

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ

до виконання практичних робіт
із навчальної дисципліни

«СИСТЕМНА ІНТЕГРАЦІЯ ТА АДМІНІСТРУВАННЯ
ІНФОРМАЦІЙНИХ СИСТЕМ»

*(для здобувачів першого (бакалаврського) рівня вищої освіти денної,
заочної і дистанційної форм навчання
зі спеціальностей F6 – Інформаційні системи та технології,
F3 – Комп'ютерні науки)*

Харків
ХНУМГ ім. О. М. Бекетова
2026

Методичні рекомендації до виконання практичних робіт із навчальної дисципліни «Системна інтеграція та адміністрування інформаційних систем» (для здобувачів першого (бакалаврського) рівня вищої освіти денної, заочної і дистанційної форм навчання зі спеціальностей F6 – Інформаційні системи та технології, F3 – Комп'ютерні науки) / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова ; уклад. Н. Д. Сізова. – Харків : ХНУМГ ім. О. М. Бекетова, 2026. – 59 с.

Укладач д-р фіз.-мат. наук, проф. Н. Д. Сізова

Рецензент

М. Ю. Карпенко, кандидат технічних наук, доцент кафедри комп'ютерних наук та інформаційних технологій Харківського національного університету міського господарства імені О. М. Бекетова

Рекомендовано кафедрою комп'ютерних наук та інформаційних технологій, протокол № 9 від 23 січня 2026 р.

ЗМІСТ

Вступ	4
Практична робота № 1 Інтеграційні процеси в інформаційних системах.....	5
Практична робота № 2 Основні методи інтеграції інформаційних систем.....	7
Практична робота № 3 Основні архітектурні моделі інтегрованих.....	9
Практична робота № 4 Розробка плану приміщень та плану комп'ютерної мережі.....	12
Практична робота № 5 Алгоритм пошуку найкоротшого шляху у комп'ютерних мережах.....	19
Практична робота № 6 Адресація в мережах	26
Практична робота № 7 Передача даних у мережі.....	37
Практична робота № 8 Протокол FTP. Організація Ftp-сервера на основі WIN7	45
Практична робота № 9 Інтеграція баз даних (на прикладі mysql ↔ postgresql).....	50
Практична робота № 10 Характеристика віртуальних машин (VM).....	53
Практична робота № 11 Адміністрування Ос Windows Server.....	55
Список використаних джерел	58

ВСТУП

Дисципліна «Системна інтеграція та адміністрування інформаційних систем» є однією з ключових у підготовці фахівців з інформаційних технологій, оскільки формує практичні навички побудови, об'єднання та супроводу сучасних інформаційних інфраструктур. У реальних умовах діяльності підприємств та організацій інформаційні системи зрідка функціонують ізольовано – вони взаємодіють між собою, обмінюються даними та потребують централізованого управління і підтримки.

Метою виконання практичних робіт є закріплення теоретичних знань та набуття практичних компетентностей у сфері інтеграції програмних, апаратних і мережевих компонентів інформаційних систем, а також адміністрування їх роботи в корпоративному середовищі. Особлива увага приділяється сучасним підходам до побудови IT-інфраструктур, використанню мережевих сервісів, хмарних технологій, засобів віртуалізації, моніторингу та забезпечення інформаційної безпеки.

У процесі виконання практичних робіт здобувачі освіти навчаться:

- аналізувати структуру та складові інформаційних систем підприємства;
- здійснювати інтеграцію різнорідних програмних рішень і сервісів;
- налаштовувати мережеві служби та серверні компоненти;
- виконувати базове адміністрування операційних систем і прикладних сервісів;
- забезпечувати надійність, доступність і безпеку інформаційної інфраструктури;
- використовувати інструменти моніторингу та діагностики IT-середовища.

Методичні рекомендації містять опис практичних завдань, теоретичні відомості, послідовність виконання робіт, контрольні запитання та вимоги до оформлення результатів. Матеріали орієнтовані на формування системного мислення, розуміння принципів взаємодії компонентів інформаційних систем та підготовку здобувачів до вирішення прикладних завдань у професійній діяльності системного адміністратора та фахівця з системної інтеграції.

Виконання практичних робіт сприятиме розвитку навичок самостійного аналізу, налаштування та оптимізації інформаційних систем, що є необхідною складовою професійної компетентності сучасного IT-фахівця.

Матеріал орієнтований на студентів зі спеціальностей F6 – Інформаційні системи та технології, F3 – Комп'ютерні науки та може бути використаний здобувачами інших спеціальностей як навчальний і довідковий ресурс під час підготовки до практичних занять, виконання лабораторних робіт і самостійного вивчення дисципліни.

Практична робота № 1

Інтеграційні процеси в інформаційних системах

Мета: ознайомитися з основними підходами та технологіями інтеграції, навчитися обґрунтовано обирати інтеграційний підхід залежно від задачі.

Завдання:

1. Зробити аналіз інтеграційної ситуації для навчального заклад або виробничої компанії, де використовуються: бухгалтерська система; CRM / LMS; вебпортал; аналітична система.
2. Визначити, які системи мають інтегруватися; вказати типи даних для обміну; запропонувати можливі підходи інтеграції.

Загальні відомості

Інтеграційні процеси в інформаційних системах (ІС) – це сукупність методів, технологій і організаційних рішень, спрямованих на **об'єднання різнорідних інформаційних ресурсів, програмних компонентів і бізнес-процесів** в єдине узгоджене середовище [1].

Мета інтеграції – забезпечити цілісність даних, безперервність процесів і ефективну взаємодію між підсистемами.

Сучасні організації використовують багато ІС, створених у різний час, на різних платформах і з різними підходами до зберігання даних. Це зумовлює:

- фрагментацію інформації;
- дублювання даних і функцій;
- складність управління та масштабування;
- затримки в ухваленні управлінських рішень.

Інтеграція дозволяє подолати ці проблеми та сформувати **єдиний інформаційний простір**.

Рівні інтеграції інформаційних систем

Інтеграційні процеси реалізуються на кількох рівнях:

1. Інтеграція даних – об'єднання та синхронізація даних між БД і сховищами (ETL, реплікація, Data Warehouse, Data Lake).
2. Інтеграція додатків визначає взаємодію між програмними компонентами через API, вебсервіси, middleware, ESB.
3. Інтеграція бізнес-процесів – узгодження та автоматизація процесів між системами (BPM, workflow-движки).
4. Інтеграція користувацького інтерфейсу – це єдиний доступ користувача до різних систем (портали, SSO, low-code платформи).

Основні підходи до інтеграції (табл. 1.1):

1. Точкова (Point-to-Point) – проста, але погано масштабується.
2. Сервісно-орієнтована (SOA), що передбачає використання сервісів із чіткими інтерфейсами.
3. На основі корпоративної шини даних (ESB), що передбачає централізоване управління обміном.

4. API-орієнтована інтеграція – гнучка та поширена в сучасних вебсистемах.

5. Подієво-орієнтована (Event-Driven), що передбачає реакцію систем на події в реальному часі.

В інтеграційних процесах використовуються такі технології та інструменти інтеграції:

- 1) REST / SOAP вебсервіси;
- 2) JSON, XML, GraphQL;
- 3) Message brokers (RabbitMQ, Kafka);
- 4) iPaaS-платформи (MuleSoft, Boomi, Azure Logic Apps);
- 5) контейнери та мікросервіси (Docker, Kubernetes).

Таблиця 1.1 – Порівняльна таблиця підходів до інтеграції ІС

Підхід інтеграції	Суть підходу	Переваги	Недоліки	Типові технології	Сфера застосування
Інтеграція на рівні даних	Прямий доступ до БД іншої системи	Простота реалізації, висока швидкість	Порушення цілісності, залежність від структури БД	SQL, ETL, реплікація	Старі системи, звітність
Точкова інтеграція (Point-to-Point)	Пряме з'єднання між системами	Мінімальні витрати на старті	Погана масштабованість	API, файли, скрипти	Невеликі ІС
Інтеграція на рівні додатків	Обмін через логіку програм	Контроль бізнес-логіки	Складність підтримки	RPC, API	Корпоративні ІС
Сервісно-орієнтована інтеграція (SOA)	Використання сервісів	Гнучкість, повторне використання	Високі вимоги до проєктування	SOAP, REST	Великі підприємства
Інтеграція через ESB	Центральна шина обміну	Централізоване управління	Вартість, складність	Mule ESB, WSO2	Корпоративні середовища
Подієва інтеграція	Обмін через події	Масштабованість, реактивність	Складність налагодження	Kafka, RabbitMQ	Реальний час
Мікро-сервісна інтеграція	Незалежні сервіси	Гнучкість, масштабованість	Складна архітектура	REST, gRPC	Хмарні ІС
Інтеграція бізнес-процесів	Узгодження процесів	Прозорість управління	Потребує BPM-інструментів	BPMN, BPEL	ERP, BPM-системи

Контрольні запитання

1. Чому точкова інтеграція не підходить для великих систем?
2. У яких випадках доцільно застосовувати ESB?
3. Чим подієва інтеграція відрізняється від сервісної?
4. Які інтеграційні підходи є найбільш перспективними для індустрії

4.0?

Практична робота № 2

Основні методи інтеграції інформаційних систем

Мета: ознайомитися з основними методами інтеграції інформаційних систем, зрозуміти їхні відмінності, переваги та недоліки.

Завдання: побудувати схему вертикальної, горизонтальної і гібридної інтеграції для однієї з систем – банківської, медичної, логістичної.

Загальні відомості

Системна інтеграція – це процес об'єднання різних ІТ-систем, програмного забезпечення та апаратних засобів у єдину цілісну структуру, яка працює як злагоджений механізм.

Стратегія системної інтеграції – це план, спрямований на організацію і об'єднання різних процесів, технологій і систем внутрішнього середовища підприємства. Стратегія системної інтеграції визначає, як саме будуть інтегровані різні компоненти підприємства для покращення ефективності операцій, підвищення якості продукту або послуги, а також забезпечення більшої гнучкості в управлінні.

Метод системної інтеграції – конкретний спосіб об'єднання систем чи компонентів підприємства. Він може передбачати використання стандартів і протоколів, розробку інтеграційних рішень, розроблення **API** (Application programming interface) для взаємодії між різними системами тощо.

Основні типи інтеграції:

1. **Горизонтальна:** створення окремого шару (шини), який є посередником між усіма системами.
2. **Вертикальна:** системи інтегруються за функціональним принципом (від простих до складних вузлів).
3. **«Точка – точка»:** пряме з'єднання двох програм (швидко, але складно масштабувати).
4. **Зіркоподібна:** центральна система керує всіма іншими підключеними модулями.

Вертикальна інтеграція передбачає поєднання різних рівнів управління або функціональних сфер діяльності підприємства. Наприклад, це може бути інтеграція між системами управління виробництвом та системами управління складом. Вона дозволяє оптимізувати процеси в межах одного підприємства, зменшуючи залежність від зовнішніх постачальників або покупців.

Вертикальна інтеграція – це концепція, яка полягає в побудові послідовного пов'язання між різними рівнями ієрархії інформаційних і виробничих систем підприємства для оптимізації обміну даними, управління та ухвалення рішень.

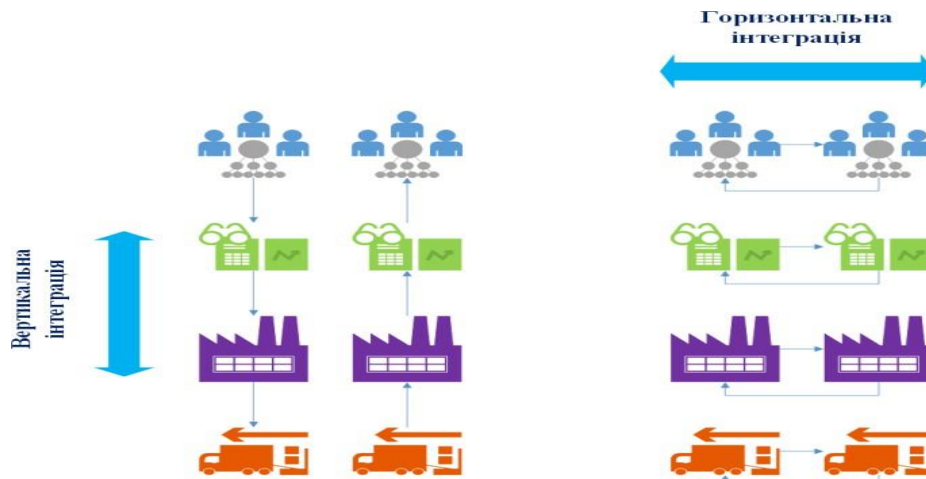


Рисунок 2.1 – Вертикальна та горизонтальна інтеграція систем підприємства [1]

Приклад такої інтеграції може бути побудований за такими рівнями:

1. Сенсори (датчики) збирають первинні дані з виробничого обладнання, процесів, навколишнього середовища.
2. MES (Manufacturing Execution System) – система управління виробничими операціями, що обробляє та аналізує дані, отримані від сенсорів, контролює процес виробництва в режимі реального часу.
3. ERP (Enterprise Resource Planning) – система планування підприємства, яка інтегрує виробничі, фінансові, кадрові та інші бізнес-процеси, отримує дані від MES для стратегічного планування і управління ресурсами.
4. BI (Business Intelligence) – система бізнес-аналітики, яка збирає всі агреговані дані з ERP, MES та інших систем для глибокого аналізу, візуалізації та підтримки ухвалення управлінських рішень.
5. Це дозволяє створити наскрізний потік інформації від безпосереднього вимірювання й контролю на виробничому рівні до стратегічного аналізу та управління на рівні керівництва. Такий підхід підвищує ефективність роботи підприємства, зменшує час реакції на зміни та поліпшує якість ухвалених рішень.

Вертикальна інтеграція виглядає так: сенсори → MES → ERP → BI.

Горизонтальна інтеграція – інтеграція між підсистемами одного рівня виглядає так: CRM ↔ ERP ↔ SCM.

Горизонтальна інтеграція – це інтеграція між підсистемами або компонентами, що знаходяться на одному рівні в архітектурі системи.

Прикладом такої інтеграції є взаємодія між різними бізнес-системами, такими як CRM (система управління взаємовідносинами з клієнтами), ERP (система планування ресурсів підприємства) та SCM (система управління ланцюгом постачання). Горизонтальна інтеграція забезпечує обмін даними і координацію процесів між цими підсистемами, що дозволяє підвищити ефективність і зменшити дублювання інформації.

Гібридна інтеграція передбачає комбінування елементів горизонтальної та вертикальної інтеграції.

Контрольні запитання

1. У чому полягає різниця між вертикальною та горизонтальною інтеграцією?
2. Які ризики існують у разі вертикальної інтеграції?
3. Коли доцільно застосовувати гібридний підхід?
4. Як інтеграція впливає на управління інформаційними системами підприємства?

Практична робота № 3 Основні архітектурні моделі інтегрованих інформаційних систем

Мета: ознайомитися з основними архітектурними моделями інтегрованих систем, їх структурою, перевагами та недоліками.

Завдання: побудувати схеми інтеграційної архітектури системи Windows.

Загальні відомості

Інтегрована система – це інформаційна система, що об'єднує різні підсистеми та модулі в єдине середовище для забезпечення узгодженості даних та ефективності управління.

Принципи інтеграційної архітектури ОС:

1. Модульність: ядро і сервіси реалізуються у вигляді незалежних модулів.
2. Універсальність інтерфейсів: системні виклики та API забезпечують однаковий доступ до різних ресурсів.
3. Міжпроцесна комунікація, що передбачає узгодженість і безпечний обмін даними між процесами.
4. Масштабованість, що передбачає підтримку розподілених систем, хмарних інтеграцій.
5. Безпека, що передбачає контроль доступу, шифрування, ізоляцію процесів.

Інтегровані інформаційні системи можуть будуватись на основі різних архітектур. Основні типи архітектур:

1. Монолітна архітектура – єдина система, яка охоплює всі бізнес-процеси. Усі модулі об'єднані в єдине програмне середовище.
2. Клієнт-серверна архітектура – взаємодія клієнтських додатків із центральним сервером. Поділ системи на серверну частину (логіка та дані) і клієнтську частину (інтерфейс користувача).
3. Багаторівнева архітектура (n-tier) – розділення на рівні: презентації, бізнес-логіки, даних.

4. Сервісно-орієнтована архітектура (SOA) передбачає використання окремих сервісів, що взаємодіють через стандартизовані інтерфейси. Компоненти системи функціонують як сервіси з чітко визначеними інтерфейсами.

5. Архітектура на основі мікросервісів передбачає поділ системи на незалежні сервіси з власними БД та API. Система складається з незалежних сервісів, що взаємодіють через API.

Монолітна архітектура (Monolithic Architecture) – це традиційний підхід до розробки програмного забезпечення, за якого весь додаток розробляється як одна єдина технологічна система. Всі компоненти додатка взаємодіють один з одним і розгортання відбувається на одному сервері або групі серверів.

Інтеграційну архітектуру операційної системи (ОС) можна розглядати як сукупність компонентів і механізмів, що забезпечують взаємодію між апаратним забезпеченням, системними службами, прикладними програмами та зовнішніми сервісами. Вона орієнтована на інтеграцію різних рівнів та модулів у єдине цілісне середовище.

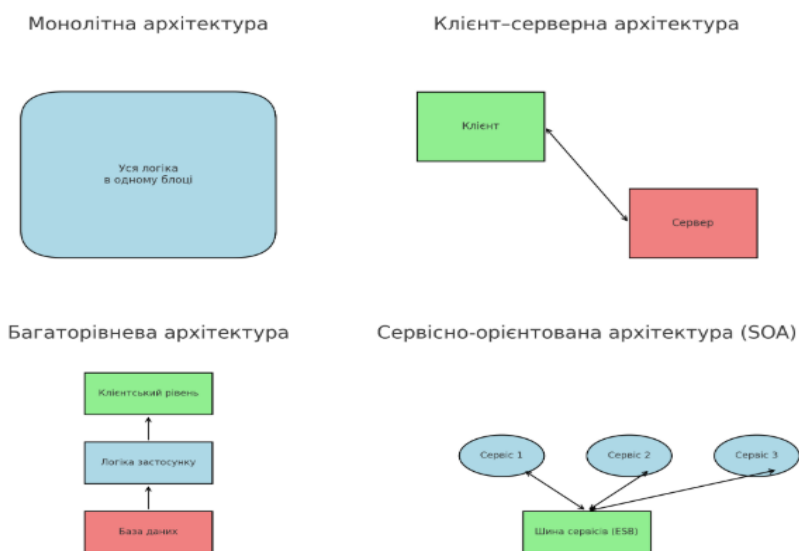


Рисунок 3.1 – Основні типи архітектур

Кожна архітектура має свої переваги і обмеження. Монолітна підходить для невеликих проєктів, де важлива швидкість розробки. Клієнт-серверна є базовим стандартом для розподілених систем. Багаторівнева архітектура застосовується у великих корпоративних рішеннях з високими вимогами до підтримки та масштабування. Сервісно-орієнтована (SOA) найбільш гнучка і підходить для інтеграції у хмарному середовищі та сучасних бізнес-процесах. Архітектура інтегрованих систем визначає їхню ефективність, масштабованість та витрати на підтримку. Вибір конкретної моделі залежить від сфери застосування та вимог до продуктивності й надійності.

Основні рівні інтеграційної архітектури ОС

1. Апаратний рівень (Hardware Layer): процесор, пам'ять, контролери, пристрої введення / виведення; драйвери пристроїв інтегрують фізичне обладнання у функціонал ОС.

2. Ядро операційної системи (Kernel Layer):

- диспетчер процесів;
- керування пам'яттю;
- драйвери пристроїв;
- файлові системи;
- системні виклики для взаємодії з прикладними програмами.

Ядро є ключовим інтеграційним компонентом між апаратним забезпеченням і прикладними сервісами.

3. Системні бібліотеки та сервіси (System Libraries & Services Layer):

- стандартні бібліотеки (glibc, .NET libraries, WinAPI тощо);
- інтерфейси для роботи з мережею, базами даних, графічними системами;
- сервіси синхронізації, безпеки, логування.

4. Інтеграційний рівень сервісів та інтерфейсів (Integration & Middleware Layer):

- механізми міжпроцесної взаємодії (IPC);
- API та SDK для розробників;
- драйвери віртуалізації та контейнеризації (наприклад, Docker, Hyper-V);
- інтеграція з хмарними сервісами та віддаленими інфраструктурами.

5. Прикладний рівень (Application Layer):

- користувацькі програми (офісні пакети, браузер, системи управління підприємством);
- взаємодія з користувачем через GUI / CLI;
- інтеграція з мобільними та вебдодатками.

Таблиця 3.1 – Порівняння архітектур

Тип архітектури	Особливості	Переваги	Недоліки	Приклади застосування
1	2	3	4	5
Монолітна	Єдина програма	Простота розгортання	Складність масштабування	Старі ERP-системи
Клієнт-серверна	Сервер + клієнти	Централізоване управління	Високе навантаження на сервер	Банківські системи
Багаторівнева (n-tier)	Поділ на рівні: UI, логіка, дані	Гнучкість, масштабованість	Складність підтримки	Корпоративні ІС

Продовження таблиці 3.1

1	2	3	4	5
SOA	Сервіси з відкритими інтерфейсами	Повторне використання, інтеграція	Потреба у стандартизації	Інтеграція ERP + CRM
Мікросервісна	Незалежні сервіси з API	Масштабованість, стійкість до збоїв	Ускладнене управління	Amazon, Netflix
SOA	Сервіси з відкритими інтерфейсами	Повторне використання, інтеграція	Потреба у стандартизації	Інтеграція ERP + CRM

Контрольні запитання

1. Надайте визначення інтегрованої системи та поясніть її роль у сучасних підприємствах.
2. Опишіть особливості і сфери застосування архітектур інтегрованих систем.
3. Назвіть основні рівні інтеграційної архітектури ОС.

Практична робота № 4

Розробка плану приміщень та плану комп'ютерної мережі

Мета роботи: отримати навички проектування плану приміщень установи і плану комп'ютерної мережі з використанням інструментального засобу, наприклад, Microsoft Office Visio. Ознайомитися з безкоштовними аналогами MS Visio.

Завдання: створити проєкт локальної мережі приміщення підприємства, установи чи організації, які розглядалися в практичній роботі № 1, приміщення має містити не менше трьох кімнат та відрізняється від приміщень, обраних іншими студентами.

Вихідними даними для цього є: кількість кімнат на поверсі, робочі місця користувачів комп'ютерної мережі та розподіл робочих місць.

На основі вихідних даних необхідно спроектувати план одного поверху, враховуючи, що одна з кімнат поверху повинна бути серверною кімнатою з одним робочим місцем для адміністратора мережі (серверна кімната входить до переліку кімнат із вихідних даних). Також необхідно врахувати всі вимоги щодо розташування серверної кімнати (двері, вікна тощо).

Загальні відомості

Пасивне мережеве обладнання. Під час проектування комп'ютерних мереж в офісних приміщеннях використовують кабельні лотки та пластикові коробки.

Кабельний лоток – це відкрита конструкція, призначена для монтажу дротів і кабелів.

Короб кабельний – конструкція з пластмаси для монтажу кабельних мереж усередині приміщення. Пластикові коробки поділяються на кілька основних видів:

1) *кабельний канал (кабель-канал)* має просту конструкцію, він досить дешевий, деякі моделі дозволяють встановлювати розетки всередину кабель-каналу;

2) *парапетні коробки* встановлюються на рівні робочого місця, внутрішній простір такого короба розділений на секції, він має подвійну стінку, і практично всі види парапетного короба підтримують монтаж розеток;

3) *короб на підлогу* – це короб для монтажу на підлогу, що має посилену конструкцію та стійку до стирання поверхню.

Вимоги до серверної кімнати. Серверна кімната – це приміщення для великого телекомунікаційного або серверного обладнання. Розміри серверної повинні відповідати вимогам до розташовуваного в ній обладнання. Якщо таких даних на момент вибору приміщення немає, то розрахунки ведуться з огляду на площу робочих місць, що обслуговуються: на кожні її 10 м² обираються 0,07 м² для серверної. Мінімальна площа апаратної дорівнює 14 м².

Серверна кімната повинна розташовуватися в приміщенні, яке не має зовнішніх стін будинку. Для забезпечення катастрофостійкості приміщень критичного електронного, електричного або механічного обладнання та комп'ютерів ці приміщення не допускається розміщати у підвальних поверхах або нижче очікуваного рівня повідкових вод, і на верхніх поверхах будинку, оскільки вони сильніше за інших страждають у випадку пожежі.

Конструкція стін приміщення повинна бути герметичною, при цьому стіни та двері повинні мати вогнестійкість не менше ніж 45 хвилин, а міжповерхові перекриття, окрім цього, повинні мати гідроізоляцію. Ширина дверей у серверну повинна бути не менше ніж 910 мм, висота – 2 000 мм. Конструкція дверей має певні обмеження: полотно повинне відкриватися назовні на 180°, а дверна коробка не повинна мати порогу. Під час використання в серверній великогабаритного обладнання передбачається встановлення двостулкових дверей. Для забезпечення герметичності в конструкції дверей повинна бути ущільнювальна прокладка, а для підвищення рівня захисту від злому необхідно передбачити протиз'ємне пристосування.

У серверній не повинно бути вікон. Обов'язковою умовою в цьому приміщенні є наявність фальш-підлоги, що витримує навантаження від обладнання, що встановлюється, і працюючих з ним людей. Рекомендована відстань між плитою на підлозі та фальш-підлогою – 400 мм, при цьому прозір між фальш-підлогою і фальш-стелею повинен бути не менше ніж 2 440 мм. Фальш підлогу рекомендується робити з легкоз'ємних модулів. Матеріал, із якого вона виготовлена, повинен бути міцним, зносостійким, мати погану займистість і мати електричний опір відносно землі від 1 Ом до 20 Ом.

Використання килимових покриттів у таких приміщеннях суворо заборонене. Перекриття під фальш-підлогою повинне бути герметизованим або пофарбованим.

Нумерація (маркування) розеток. Усі розетки в комп'ютерній мережі повинні бути пронумеровані. Причому, номер розетки повинен бути зазначений (приклеєний, підписаний) безпосередньо поруч із розеткою. Для кожного користувача комп'ютерної мережі повинні бути зарезервовані 2 розетки: комп'ютерна для під'єднання комп'ютера користувача до комп'ютерної мережі та телефонна для підключення телефону. Правила нумерації розеток не регламентуються, але необхідно підкреслити, що кожна розетка повинна мати свій унікальний номер, а також пошук фізичного розташування розетки повинен бути не складним. Пропонується така складена нумерація розеток – 01-01-K01:

- перша і друга цифри – номер поверху;
- третя та четверта цифри – номер кімнати;
- п'ятий символ – тип розетки (К – комп'ютерна, Т – телефонна);
- шоста і сьома цифри – порядковий номер розетки.

Типи кабельних сегментів. Під час проєктування комп'ютерної мережі необхідно враховувати характеристики кабельних сегментів. Кабельний сегмент – відрізок кабелю або ланцюг відрізків кабелів, електрично (оптично) з'єднаних один з одним, що забезпечують з'єднання двох або більше вузлів мережі. Особливо важливо враховувати довжину кабельного сегмента. У таблиці 4.1 надані основні характеристики кабельних сегментів [3].

Таблиця 4.1 – Характеристики кабельних сегментів

Номер	Стандарт	Швидкість передачі даних	Тип кабелю, що використовується	Максимальна довжина сегмента
1	Ethernet 10Base-2	10 Мбіт/с	тонкий коаксіальний	185 м
2	Ethernet 10Base-5	10 Мбіт/с	товстий коаксіальний	500 м
3	Ethernet 10Base-F	10 Мбіт/с	волоконно-оптичний	2 км
4	Ethernet 10Base-T	10 Мбіт/с	вита пара	100 м
5	Ethernet 100Base-FX	100 Мбіт/с	волоконно-оптичний	2 000 м
6	Ethernet 100Base-T	100 Мбіт/с	вита пара	100 м
7	Ethernet 100Base-T2	100 Мбіт/с	UTP 3	100 м
8	Ethernet 100 Base-T4	100 Мбіт/с	UTP5, STP	100 м
9	Ethernet 1000 Base-CX	1 000 Мбіт/с	STP	25 м
10	Ethernet 1000 Base-LX	1 000 Мбіт/с	волоконно-оптичний	одномод. 5 000 м багатомод. 550 м
11	Ethernet 1000 Base-T	1 000 Мбіт/с	UTP 5	100 м

Таблиця 4.2 – Вихідні дані

Варіант № 1		Варіант № 2		Варіант № 3		Варіант № 4	
Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць
1	7	1	1	1	4	1	4
2	6	2	6	2	8	2	8
3	9	3	7	3	10	3	8
4	5	4	10	4	3	4	3
5	5	5	5	5	5	5	5
6	2	6	7	6	4	6	8
7	1			7	1	7	1
Варіант № 5		Варіант № 6		Варіант № 7		Варіант № 8	
Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць
1	5	1	5	1	25	1	30
2	8	2	7	2	5	2	3
3	10	3	12	3	1	3	2
4	5	4	1	4	7	4	1
5	5	5	9	5	15	5	1
6	3	6	5	6	3	6	4
7	1	7	1				
Варіант № 9		Варіант № 10		Варіант № 11		Варіант № 12	
Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць
1	1	1	3	1	1	1	10
2	7	2	1	2	3	2	5
3	10	3	5	3	10	3	1
4	12	4	7	4	7	4	8
5	3	5	9	5	14	5	9
6	4	6	5	6	5	6	4
7	6	7	8	7	6	7	4
8	2	8	1				
Варіант № 13		Варіант № 14		Варіант № 15		Варіант № 16	
Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць
1	6	1	5	1	25	1	18
2	8	2	7	2	5	2	3
3	9	3	3	3	1	3	2
4	5	4	1	4	4	4	5
5	5	5	9	5	12	5	1
6	1	6	5	6	3	6	4
7	3	7	3				

Варіант № 17		Варіант № 18		Варіант № 19		Варіант № 20	
Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць	Номер кімнати	К-ть робочих місць
1	15	1	1	1	14	1	10
2	8	2	4	2	5	2	3
3	5	3	10	3	3	3	2
4	5	4	2	4	7	4	6
5	5	5	3	5	13	5	1
6	3	6	5	6	1	6	4
7	1	7	3				

Порядок виконання роботи

1. Визначити форму периметра зовнішніх несучих стін будівлі.
2. Спроектувати план поверху, тобто визначити розташування кімнат на поверсі. Необхідно також підписати номери кімнат. На поверсі повинні бути коридори для переміщень, серверна кімната, місця для комунікацій.
3. Показати розміри кімнат. Це необхідно для визначення порядку довжинкабельних сегментів від серверної до офісних кімнат.
4. Ґрунтуючись на вихідних даних визначити робочі місця користувачів комп'ютерної мережі. Для цього необхідно використовувати відповідні елементи Microsoft Office Visio: столи, стільці, комп'ютери тощо.
5. Визначити місце розташування коробів, лотків, комп'ютерних мережевих розеток. Короба, лотки й розетки необхідно пронумерувати.
6. Заповнити кабельний журнал, у якому необхідно вказати відповідність мережевого обладнання, порту мережевого обладнання, мережевої комп'ютерної розетки, номера кімнати та ім'я комп'ютера. Приклад кабельного журналу подано у таблиці 4.3.

Таблиця 4.3 – Приклад кабельного журналу*

№ з/п	Назва пристрою	Номер порту	Номер розетки	Ім'я комп'ютера	Номер кімнати
1	2	3	4	5	6
1	KM01	01	01-01-K01	01-01-01	01
		02	01-01-K02	01-01-02	
		03	01-01-K03	01-01-03	
2	KM02	01	01-01-T01	01-01-01	
		02	01-01-T02	01-01-02	
		03	01-01-T03	01-01-03	
3	KM03	01	01-02-K04	01-02-04	02
		02	01-02-K05	01-02-05	
		03	01-02-K06	01-02-06	
		04	01-02-K07	01-02-07	

Продовження таблиці 4.3

1	2	3	4	5	6
4	KM04	01	01-02-T04	01-02-04	
		02	01-02-T05	01-02-05	
		03	01-02-T06	01-02-06	
		04	01-02-T07	01-02-07	
5	MP01	01	01-05-K36	01-05-36	03

* Умовні позначення: КМ – комутатор, МР – маршрутизатор.

Приклад 1. Створити локальну мережі підрозділу з виробництва напівфабрикатів супермаркету «Наш продукт».

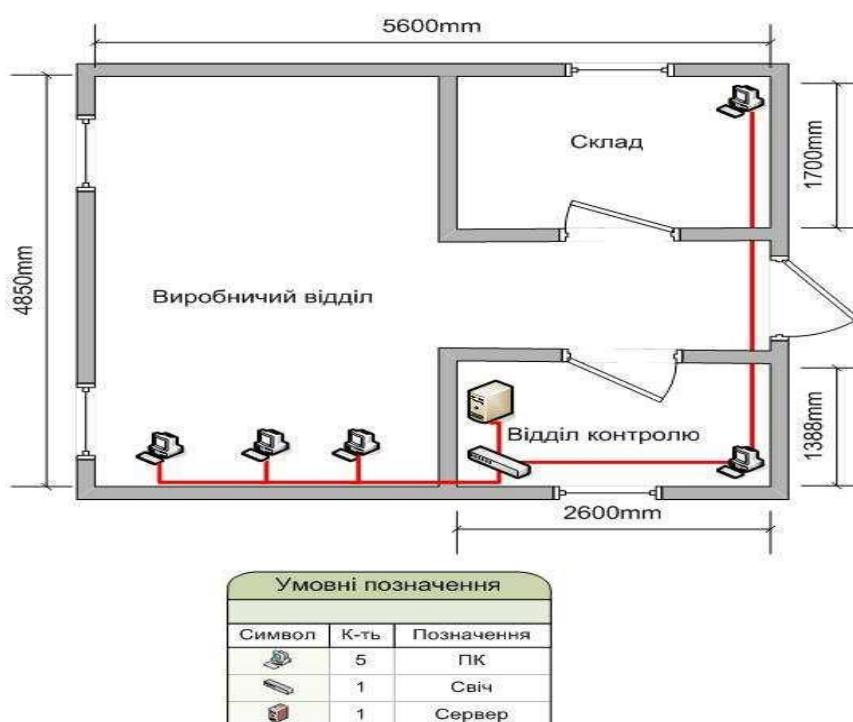


Рисунок 4.4 – Приклад проєкту локальної мережі приміщення підприємства

Приклад 2. Розробити план приміщень та плану комп'ютерної мережі управління постачанням книг до громадських бібліотек:

Варіант № 99	
Номер кімнати	Кількість робочих місць
1	25
2	5
3	1
4	7
5	15
6	3

Таблиця 4.4 – Характеристика приміщень

Кімната № 1	Кімната № 2	Кімната № 3
«Програмне забезпечення та веброзробка»: – робочі місця – 25 комп'ютерів; – площа кімнати – 28,5 м ² ; – кількість вікон – 4; – кількість дверей – 1	«Обробка та розподіл замовлень»: – робочі місця – 5 комп'ютерів; – площа кімнати – 8,1 м ² ; – кількість вікон – 2; – кількість дверей – 1	«Сервер»: – кількість робочих місць – 1; – площа кімнати – 4,5 м ² ; – кількість вікон = 0; – кількість дверей = 1
Кімната № 4	Кімната № 5	Кімната № 6
«Тестування та технічна підтримка»: – кількість робочих місць – 7; – площа кімнати – 7,92 м ² ; – кількість вікон – 2; – кількість дверей – 1	«Пакування, реєстрація та відправлення»: – кількість робочих місць – 15; – площа кімнати – 18,45 м ² ; – кількість вікон – 7; – кількість дверей – 1	«Адміністрація»: – кількість робочих місць – 3; – площа кімнати – 3,84 м ² ; – кількість вікон – 1; – кількість дверей – 1

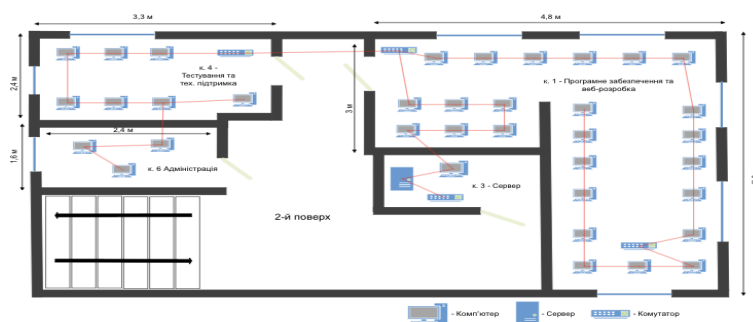


Рисунок 4.5 – План приміщень та комп'ютерної мережі

На основі вихідних даних необхідно спроектувати план одного поверху, враховуючи, що одна з кімнат поверху повинна бути серверною кімнатою з одним робочим місцем для адміністратора мережі (серверна кімната входить до переліку кімнат із вихідних даних). Також необхідно врахувати всі вимоги щодо розташування серверної кімнати (двері, вікна тощо). Розробка плану здійснюється за варіантом (табл. 4.2).

Контрольні запитання

1. Зробіть аналіз безкоштовних аналогів MS Visio.
2. Назвіть основні вимоги до створення серверної кімнати.
3. Яким чином нумеруються комп'ютерні і телефонні розетки у комп'ютерній мережі?
4. Перерахуйте основні характеристики типів кабельних сегментів.
5. З якою метою необхідно вказувати розміри кімнат на плані поверху?
6. Що таке кабельний лоток і пластиковий короб?
7. Що входить до робочого місця користувача комп'ютерної мережі?

Практична робота № 5

Алгоритм пошуку найкоротшого шляху у комп'ютерних мережах

Мета: навчитися реалізовувати алгоритм Дейкстри для пошуку найкоротшого шляху у комп'ютерних мережах від одного вузла (однієї вершини графа) до інших.

Завдання: визначити найкоротшій шлях у мережі відповідно до варіанта.

Загальні відомості

Алгоритм Дейкстри – це алгоритм на графах, відкритий Дейкстрою. Знаходить найкоротший шлях від однієї вершини графа до всіх інших вершин. Класичний алгоритм Дейкстри працює тільки для графів без циклів від'ємної довжини [4].

Варіанти завдань

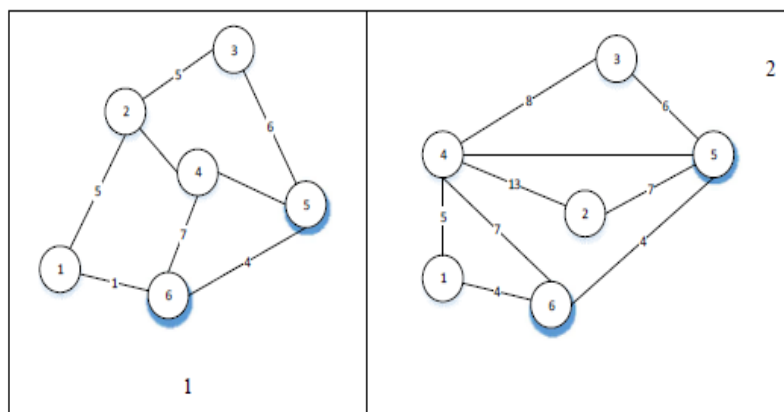
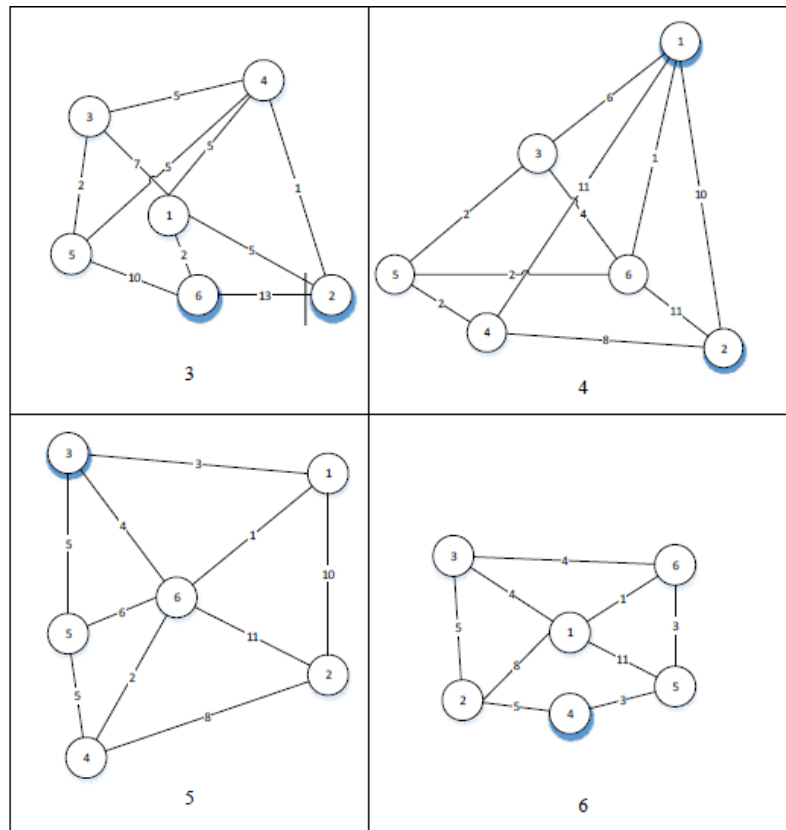


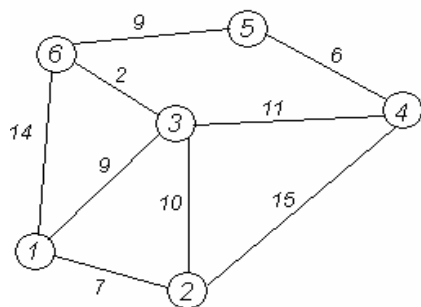
Рисунок 5.1 – Варіанти завдань



Продовження рисунка 5.1

Приклад. Подано неорієнтований зв'язний граф $G(V, U)$. Знайти відстань від вершини **a** до всіх інших вершин V .

Зберігається поточна мінімальна відстань до всіх вершин V (від поданої вершини **a**) і на кожному кроці алгоритму зменшується відстань. Спочатку встановлюються відстані до всіх вершин, які рівні нескінченності, а до вершини **a** – нулю.



а)

0	7	9	0	0	14
7	0	10	17	0	0
9	10	0	11	0	2
0	17	11	0	6	0
0	0	0	6	0	9
14	0	2	0	9	0

б)

Рисунок 5.2 – Графічне і матричне представлення даних:

а – графічне; б – матричне

Необхідно знайти відстані від першої вершини до всіх інших. Кругечками позначені вершини, лініями – шляхи між ними («дуги»). Над дугами позначена їхня «ціна» – довжина шляху. Надписом над кругечком позначена поточна найкоротша відстань до вершини.

Порядок виконання

Крок 1. Ініціалізація. Відстань до всіх вершин графа $V = \infty$. Відстань до $a = 0$.

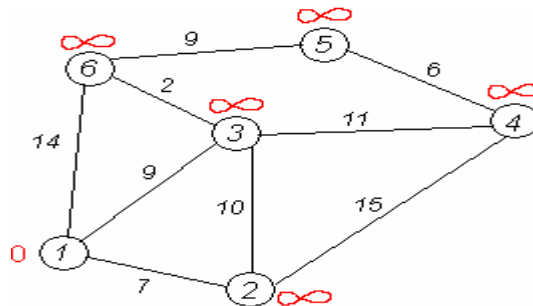


Рисунок 5.3 – Неопрацьовані вершини графа

Жодна вершина графа ще не опрацьована.

Крок 2. Знаходимо таку вершину (з іще не оброблених), поточна найкоротша відстань до якої мінімальна. У нашому випадку це вершина 1. Обходимо всіх її сусідів і, якщо шлях в сусідню вершину через 1 менший за поточний мінімальний шлях в цю сусідню вершину, то запам'ятовуємо цей новий, коротший шлях як поточний найкоротший шлях до сусіда.

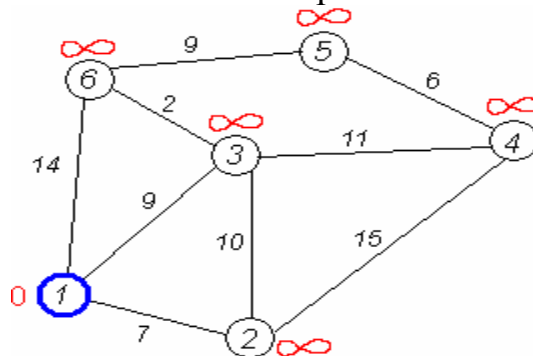


Рисунок 5.4 – Визначення поточної найкоротшої відстані

Крок 3. Перший за порядком сусід першої вершини – друга вершина. Шлях до неї через першу вершину дорівнює найкоротшій відстані до першої вершини плюс довжина дуги між першою та другою вершиною, тобто $0 + 7 = 7$. Це менше поточного найкоротшого шляху до другої вершини, тому найкоротший шлях до другої вершини дорівнює 7.

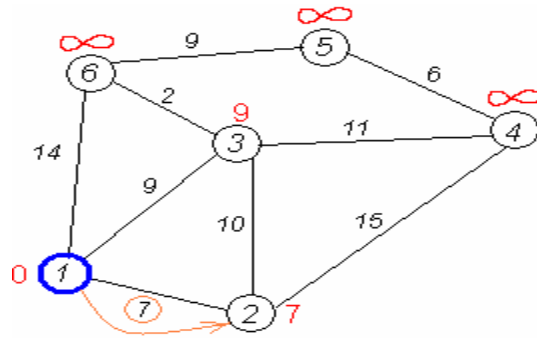


Рисунок 5.5 – Визначення другої вершини графа

Кроки 4, 5. Аналогічну операцію проробляєм з двома іншими сусідами першої вершини – третьою та шостою.

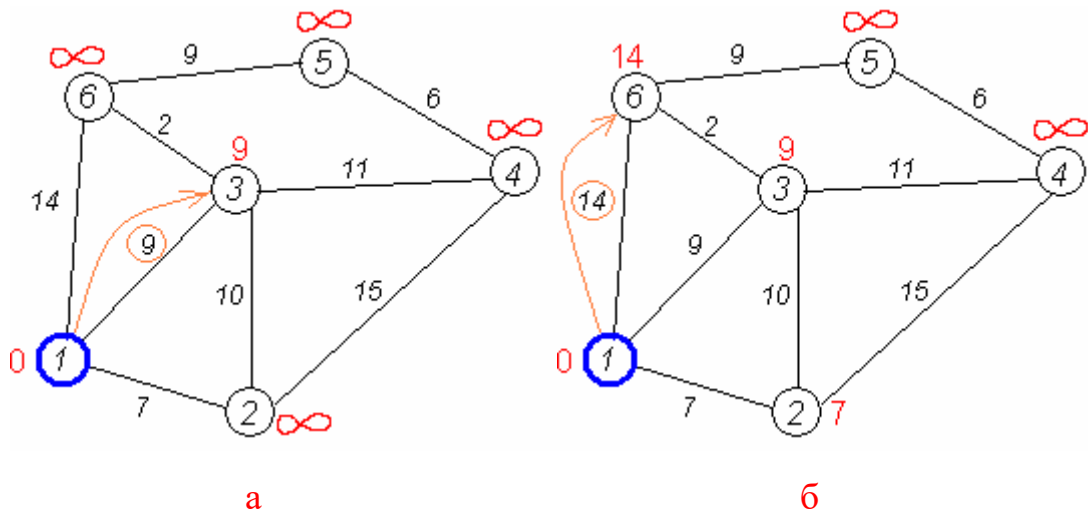


Рисунок 5.6 – Визначення відстаней від сусідніх вершин:
а – вершина 3; б – вершина 6

Крок 6. Всі сусіди першої вершини перевірені. Поточна мінімальна відстань до першої вершини вважається остаточною і обговоренню не підлягає (те, що це дійсно так, вперше довів Дейкстра), тому викреслимо її з графа, щоб відмітити цей факт.

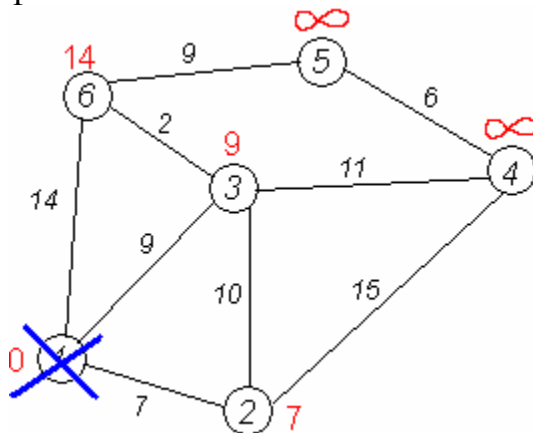


Рисунок 5.7 – Поточна мінімальна відстань до першої вершини

Крок 7. Практично відбувається повернення до кроку 2. Знову знаходимо «найближчу» необроблену (невикреслену) вершину. Це друга вершина з поточною найкоротшою відстанню до неї, яка дорівнює 7. І знову намагаємося зменшити відстань до всіх сусідів другої вершини, намагаючись пройти в них через другу. Сусідами другої вершини є перша, третя, четверта.

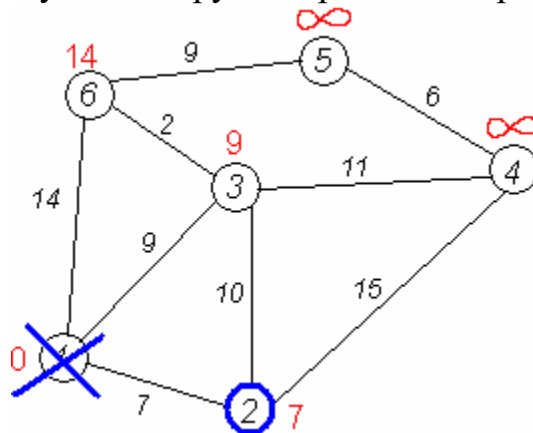


Рисунок 5.8 – Визначення «найближчої» необробленої (невикресленої) вершини

Крок 8. Перший (по порядку) сусід другої вершини – перша вершина. Але вона вже оброблена (або викреслена – див. крок 6). Тому з першою вершиною нічого не робимо.

Крок 8 (з іншими вхідними даними). Інший сусід вершини другої – вершина четверта. Якщо йти в неї через другу, то шлях буде дорівнювати найкоротшій відстані до другої плюс відстань між другою і четвертою вершинами становить: $7 + 15 = 22$. Оскільки $22 < \infty$, встановлюємо відстань до четвертої вершини рівним 22.

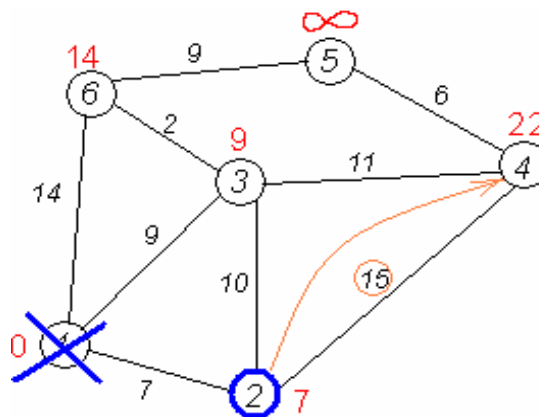


Рисунок 5.9 – Визначення відстані для вершини 4

Крок 9. Ще один сусід вершини 2 — вершина 3. Якщо йти в неї через 2-у, то шлях буде $= 7 + 10 = 17$. Але 17 більше за відстань, що вже запам'ятали раніше до вершини № 3 і дорівнює 9, тому поточну відстань до 3-ої вершини не змінюємо.

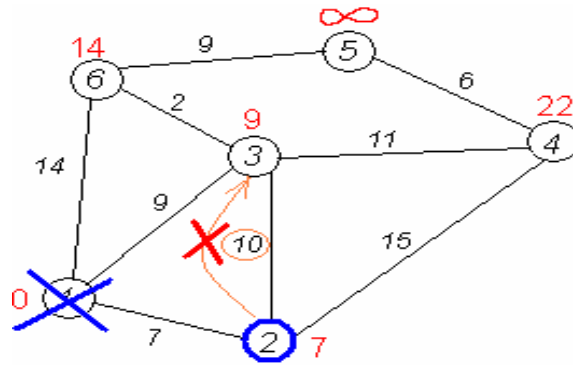


Рисунок 5.10 – Визначення поточної відстані від 2-ї до 3-ої вершини.

Перегляд сусідніх точок

Крок 10. Всі сусіди другої вершини переглянуті, заморожуємо відстань до неї і викреслюємо її з графа.

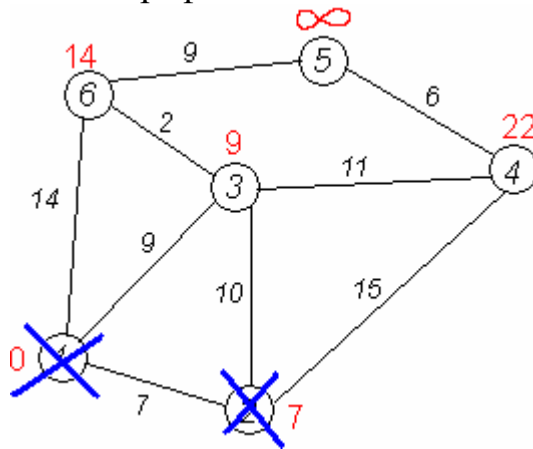


Рисунок 5.11 – Граф з викресленими двома вершинами

Кроки 11–15. За вже «відпрацьованою» схемою повторюємо кроки 2–6. Тепер «найближчою» виявляється вершина № 3. Після її «обробки» отримаємо результати.

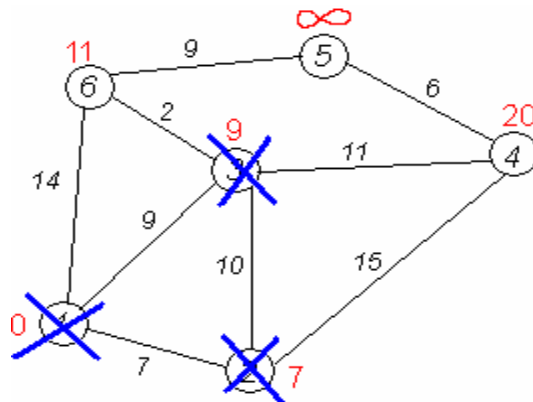


Рисунок 5.12 – Граф з викресленими трьома вершинами

Наступні кроки. Проробляємо те саме з вершинами, що залишилися (по порядку: шостою, четвертою і п'ятою).

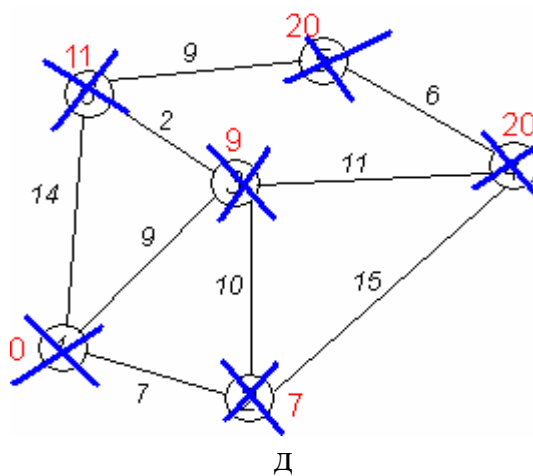
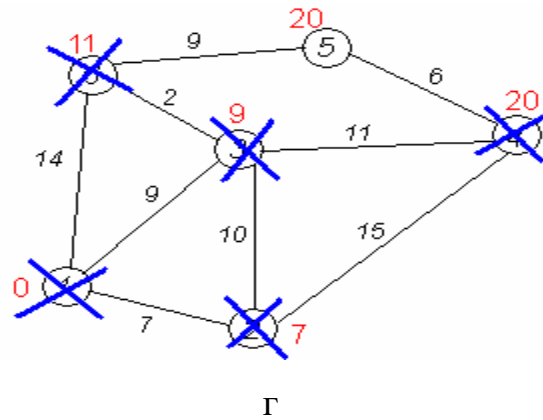
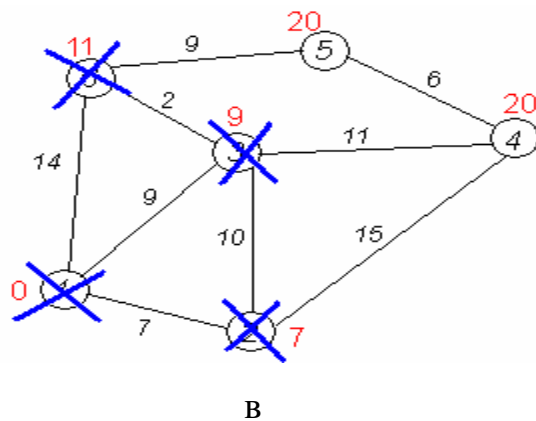
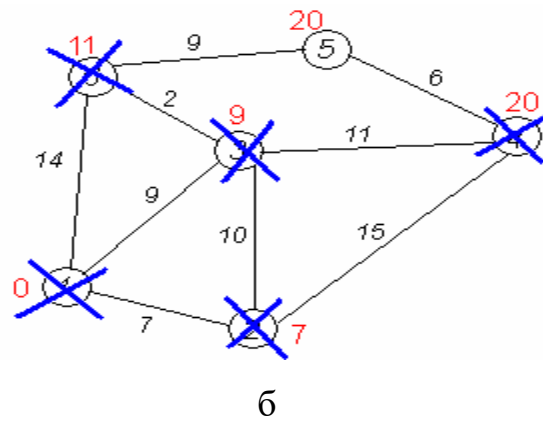
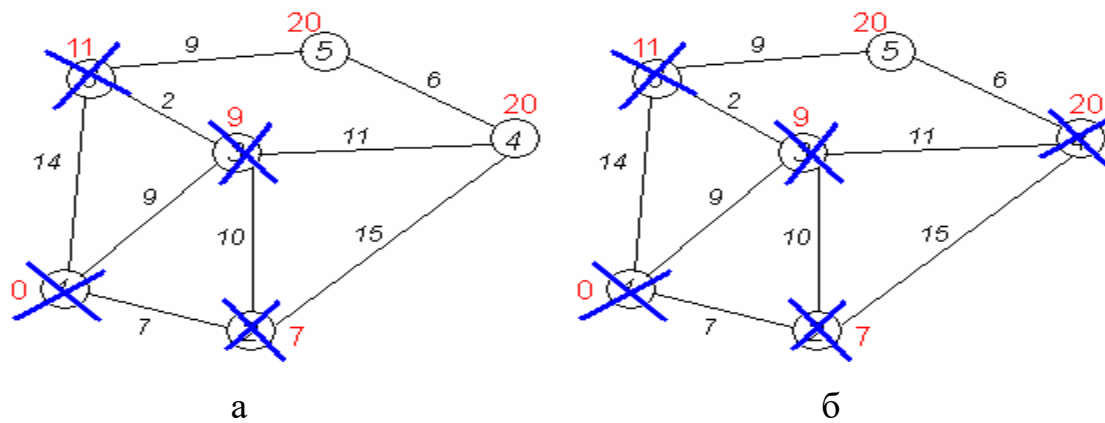


Рисунок 5.13 – Граф з викресленими вершинами:
а – чотирма вершинами; б – п'ятьма вершинами; в – з чотирма вершинами;
г – з п'ятьма вершинами; д – з шістьма вершинами.

Завершення виконання алгоритму. Алгоритм закінчує роботу, коли викреслені всі вершини. Результат його роботи видно на останньому рисунку: найкоротший шлях від першої вершини до другої становить 7, до третьої – 9, до четвертої – 20, до п'ятої – 20, до шостої – 11 умовних одиниць.

Порядок виконання

1. Написати програму, яка реалізує алгоритм Дейкстри.
2. Відкомпілювати та налагодити програму.
3. Протестувати роботу програми, використовуючи різні вхідні дані.
4. Відповісти на контрольні запитання.

Контрольні запитання

1. Які елементи можна розмістити на рисунку для комп'ютерної мережі, якщо вважати, що це мережа?
2. Що таке граф?
3. Які є види графів?
4. Як можуть задаватися графи?
5. Що таке вершина у графі?
6. Що таке ребро у графі?
7. Які алгоритми для пошуку мінімальної відстані у графі вам відомі?

Практична робота № 6 Адресація в мережах

Мета: знайомство з адресацією у мережах.

Завдання: створіть самостійно мережу класу A , розділити її на m підмереж, у першій, другій, третій і четвертій підмережах повинне бути n вузлів, у п'ятій підмережі – p вузлів, у шостій підмережі – r вузлів.

Загальні відомості

Мережа – це загальне поняття, яке має різні втілення й тлумачення.

Мережа – це об'єднання однорідних об'єктів, яке має правила взаємодії його членів між собою й з об'єктами поза ним. Однорідність членів мережі дозволяє взаємодіяти з ними в єдиний спосіб, об'єднання – взаємодіяти з мережею як єдиним об'єктом.

Мережі можуть бути елементами більших, загальніших мереж, утворюючи ієрархії.

Мережа здійснює необхідні перетворення даних під час передачі. Різновиди повідомлень, що передаються: текстові, звукові, зображення тощо, а також спеціальні повідомлення для керування мережею (повідомлення про зміни, помилки, несправності тощо).

Мережа – це система каналів зв'язку, призначена для передачі інформації шляхом передачі даних у вигляді повідомлень.

Приклади мереж:

- комп'ютерна мережа;
- телефонна мережа;
- мережа стільникового зв'язку;
- мережа кабельного телебачення.

В якості каналів передачі даних у мережах використовуються різноманітні засоби й середовища: телефонні кабелі, спеціальні мережеві кабелі (коаксіальні кабелі, кручені пари, волоконно-оптичні кабелі), радіохвилі, світлові сигнали.

Комп'ютерна мережа – мережа, утворена з ліній зв'язку та мережевого обладнання так, що приєднані до них комп'ютери можуть обмінюватися один з одним ресурсами (процесорний час, пам'ять, дані тощо). Кожен комп'ютер з'єднується з комунікаційною мережею за допомогою спеціального пристрою – мережевого адаптера, який називається (мережевим) інтерфейсом.

Комп'ютерні мережі здатні забезпечувати:

- швидку передачу даних на великі відстані;
- оперативний пошук інформації на віддалених комп'ютерах;
- обмін текстової, звукової та відеоінформації у режимі реального часу;
- інтерактивний та оперативний зв'язок між віддаленими комп'ютерами.

Вузол (від англ. «node») є одним з елементів комп'ютерних мереж – це пристрій, з'єднаний з іншими пристроями як частина комп'ютерної мережі. Вузлом може бути комп'ютер, мобільний засіб зв'язку або спеціальний мережевий пристрій.

Хост (від англ. «host» – «господар») – це комп'ютер, який має власне під'єднання до мережі й містить зображення інформації, призначене для відправки іншим об'єктам.

Вузли в мережах обмінюються даними за певними алгоритмами, які відповідають призначенню вузлів і реалізовані апаратно або програмно. Реалізацію алгоритмів обміну даними в мережах називають **мережевими протоколами**. Існують сотні різноманітних протоколів і робота зі створення нових протоколів та вдосконалення існуючих ведеться постійно.

Для ідентифікації абонентів і знаходження місця їх розташування в мережі кожному з них привласнюється ознака – адреса.

Під адресою розуміється умовний номер, знання якого є достатнім для доставки повідомлення необхідному абонентові.

Сукупність правил надання адрес абонентів, з обліком місць їх розташування й структури мережі називається *способом адресування*.

Існуючий стек протоколів пакетних мереж використовує три типи адрес:

- локальні (що мають назву апаратні фізичні адреси);
- мережні (IP-адреси);
- символічні (доменні імена).

Фізична локальна адреса – це адреса, що використовується для доставки даних у локальних межах (підмережах), які є елементами складної мережі.

Фізична локальна адреса – це **MAC-адреса (Media Access Control)**, яка призначається мережним адаптерам і мережним інтерфейсам маршрутизаторів, інтерфейсам ПК й усім інтерфейсам елементів мережі. MAC-адреса призначається виробником обладнання і є унікальною, оскільки

управляється централізовано. Для усіх відомих технологій ЛОМ MAC-адреса (рис. 6.1) має формат 48 біт (6 октетів), наприклад: 11-A0-17-3D-BC-01.

MAC-адресу можна розділити на дві частини (рис. 6.1). У першій частині вказується *унікальний ідентифікатор виробника устаткування* (Organizationally Unique Identifier, OUI). Цей унікальний ідентифікатор привласнюється виробникові інститутом IEEE.

Останні 24 біти MAC-адреси призначаються безпосередньо виробником устаткування. Перший біт MAC-адреси (I/G) вказує, чи є адреса індивідуальною або груповою:

- 0 (індивідуальна) – адреса, асоційована з певним мережним пристроєм;
- 1 (групова) – адреса, асоційована з кількома або всіма вузлами цієї мережі.

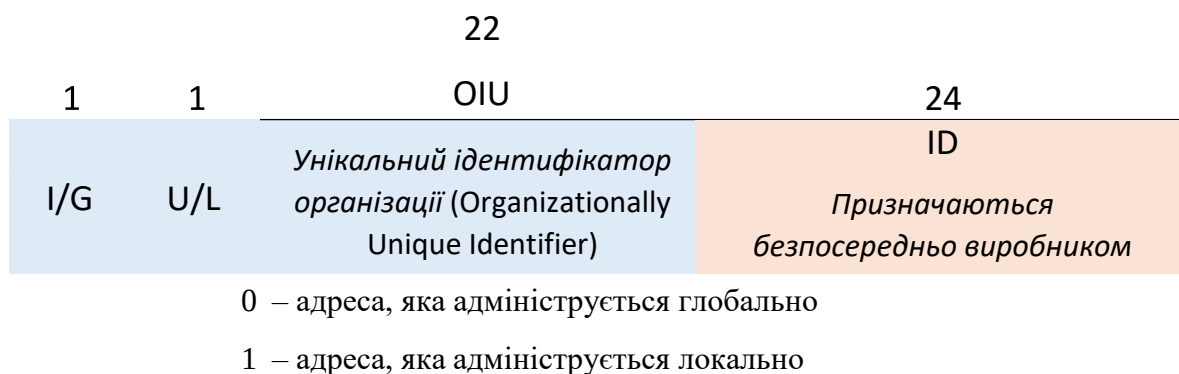


Рисунок 6.1 – Формат MAC-адреси [3]

Існують два види групових адрес:

– *багатоадресна* або *групова (multicast)* – адреса, асоційована із групою вузлів мережі;

– *широкомовна (broadcast)* – адреса, асоційована з усіма вузлами мережі. Її значення – $(0 \times 11) \text{ F-FF-FF-FF-FF-FF}$.

Другий біт MAC-адреси (U/L) указує, чи є MAC-адреса глобально або локально адміністрованою:

– 0 (глобально адміністрована MAC-адреса пристрою) – глобально унікальна (адмініструється IEEE) і звичайно «зашиита» в апаратуру;

– 1 (локально адміністрована MAC-адреса) вибирається довільно й може не містити інформації про виробника цього устаткування (OUI).

Деякі виробники мережних адаптерів підтримують можливість змінювати MAC-адреси пристрою.

Мережні адреси (IPv4)

IP-адреса або **адреса третього рівня** – це логічна адреса, яка не прив'язується до конкретної апаратури (мережної карти, інтерфейсу тощо) і призначається адміністратором мережі: протокол IP версії 4 (IPv4) використовує 32-бітні адреси; класова адресація IPv4.

Розмір IPv4-адреси був обраний довжиною в 32 біта (при цьому можна адресувати $2^{32} \approx 4,3$ млрд пристроїв).

Хронологічно першим методом поділу IP-адрес є так звана **класова** модель IP-адресації, яка частково вирішила проблему нераціонального використання адресного простору. Згідно з цією моделлю, увесь простір IP-адрес ділиться на **5 класів** залежно від визначення перших чотирьох біт IPv4-адреси. Класам привласнені імена від А до Е (табл. 6.1).

Перші 3 класи А, В и С використовуються для індивідуальної (unicast) адресації мереж і вузлів, клас D – для багатоадресного або групового (multicast) розсилання, клас Е зарезервований для експериментів. Класи А, В и С мають різну довжину мережної частини адреси.

Для мереж класу А (табл. 6.1) під ідентифікатор мережі приділяється 1 байт (перший октет), 3 байти (3 октети), що залишилися використовуються для ідентифікатора вузла, причому старший (лівий) біт ідентифікатора мережі завжди рівний 0.

Таблиця 6.1 – Класи IP-адрес

Клас	Перші біти	Найменший номер мережі	Найбільший номер мережі	Максимальна кількість вузлів у мережі
А	0	1.0.0.0	126.0.0.0	2^{24} (поле 3 байти)
В	10	128.0.0.0	191.255.0.0	2^{16} (поле 2 байти)
С	110	192.0.0.0	223.255.255.0	2^8 (поле 1 байт)
D	1110	224.0.0.0	239.255.255.255	Групові адреси
E	11110	240.0.0.0	247.255.255.255	Зарезервований

Оскільки **перший біт** ідентифікатора мережі завжди дорівнює нулю, то 7 біт, що залишилися дозволяють адресувати 128 (2^7) різних мереж. Однак через те, що адреси 0.0.0.0 і 127.0.0.0 є спеціальними IPv4-адресами, кількість доступних **мереж** класу А рівно 126 ($2^7 - 2$). У кожній мережі класу А можна адресувати до 16 777 214 ($2^{24} - 2$) вузлів. Дві адреси віднімаються внаслідок того, що вони використовуються у спеціальних цілях і не можуть бути призначені пристрою (перший – адреса мережі, останній – ширококомовна адреса).

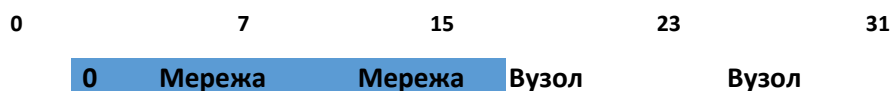


Рисунок 6.2 – Формат IPv 4-адреси класу А [3]

Мережі класу В (рис. 6.3) визначаються значеннями 10 у двох старших бітах адреси. Перші два байти в адресі використовуються для ідентифікатора мережі, два байти, що залишилися – для ідентифікатора вузла. У результаті

кількість доступних мереж класу В становить 16 384 (2^{14}) і з кількістю вузлів у кожній мережі рівним 65 534 ($2^{16} - 2$).

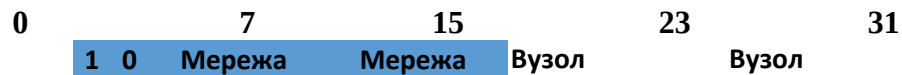


Рисунок 6.3 – Формат IPv 4-адреси класу В [3]

Для мереж класу С (рис. 6.4) під ідентифікатор мережі приділяється три байти в той час, як під ідентифікатор вузла тільки один байт. Три старші біти першого октету **завжди рівні 110**, дозволяючи визначити, що адреса ставиться саме до класу С. Таким чином, одержуємо 2 097 152 (2^{21}) мереж, у кожній з яких перебуває 254 ($2^8 - 2$) вузла.

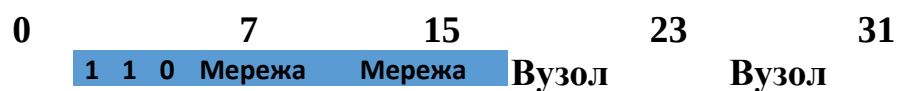


Рисунок 6.4 – Формат IPv 4-адреси класу С [3]

Мережі класу D (рис. 6.5) визначаються значеннями 1 110 у перших чотирьох бітах адреси, інші біти використовуються для адресації багатоадресної групи. Адресний простір класу D зарезервований для групового розсилання й використовується для адресації групи вузлів. Ідентифікаторів мереж і вузлів у IPv4-адресі класу D не виділяють.



Рисунок 6.4 – Формат IPv4-адреси класу D [3]

Мережі класу E (рис. 6.5) є експериментальними й у цей час не використовуються. Адреси в цьому класі визначаються значеннями 1 111 у перших чотирьох бітах.



Рисунок 6.5 – Формат IPv 4-адреси класу E [3]

Застосування IP-адресування на основі класового розподілу обмежує можливості раціонального використання всього переліку (пулу) адрес у кожній мережі, тому в подальшому стали використовувати **два** додаткові варіанти IPv4: на основі підмереж та основі безкласового адресування.

Із появою трирівневої ієрархії IPv4-адрес потрібні були додаткові методи, які дозволяли б визначити, яка частина адреси вказує на

ідентифікатор підмережі, а яка – на **ідентифікатор вузла**. Було запропоновано використовувати бітову маску (bit mask), яка відокремлювала б частину адресного простору ідентифікаторів вузлів від адресного простору ідентифікаторів підмережі. Така бітова маска називається **маскою підмережі** (subnet mask).

Маска підмережі (рис. 6.6) – це 32-бітне число, двійковий запис якого містить одиниці в тих розрядах, які повинні визначатися як ідентифікатор мережі. Оскільки ідентифікатор мережі є цільною частиною IPv4-адреси, послідовність одиниць у масці підмережі повинна бути також безперервною.



Рисунок 6.6 – Формування маски підмережі [3]

Щоб одержати адресу мережі, знаючи IPv4-адресу й маску підмережі, необхідно застосувати до них **операцію логічне «І»** (рис. 6.7). Інакше кажучи, у тих позиціях IPv4-адреси, в яких у масці підмережі є двійкові 1, перебуває ідентифікатор **мережі**, а де двійкові 0 – ідентифікатор **вузла**.

Для мереж класу А, В та С визначені **фіксовані** маски підмережі, які жорстко визначають кількість можливих IPv4-адрес. Технологія поділу мережі дає можливість створювати більшу кількість мереж із меншою кількістю вузлів у них, що дозволяє ефективно використовувати адресний простір. Для обчислення кількості підмереж використовується формула 2^s , де s – кількість біт, зайнятих під ідентифікатор мережі з частини, відведеної під ідентифікатор вузла.

Кількість вузлів у кожній підмережі обчислюється по формулі $2^n - 2$, де n – кількість біт, які **залишилися** в частини, що ідентифікує вузол, а дві адреси – адреса підмережі й широкомовна адреса – у кожній отриманій підмережі зарезервовані.

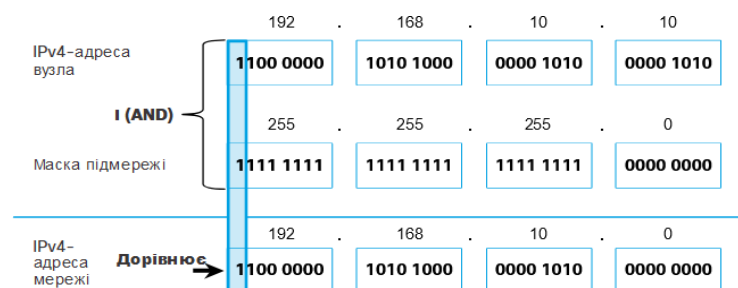


Рисунок 6.7 – Одержання адреси мережі з IP-адреси й маски підмережі [3]

Адресування на основі підмереж

Для більш ефективного використання адресного простору були внесені зміни в існуючу класову систему адресації. У RFC 950 була описана процедура розбивки мереж на підмережі, і в структуру IPv4-адреси був доданий ще один рівень ієрархії – *підмереж* (subnetwork).



Рисунок 6.8 – Структура IPv4-адреси

Поява ще одного рівня ієрархії (рис. 6.8) не змінила самої IPv4-адреси, вона залишилася 32-розрядною, а частина адреси, що раніше відводилася під ідентифікатор вузла, була розділена на дві частини – ідентифікатор підмережі й ідентифікатор вузла.



Рисунок 6.9 – Трирівнева ієрархія IP-адреси [3]

Розбивка однієї великої мережі на більш дрібні (рис. 6.9) дозволяє:

- раціонально використовувати адресний простір (тобто виділити для сегмента мережі блок адрес не цілком класу А, В або С, а тільки частину класової мережі);
- підвищити безпеку й керованість мережі (за рахунок зменшення розмірів сегментів та ізоляції трафіка сегментів один одного).

Приклад 1. Організації необхідно розбити мережу 192.168.1.0 на 20 підмереж по 6 комп'ютерів у кожній. Для початку необхідно визначити, до якого класу належить адреса. 192.168.1.0 – це клас С, відповідно, стандартна маска підмережі для класу С дорівнює 255.255.255.0 і під ідентифікатор вузла відведений четвертий октет. Потім визначається кількість біт четвертого октету, потрібних для формування 20-ти підмереж. Оскільки знайти число, за якого ступінь 2 буде дорівнювати 20 неможливо, вибираємо найближче більше число $2^5 = 32$. Таким чином, 5 перших біт 4-го октету будуть використані для ідентифікації підмережі, а 3 біта, що залишилися – для ідентифікації вузлів у них (рис. 6.10).

-



Рисунок 6.10 – Розбивка мережі на підмережі [3]

Щоб уникнути проблем з адресацією й маршрутизацією всі мережні пристрої TCP/IP в одному сегменті мережі повинні використовувати ту саму маску підмережі.

11000000	10101000	00000001	00000000	Маска підмережі 255.255.255.0
11111111	11111111	11111111	00000000	Підмережа 1
↓				
11000000	10101000	00000001	00000000	Маска підмережі 255.255.255.248
11111111	11111111	11111111	11111000	Підмережа 2
↓				
11000000	10101000	00000001	00001000	Маска підмережі 255.255.255.248
11111111	11111111	11111111	11111000	Підмережа 3
↓				
11000000	10101000	00000001	00010000	Маска підмережі 255.255.255.248
11111111	11111111	11111111	11111000	Підмережа 4
↓				
11000000	10101000	00000001	00011000	Маска підмережі 255.255.255.248
11111111	11111111	11111111	11111000	...
...				
11000000	10101000	00000001	10011000	Підмережа 20
11111111	11111111	11111111	11111000	192.168.1.152
				Маска підмережі 255.255.255.248

Рисунок 6.11 – Приклад розбивки мережі 192.168.1.0 на підмережі [3]

Безкласове адресування

Класова модель IPv4-адресації виявилася нераціональною з погляду ефективного використання адресного простору. Навіть під час використання підмереж, у випадку класової адресації мережі можна було розбити тільки на підмережі **однакового** розміру.

Наприклад, велика кількість вузлів є надлишковою для підмережі, яка зв'язує два маршрутизатори по каналу «точка – точка». У цьому випадку необхідно всього дві IPv4-адреси для адресації інтерфейсів сусідніх маршрутизаторів. Таким чином, розбивка мережі на підмережі різного розміру дозволила б раціонально використовувати адресний простір.

Поступово зі зростанням мережі «Інтернет» відбулася відмова від класової схеми, і була обрана *безкласова модель IPv4-адресації*, у якій **немає прив'язки** до класу мережі й маски підмережі за замовчуванням. Безкласова адресація використовує *маски підмережі змінної довжини* (Variable Length Subnet Mask, VLSM) і *технологію безкласової міждоменої маршрутизації* (Classless Inter Domain Routing, CIDR). Термін «маска змінної довжини» означає, що мережа може бути розбита на підмережі з різними масками підмережі. Основна ідея застосування VLSM полягає в тому, що можна розбити мережу на підмережі, потім підмережі розбити ще на підмережі точно в такий же спосіб, як була розбита первісна мережа. Тобто мережа може бути розбита на підмережі різних розмірів, з різними масками. Маски підмережі є основою методу безкласової маршрутизації й записуються у вигляді нотації «IP-адреса/довжина префіксу». Число після знака «/» означає кількість одиничних розрядів у масці підмережі. Наприклад, мережна адреса 192.168.1.8 з маскою підмережі 255.255.255.248 також може бути записана, як 192.168.1.8/29. Число 29 вказує, що у масці підмережі 255.255.255.248 є 29 одиничних біт.

Розподіл мережі на підмережі з використанням масок змінної довжини аналогічно традиційному розподілу на підмережі.

Приклад 2. Розбиття мереж на підмережі (6.12).

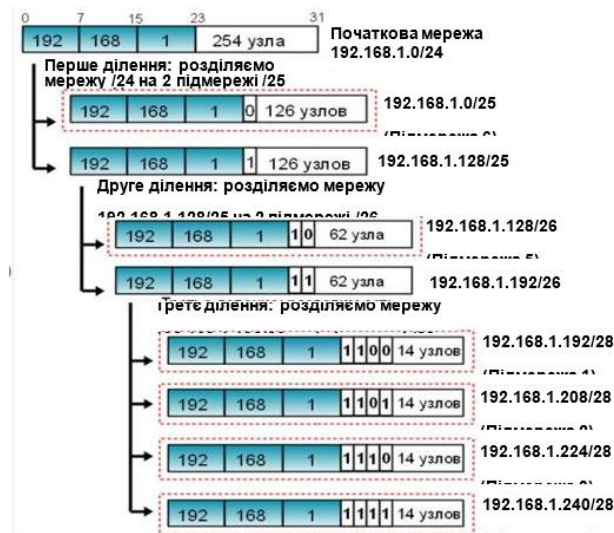


Рисунок 6.12 – Приклад розбивки мережі 192.168.1.0/24 на підмережі [3]

Приклад 3. Розділити мережу класу С 192.168.1.0/24 її на 6 підмереж. У підмережах 1, 2, 3 і 4 повинно бути 10 вузлів, а 5-й підмережі – 50 вузлів, в 6-й підмережі – 100. Теоретично для мережі 192.168.1.0/24 припустима кількість вузлів дорівнює 254.

Спочатку необхідно розділити мережу 192.168.1.0/24 на дві підмережі. Для цього з 4-го октету необхідно зайняти 1 біт для ідентифікатора підмережі, таким чином, для ідентифікації вузлів залишиться 7 біт. У підсумку виходить дві підмережі 192.168.1.0/25 і 192.168.1.128/25, у кожній з яких може бути по 126 ($2^7 - 2$) вузлів.

Першу з них залишимо, тому що потрібно, щоб у 6-й підмережі було 100 вузлів, а другу розділимо ще на дві підмережі. Для цього заберемо 1 біт з тих 7-ми біт відведених під ідентифікатор вузла, що залишилися. Таким чином, виходить дві підмережі 192.168.1.128/26 і 192.168.1.192/26, у кожній з яких **припустима кількість** вузлів рівно 62 ($2^6 - 2$). Першу підмережу необхідно залишити для 5-ої підмережі, у якій повинно бути 50 вузлів, а із другої підмережі сформувані ще чотири підмережі. Для цього займемо ще 2 біти з 6-ти біт, відведених під ідентифікатор вузла, що залишилися. У результаті одержимо **чотири** підмережі з 14 ($2^4 - 2$) вузлами в кожній, що дозволить адресувати необхідну кількість вузлів, необхідних для підмереж 1, 2, 3 і 4.

Приклад 4. Необхідно розробити схему IP-адресації для міської комп'ютерної мережі, яка об'єднує три відділення фірми:

- відділення № 1 знаходиться в одноповерховому будинку;
- відділення № 2 – у триповерховому;
- відділення № 3 – у десятиповерховому будинку.

Кожне відділення – це окрема підмережа. Для адресації використовувати мережу 192.168.0.0/16 та змінну довжину маски підмережі. Обов'язковою умовою є використання мінімальної можливої довжини хостової частини.

Вхідні дані:

Варіант № 1	
№ кімнати	Кількість робочих місць
1	7
2	6
3	9
4	5
5	5
6	2
7	1

Кількість підмереж

Відділення № 1: 1, 2 кімнати ($7 + 6 = 13$ робочих місць).

Відділення № 2: 3, 4 кімнати ($9 + 5 = 14$ робочих місць).

Відділення № 3: 5, 6, 7 кімнати ($5 + 2 + 1 = 8$ робочих місць).

Підмережа № 1 (13 робочих місць):

- IP-адреса підмережі: 192.168.0.0;
- маска підмережі: 4 біти, 255.255.255.240;
- IP-адреса першого вузла: 192.168.0.1;
- IP-адреса останнього вузла: 192.168.0.14;
- Broadcast: 192.168.0.15.

Максимальна кількість вузлів у мережі: $2^{(32 - 28)} - 2 = 16 - 2 = 14$.

Підмережа № 2 (14 робочих місць):

- IP-адреса підмережі: 192.168.0.16/28;
- Маска підмережі: 4 біти, 255.255.255.240;
- IP-адреса першого вузла: 192.168.0.17;
- IP-адреса останнього вузла: 192.168.0.30;
- Broadcast: 192.168.0.31.

Максимальна кількість вузлів у мережі: $2^{(32 - 28)} - 2 = 16 - 2 = 14$.

Підмережа № 3 (8 робочих місць):

- IP-адреса підмережі: 192.168.0.32/29;
- Маска підмережі: 3 біти, 255.255.255.248;
- IP-адреса першого вузла: 192.168.0.33;
- IP-адреса останнього вузла: 192.168.0.38;
- Broadcast: 192.168.0.39.

Максимальна кількість вузлів у мережі: $2^{(32 - 29)} - 2 = 8 - 2 = 6$.

Таблиця 6.2 – Результати розрахунків

Номер підмережі	IP-адреса підмереж	Маска підмереж	IP-адреса першого вузла	IP-адреса останнього вузла	Broadcast	Максимальна кількість вузлів у мережі
1	192.168.0.0	255.255.255.240	192.168.0.1	192.168.0.14	192.168.0.15	14
2	192.168.0.16/28	255.255.255.240	192.168.0.17	192.168.0.30	192.168.0.31	14
3	192.168.0.32/29	255.255.255.248	192.168.0.33	192.168.0.38	192.168.0.39	6

Контрольні запитання

1. Дайте визначення MAC-адреси, її призначення і характеристику.
2. Що таке IP-адреса? Опишіть її структуру.
3. Що таке маска підмережі? Яке її призначення?
4. Як визначається адреса мережі, кількість підмереж, кількість вузлів у кожній підмережі?

Практична робота № 7 Передача даних у мережі

Мета: ознайомлення з методами передачі даних у мережі, мережевою архітектурою, мережевими протоколами, мережевими ОС.

Завдання:

1. Ознайомитись зі структурою мережі, яка функціонує в навчальному закладі (інституті):

- тип мережі (однорангова, з виділеним сервером);
- кількість та функціональне призначення серверів мережі;
- вказати тип мережевої операційної системи (сервера, робочої станції);
- визначити топологію комп'ютерної мережі;
- визначити тип використовуваного обладнання (тип кабелю, мережевий адаптер).

2. Вказати переваги та недоліки мережі з виділеним сервером та однорангової мережі:

Переваги	Недоліки

2. Підключитися до локальної мережі за допомогою мережевого адаптера.

Загальні відомості

Передавання даних у комп'ютерній мережі може здійснюватися різними методами.

Методи конкурентного доступу

Конкурентний доступ передбачає передавання всіма вузлами мережі, які мають дані для передавання, тоді, коли ці дані треба передати. При цьому нерідко трапляється, що кілька вузлів починають передавання одночасно. Така ситуація має назву **колізії**. У цьому випадку всі передачі примусово припиняються і відновлюються через випадковий проміжок часу.

Вперше такий підхід використано у мережі університету штату Гаваї (США). Система отримала тоді назву ALOHA. Реальна пропускна здатність такої мережі складала 19 % від теоретичної. Перевагою цього підходу є простота його реалізації. Недоліком – недовикористання можливостей каналів зв'язку.

Маркерні методи доступу

Суть цього методу доступу полягає в тому, що в мережі постійно циркулює спеціальний службовий сигнал (імпульс) – **маркер**. Станція, яка в цей момент має маркер, отримує право на передавання.

Цей метод реалізується по-різному у шинних і кільцевих мережах, тому про них і говорять у множині.

Маркерний доступ у шинній мережі. Якщо кадр, який потрапляє на станцію, є маркерним, то у випадку наявності даних для передавання станція передає спочатку ці дані, а потім (після закінчення передавання даних) передає

маркер. Якщо даних для передавання немає, то станція передає лише маркер. Маркер повинен обов'язково містити адресу приймача.

Маркерний доступ у кільцевій мережі. Така мережа називається мережею з ретрансляцією і передаванням маркера. По кільцю постійно передаються вільні (пусті) або доповнені даними маркери. У кільцевій мережі маркер не має поля адреси, оскільки маркери передаються по кільцю по чергові всім станціям.

Недоліки та переваги цих мереж ті ж, що у мереж із доступом методом опитування.

Метод уставляння регістра

В університеті штату Огайо (США) для доступу до середовища передавання була запропонована мережа з уставлянням регістра. Особливістю таких мереж була нова конструкція передавача станцій.

Структурну схему передавача наведено на рисунку 7.1.

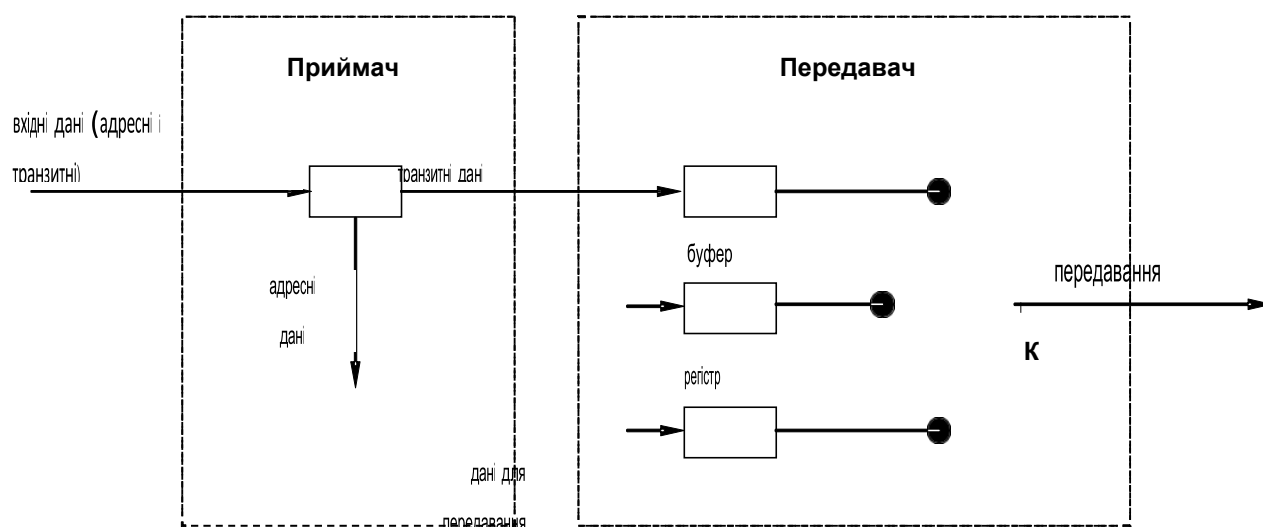


Рисунок 7.1 – Структурна схема передавача для використання методу уставляння регістра

Ця система працює у такий спосіб:

1. Перемикач К на виході передавача по чергові з'єднується з кожним із трьох виходів: прямим для транзитних даних, буфером транзитних даних і регістром передавача, де знаходяться дані для передавання з цього комп'ютера.
2. Якщо комп'ютер не має даних для передавання, то передаються транзитні дані (за умови її наявності).
3. Якщо комп'ютер має дані для передавання, то з регістра передається кадр даних разом із регістром, а транзитні дані (черговий кадр) тим часом зберігаються в буфері транзитних даних.
4. У наступних тактах дані з комп'ютера передаються з регістра разом із регістром, а транзитні з буфера транзитних даних. Передавання з прямого виходу для транзитних даних немає.
5. Так триватиме доти, поки не будуть передані всі дані з комп'ютера.

6. Після закінчення передавання даних із комп'ютера відновлюється звичайне передавання транзитних даних.

Таким чином, працюють передавачі всіх комп'ютерів у мережі та всі дані, що в ній передаються, мають поле регістру, де вказується відправник даних.

Перевагою такого методу доступу є брак будь-якого зайвого службового трафіку, що дозволяє використовувати канали зв'язку максимально ефективно, адже в такій мережі передаються лише корисні сигнали.

Метод доступу із запитом пріоритету

Метод доступу із запитом пріоритету використовується в мережах, де одночасно можуть передаватись як файли, так і потоки.

Файл має фіксований розмір і його передавання у будь-якому випадку або закінчується успішно, або переривається. В останньому випадку передавання потрібно повторювати.

Потоки (аудіо чи відео) розмірів не мають перерви в їх передаванні та впливають лише на сприйняття абонентом (якість трансляції), але не спричиняють повторення передавання. Потоки, як і файли, передаються окремими фреймами.

Пріоритет даних важливий в обох випадках для поєднання передавання файлів із передаванням потоків одними й тими ж лініями зв'язку за умови збереження якості передавання як файлів, так і потоків.

Зазвичай такі проблеми виникають у деревоподібних мережах (як локальних, так і глобальних) (рис. 7.2).

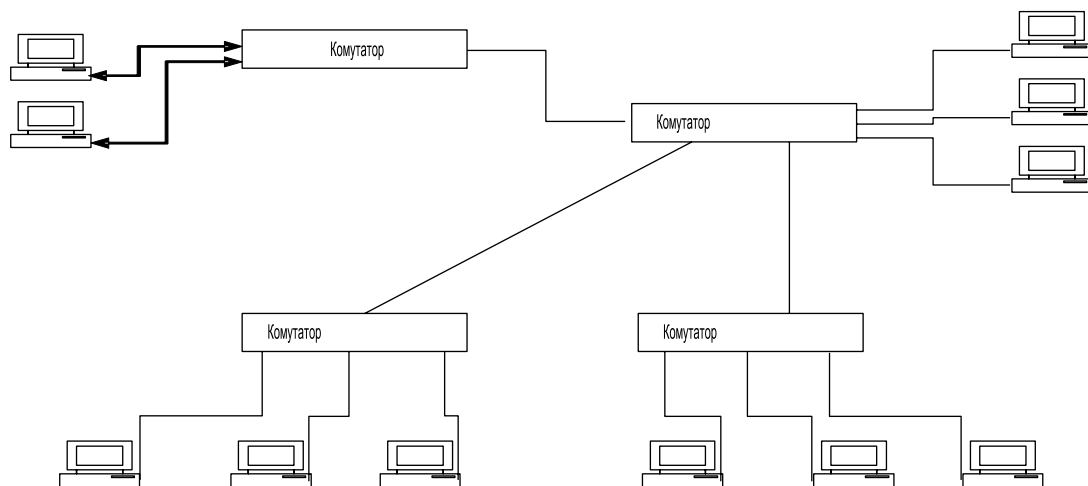


Рисунок 7.2 – Схема мережі деревоподібної топології

Визначено два пріоритети: нормальний і високий.

Нормальний пріоритет призначений для передавання файлів, високий – для передавання потоків.

Усі дані містять інформацію про пріоритет. Відповідно до цього відбувається обмін даними. Практично це означає, що під час приймання потоку приймання файлів або неможливе зовсім, або ускладнене. Тут усе

залежить від технічних характеристик апаратури, якості ліній зв'язку та якості програмного забезпечення.

Мережеві архітектури

Поєднання певного типу кабелю, мережевих адаптерів, мережевої топології та методу доступу до середовища передавання називають **архітектурним вирішенням** комп'ютерної мережі або **архітектурою мережі**. Архітектура описує конкретний стандарт мережі.

Мережеві архітектури також називають протоколами низького рівня. Розглянемо основні мережеві архітектури.

Ethernet/IEEE 802.3. Це протоколи взаємодії на фізичному і каналному рівнях. При цьому використовується конкурентний доступ до передавального середовища. В архітектурі IEEE 802.3 передбачається використання 3-х типів коаксіального кабелю і 2-х типів витой пари та різних топологій залежно від фізичного рівня.

В архітектурі **Ethernet** використовується коаксіальний кабель та шинна топологія. **Шина** – це канал передавання даних, окремі частини якого називаються **сегментами**. Один сегмент мережі може містити до 100-а робочих станцій, кілька таких сегментів можна з'єднати повторювачами (repeater). В одній мережі Ethernet можуть працювати до 1 024 робочих станцій. У разі такого методу організації доступу передавання даних починається тільки тоді, коли є вільний канал; в інших випадках передавання затримується. Швидкодія такої мережі знижується під час роботи більш як 100-а комп'ютерів. Швидкість передавання даних у мережах з архітектурою Ethernet – 10 Мбіт/с. Використання сучасних типів витой пари, кабельних центрів та мережевих адаптерів дозволяє будувати швидкий Ethernet – **FastEthernet** із максимальною пропускною здатністю 100 Мбіт/с. Останнім часом широко використовується технологія **Gigabit Ethernet** (швидкість 1 Гбіт/с). Раніше вона ґрунтувалася на волоконно-оптичному середовищі передавання даних, а зараз все частіше використовують спеціальну виту пару.

Arcnet. Використовується в топології «зірка». Повідомлення до комп'ютерів передаються за допомогою **маркерів**. Під час відправлення повідомлень до них приєднуються маркери, і дані посилаються в мережу; під час отримання повідомлень аналізується заголовок маркера, і в разі його відповідності адресі комп'ютера повідомлення відокремлюються та залишаються на комп'ютері.

Token Ring/IEEE 802.5. Архітектури Token Ring та IEEE 802.5 істотно не відрізняються. У них застосовується кільцева топологія з передаванням маркера. Схожа з архітектурою Arcnet, але тут можна встановити пріоритет отримання маркера окремим комп'ютером.

FDDI (Fiber Distributed Data Interface) та CDDI. FDDI – волоконно-оптичний інтерфейс розподілу даних. Функціонує за схемою передавання маркера в логічному кільці з волоконно-оптичними кабелями (метод кільцевих слотів). Швидкість передавання – 100 Мбіт/с. Топологія – подвійне кільце (можлива мішана). Максимальна кількість станцій в мережі – 1 000, довжина

кільця – до 100 км. *CDDI* – електрична реалізація архітектури *FDDI* на витій парі. Довжина сегмента – 100 м.

LocalTalk. Це **протокол фізичного і каналного рівнів**, що належить Apple Computer. Він сконструйований для маленьких робочих груп і вбудований у кожний комп'ютер Apple Macintosh. У LocalTalk використовується шинна топологія на екранованій витій парі. Метод доступу – конкурентний. На відміну від інших протоколів цікавим є динамічне призначення мережевих адрес. Швидкість передавання даних – 230 Кбіт/с.

Поєднання певного типу кабелю, мережевих адаптерів, мережевої топології та методу доступу до середовища передавання називають **архітектурним вирішенням** комп'ютерної мережі або **архітектурою мережі**. Архітектура описує конкретний стандарт мережі.

В **архітектурі Ethernet** використовується коаксіальний кабель та шинна топологія. **Шина** – це канал передавання даних, окремі частини якого називаються *сегментами*.

Мережеві протоколи

Для забезпечення необхідної сумісності на кожному із семи можливих рівнів архітектури комп'ютерної мережі діють спеціальні стандарти, які називаються **протоколами**.

Протокол – це сукупність правил синхронного функціонування об'єктів однойменного рівня під час обміну даними між станцією-відправником і станцією-одержувачем; формати блоків даних, що передаються; контроль помилок, методи кодування даних тощо.

У протоколах описуються характер апаратних взаємозв'язків компонентів мережі (**апаратні протоколи** або **протоколи низького рівня**) і характер використання програм і даних (**програмні протоколи** або **протоколи верхнього рівня**). Фізично підтримка забезпечується через апаратні пристрої (**інтерфейси**) і програмні засоби (**програми підтримки протоколів**). Програми, за якими здійснюється підтримка протоколів, також називаються **протоколами**.

Інтерфейс – це опис процедури взаємозв'язків, об'єктів, суміжних рівнів (наприклад, системи і середовища) для управління фізичним передаванням даних, а також формати даних, що передаються.

Наприклад, якщо два комп'ютери з'єднані між собою прямим з'єднанням, то на нижчому (фізичному) рівні протокол їхнього зв'язку визначають конкретні пристрої фізичного порту (паралельного чи послідовного) і механічні компоненти (роз'єм, кабель тощо). На більш високому рівні зв'язки між комп'ютерами визначаються через програмні засоби, за допомогою яких здійснюється управління передаванням даних через порти. Для стандартних портів такі програми знаходяться в базовій системі «введення – виведення» (BIOS). На найвищому рівні протокол взаємозв'язків забезпечується застосування операційної системи. Наприклад, для Windows це стандартна програма **Пряме кабельне з'єднання**

У комп'ютерах локальної мережі зазвичай використовується єдиний комплект протоколів для всіх комп'ютерів. У глобальній мережі можуть з'єднуватися як окремі комп'ютери, так і окремі локальні мережі, зокрема і з різними протоколами.

Розглянемо найпоширеніші *протоколи верхнього рівня* (програмні протоколи).

Apple Talk. Це сімейство мережевих протоколів Apple Computer. В ньому підтримуються архітектури Ethernet, Token Ring та LocalTalk. «Серцем» Apple Talk є AppleTalk Filing Protocol (AFP – файловий протокол AppleTalk). За допомогою AFP можна легко виконувати операції файлової системи, пропонується система безпеки. Apple підтримує різноманітні протоколи і служби рівня прикладних програм. Найчастіше використовують три програми – AppleShare File Server, AppleShare Print Server та AppleShare PC. У першій із них використовуються служби AFP для доступу до віддалених файлів. За її допомогою користувачам надається можливість зберігання файлів на мережевому сервері та отримання до них доступу. За допомогою AppleShare Print Server надсилаються дані на принтери, що знаходяться в мережі AppleTalk. За допомогою AppleShare PC можна отримати доступ до файлів, що знаходяться в файловій системі AppleTalk; надається також можливість MS-DOS-комп'ютерам друкувати матеріали на принтерах, сумісних із AppleTalk.

NetWare. Це група мережевих протоколів і відповідних сервісних служб, створених фірмою Novell. У NetWare описуються специфікації для рівнів 3–7 моделі OSI (мережевий, транспортний, рівень подання даних, прикладний), використовується з багатьма середовищами з різними типами доступу. Протоколи NetWare для рівнів 3 і 4 називають Internetwork Packet exchange (IPX – міжмережевий пакетний обмін), Sequenced Packet eXchange (SPX – упорядкований обмін пакетами).

TCP/IP. TCP/IP – сукупність двох протоколів, що регулюють передавання даних між комп'ютерами в мережі «Інтернет». Свою назву протокол TCP/IP дістав від двох протоколів: Transmission Control Protocol (TCP) – протокол транспортного рівня, що визначає, як відбувається передавання даних, та Internet Protocol (IP) – адресний протокол мережевого рівня, що визначає, куди передаються дані. Комплекс протоколів TCP/IP містить 4 верхні рівні моделі OSI. Протоколи базуються на IP-протоколах негарантованого доставляння пакетів без встановлення з'єднання. Згідно з *протоколом TCP* дані, що відправляються, «нарізуються» на невеликі пакети, після чого кожен пакет маркується так, щоб у ньому були адреси одержувача та відправника, а також інші дані, необхідні для правильного збирання документа на комп'ютері одержувача.

У *протоколі IP* передбачається, що в кожного користувача має бути своя унікальна адреса (*IP-адреса*). Вона подається чотирма байтами, наприклад: 104.17.126.10. Структура IP-адреси така, що кожен комп'ютер, через який проходить пакет, може за цими чотирма числами визначити маршрут відправлення пакетів у цей момент з урахуванням умов зв'язку і пропускних характеристик лінії. **Пакети можуть передаватися незалежно один від**

одного різними лініями зв'язку. Незважаючи на те, що в мережі «Інтернет» використовуються й інші протоколи, її часто називають *TCP/IP-мережею*, оскільки ці два протоколи, безумовно, є найважливішими.

Мережева операційна система – це пакет програм, за допомогою якого забезпечується функціонування мережі та управління нею, що дає змогу клієнтам користуватись мережевими сервісами.

Основним призначенням мережевих ОС є:

- забезпечення спільного використання та розподілу ресурсів мережі;
- адміністрування мережі;
- передавання повідомлень між вузлами мережі;
- надійного зберігання даних;
- вирішення завдань, пов'язаних із функціонуванням системи.

Важливою функцією мережевої ОС є забезпечення системи **захисту** – конфіденційності даних, розмежування прав доступу до ресурсів, парольний захист, виявлення спроб несанкціонованого доступу, відслідковування дій користувача, ведення журналів системних подій тощо.

Через мережеву ОС забезпечується підтримка функціонування різноманітних периферійних пристроїв, мережевих адаптерів, протоколів та можливість їхнього конфігурування.

У *мережах із централізованим управлінням (із виділенням сервером)* мережева ОС є головною (або єдиною) системою, за допомогою якої здійснюється управління ресурсами мережі. Такі системи мають високу продуктивність та функціональні характеристики, в них використовуються власні дискові та файлові системи, що оптимізовані для мережевого використання. Клієнтська частина мережевої ОС реалізується у вигляді спеціальної програми (оболонки), за допомогою якої обслуговуються мережеві запити. Ця програма функціонує під управлінням вихідної ОС робочої станції, або ж є органічною частиною ОС.

В *однорангових мережах* кожен вузол мережі може одночасно використовуватися в ролі сервера та клієнта. **Однорангові мережі (peer-to-peer)** – це такі мережі, в яких окремі елементи (вузли) обмінюються між собою файлами і зберігають один і той же набір даних.

Під'єднання до локальної мережі за допомогою мережевого адаптера

Під час інсталяції Windows за допомогою операційної системи автоматично відшукується мережевий адаптер і здійснюється під'єднання до локальної мережі. Як і засоби забезпечення під'єднання інших типів, цей знаходиться в папці **Панель керування / Усі елементи панелі керування / Центр мережних підключень і спільного доступу**. За замовчуванням під'єднання до локальної мережі завжди наготові. Тільки під'єднання цього типу створюються автоматично. При цьому в папці **Центр мережних під'єднань і спільного доступу** з'являється ярлик під'єднання до мережі.

Щоб переглянути стан під'єднання до локальної мережі (чи наготові під'єднання, чи воно фізично не встановлене), потрібно викликати контекстне

меню для ярлика під'єднання до локальної мережі і вибрати пункт «Стан» (рис. 7.3).

Після виконання цих дій з'явиться допоміжне вікно стану. У цьому вікні видно стан, тривалість, швидкість під'єднання (рис. 7.4).

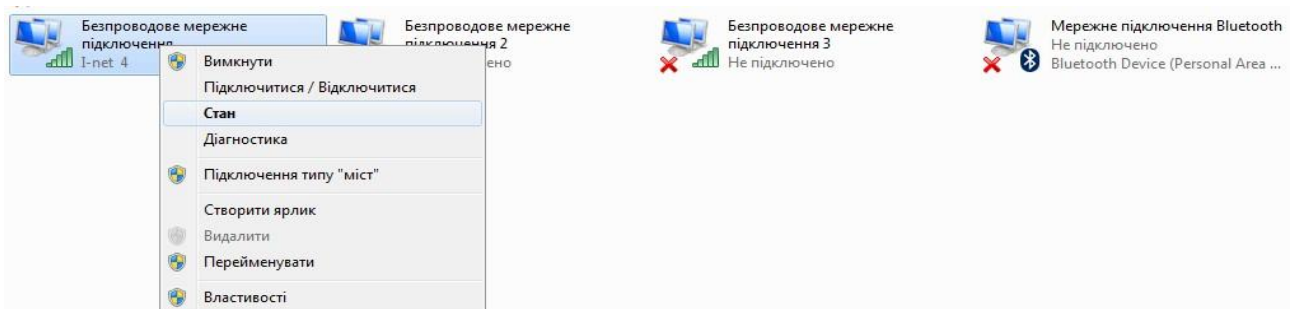


Рисунок 7.3 – Допоміжне вікно стану

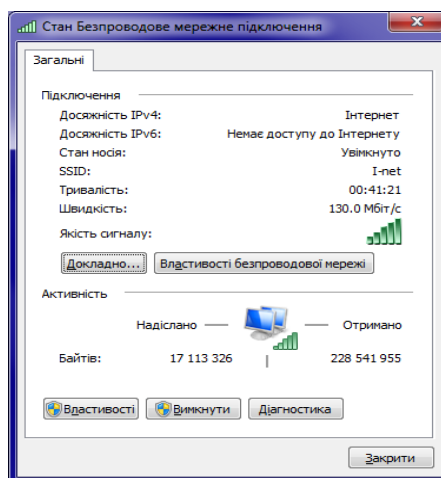
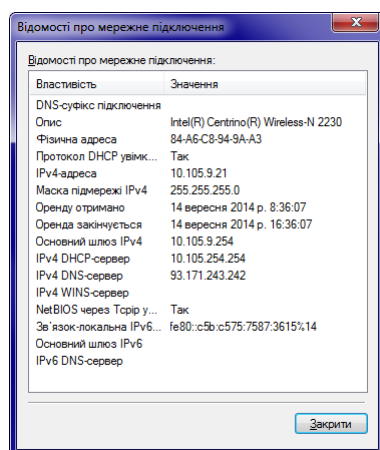
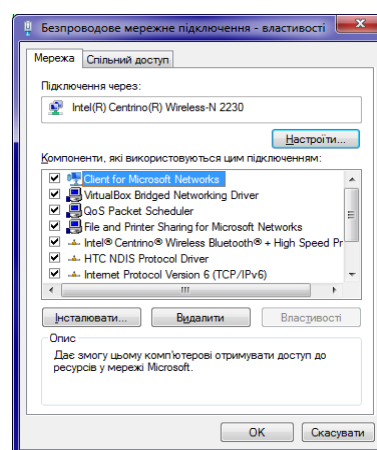


Рисунок 7.4 – Стан під'єднання до локальної мережі [6]



а)



б)

Рисунок 7.5 – Відомості про мережеве підключення: а – кнопка «Докладно»;

б – кнопка «Властивості»

Під час роботи з Windows автоматично створюється під'єднання до локальної мережі для кожного знайденого мережевого адаптера. Якщо на комп'ютері встановлено кілька мережевих адаптерів, то зручно для кожного під'єднання до локальної мережі призначити ім'я, за допомогою якого ідентифікується мережа, під'єднання до якої здійснено через адаптер.

Контрольні запитання

1. Опишіть методи конкурентного доступу передачі даних.
2. Дайте визначення протоколів низького рівня.
3. Опишіть протоколи верхнього рівня.
4. Що таке мережеві операційні системи?

Практична робота № 8 **Протокол FTP. Організація FTP-сервера на основі WIN7**

Мета: навчитися організовувати FTP-сервер та забезпечувати доступ до нього різним групам користувачів; виконувати обмін інформацією з використанням доступу за протоколом FTP .

Завдання: створити FTP сервер для обміну інформацією між користувачами .

Варіанти завдань

1. Служби доменів Active Directory (AD DS).
2. Служби Active Directory полегшеного доступу до каталогів (AD LDS).
3. Служби управління правами Active Directory (AD RMS).
4. Сервер додатків.
5. DHCP-сервер.
6. DNS-сервер.
7. Факс-сервер.
8. Файлові служби.
9. Hyper-V.
10. Служби мережевих політик і доступу.
11. Служби друку.
12. Служби віддалених робочих столів.
13. Служби терміналів.
14. Служби UDDI.
15. Вебсервер (IIS).
16. Служби Windows Server Update Services (WSUS).
17. Поштовий сервер.

Загальні відомості

Основою функціонування інтернету є робота кількох протоколів.

Протокол мережі – це набір правил, що дозволяє обмін даними між пристроями, це основа забезпечення доставки пакета, певні дії реалізації конкретного процесу.

MAC (Media Access Control). Протокол MAC – це протокол **низького рівня**, унікальний ідентифікатор пристрою. Його застосовують як ідентифікацію пристроїв у локальній мережі. Кожен пристрій, який підключено до інтернету, має свою унікальну MAC-адресу. Ця адреса задана виробником. Це протокол рівня з'єднання, з яким часто доводиться стикатися кожному користувачеві. **Це фізична адреса обладнання або інтерфейсу деяких мереж.**

Ця адреса видається безкоштовно на кожен пристрій. Основна частина мережеских протоколів використовують одне з трьох просторів MAC, під керуванням IEEE: MAC-48, EUI-48, EUI-64. Приклад написання MAC адреси: 00-50-B6-5B-CA-6A. Перші 6 знаків позначають виробника, а друга частина адреси – це серійний номер інтерфейсу.

Навіщо потрібна MAC-адреса? Цей протокол використовується для ідентифікації вузла мережі та відповідно доставки інформації саме у цей вузол.

IP (Internet Protocol) – проти MAC, розташовується до рівня вище. IP-адреси є унікальними для кожного пристрою і дають можливість комп'ютерам знаходити та визначати один одного в мережі. IP належить мережевому рівню моделі TCP/IP, це стандартний протокол Інтернету.

Мережі зв'язуються в складні структури, а за допомогою цього протоколу машини визначають можливі шляхи цільового пристрою (ці шляхи можуть змінюватися під час роботи). Протокол реалізується за допомогою відомих видів IPv4 та IPv6.

Pv4 (Internet Protocol version 4) – четверта версія мережевого протоколу IP. Використовує 4-байтну форму запису адрес пристроїв у комп'ютерній мережі.

IPv6 (Internet Protocol version 6) – нова версія IP-протоколу – IP-версії 6. Розробка протоколу IPv6 почалася 1992 року, а з 2003 року його підтримку забезпечують виробники більшості телекомунікаційного устаткування (корпоративного рівня). IPv6 – новий крок у розвитку Інтернету. Цей протокол розроблено з урахуванням вимог до Глобальної мережі, що постійно зростають. З лютого 2011 року IANA виділила останні п'ять блоків IP-адрес /8 (IPv4).

Найбільш суттєва різниця між IPv4 та IPv6 (табл. 8.1) полягає в тому, що раніше на інтернет-адресу виділяли 4 байти (32 біти), що відповідає стандартній на сьогодні чотириблоковій адресі IP, а протокол IPv6 виділяє на адресу 16 байтів (128 бітів). Це відповідає 340-ка секстильйонам адрес ($3,4 \times 10^{38}$) або по 5×10^{28} адрес на кожному людину.

IP-адреса може бути як динамічною (тимчасовою), так і статичною, що прикріплюється на комп'ютер у разі підключення до інтернету. Наприклад, на

VPS ви отримаєте статичну IP-адресу, а з використанням деяких VPN – динамічну IP-адресу.

Таблиця 8.1 – Порівняльна таблиця для IPv6 та IPv4

IPv4	IPv6
Розмір пакету: 576 байт	Розмір пакету: 1 280 байт
Безпека не передбачена	Включено аутентифікацію та шифрування
Використання в мережах: 99 %	Використання в мережах: 1 %
Доступний у всіх хостинг-провайдерів	Доступний не у всіх хостинг-провайдерів
Чотири десяткові числа розділені крапкою	Шестизначне число з розділенням двокрапкою

Поняття IP-адреси (виділеної адреси) **дуже часто використовується у сфері хостингу та оренди серверів**, адже кожен сервер має свою унікальну виділену IP-адресу або навіть свою підмережу адрес. Протоколи передачі даних для початківців, зокрема IP-адреси, можна використовувати на хостинг послугах для сайтів. На віртуальному хостингу також можна замовити виділену адресу для свого сайту, що надається багато переваг для роботи ресурсу в мережі та його подальшого просування.

ICMP (Internet control message protocol)

Цей протокол з'єднання призначений для того, щоб пристрої могли обмінюватись повідомленнями. Це можуть бути повідомлення про помилки чи інформаційні оповіщення. Дані цього протоколу не передають інформацію. Цей протокол перебуває рівнем вище, ніж протокол IP.

TCP (Transmission control protocol)

Один з основних протоколів мережі «Інтернет», що знаходиться на одному рівні з попереднім протоколом ICMP. Він керує передачею даних.

Бувають ситуації, коли пакети можуть приходити не в тому порядку або взагалі губитися. Але протокол TCP забезпечує правильний порядок доставки і дозволяє виправити помилки передачі пакетів. Інформація подається у правильному порядку для програми. З'єднання здійснюється за допомогою спеціального алгоритму, який передбачає надсилання запиту та підтвердження відкриття з'єднання двома комп'ютерами. Мінус роботи цього протоколу – низька швидкість, оскільки для надійної передачі необхідно більше часу.

Багато додатків використовують TCP, сюди належать SSH, WWW, FTP тощо. Зазвичай протокол TCP вбудований в ядро операційної системи.

Роль TCP протоколу (вебпротоколу). Забезпечення надійної доставки сегментів та його впорядкування, контроль швидкості передачі, робота із сесіями.

UDP (user datagram protocol)

UDP – відомий протокол, чимось схожий на TCP, який також функціонує на транспортному рівні. Основна відмінність – ненадійна передача даних: дані не проходять перевірку під час отримання. У деяких випадках цього цілком достатньо.

Протокол UDP. Принцип роботи. За рахунок відправки меншої кількості пакетів, UDP працює швидше ніж TCP. Немає потреби встановлювати з'єднання, і протокол використовується для відправлення пакетів відразу на кілька пристроїв або IP-телефонії. Це спрощений протокол за рахунок того, що деякі фрагменти під час передачі можуть бути втрачені.

HTTP (hypertext transfer protocol)

Протокол програми HTTP (hypertext transfer protocol) є основою роботи всіх сайтів у мережі. HTTP дає можливість запитувати необхідні ресурси віддаленої системи, наприклад, вебсторінки та файли. HTTPS – це окремий протокол мережі, а той самий HTTP, але зі спеціальним налаштуванням шифрування.

HTTP – порт 80, HTTPS – порт 443, останній використовується для конфіденційної передачі даних у мережі.

HTTPS забезпечується наявністю сертифіката захисту SSL, що підключається на боці хостинг-провайдера. SSL є безкоштовними, наприклад, від Lets Encrypt та платними від різних постачальників послуг, наприклад, Sectigo.

DNS (domain name system) використовується для перетворення зрозумілих адрес, що легко прочитуються, у складні IP-адреси, які важко запам'ятати і навпаки. За допомогою DNS ми отримуємо доступ до інтернет-ресурсу за його доменом.

Поняття DNS часто можна зустріти під час роботи з доменом. Наприклад, зі зміною хостингу або реєстратора домену відбувається процес оновлення DNS зони, який може тривати від 2 год до 72 год.

DNS сервер має **кілька типів записів**:

- 1) CNAME часто використовується для прив'язки субдомену, вказує альтернативний варіант основному домену;
- 2) NS – адреса DNS-сервера (наприклад, dns1.hyperhost.ua);
- 3) A-запис – IP-адреса сайту;
- 4) MX-запис – адреса поштового сервера;
- 5) TXT-запис – текстова інформація про домен, часто додається для проходження різноманітних верифікацій домену;
- 6) SPF – список серверів, які можуть надсилати пошту від конкретного домену;
- 7) SOA – запис з інформацією про сервер.

SSH (secure shell) також належить до протоколу рівня додатків. Він розроблений для забезпечення віддаленого керування системою захищеного каналу. Цей протокол використовується для багатьох додаткових технологій.

SSH – це насамперед безпечний протокол передачі, тому, якщо є можливість використовувати саме його, то не варто користуватися іншими протоколами, такими як ФТП.

Переважно ССШ використовують для **віддаленої роботи із серверами**, а також для надійного перенесення великої кількості даних. Наприклад, зі зміною хостинг-провайдера та перенесення сайту від одного хостера до іншого, для швидкої організації процесу вам знадобиться саме SSH доступ до старого хостингу.

SSH працює за допомогою спеціальних команд, які запускаються з командного рядка. Ця команда може трохи відрізнятися залежно від використаної ОС, а також SSH клієнта.

POP3 (Post Office Protocol) – це стандартний протокол, який використовується для отримання повідомлень електронної пошти. Протокол поштового з'єднання призначено для обробки запитів на отримання пошти від клієнтських поштових програм. POP3 призначений для отримання пошти з будь-якої точки доступу до Інтернету. POP3 – **найпопулярніший тип облікового запису електронної пошти**. POP3 дозволяє завантажувати листи на пристрої, а потім видаляти їх із сервера. Мінуси використання POP3: листи на сервері не зберігаються, зокрема надіслані. Подивитися надіслані можна лише з пристрою, з якого користувач надіслав листа. Усі налаштування зберігання листів налаштовуються окремо в поштовому клієнті користувача.

SMTP (Simple Mail Transfer Protocol) – протокол передачі пошти. Основне завдання сервера SMTP: повернення або підтвердження прийому, або оповіщення про помилку, або запит на додаткові дані. За допомогою SMTP протоколу перевіряється коректність системи, в якій надсилається вихідне повідомлення. За цим протоколом закріплено два порти: 25 та 587. Другий варіант протоколу із захищеним SSL, хоча 25 на практиці використовується набагато частіше. Використання особистого SMTP сервера дуже просто застосувати, в мережі дуже багато доступних інструкцій та фахівців по роботі з цим типом протоколу.

Особистий SMTP-сервер переважно використовують для організації масових розсилок, особистих листування або трансакційних листів.

FTP (file transfer protocol) FTP (file transfer protocol)

FTP (file transfer protocol) використовується для передачі даних. Функціонує на рівні додатків, що забезпечує передачу файлу від одного комп'ютера до іншого. FTP вважається небезпечним, його не варто застосовувати для передачі особистих даних. Тому, якщо є така можливість, то замість FTP необхідно використовувати SSH-протокол як основний тип протоколів передачі даних.

Для з'єднання FTP використовуються спеціальні програми, наприклад, Far Manager, Total Commander, FileZilla. Незважаючи на свою небезпечність FTP, протокол все одно є найпопулярнішим серед вебмайстрів. На послугах віртуального хостингу FTP надається за замовчуванням для з'єднання з серверами та передачі даних.

FTP – один із найстаріших прикладних протоколів, який використовується для доступу до віддалених хостів. Саме завдяки FTP відбувається організація обміну файлами.

Контрольні запитання

1. Особливості створення з'єднань за FTP-протоколом.
2. Послідовність кроків зі створення FTP-сервера.
3. Чи можуть на комп'ютері існувати два FTP-сервери?
4. Назвіть протоколи для забезпечення функціонування процесу передавання файлів в мережі.

Практична робота № 9

Інтеграція баз даних (на прикладі MySQL ↔ PostgreSQL)

Мета роботи: ознайомлення з методами інтеграції різномірних баз даних, навчитися створювати зв'язок між MySQL і PostgreSQL для передачі, синхронізації та аналітики даних за допомогою інструментів ETL.

Завдання:

1. Додати у процесі інтеграції фільтрацію даних (наприклад, переносити лише співробітників із зарплатою > 27 000).
2. Реалізувати логуювання (виведення кількості перенесених записів).
3. Розширити ETL-процес – додати автоматичне оновлення таблиці у PostgreSQL зі змінами у MySQL.
4. Побудувати візуалізацію у Power BI або Python (matplotlib, plotly) для порівняння зарплат у двох БД.

Загальні відомості

Інтеграція баз даних – це процес об'єднання даних із кількох джерел у єдину узгоджену систему для аналітики або управління [8].

Основні етапи інтеграції (ETL):

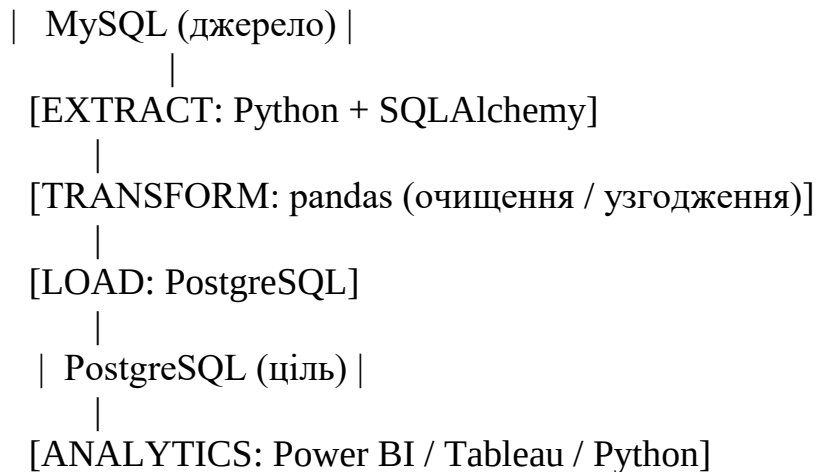
1. **Extract (витяг)** – отримання даних із різних джерел.
2. **Transform (перетворення)** – очищення, узгодження форматів, конвертація типів.

3. **Load (завантаження)** – запис результатів у цільову базу даних.

Для інтеграції MySQL ↔ PostgreSQL можна використовувати:

1. **Python-бібліотеки:** pandas, sqlalchemy, psycopg2, mysql.connector;
2. **ETL-платформи:** Apache Nifi, Talend, Pentaho Data Integration;
3. **Сервіси реплікації:** Debezium, Airbyte.

Архітектура процесу інтеграції. Схема процесу (ETL + аналітика):



Порядок виконання

1. Підготувати дві бази даних:
 - MySQL: база source_db, таблиця employees;
 - PostgreSQL: база target_db, таблиця employees_all.
2. Створити таблиці з однаковою структурою:
MySQL: CREATE DATABASE source_db;

```
USE source_db;
CREATE TABLE employees
(id INT PRIMARY KEY AUTO_INCREMENT,
 name VARCHAR(100),
 position VARCHAR(100),
 salary DECIMAL(10,2));

INSERT INTO employees (name, position, salary)
VALUES
('Іван Петренко', 'Інженер', 25000),
('Олена Коваль', 'Аналітик', 27000),
('Максим Левченко', 'Розробник', 30000);
```

PostgreSQL:
CREATE DATABASE target_db;

```
CREATE TABLE employees_all
(id SERIAL PRIMARY KEY,
 name VARCHAR(100),
 position VARCHAR(100),
 salary NUMERIC(10,2));
```

Реалізація інтеграції (на Python). Код інтеграції між MySQL → PostgreSQL:

```
– import pandas as pd;  
– from sqlalchemy import create_engine.
```

```
# 1. Підключення до MySQL  
mysql_engine  
create_engine(«mysql+mysqlconnector://root:password@localhost/source_db»).
```

```
# 2. Підключення до PostgreSQL  
pg_engine  
create_engine(«postgresql+psycopg2://postgres:password@localhost/target_db»).
```

```
# 3. Зчитування даних із MySQL  
df = pd.read_sql(«SELECT * FROM employees», mysql_engine).
```

```
# 4. (Опціонально) Трансформація даних  
df['salary'] = df['salary'] * 1.05 # наприклад, індексація зарплати.
```

```
# 5. Завантаження в PostgreSQL  
df.to_sql('employees_all', pg_engine, if_exists='append', index=False).
```

```
print(«Дані успішно перенесено з MySQL у PostgreSQL!»).
```

Перевірка результату

PostgreSQL:
SELECT * FROM employees_all;

Таблиця 9.1 – Очікуваний р

Результат:

ID	Name	Position	Salary
1	Іван Петренко	Інженер	26 250.0
2	Олена Коваль	Аналітик	28 350.0
3	Максим Левченко	Розробник	31 500.0

ID	Name	Position	Salary
1	Іван Петренко	Інженер	26 250.0
2	Олена Коваль	Аналітик	28 350.0
3	Максим Левченко	Розробник	31 500.0

Контрольні запитання

1. Які основні етапи ETL-процесу?
2. Чим відрізняється фізична інтеграція БД від віртуальної?
3. Які переваги використання Python для інтеграції БД?
4. Як вирішуються проблеми різних типів даних між MySQL і PostgreSQL?

Практична робота № 10 Характеристика віртуальних машин (VM)

Мета: ознайомлення з особливостями віртуальних машин Hyper-V, VMware, VMware Workstation.

Завдання: зробити аналіз віртуальних машин Hyper-V, VMware, VMware Workstation, описати їх можливості, переваги, недоліки, використання, застосування.

Загальні відомості

Сучасні технології віртуалізації відкривають нові можливості для роботи з операційними системами та програмним забезпеченням. Одним з найпопулярніших інструментів у цій галузі є Oracle VM VirtualBox – це безкоштовна програма, яка дозволяє працювати з віртуальними машинами на різних платформах. Із його допомогою можна встановлювати кілька ОС на одному комп'ютері, створювати та тестувати програмне забезпечення, а також мінімізувати ризики під час запуску потенційно небезпечного софту.

Віртуальна машина (VM) – це цифровий аналог фізичного комп'ютера, який створюється з допомогою VirtualBox або іншої програми для віртуалізації. По суті, це окрема операційна система, яка запущена всередині основної (хостової) ОС, вона використовує апаратні ресурси ПК, включно з процесором, жорстким диском і мережею.

Віртуальна машина працює у **власному ізольованому** середовищі, яке дозволяє встановлювати та запускати **різні** операційні системи без внесення змін до основної ОС. Це також **запобігає** конфліктам між різними системами та підтримує їхню коректну роботу.

Коли у віртуальній машині встановлюється та запускається гостьова операційна система, вона **взаємодіє** з основною ОС через спеціальний віртуалізаційний механізм, а керування здійснюється через зручний інтерфейс, аналогічний звичайним програмам.

Ключову роль цьому процесі відіграє **програма управління** віртуальними машинами (наприклад, VirtualBox). Вона створює **автономне** середовище для роботи Windows (або інших операційних систем) за допомогою програмних або апаратних механізмів віртуалізації.

І хоча гостьова ОС створена **всередині** віртуальної машини, вона отримує практично ті ж можливості, що й фізичний комп'ютер. Але додатково дозволяє **розширити** функціональні можливості ІТ-інфраструктури:

- тестувати програмне забезпечення;
- запускати несумісні програми;
- безпечно працювати з потенційно небезпечними файлами.

Завдяки таким перевагам віртуальні машини дозволяють **зробити** ІТ-інфраструктуру гнучкішою, скоротити витрати на обладнання, оптимізувати процеси, підвищити продуктивність серверів та робочих станцій.

Програма для створення віртуальних машин користується попитом серед ІТ-фахівців, які працюють із різними операційними системами. Вона особливо корисна в тих випадках, коли **потрібно** встановити та протестувати нову ОС, але є сумніви щодо її якості чи сумісності з комп'ютером.

Застосування віртуальних машин особливо корисне у таких ситуаціях:

- встановлення додаткових операційних систем на фізичний комп'ютер;
- встановлення та тестування програмного забезпечення, а також аналіз його роботи у різних середовищах;
- запуск програм та ігор, які призначені для конкретної ОС (наприклад, Windows), і не можуть правильно працювати на основній машині;
- безпечний запуск невідомих програм у ізолюваному середовищі;
- використання програм, які не сумісні з основною ОС, але можуть бути запущені на ВМ;
- створення та емуляція комп'ютерних мереж у складних конфігураціях без додаткового налаштування обладнання.

Завдяки таким можливостям віртуальні машини стали зручним та безпечним інструментом для ІТ-фахівців.

Віртуальні машини поступаються фізичним комп'ютерам у деяких аспектах, але вони мають низку переваг, які роблять їх незамінними в ІТ-сфері.

Переваги віртуалізації:

1. Снапшоти та відкати системи: перед тестуванням нових програм можна створювати снапшоти (резервні копії стану ОС). Якщо програма виявилася несумісною або містить критичні помилки, ВМ можна швидко повернути до попереднього стану. Це займає менше часу та місця на диску порівняно з відновленням фізичної системи.

2. Ізольоване автономне середовище – віртуальна машина повністю ізольована від основної системи, що дозволяє безпечно запускати програми, копіювати її на інші пристрої та зберігати індивідуальні налаштування.

3. Гнучкість у налаштуванні ОС – можна легко перенести дані та конфігурації між віртуальною машиною та фізичним комп'ютером, що спрощує налаштування та розгортання програмного забезпечення.

4. Економія ресурсів – тестування програм на ВМ не займає місце на основному жорсткому диску, тому що для зберігання даних використовується виділена віртуальна пам'ять. Всі зміни фіксуються в log-файлі та видаляються після завершення роботи.

5. Збереження стану ОС – віртуальна машина зберігає свій поточний стан навіть у разі вимкнення основного комп'ютера. У разі наступного запуску ВМ можна продовжити роботу з місця.

6. Одночасний запуск різних ОС – не потрібно перезавантажувати комп'ютер для перемикання між системами, що значно прискорює роботу.

7. Оптимізація витрат – замість купівлі кількох фізичних комп'ютерів можна встановлювати та запускати кілька ВМ на одному пристрої, що скорочує витрати на обладнання та інфраструктуру.

Недоліки віртуальних машин:

1. Високі вимоги до ресурсів: для роботи кількох ВМ одночасно потрібне потужне обладнання з достатнім обсягом оперативної пам'яті та процесорних потужностей.

2. Низька продуктивність порівняно з фізичним ПК – традиційні операційні системи працюють швидше, ніж їхні віртуалізовані аналоги.

3. Обмежена підтримка апаратного прискорення – багато віртуальних машин мають труднощі з роботою графічних процесорів, відеоадаптерів і 3D-прискорення.

4. Залежність від програмних розробок – технології віртуалізації постійно розвиваються, але все ж таки поступаються фізичним системам у продуктивності та підтримці спеціалізованого обладнання.

Незважаючи на деякі недоліки віртуальних машин, грамотне управління ресурсами допомагає мінімізувати їхні слабкі сторони та максимально використати переваги.

Контрольні запитання

1. Технології віртуалізації. Віртуальні машини.
2. Порівняльний аналіз віртуальних машин.
3. Ефективне використання обчислювальних ресурсів.
4. Скорочення витрат на програмне забезпечення.

Практична робота № 11 **Адміністрування ОС Windows Server**

Мета роботи: ознайомити студентів із ролями та їх адмініструванням у ОС Windows Server.

Завдання:

1. Ознайомитись з теоретичними відомостями по службам у ОС Windows Server.
2. Встановити і налаштувати сервіс на ВМ відповідно до індивідуального завдання.
3. Вказати і вивчити основні можливості сервісу.

Загальні відомості

Установка, налаштування і використання системи Windows Server залежить від тих завдань, які повинна виконувати конкретна інсталяція. Типові завдання системи корпорація Microsoft об'єднала у вигляді «ролей» сервера. Всі ролі можна побачити в разі запуску майстрів «Майстер налаштування сервера» або «Управління даним сервером». До основних ролей належать:

1. Файловий сервер (сервер, що надає доступ до файлів і його керівник; вибір цієї ролі дозволить вам швидко набудувати параметри квотування та індексування).

2. Сервер друку (сервер, організуючий доступ до мережеских принтерів і керівник чергами друку і драйверами принтерів; вибір цієї ролі дозволить вам швидко набудувати параметри принтерів і драйверів).

3. Сервер додатків (сервер, на якому виконуються вебслужби XML, вебдодатки і розподілені застосування. У разі призначення серверу цієї ролі на ньому автоматично встановлюються IIS, COM+ і Microsoft .NET Framework. За бажання ви можете додати до них серверні розширення Microsoft FrontPage, а також увімкнути або вимкнути ASP.NET);

4. Поштовий сервер (сервер, на якому працюють основні поштові служби POP3 (Post Office Protocol 3) і SMTP (Simple Mail Transfer Protocol), завдяки чому поштові POP3-клієнти домену можуть відправляти і отримувати електронну пошту. Вибравши цю роль, ви визначаєте домен за замовченням для обміну поштою і створюєте поштові скриньки).

5. Сервер терміналів (сервер, що виконує завдання для клієнтських комп'ютерів, які працюють в режимі термінальної служби; вибір цієї ролі приводить до установки служб терміналів, що працюють в режимі сервера додатків).

6. Сервер видаленого доступу / сервер віртуальної приватної мережі (сервер, що здійснює маршрутизацію мережевого трафіку і керівник телефонними з'єднаннями і з'єднаннями через віртуальні приватні мережі (virtual private network, VPN); вибравши цю роль, ви запуснете майстер налаштування сервера маршрутизації і видаленого доступу (Routing and Remote Access Server Setup Wizard). За допомогою параметрів маршрутизації і видаленого доступу ви можете вирішити тільки вхідні та вихідні підключення або повністю заборонити доступ ззовні).

7. Служба каталогів (контролер домену Active Directory – сервер, на якому працюють служби каталогів і розташовується сховище даних каталога; контролери домена також відповідають за вхід у мережу і пошук у каталозі; під час вибору цієї ролі на сервері будуть встановлені DNS і Active Directory);

8. Система доменних імен (сервер, на якому запущена служба DNS, що вирішує імена комп'ютерів у IP-адреси і навпаки; у разі вибору цієї ролі на сервері буде встановлена DNS і запущений майстер налаштування DNS-сервера).

9. Сервер протоколу динамічної настройки вузлів (сервер, на якому запущена служба DHCP (Dynamic Host Configuration Protocol), що дозволяє

автоматизувати призначення IP-адрес вузлам мережі. У разі вибору цієї ролі на сервері буде встановлена служба DHCP і запущений майстер створення області.

10. Сервер Windows Internet Naming Service (сервер, на якому запущена служба WINS (Windows Internet Name Service), що вирішує імена NETBIOS IP-адрес і навпаки; вибір цієї ролі приводить до установки служби WINS);

11. Сервер – потокове мультимедіа-віщання (сервер, що надає мультимедійні потоки іншим системам мережі або інтернету; вибір цієї ролі приводить до установки служб Windows Media; ця роль підтримується тільки у версіях Standard Edition і Enterprise Edition).

Зі способами вирішення адміністративних завдань найтіснішим чином зв'язана і архітектура системи безпеки Windows Server. Active Directory і адміністративні шаблони дозволяють застосовувати параметри безпеки до всіх робочих станцій і серверів компанії. Іншими словами, ви налаштовуєте захист даних не кожного конкретного комп'ютера, а всього підприємства загалом.

Порядок виконання роботи

1. Ознайомитись з теоретичними відомостями по службам у ОС Windows Server.

2. Встановити і налаштувати сервіс на VM відповідно до індивідуального завдання.

3. Вказати і вивчити основні можливості сервісу.

Контрольні запитання

1. Подайте визначення мережевих служб.

2. Опишіть служби доменів Active Directory (AD DS).

3. Як працює сервер протоколу динамічного налаштування вузлів, сервер, на якому запущена служба DHCP (Dynamic Host Configuration Protocol)?

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Інтегровані інформаційні системи [Електрон. ресурс] : конспект лекцій / О. В. Бойко. – Електрон. текст. дані. – Суми : Сумський державний університет, 2023. – 130 с. – Режим доступу: <https://essuir.sumdu.edu.ua/server/api/core/bitstreams/6af45ef5-ef68-469f-bb88-12fd6e75b4b3/content>, вільний (дата звернення: 09.03.2026). – Назва з екрана.
2. Адміністрування комп'ютерних мереж та операційних систем [Електрон. ресурс] : методичне видання для студентів за спеціальністю 121 – Інженерія програмного забезпечення факультету інформаційних технологій УжНУ / В. В. Поліщук. – Електрон. текст. дані. – Ужгород: 2019. – 60 с. – Режим доступу: <https://www.scribd.com/document/707387298/%D0%90%D0%B4%D0%BC%D1%96%D0%BD%D1%96%D1%81%D1%82%D1%80%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F-%D0%9A%D0%9C-%D1%96-%D0%9E%D0%A1-%D0%9B%D0%B5%D0%BA%D1%86%D1%96%D1%97>, вільний (дата звернення: 18.03.2026). – Назва з екрана.
3. Комп'ютерні мережі. Книга 1 [Електрон. ресурс] : посібник / А. Г. Микитишин, М. М. Митник, П. Д. Стухляк, В. В. Пасічник. – Електрон. текст. дані. – Київ: «Ліра», 2024. – 217 с. – Режим доступу: <https://magnolia.lviv.ua/wp-content/uploads/2024/03/Komp.-merezhi.-Tom2-zmist.pdf>, вільний (дата звернення: 15.04.2026). – Назва з екрана.
4. Дискретна математика [Електрон. ресурс]. – Електрон. текст. дані. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 154 с. – Режим доступу: <http://mathhelpplanet.com/static.php?p=osnovnye-zakony-raspredeleniya-nepreryvnyh-sluchainyh-velichin>, вільний (дата звернення: 20.03.2026). – Назва з екрана.

Електронне навчальне видання

Методичні рекомендації
до виконання практичних робіт
із навчальної дисципліни

СИСТЕМНА ІНТЕГРАЦІЯ ТА АДМІНІСТРУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

*(для здобувачів першого (бакалаврського) рівня вищої освіти денної,
заочної і дистанційної форм навчання
зі спеціальностей F6 – Інформаційні системи та технології,
F3 – Комп'ютерні науки)*

Укладач **СІЗОВА** Наталія Дмитріївна

Відповідальний за випуск *М. В. Новожилова*
Редактор *Б. О. Хільська*
Комп'ютерне верстання *Н. Д. Сізова, Т. П. Воробйова*

План 2026, поз. 214М

Підп. до друку 19.06.2026. Формат 60 × 84/16.

Ум. друк. арк. 3,5.

Видавець і виготовлювач:
Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Чорноглазівська, 17, Харків, 61002.
Електронна адреса: office@kname.edu.ua
Свідоцтво суб'єкта видавничої справи:
ДК № 8386 від 14.07.2025.