцій процесора віртуальної машини може бути довільним.
- широкі можливості контролю за програмами
- легкість модифікацій та відновлення

Основний недолік:

віртуальна машина не така ефективна як реальна, тому що доступ до апаратури в ній відбувається опосередковано.

Різні ВМ, на кожній з яких може бути встановлена своя власна ОС (які також називається гостьовими ОС), часто використовуються для серверного об'єднання: різні сервіси (що повинні виконуватися на окремих машинах, щоб запобігти взаємовтручанню) запускаються в різних ВМ, проте на одній фізичній машині, що дозволяє економити апаратні ресурси.

Прикладні віртуальні машини

Прикладні віртуальні машини виконують звичайні програми всередині ОС. Вони зазвичай створюються коли програма запускається та знищуються після її завершення. Їхня ціль — забезпечити платформно-незалежне програмне середовище, яке дозволяє абстрагуватися від конкретної апаратури та операційної системи, на якій виконується програма.

Гіпервізор

Гіпервізор або Монітор віртуальних машин – комп'ютерна програма або обладнання процесора, що забезпечує на одному фізичному комп'ютері (який зветься хост-машина або хост-комп'ютер, англ. host computer) одночасне паралельне виконання декількох віртуальних машин, на кожній з яких виконується власна операційна система. Гіпервізор забезпечує взаємну

ізоляцію операційних систем, що виконуються на віртуальних машинах, шляхом розділення фізичних та логічних пристроїв між декількома віртуальними машинами.

Гіпервізор також може надавати операційним системам, що працюють під його керуванням, засоби зв'язку і взаємодії між собою (наприклад, через обмін файлами або через віртуальні мережеві з'єднання) так, ніби ці ОС виконувалися на різних фізичних комп'ютерах, та створювати логічні пристрої, моделюючи апаратне забезпечення на хост-машині, яке насправді не існує.

Гіпервізор як система

Гіпервізор сам по собі є мінімалістичною операційною системою, що складається з мікро- або нано- ядра та інструментів керування віртуальними машинами. Він створює віртуальні машини, на яких можуть бути запущені окремі операційні системи, віртуалізуючи або емулюючи апаратне забезпечення (в тому числі процесор, оперативну пам'ять та пристрої введення/виведення), і керує цими віртуальними машинами.

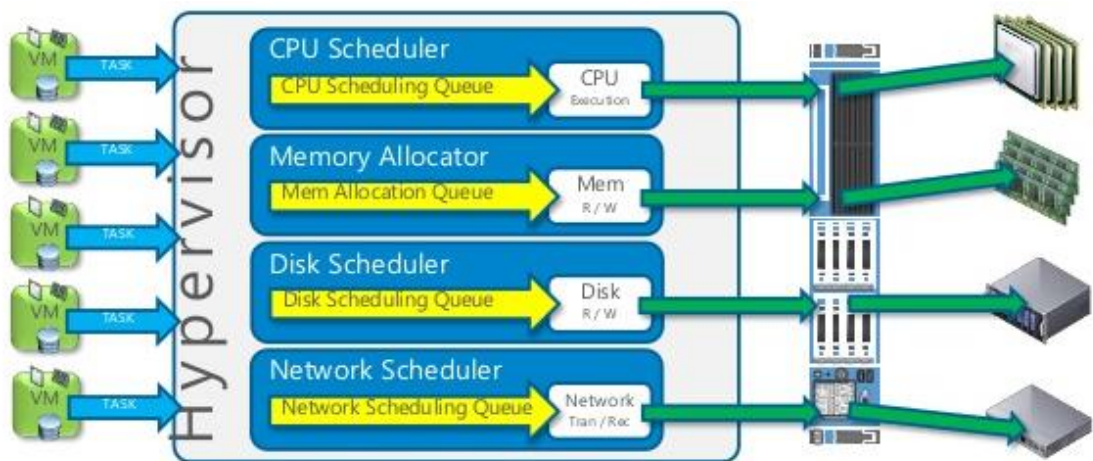


Рис. 3 – Гіпервізор як менеджер ресурсів

Гіпервізор забезпечує взаємну ізоляцію віртуальних машин шляхом надання пристроїв (у тому числі процесора та оперативної пам'яті) в один з способів (рис. 3):

- розділення часу, коли деякий пристрій надається окремій віртуальній машині на деякий час (звичайно декілька мілісекунд);
- закріплення, коли деякий пристрій надається окремій віртуальній машині в одноосібне володіння;
- емуляції, коли деякий пристрій створюється програмно для кожної окремої віртуальної машини.

Гіпервізор дозволяє незалежне «включення», «перезавантаження», «вимкнення» кожної з віртуальних машин з тією чи іншою ОС. В цьому разі операційна система, що працює у віртуальній машині під управлінням гіпервізора, може «знати» (проте – не обов’язково), що вона виконується у віртуальній машині, а не на реальному апаратному забезпеченні.

Типи гіпервізорів

Автономний гіпервізор (Тип 1)

Має свої вбудовані драйвери пристроїв, моделі драйверів і планувальник, і тому не залежить від базової ОС. Оскільки автономний гіпервізор працює безпосередньо на обладнанні (рис. 2 справа), він більш продуктивний.

Приклади: VMware ESX, PR/SM

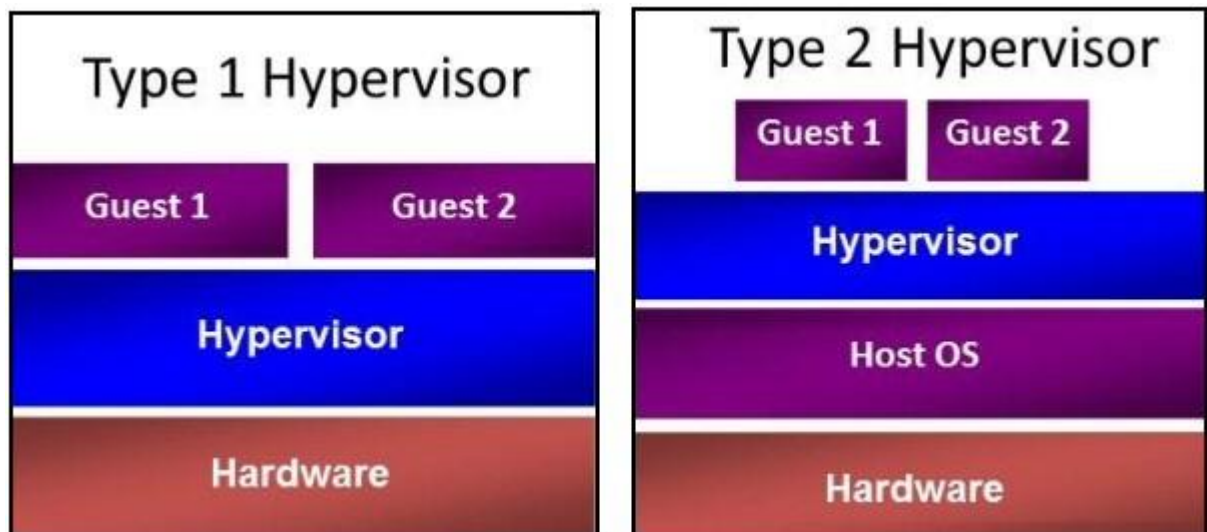


Рис. 4 –Типи гіпервізорів

На основі базової ОС (Тип 2)

Це компонент, який працює під керуванням основної ОС. Гостьовий код може виконуватися безпосередньо на фізичному процесорі, але доступ до пристроїв вводу-виводу комп'ютера з гостьової ОС здійснюється через другий компонент, звичайний процес основної ОС – монітор рівня користувача.

Приклади: Microsoft Virtual PC, VMware Workstation, QEMU, Parallels, VirtualBox.

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Поняття хмарних обчислень.
2. Моделі топології хмар.
3. Моделі хмарних сервісів.
4. Системні та прикладні віртуальні машини.
5. Гіпервізор.

Лекція № 17

Тема: Технологія супутникової навігації на прикладі системи GPS**Зміст**

Принцип вимірювання транзитного часу сигналу	3
Основні принципи супутникової навігації.....	3
Час проходження сигналу	6
Визначення позиції на мапі	6
Поява похибки часу та її корекція	8
Технологія супутникової навігації на прикладі системи GPS	9
Опис системи.....	9
Розміщення та рух супутників	9
Супутники GPS	11
Конструкція супутника	11
Аналіз зв'язку	11
Сигнали супутника.....	12
Блокова діаграма генерації сигналу супутника	13
Детальна блокова система генерації сигналу супутника	14
Керуючий сегмент.....	15
Сегмент користувача	15
Повідомлення GPS	17
Введення	17
Структура навігаційного повідомлення	18
Інформація в підфреймах	19
TLM і HOW	19
Розбиття на 25 сторінок	19
Порівняння даних альманаху і ефемерид	20
Контрольні питання за темою.....	21
Рівень модуля	21
Рівень курсу	21

Джерела:

1. GPS. Essentials of Satellite Navigation. Compendium. GPS-X-02007-D. Copyright © 2009, u-blox AG
Режим доступу: https://www.u-blox.com/sites/default/files/products/documents/GPS-Compendium_Book_%28GPS-X-02007%29.pdf

Принцип вимірювання транзитного часу сигналу

Якийсь раз під час грозовий ночі Ви, безсумнівно, намагалися зрозуміти, як далеко Ви знаходитесь від спалаху блискавки. Відстань можна встановити досить легко (Рис. 1): відстань = це час від моменту спалаху блискавки (стартовий час) до появи грому (кінцевий час), помножений на швидкість звуку (приблизно 330 м/с.). Різниця між стартовим та кінцевим часом і є *транзитний час*.

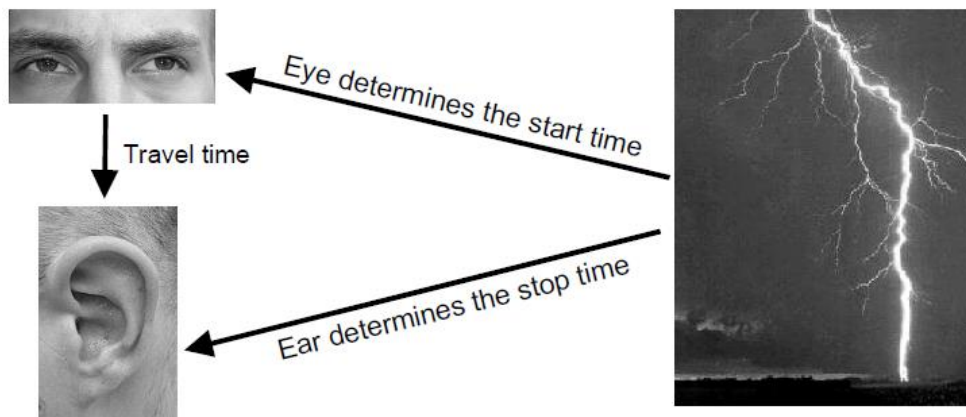


Рис. 1 – Визначення відстані по спалаху блискавки

Відстань = транзитний час * швидкість звуку

Система GPS функціонує відповідно до такого ж принципу. Для того щоб обчислити точну позицію, потрібно всього лише виміряти транзитний час сигналу між точкою спостереження і чотирма іншими супутниками, чії позиції відомі.

Основні принципи супутникової навігації

Все супутникові навігаційні системи використовують загальні принципи визначення координат:

1. Супутники з відомою позицією передають регулярний сигнал.
2. Позиція приймача обчислюється на основі вимірювання часу поширення радіохвиль (електромагнітні сигнали поширюються зі швидкістю світла $c = 300000 \text{ км/с}$),.

Пояснимо ці принципи на простій моделі. Уявіть, що ми в автомобілі та хочемо визначити своє місцезнаходження на довгій, прямій вулиці. В кінці вулиці є радіопередавач, який посилає тактовий імпульс кожену секунду. В автомобілі є годинник, який синхронізовано з годинником радіопередавача. Вимірюючи час від передавача до машини, ми можемо визначити нашу позицію на вулиці (Рис. 2).

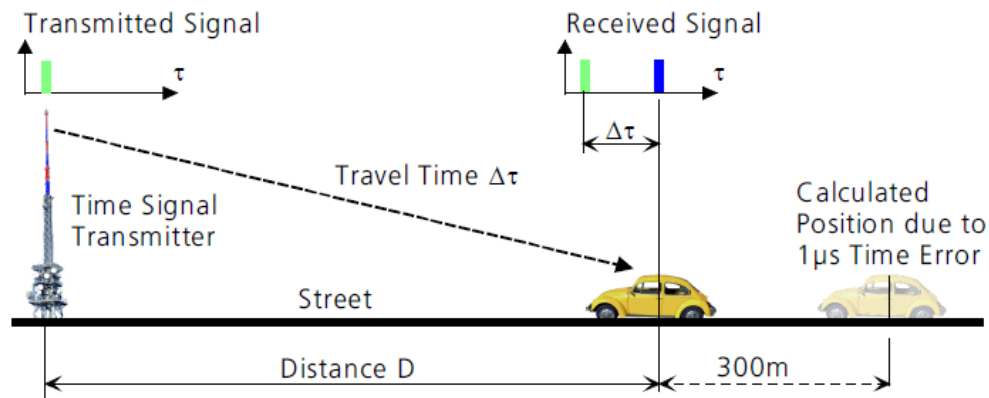


Рис.2 – У найпростішому випадку відстань визначається часом, за який радіоімпульс пройшов шлях від передавача до авто

Відстань D обчислюється шляхом множення часу поширення $\Delta\tau$ на швидкість світла c :

$$D = \Delta\tau \cdot c$$

Оскільки синхронізація годин в машині з передавачем неідеальна, існує різниця між обчисленою відстанню і фактичною. В навігації це некоректне значення відстані, яке обчислено з використанням неточного годинника приймача, називають псевдовідстанню. Фактичну відстань називають просто «відстань». У нашому прикладі, якщо помилка за часом становить 1 мікросекунду (1мкс), то це визначає, що між псевдовідстанню та фактичною відстанню буде різниця в 300 м.

Ми могли б вирішити цю проблему, оснастивши наш автомобіль точним атомним годинником, але це значно вплине на наш бюджет. Іншим рішенням буде використання другого синхронізованого передавача, відстань до якого A від першого передавача відома. Вимірюючи обидва часи поширення, можна точно визначити відстань, незважаючи на неточний бортовий годинник.

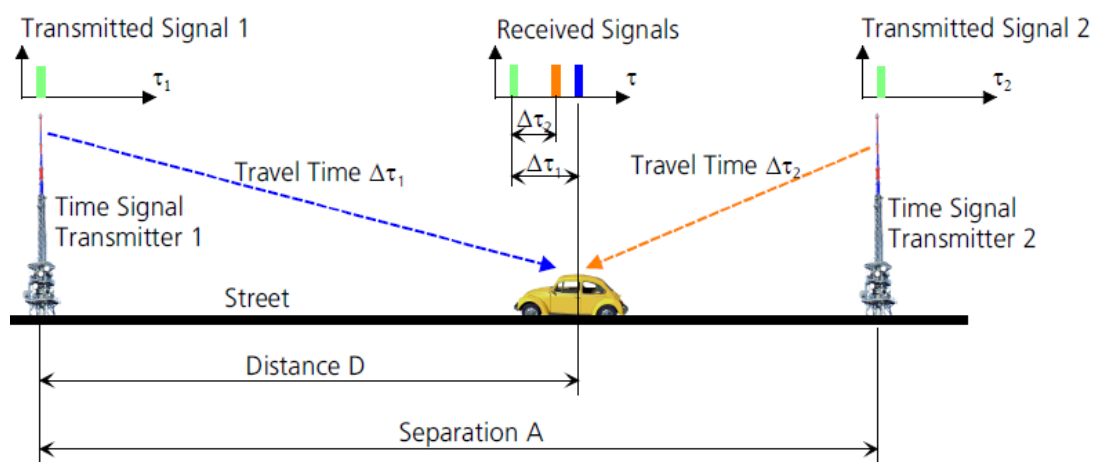


Рис.3 – З двома передавачами можна обчислити точну позиція, незважаючи на похибку годинника авто

$$D = \frac{(\Delta\tau_1 - \Delta\tau_2) \cdot c + A}{2}$$

В цьому разі необхідно лише виміряти різницю між приходом двох сигналів. Потреби в вимірюванні абсолютного часу, щоб знайти Δt вже немає.

Таким чином, щоб точно обчислити позицію і час уздовж лінії (приймаємо, що лінія продовжується тільки в одному напрямку), нам необхідно два передавачі сигналів часу. З цього ми можемо зробити наступний висновок: в разі несинхронізованого бортового годинника, що використовуються при розрахунку позиції, необхідно мати таке число передавачів сигналів часу, яке перевищує число невідомих вимірювань на одиницю.

Приклад:

- На площині (два виміри) нам необхідно три передавача сигналів часу.
- В тривимірному просторі нам необхідно чотири передавача сигналів часу.

Системи супутникової навігації використовують супутники як передавачі сигналів часу. Необхідно мати зв'язок, як мінімум, з 4 супутниками (рис. 4) для визначення тривимірних координат (Довгота, Широта, Висота) протягом всього часу. Ми розглянемо це більш детально в наступних розділах.

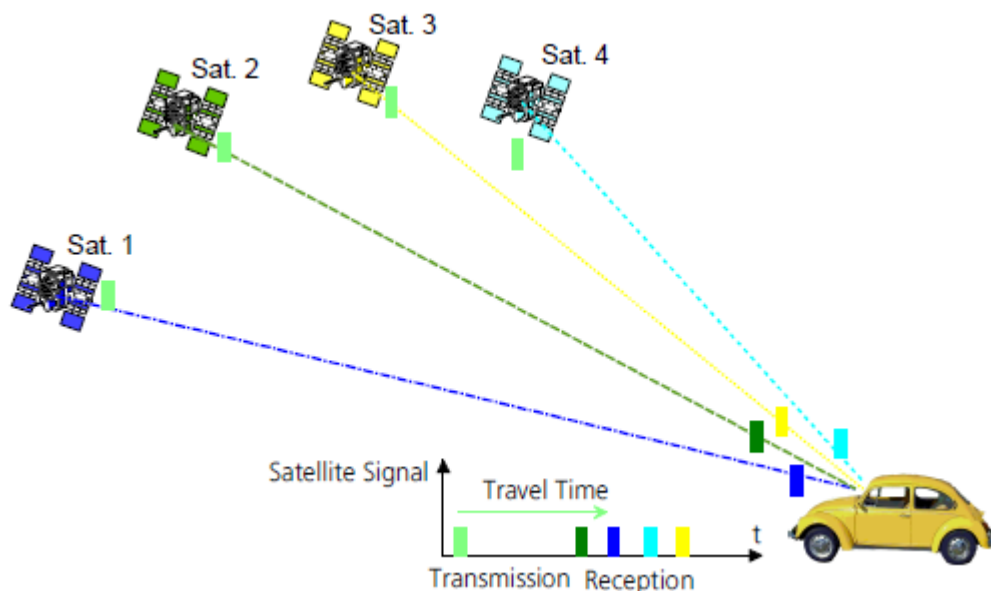


Рис.4 – Необхідно чотири супутники для визначення довготи, широти, висоти та часу

Час проходження сигналу

Системи супутникової навігації використовують високо розташовані супутники, які розміщуються таким чином, щоб з будь-якої точки на землі можна було провести лінію, як мінімум, до чотирьох супутників.

Кожен з цих супутників має до чотирьох атомних годинників на борту. Атомні годинники є в даний час найбільш точним інструментом вимірювання часу. Вони втрачають максимум одну секунду за 30,000 – 1,000,000 років. Для того щоб зробити їх ще більш точними, виконують корекцію або синхронізацію з різних керуючих точок на Землі. Кожен супутник передає свою точну позицію і точний час на Землю з частотою 1575.42 МГц. Ці сигнали передаються зі швидкістю світла (300,000 км/с) і, отже, буде потрібно приблизно 67,3мс для досягнення земної поверхні прямо під супутником. Сигналу необхідно 3,33мкс на кожен додатковий кілометр. Якщо Ви хочете встановити вашу позицію на землі (або в морі, або десь ще), все, що Вам потрібно – це приймач та точний годинник. Шляхом порівняння часу отримання супутникового сигналу з часом його відправки супутником можливо визначити транзитний час цього сигналу (Рис 5).

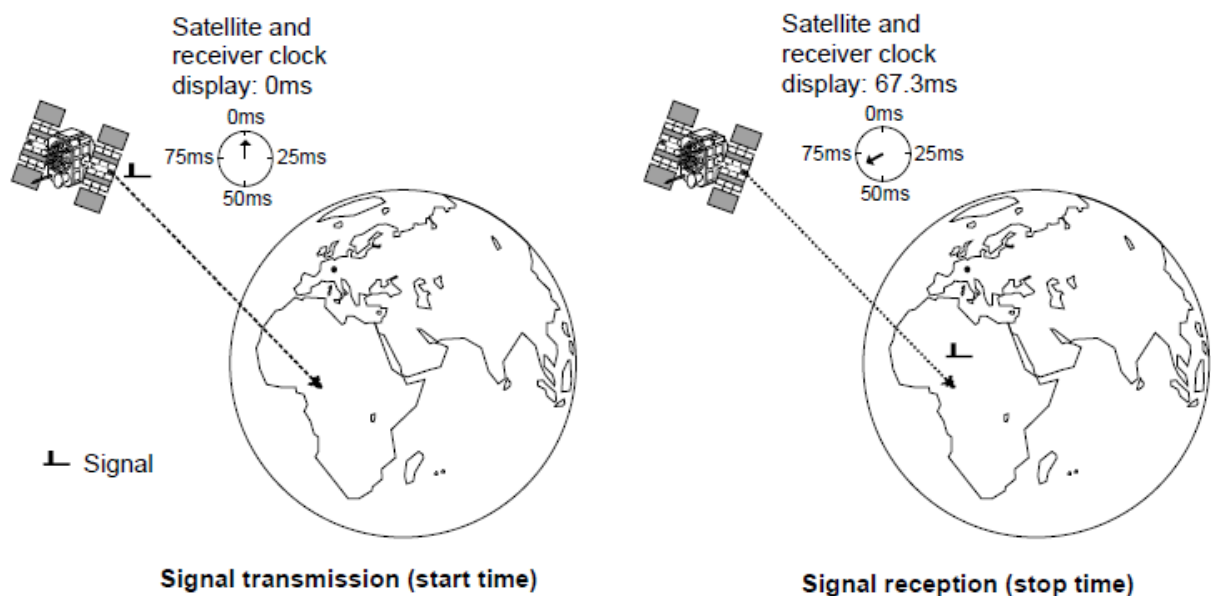


Рис.5 Визначення транзитного часу сигналу

Як в прикладі з машиною, Відстань E до супутника можна визначити, використовуючи транзитний час:

$$\text{Відстань} = \text{транзитний час} * \text{швидкість світла}$$

Визначення позиції на mapі

Уявіть собі, що Ви йдете через велике плато і хочете знати, де Ви. Два супутника, розташовані прямо над Вами, передають свій бортовий час і позиції. Використовуючи транзитний час сигналу від кожного з обох супутників, Ви можете намалювати два кола з радіусами D_1 і D_2 навколо

супутників. Кожен радіус відповідає обчисленій відстані до супутника. Всі можливі місця, які знаходяться на даній відстані від супутника, розташовані на цій окружності. Звичайно, що позиція вище супутників виключена, таким чином позиція приймача – в точці перетину кіл під супутниками (Рис. 6). Отже, двох супутників досить, щоб визначити позицію в площині X/Y.

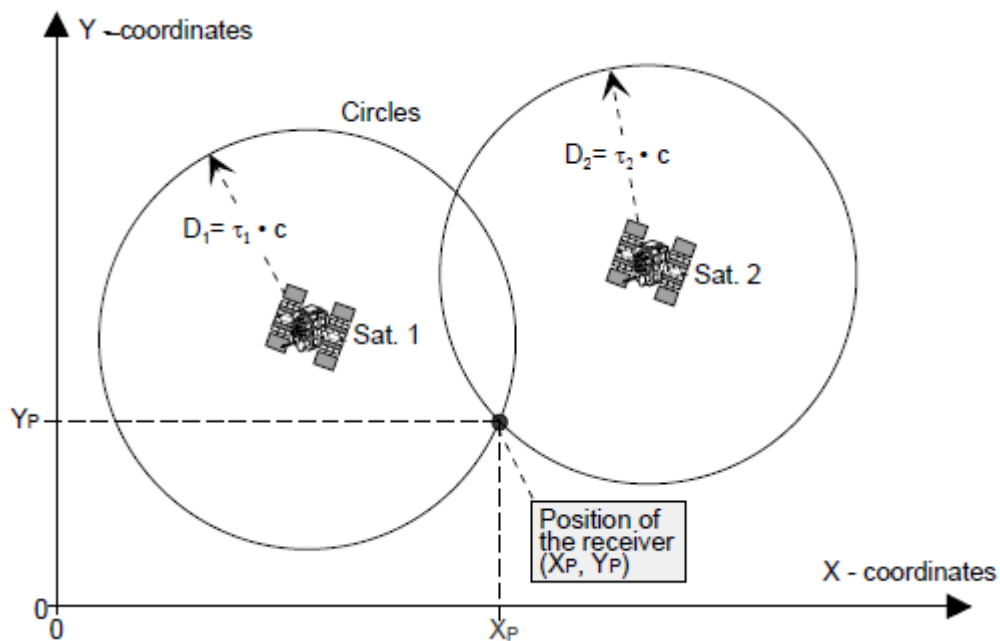


Рис. 6 – Позиція приймача знаходиться в місці перетину двох окружностей.

Але ж насправді, позиція повинна бути визначена в тривимірному просторі, а не на площині. Різниця між площиною і тривимірним простором полягає в додатковому вимірі (висота Z), відповідно додатковий третій супутник повинен бути доступний для визначення дійсної позиції. Якщо відстані до трьох супутників відомі, то всі можливі позиції розташовані на поверхні трьох сфер, чий радіуси відповідають розрахованим відстаням. Шукана позиція – місце перетину всіх трьох сфер (Рис. 7).

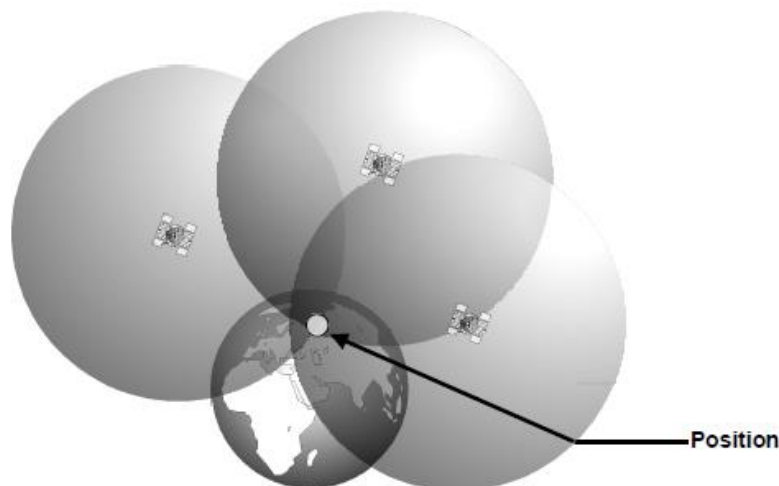


Рис. 7 – Позиція визначається як точка перетину трьох сфер

Поява похибки часу та її корекція

До цього часу ми вважали, що вимір транзитного часу сигналу був точним. Тим не менш, це не так. Приймачу для точного вимірювання часу необхідна синхронізація. Якщо транзитний час має похибку 1нс – позиційна помилка складе 300м . Незважаючи на те, що часи на борту всіх трьох супутників синхронізовані, позиційна похибка через вимірювання транзитного часу буде мати те ж саме значення 300м . Математика є єдиною річчю, яка може нам допомогти.

Згадаймо, що, якщо є N невідомих змінних, то нам потрібно N незалежних рівнянь. Якщо вимірювання часу супроводжується постійною невідомою помилкою (Δt), то в нас є чотири невідомих змінних в 3-х мірному просторі:

- Довгота (X)
- Широта (Y)
- Висота (Z)
- Похибка в часі (Δt),

З цього випливає, що в 3-х мірному просторі необхідно 4 супутники задля визначення точної позиції.

Супутникові навігаційні системи сконструйовані таким чином, щоб з будь-якої точки на Землі було видно як мінімум 4 супутника (Рис. 8). Таким чином, незважаючи на наявність похибки годинника приймача й результуючих помилок за часом, позиція обчислюється з точністю приблизно $3 - 5\text{ м}$.

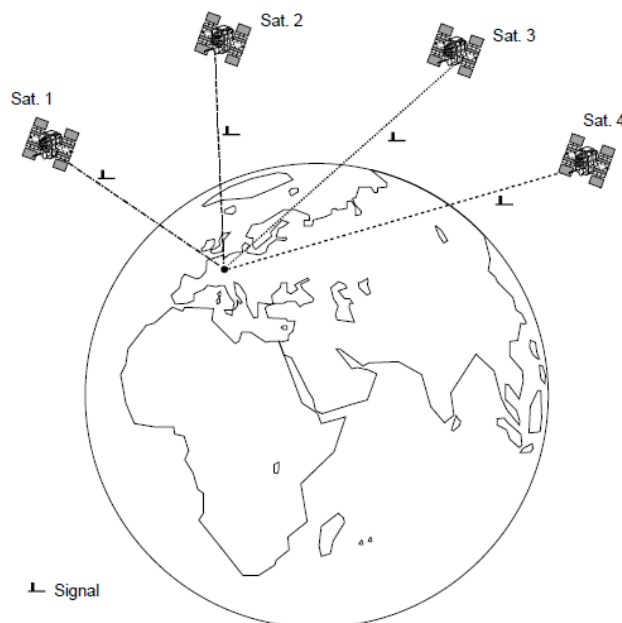


Рис.8 Необхідно чотири супутники для визначення позиції в 3-D просторі

Технологія супутникової навігації на прикладі системи GPS

Опис системи

В цьому розділі буде розглянуто різні сегменти технології супутникової навігації на прикладі системи GPS.

Система глобального позиціонування (GPS) включає в себе три сегменти (Рис. 9):

- космічний сегмент (всі робочі супутники),
- керуючий сегмент (всі наземні станції системи: основна керуюча і додаткові для контролю),
- сегмент користувача (всі цивільні і військові GPS користувачі).

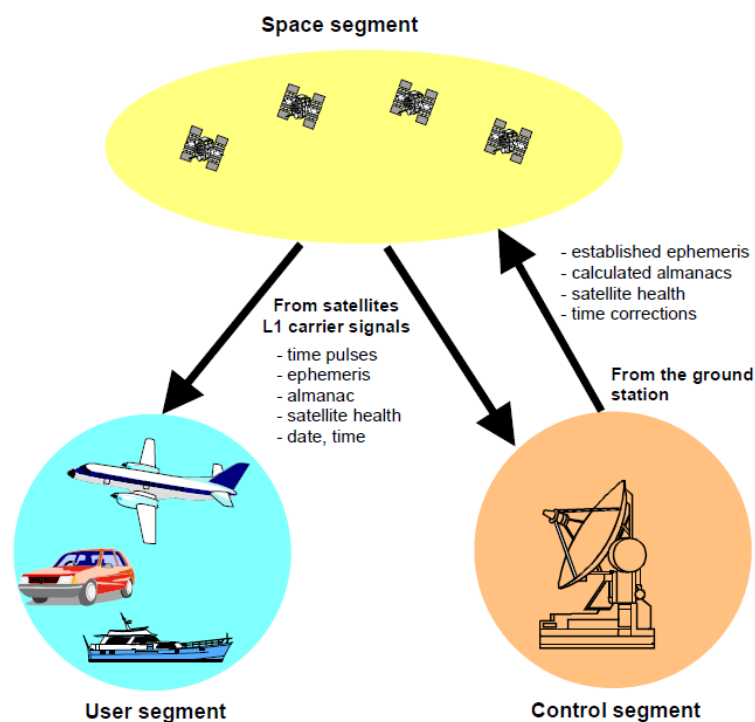


Рис. 9 – Три сегменти GSM

Як видно з рисунка 9 зв'язок між космічним сегментом і сегментом користувача є одностороннім. Керуючі станції на землі мають двосторонній зв'язок із супутниками.

Розміщення та рух супутників

Космічний сегмент складається з 32 діючих супутників (Рис. 10) з орбітами в 6 різних площинах (від чотирьох до п'яти супутників в площині). Супутники знаходяться на висоті 20,180 км над землею поверхнею і нахилені на 55° до екватора. Кожен супутник здійснює круг по орбіті за 12 годин. Через обертання Землі, супутник буде в своєму початковому положенні (Рис. 11) після приблизно 24 годин (23 години 56 хвилин, щоб бути точним).

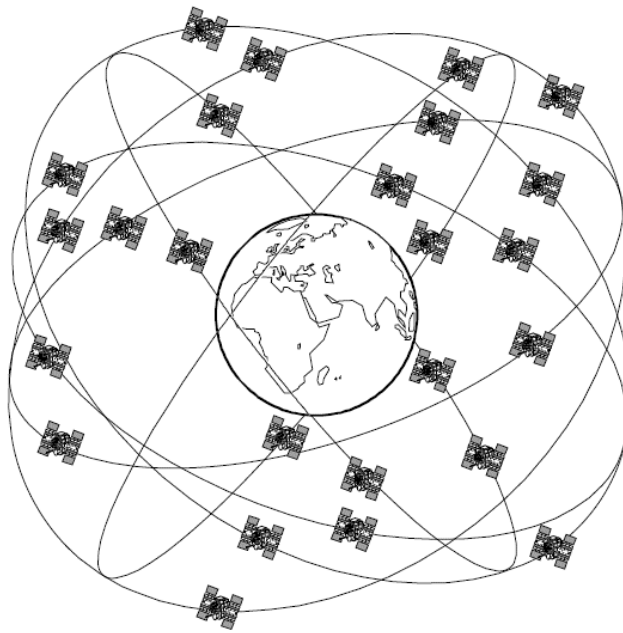


Рис. 10: Орбіти GPS супутників в 6 різних площинах

Супутникові сигнали можуть бути прийняті будь де в межах зони дії супутника. Рис. 13 показує зону дії супутника (затінена область), розташованого прямо над нульовим меридіаном.

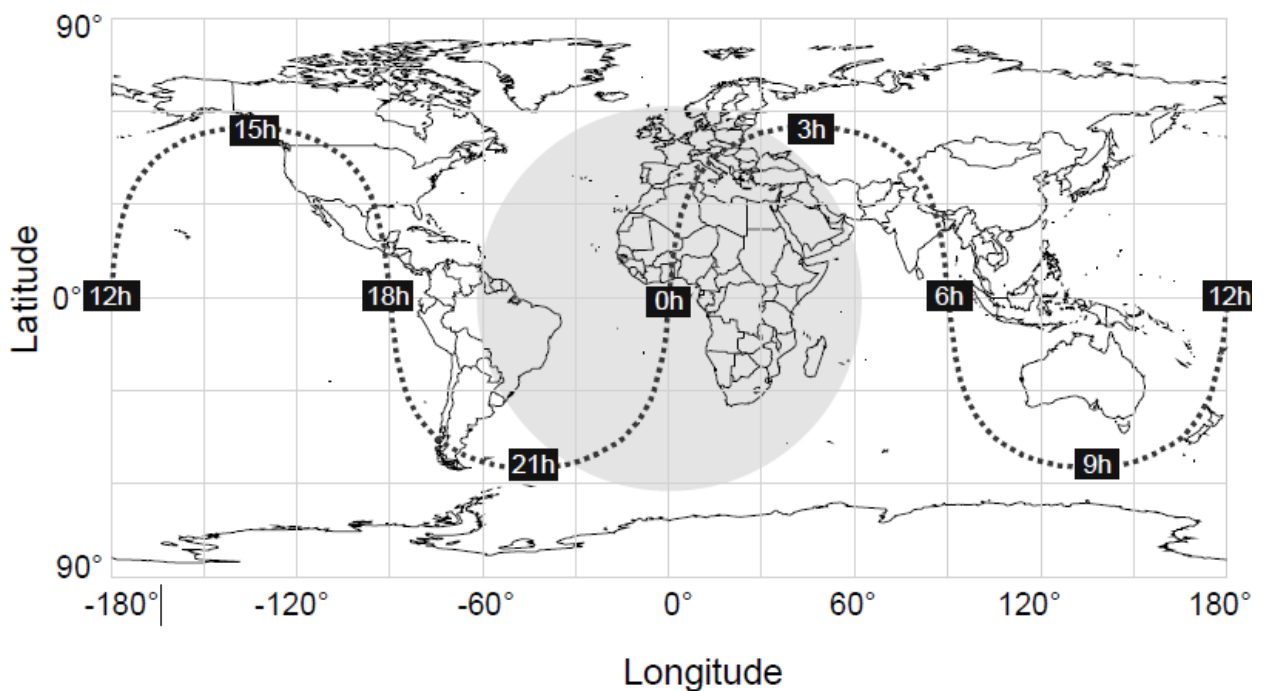


Рис. 11 – Умовний слід на поверхні Землі під час 24-годинний руху GPS супутника та зона його дії

Розміщення супутників в будь-який момент часу можна побачити на рис. 12. Таке розміщення є результатом вдалого розподілу орбіт та висот супутників, яке забезпечує зв'язок, як мінімум, з чотирма супутниками в будь-який час в будь якій точці світу.

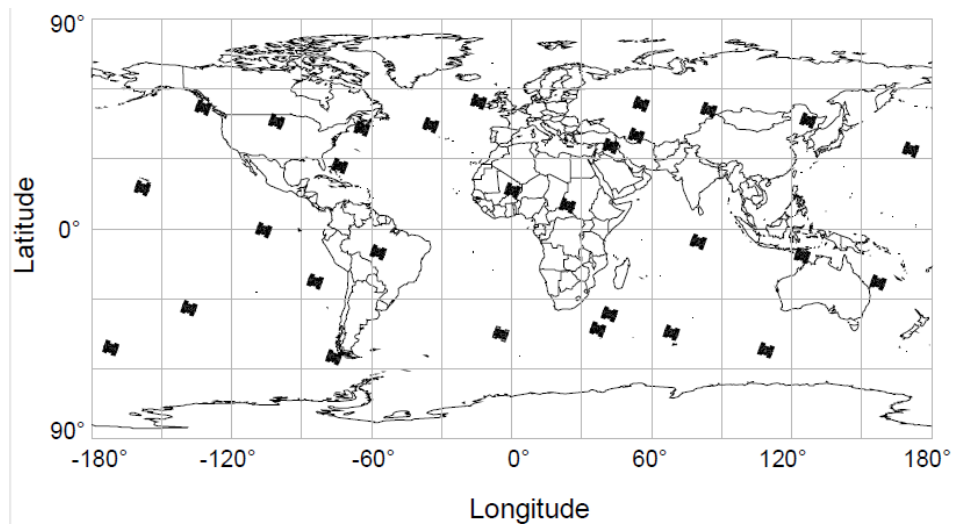


Рис. 12 – Позиції супутників в якийсь певний момент часу (приклад)

Супутники GPS

Конструкція супутника

Всі супутники використовують бортові атомні годинники задля забезпечення синхронізованих сигналів, які випромінюються строго на одній частоті (1575.42 МГц). Мінімальна потужність сигналу, отриманого на Землі, приблизно від -158dBW до -160dBW.

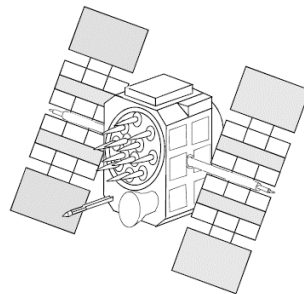


Рис. 13 – Супутник GPS

Аналіз зв'язку

Аналіз зв'язку між супутником і користувачем потрібен для установки необхідного рівня потужності передачі. Відповідно до специфікації, мінімальна потужність не повинна бути нижче до -160dBW (-130dBm). Для гарантії підтримки цього рівня потужність передачі супутника 1 повинна бути не менш ніж 21.9 Вт.

Потужність отриманого сигналу до -160dBW (-130dBm) – дуже невелика. Максимальна спектральна щільність потужності цього сигналу дорівнює -190 dBm/Hz (рис 16). Спектральна щільність потужності термічного шуму становить 174 dBm/Hz (за температури 290K). Таким чином, максимальна потужність отриманого сигналу приблизно на 16 dBm нижче рівня термічного шуму.

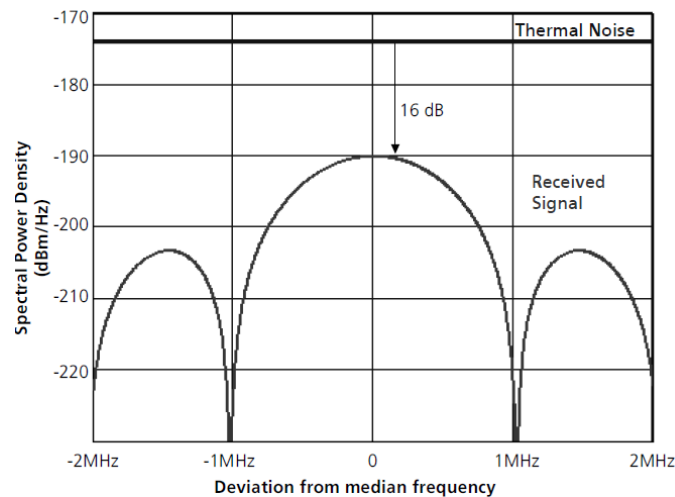


Рис. 14 – Спектральна щільність потужності отриманого сигналу у порівнянні з термічним шумом

Сигнали супутника

Уся інформація, яка наведена нижче (так зване «навігаційне повідомлення») передається супутником зі швидкістю 50 біт в секунду:

- Сигнал супутникового часу та сигнали синхронізації,
- Точні дані орбіти супутника (так звані «ефемериди»),
- Корекційна інформація для визначення точного часу
- Наближені орбітальні дані щодо усіх супутників (альманах)
- Дані щодо іоносфери
- Інформація про технічний стан супутника

Час, який потрібен для передачі даної інформації, складає 12.5 хвилин.

Використовуючи навігаційне повідомлення, приймач може визначити час передачі кожного сигналу і точну позицію супутника на момент передачі.

Кожен з 32 супутників передає унікальну сигнатуру. Ця сигнатура складається з послідовності 1023 нулів і одиниць (Рис. 15). Це є код псевдовипадкового шуму (Pseudo Random Noise – **PRN**), який має статистичні властивості сигналу випадкового шуму, проте генерується штучно – на основі стандартного алгоритму. Алгоритм формує ту чи іншу послідовність 1023 нулів і одиниць в залежності від певного набору чисел (зазвичай 2-3 числа), які є параметрами алгоритму. Підставивши в алгоритм іншу параметри, отримуємо іншу послідовність нулів і одиниць. Якщо ж алгоритм починає роботу з одними ж тими параметрами, то він завжди буде генерувати одну й ту ж саму послідовність. Таким чином, аби на двох пристроях сформувати однакову послідовність достатньо задати їм один й той же алгоритм та однакові параметри цього алгоритму. Якщо ж усі пристрої оснащені одним й тим самим алгоритмом, то для формування потрібної послідовності пристрою треба передати лише параметри цього алгоритму, тобто всього декілька чисел.

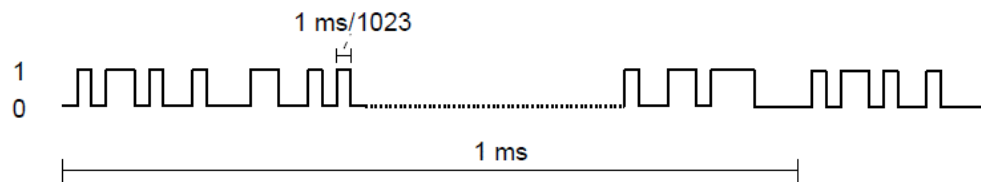


Рис.15 – Псевдовипадковий шум

Ця сигнатура, або код з 1023 біт, має тривалість 1мс та генерується безперервно, повторюючись кожні 1мс. Ця сигнатура – це є унікальний ідентифікатор конкретного супутника й він використовується з наступною метою:

- Ідентифікація: унікальна сигнатура означає, що приймач знає, від якого супутника отримано сигнал.
- Вимірювання транзитного часу сигналу

Блокова діаграма генерації сигналу супутника

На борту супутників знаходяться чотири дуже точних атомних годинника. Тактові імпульси і частоти, необхідні для функціонування системи, є похідними від резонансної частоти атомного годинника (рис.18 і 19):

- Частота передавання даних 50 Гц,
- Послідовність бітових сигналів PRN-коду з частотою 1.023 МГц, яка модулює дані, використовуючи логічну операцію «Виключне АБО» (exclusive OR – XOR),
- Частота опорного колювання (1575.42 МГц).

Дані, промодульовані PRN-кодом, потім модулюються ще раз опорним колюванням 1575.42 МГц за методом двійкової фазової маніпуляції (binary phase-shift keying - **BPSK**), який було розглянуто раніше в лекції 6.

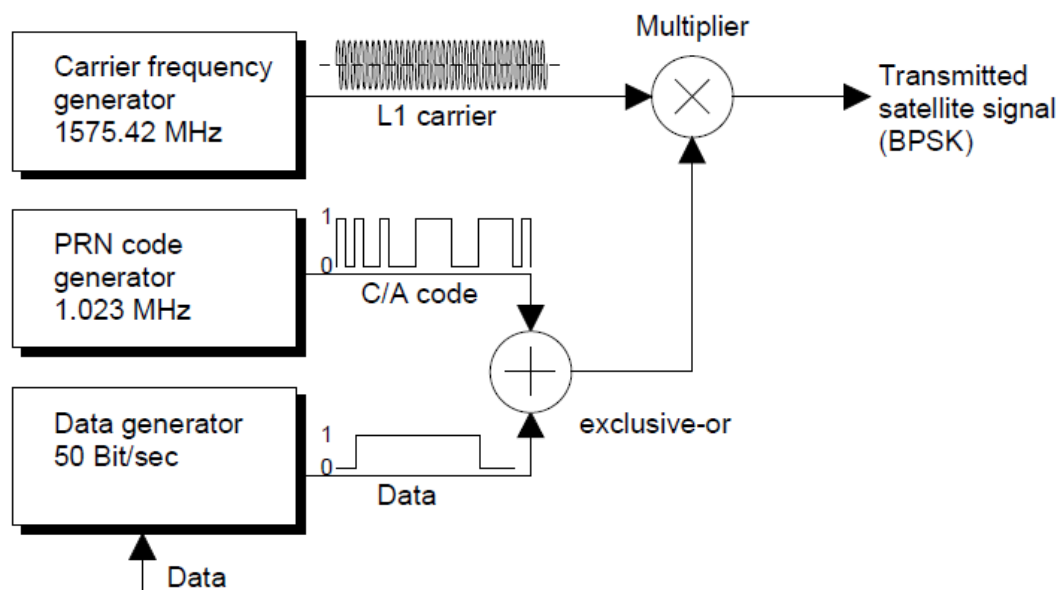


Рис. 16 – Спрощена блокова діаграма модуляції GPS-сигналу

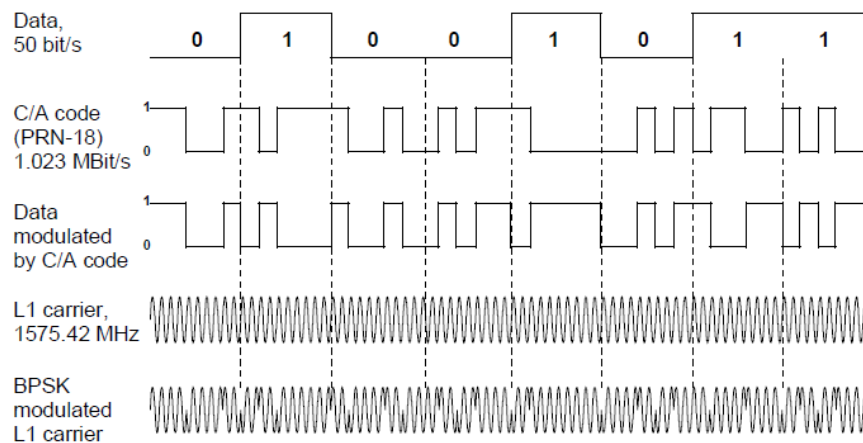


Рис. 17 Структура GPS-сигналу

Детальна блокова система генерації сигналу супутника

Принцип модуляції, представлений вище, був відкритий в 1940 році в США, актрисою Hedy Lamarr і піаністом George Anthell.

Атомні годинники на борту супутника мають стійкість понад $2 \cdot 10^{-11}$. Основна частота 10.23 МГц походить від резонансної частоти атомного годинника. У свою чергу, опорна частота, частота даних, частота генерації псевдовипадкового шуму (PRN) походять від основної частоти (Рис. 18). Тобто всі 32 супутники передають на частоті 1575.42 МГц, використовуюючи процес, відомий під назвою CDMA (Code Division Multiple Access). Генератор PRN-коду має частоту 1.023 МГц, а сам код складається з 1023 імпульсів (послідовних бітів) і має тривалість 1мс.

Процес модуляції, описаний вище, називається DSSS модуляцією, при цьому PRN-код відіграє важливу роль в цьому процесі. Оскільки всі супутники передають на одній частоті (1575.42 МГц), то код PRN містить ідентифікаційну інформацію для кожного супутника. Він є унікальним для кожного супутника й постійно повторюється. Отже, кожний супутник завжди ідентифікований за відповідним йому PRN-кодом.

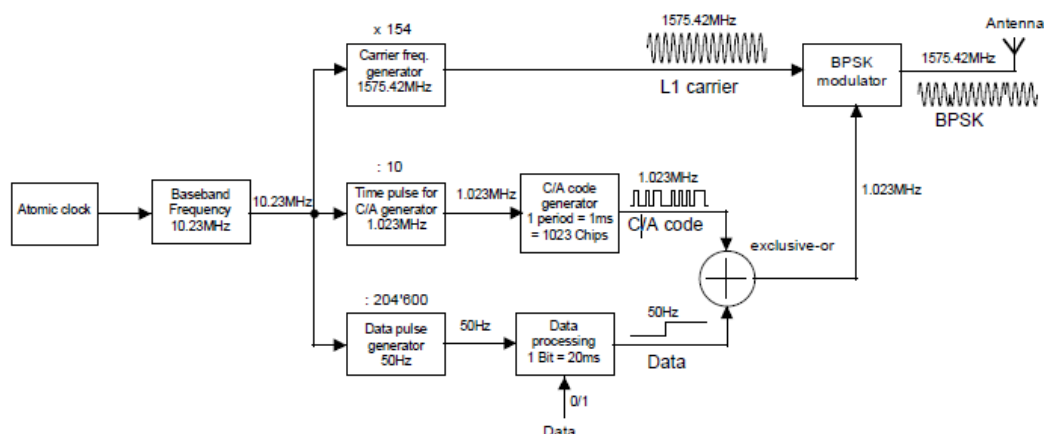


Рис.18 – Детальна блокова діаграма каналу формування GPS-сигналу супутника

Керуючий сегмент

Керуючий сегмент, складається з основної керуючої станції, розташованої в штаті Колорадо, п'яти контрольних станцій, оснащених атомними годинами, які розташовані навколо земної кулі біля екватора, і трьох керуючих наземних станцій, які передають інформацію супутникам.

Найбільш важливі завдання керуючого сегмента:

- Спостереження за переміщенням супутників і розрахунок орбітальних даних (ефемерид)
- Контроль годин супутника і прогнозування їхньої роботи
- Синхронізація часу супутника
- Ретрансляція точних орбітальних даних, отриманих від супутників
- Ретрансляція приблизних орбітальних даних, отриманих від усіх супутників (альманах)
- Ретрансляція подальшої інформації, включаючи стан супутника, похибки часу і т.і.

Сегмент користувача

Сигналам, що передані супутниками, потрібно приблизно 67 мс для того, аби досягнути користувача. Хоча сигнали рухаються зі швидкістю світла, їх транзитний час залежить від відстані між супутниками і споживачем.

Чотири різних сигнали генеруються приймачем і всі вони мають ту ж саму структуру, які отримуються з 4-х супутників. Під час синхронізації сигналів, які генеруються приймачем, з сигналами, які отримуються від супутників, вимірюється часовий зсув Δt для всіх 4-х супутників (рис. 19). Виміряні зміщення часу Δt потім використовуються для визначення точного транзитного часу сигналу.

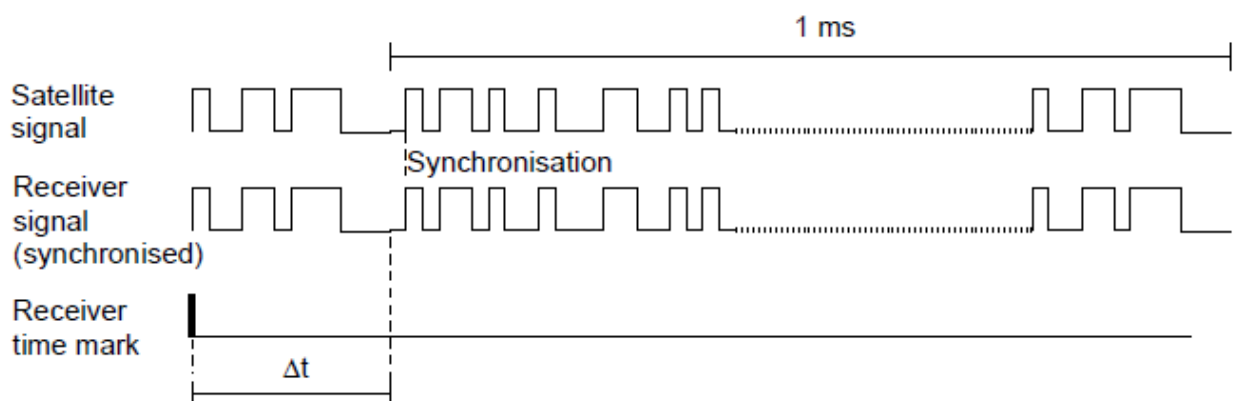


Рис.19 – Вимірювання транзитного часу сигналу

Для визначення позиції користувача потрібно радіозв'язок з чотирма іншими супутниками. Відстань до супутників визначає транзитний час сигналів. Приймач потім обчислює позицію користувача: широту φ , довготу λ ,

висоту h та час t з псевдовідстані та відомої позиції чотирьох супутників. Висловлюючись математичними термінами, це означає, що чотири невідомих змінних φ , λ , h та t визначені за допомогою відстані і позиції цих чотирьох супутників шляхом вирішення системи з 4-х нелінійних рівнянь.

Як сказано раніше, всі супутники передають на одній частоті, але з різним кодом PRN. Цей процес називається Code Division Multiple Access (CDMA). Відновлення сигналу та ідентифікація супутників відбувається за допомогою кореляції. Оскільки приймач знає всі PRN-коди, які використовуються, то систематичний зсув і порівняння кожного коду з усіма вхідними сигналами з супутників призведе до повного збігу сигналу згенерованого та сигналу отриманого від супутника, тож точка кореляції буде досягнута. (Рис. 20). Точка кореляції використовується для вимірювання фактичного транзитного часу сигналу і для ідентифікації супутника.

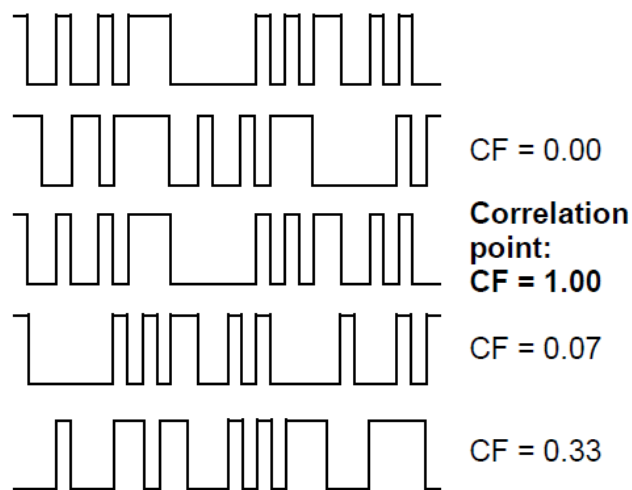


Рис.20 – Демонстрація процесу кореляції протягом 30 бітів

Ступінь кореляції виражається тут як **CF** (показник кореляції). Діапазон величини **CF** лежить від мінус одиниці до плюс одиниці і є позитивним тільки при збігу сигналів (бітова частота і фаза).

В результаті ефекту Доплера (всі супутники і приймачі рухаються відносно один одного) можливий зсув переданих сигналів на ± 6000 Гц щодо точки прийому. Визначення транзитного часу сигналу і відновлення даних вимагає не тільки розрахунку кореляції з усіма можливими кодами і фазами зміщення, але також і ідентифікацію правильної фази опорної частоти. Крім того, щоб виконати усі розрахунки кореляції, необхідно знайти точну частоту опорного колювання, яка може бути зміщеною ± 6000 Гц через ефект Доплера. За допомогою систематичного зсуву частоти і порівняння вхідних сигналів від супутника з усіма кодами (рис. 20) знаходимо потрібну точку (в якій коефіцієнт кореляції дорівнює 1) (Рис. 21).

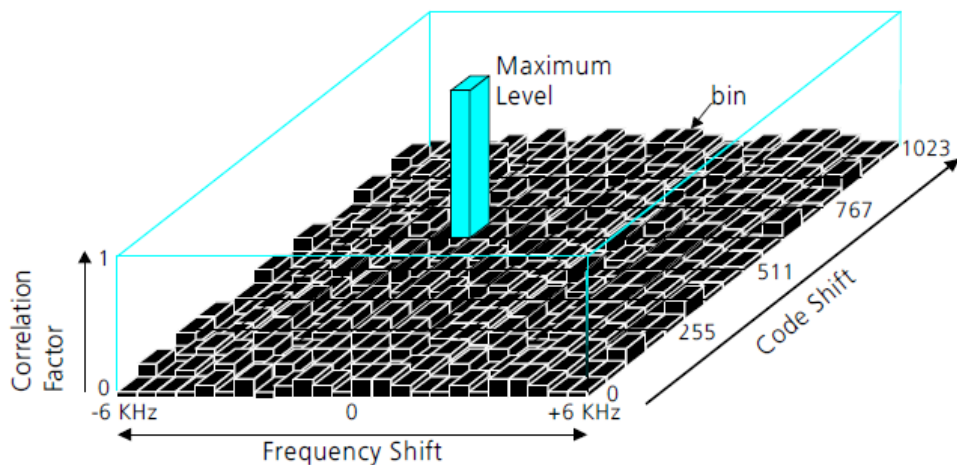


Рис. 21 – Пошук максимуму кореляції в області кодів та області частот

Спектральна щільність потужності отриманого GPS сигналу лежить приблизно на 16 dBm нижче термічного шуму (див. рис. 14). Демодуляція і концентрація отриманого GPS сигналу дає системне посилення G_g :

$$G_g = 10 \lg \frac{\text{Швидкість модуляції PRN-кодом}}{\text{Швидкість передавання даних}} = 10 \lg \frac{1023000 \text{ bps}}{50 \text{ bps}} = 10 \lg 20460 = 43 \text{ dB}$$

Після концентрації, щільність потужності використовуваного сигналу стає більше термічного шуму (Рис. 22).

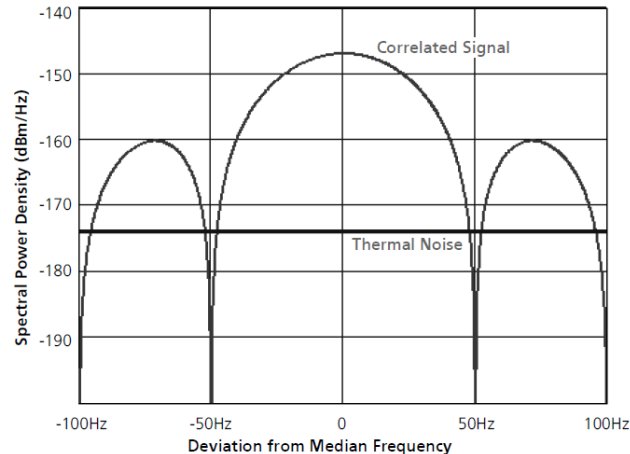


Рис.22 – Спектральна щільність потужності демодульованого сигналу у порівнянні з термічним шумом

Повідомлення GPS

Введення

Повідомлення являє собою безперервний потік даних, передаваний зі швидкістю 50 біт на секунду. Кожен супутник передає наступну інформацію на Землю:

- Системний час і скориговані значення годин
- Власні високоточні орбітальні дані (ефемериди)

- Наближені орбітальні дані для всіх супутників (альманах)
- Стан системи і т.д.

Навігаційне повідомлення потрібне для обчислення поточної позиції супутників і для визначення транзитного часу сигналу.

Потік даних модулюється опорною хвилею кожного окремого супутника. Передаванні дані поділені на логічно згруповані блоки – «фрейми», звані також кадрами або сторінками. Кожен фрейм є 1500 біт довжиною, і його передавання займає 30 секунд. Фрейми розділені на 5 підфреймів. Кожен з підфреймів є 300 біт довжиною, і його передавання займає 6 секунд. Для передавання всього альманаху потрібно 25 різних фреймів. Час передачі для альманаху – 12.5 хвилин. Приймач GPS має отримати весь альманах для роботи (зокрема – для своєї первинної ініціалізації).

Структура навігаційного повідомлення

Фрейм довжиною 1 500 біт займає 30 секунд часу для передавання. 1500 біт розділені на п'ять підфреймів довжиною 300 біт (час передачі – 6 секунд). Кожен з підфреймів, в свою чергу, розділений на 10 слів, кожен з яких довжиною 30 біт. Кожен з підфреймів починається зі слова телеметрії (TLM) і слова handover (HOW). Повне навігаційне повідомлення складається з 25 фреймів (сторінок). Структура навігаційного повідомлення проілюстрована на рис. 23.

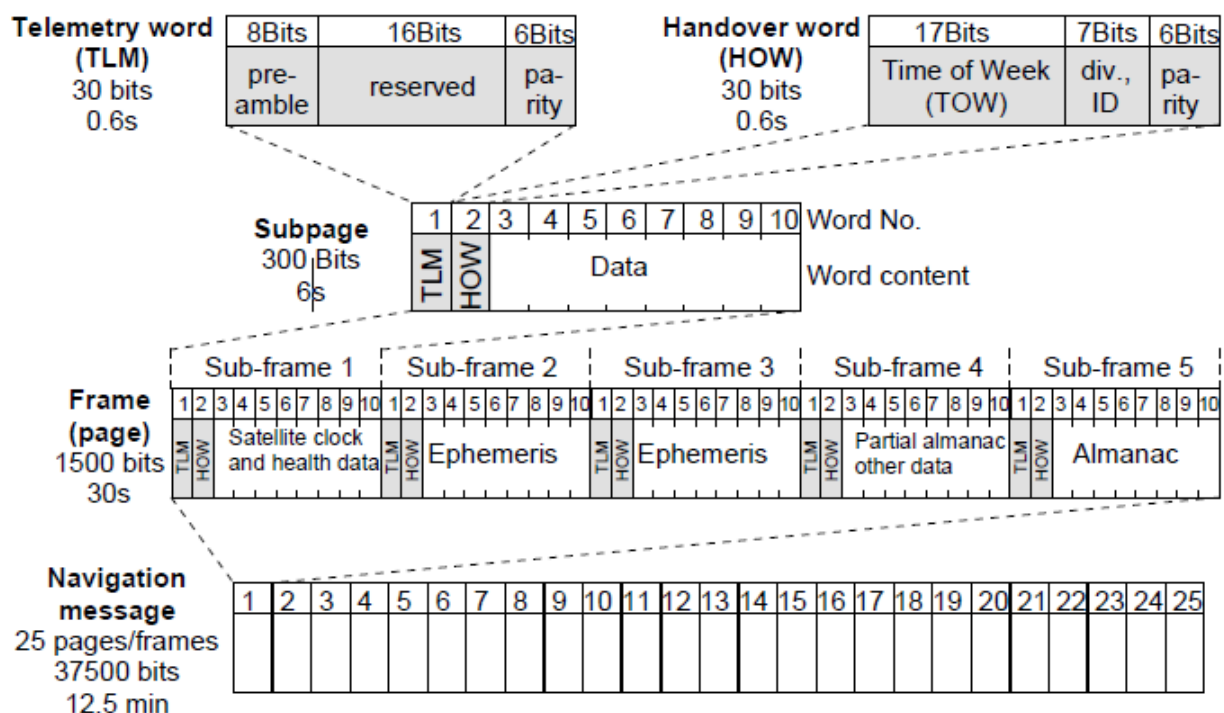


Рис. 23 – Структура повного навігаційного повідомлення

Інформація в підфреймах

Фрейм розділений на п'ять підфреймів, кожен з яких передає різну інформацію.

- підфрейм 1 містить значення часу супутника, включаючи параметри для корекції затримки транзитного часу сигналу і годин на борту супутника, а також інформацію про стан супутника і оцінку позиційної точності супутника.
- підфрейми 2 і 3 містять дані ефемерид супутника. Ці дані забезпечують дуже точну інформацію про орбіту супутнику.
- підфрейм 4 містить дані альманаху про супутники з номерами від 25 до 32 (кожен з підфреймів може передати дані тільки одного супутника).
- підфрейми 5 містять дані альманаху про супутники з номерами від 1 до 24 (кожен підфреймів може передати дані тільки одного супутника). Всі 25 сторінок передаються разом з інформацією про стан супутників з номерами від 1 до 24.

TLM і HOW

Перше слово кожного фрейму, слово телеметрії (TLM), містить послідовність з 8 біт преамбули (10001011), яка використовується для синхронізації, наступні 16 біт зарезервовані для зареєстрованих користувачів. Як і у всіх словах, кінцеві 6 біт слова телеметрії – біти парності.

Слово Handover (HOW), йде відразу за словом телеметрії в кожному з підфреймів і містить в межах своєї структури стартовий час для наступного підфрейму.

Розбиття на 25 сторінок

Повне навігаційне повідомлення потребує 25 сторінок і займає 12.5 хвилин. Сторінка або фрейм розділені на п'ять підфреймів. Щодо підфреймів 1 .. 3, то їх інформаційний вміст однаковий для всіх 25 сторінок. Це означає, що приймач має всі значення годин і дані ефемерид від супутника кожні 30 секунд.

- ✓ Єдина відмінність між підфреймами 4 і 5 – в організації переданої інформації.
- ✓ У разі підфреймів 4, сторінки 2, 3, 4, 5, 7, 8, 9 і 10 ретранслюють дані альманаху із супутників з номерами від 25 до 32. У цьому випадку, дані альманаху одного супутника передаються на одну сторінку. Сторінка 18 передає значення корекції через іоносферічної сцинтиляції, а також різницю між UTC і GPS часом. Сторінка 25 містить інформацію про конфігурацію всіх 32 супутників та про стан супутників з номерами 25 ... 32.

- ✓ У разі підфреймів 5, сторінки 1 ... 24 ретранслюють дані альманаху із супутників з номерами від 1 до 24 . У цьому випадку, дані альманаху для одного супутника передаються на одну сторінку. Сторінка 25 передає інформацію про стан супутників з номерами 1 ... 24 і дійсний час альманаху.

Порівняння даних альманаху і ефемерид

Використовуючи дані ефемерид і альманаху, можна розрахувати, як супутники рухаються по орбітах і, отже, координати певного супутника можна знайти в будь-який момент часу. Різниця між переданими величинами складається головним чином в точності чисел. В наступній таблиці (Таблиця 1), порівнюються два варіанти чисел.

Порівняння даних альманаху і ефемерид

Таблиця 1

Інформація	Ефемериди, число біт	Альманах, число біт
Велика піввісь орбітального еліпса a (квадратний корінь з a)	32	16
Ексцентриситет орбітального еліпсу e	32	16

Для пояснення термінів, використаних в Таблиці 2 дивись рис. 24.

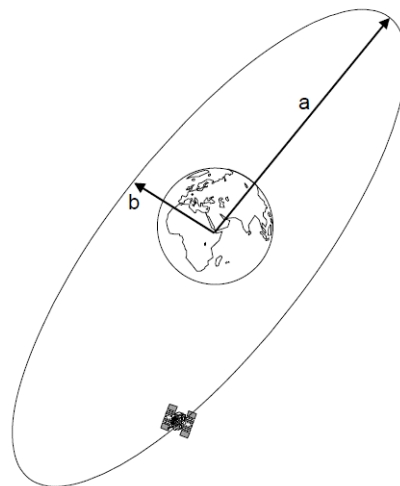


Рис. 24 – Терміни ефемерид

Велика піввісь орбітального еліпса: a

Мала піввісь орбітального еліпса: b

Ексцентриситет орбітального еліпса: $e = \sqrt{\frac{a^2 - b^2}{a^2}}$

Контрольні питання за темою

Рівень модуля

Рівень курсу

1. Технологія супутникової навігації: принцип вимірювання транзитного часу сигналу.
2. Технологія супутникової навігації: сигнали супутника та їх генерація.
3. Технологія супутникової навігації: сегмент користувача.
4. Технологія супутникової навігації: повідомлення GPS, його структура та вміст.