МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

Н. Д. Сізова

СИСТЕМНА ІНТЕГРАЦІЯ ТА АДМІНІСТРУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

КОНСПЕКТ ЛЕКЦІЙ

(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

УДК 004.891

Сізова Н. Д. Системна інтеграція та адміністрування інформаційних систем : конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти денної, заочної і дистанційної форм навчання зі спеціальностей F6 – Інформаційні системи та технології, F3 – Комп'ютерні науки) / Н. Д. Сізова ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2026. – 73 с.

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

Рецензент
М. Ю. Карпенко, кандидат технічних наук, доцент кафедри комп'ютерних наук та інформаційних технологій (Харківський національний університет міського господарства імені О. М. Бекетова)

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

ЗМІСТ

ВСТУП

У сучасному цифровому середовищі інформаційні системи стали важливою складовою діяльності підприємств, організацій і державних установ. Зростання обсягів даних, ускладнення програмно-апаратних комплексів та необхідність забезпечення їх безперервної роботи зумовлюють підвищену увагу до питань системної інтеграції та адміністрування. Саме ефективне поєднання різнорідних компонентів у єдину узгоджену систему та їхнє професійне обслуговування визначають стабільність, продуктивність і безпеку інформаційної інфраструктури.

Дисципліна «Системна інтеграція та адміністрування інформаційних систем» спрямована на формування у студентів теоретичних знань і практичних навичок щодо побудови, налаштування, супроводу та оптимізації складних інформаційних систем. Особлива увага приділяється принципам інтеграції програмного забезпечення, баз даних, мережевих технологій, хмарних сервісів, а також методам управління ресурсами та забезпечення інформаційної безпеки.

У процесі вивчення курсу розглядаються сучасні підходи до системної інтеграції, включно із сервіс-орієнтованою архітектурою (SOA), мікросервісними рішеннями, контейнеризацією та автоматизацією процесів адміністрування. Також аналізуються інструменти моніторингу, резервного копіювання, відновлення даних і управління доступом, що є критично важливими для забезпечення надійності та відмовостійкості систем.

Метою цього конспекту є систематизація основних понять, методів і технологій у сфері системної інтеграції та адміністрування інформаційних систем, а також надання практичних рекомендацій щодо їх застосування.

У матеріалі подаються дев’ять тем та список рекомендованих джерел. До кожного розділу наведені запитання для самоконтролю, що дозволить студентам перевірити свої знання за теоретичними темами.

Матеріал орієнтований на здобувачів першого (бакалаврського) рівня вищої освіти денної, заочної і дистанційної форм навчання зі спеціальностей 126 − Інформаційні системи та технології, 122 – Комп’ютерні науки та може бути використаний здобувачами інших спеціальностей як навчальний і довідковий ресурс під час підготовки до занять, виконання лабораторних робіт і самостійного вивчення дисципліни.

ТЕМА 1 Завдання і функції адміністрування інформаційних систем

1.1 Поняття інформаційної системи, адміністрування

Інформаційна система *(англ. «Information system»)* – сукупність організаційних і технічних засобів для збереження та обробки інформації з метою забезпечення інформаційних потреб користувачів. Для існування цивілізації необхідний обмін інформацією – передача знань як між окремими членами і колективами суспільства, так і між різними поколіннями. Інформаційна система може існувати і без застосування комп'ютерної техніки, це питання економічної необхідності.

Приклади інформаційних систем: система керування польотами літаків, бібліотека, аналітичний центр соціологічних досліджень, довідкова система залізничного вокзалу тощо.

Інформаційна система − організований набір елементів, що збирає, обробляє, передає, зберігає та надає дані. Інформаційна система складається з людей, обладнання, процесів, процедур, даних та операцій [1].

Кожна інформаційна система містить такі компоненти:

– структура системи;

– функції кожного елемента системи;

– вхід і вихід кожного елемента і системи загалом;

– мета і обмеження системи та її окремих елементів.

Інформаційна система не тільки відображає функціонування об'єкта управління, а й впливає на нього через органи управління. Вона є сукупністю інформаційних процесів для задоволення потреби в інформації різних рівнів ухвалення рішень.

Її метою є продукування інформації для використання (споживання) управлінським апаратом. Відповідно вона забезпечує накопичення, передачу, збереження, оброблення та узагальнення інформації «знизу вгору», а також конкретизацію інформації «згори донизу».

Призначення ІС полягає в описі об'єкта, його станів, взаємодії, що виражається через певні показники. Вона покликана своєчасно подавати органам управління необхідну і достатню інформацію для ухвалення рішень, якість яких забезпечує високоефективну діяльність об'єкта управління та його підрозділів.

До головних завдань належать:

– виявлення джерел інформації;

– збирання, реєстрація, обробка та видача інформації, що характеризує стан виробництва та управління;

– розподіл інформації між керівниками, підрозділами та виконавцями відповідно до їхньої участі в управлінні.

Ключовими елементами кожної організації є персонал, структура, робочі процедури, політика і культура.

Таким чином, можна сказати, що сукупність взаємопов'язаних елементів, що утворюють єдине ціле і призначені для реалізації інформаційних процесів, має назву інформаційної системи.

Інформаційна система має **елементи**, які взаємопов'язані між собою:

– *складові, що забезпечують збирання даних з різних джерел:* наприклад, це метеорологічні станції, метеорологічні повітряні зонди, метеорологічні супутники Землі тощо;

– *канали передавання даних:* наприклад, радіо, телевізійні, телефонні, телеграфні, комп'ютерні мережі тощо;

– *складові, що забезпечують упорядковане зберігання даних та їхнє опрацювання,* системи упорядкування і зберігання повідомлень: співробітники, обчислювальні пристрої, спеціальні програми, які на основі отриманих повідомлень створюють прогноз погоди;

– *споживачі даних:* ними можуть бути мешканці окремого регіону, країни або всієї планети, моряки, льотчики, агрономи та інші.

Важливою частиною інформаційних систем стають пристрої, які автоматизують інформаційні процеси, особливо процеси опрацювання даних. Такими пристроями, зокрема, є комп'ютери. Інформаційна система має апаратну та програмну складову.

Апаратна складова − це комплекс технічних засобів, який містить пристрої опрацювання і зберігання даних, пристрої введення і виведення, засоби комунікацій.

Програмна складова − це комплекс програм, які забезпечують реалізацію інформаційних процесів пристроями інформаційної системи. Програми є одним із видів даних.

Інформаційні системи можна класифікувати за їхніми видами.

Кожна інформаційна система містить такі компоненти:

– структура системи;

– функції кожного елемента системи;

– вхід і вихід кожного елемента і системи загалом;

– мета і обмеження системи та її окремих елементів.

Інформаційна система не тільки відображає функціонування об'єкта управління, а й впливає на нього через органи управління.

Основне призначення інформаційних систем − це своєчасне подання необхідної інформації ОПР для ухвалення адекватних і ефективних рішень під час керування процесами, ресурсами, фінансовими транзакціями, персоналом або організацією загалом. Однак у процесі розвитку інформаційних технологій, дослідження операцій і технологій моделювання, а також зі зростанням споживачів інформаційно-аналітичної підтримки самих ОПР, усе більше проявлялася потреба в системах, що не тільки подають інформацію, але й виконують деякий її попередній аналіз, здатні давати деякі поради й рекомендації, здійснювати прогнозування розвитку ситуацій, відбирати найбільш перспективні альтернативи рішень, тобто підтримувати рішення ОПР,

взявши на себе значну частину рутинних операцій, а також функції попереднього аналізу й оцінок [1–2].

Інформаційна система підтримки рішень (ІСПР) пов'язує інтелектуальні ресурси керівника зі здатностями й можливостями комп'ютера для поліпшення якості рішень. Ці системи призначені для ухвалення рішень в умовах напівструктурованих і слабко структурованих завдань.

Інтелектуальні інформаційні технології (Intellectual information technology, ІІТ) − це інформаційні технології, що допомагають людині прискорити аналіз політичної, економічної, соціальної й технічної ситуації, а також синтезувати управлінські рішення.

Використання ІІТ у реальній практиці має на увазі облік **специфіки** проблемної області, що може характеризуватися таким набором ознак:

– якість і оперативність ухвалення рішень;

– нечіткість цілей та інстиціональних границь;

– множинність суб'єктів, що беруть участь у вирішенні проблеми;

– хаотичність, флюктованість і квантованість поводження середовища;

– множинність взаємного впливу факторів один на одного;

– слабка формалізованість, унікальність, нестереотипність ситуацій;

– латентність, прихованість, наявність інформації;

– девіантність реалізації планів, значущість малих дій;

– парадоксальність логіки рішень тощо.

Інтелектуальні інформаційні системи (ІІС) мають багато переваг, які роблять їх **ефективними** в обробці, аналізі та використанні даних [3–4].

Комп'ютерна система – означає будь-який пристрій або групу взаємно поєднаних або пов'язаних пристроїв, один чи більшість з яких, відповідно до певної програми, виконує автоматичну обробку даних

Комп'ютерна система – це система, яка поєднує, з одного боку, фізичну частину обчислень, а з іншого – цифрову або нематеріальну частину обчислень.

Комп'ютерна мережа – це система розподіленої обробки інформації між комп'ютерами за допомогою засобів зв'язку. Комп'ютерна мережа – це сукупність територіально рознесених комп'ютерів, здатних обмінюватися між собою повідомленнями через середовище передачі даних.

1.2 Склад служб адміністратора системи та їх функції

Однією з умов стабільної роботи інформаційної системи є процес адміністрування. Системне адміністрування ІС – це комплекс заходів з налаштування, управління і підтримки працездатності комп'ютерної техніки, який також передбачає забезпечення безпеки інформаційних систем і даних організації. Процес адміністрування інформаційних систем містить етапи від виявлення інформаційних потреб до збору, введення та обробки даних. Він охоплює життєвий цикл ІС: проєктування, розробку, тестування, впровадження та супровід [1].

Об’єктами адміністрування є апаратне забезпечення, програмне забезпечення, мережі, бази даних та користувачі в ІС. Управління ними передбачає забезпечення сумісності, стандартизації та єдності обліку.
Моделі управління містять централізовані, децентралізовані та колективного використання ІС, а також багаторівневі з інтеграцією за рівнями. Для вирішення задач використовуються моделі теорії ігор, черг, лінійного програмування.

Адміністрування інформаційних систем (ІС) − це сукупність організаційних, технічних і програмних заходів, спрямованих на забезпечення стабільної, безпечної, ефективної та безперервної роботи інформаційної системи протягом усього її життєвого циклу.

Основні цілі адміністрування ІС:

– забезпечення доступності та надійності ІС;

– підтримка продуктивності та масштабованості;

– захист даних і ресурсів;

– мінімізація ризиків збоїв та втрат інформації;

– підтримка користувачів і бізнес-процесів.

Етапи процесу адміністрування ІС передбачають:

1. Планування, тобто визначення політик адміністрування, розподіл ролей і відповідальності, планування ресурсів та резервування.
2. Розгортання та налаштування − інсталяція серверів, ОС, СУБД, прикладного ПЗ, конфігурування мережі та доступів.
3. Моніторинг і контроль передбачає контроль стану апаратних і програмних компонентів, аналіз журналів подій (logs), контроль навантаження і продуктивності.
4. Експлуатація та підтримка: управління користувачами, обробка інцидентів; оновлення та патч-менеджмент.
5. Забезпечення безпеки, управління доступом, резервне копіювання, аудит і реагування на загрози.
6. Оптимізація та розвиток, тобто масштабування, інтеграція з іншими системами, модернізація архітектури.

Адміністратор інформаційної (комп’ютерної) системи або системний адміністратор *(від англ. «system administrator», «systems administrator»)* – працівник, посадові обов’язки якого передбачають забезпечення роботи комп’ютерної техніки, комп’ютерної мережі і програмного забезпечення в організації. Системний адміністратор може бути працівником підрозділу інформаційних технологій або окремою штатною одиницею залежно від розміру організації.

Системний адміністратор (від англ. *«system administrator», «systems administrator»*) – працівник, посадові обов’язки якого передбачають забезпечення роботи комп’ютерної техніки, комп’ютерної мережі і

програмного забезпечення в організації. Інша назва – сисадмін *(від англ. «sysadmin»)* походить від комп'ютерного сленгу. Системний адміністратор може бути, залежно від розміру організації, або працівником підрозділу інформаційних технологій або окремою штатною одиницею.

Ефективне адміністрування забезпечує надійність збереження даних, швидкість доступу до них та захист від несанкціонованого втручання

Системний адміністратор – не розробник програмного забезпечення.

Типові (функції) обов'язки системного адміністратора із точки зору адміністрування комп'ютерних систем:

– підготовка і збереження резервних копій даних, їх періодична перевірка і знищення;

– встановлення і конфігурування оновлень операційної системи і прикладного програмного забезпечення;

– встановлення і конфігурування нового апаратного і програмного забезпечення;

– створення і підтримка в актуальному стані файлу облікових записів користувачів;

– підтримання інформаційної безпеки в організації;

– документування своєї роботи;

– усунення несправностей у комп'ютерній системі.

Типові (функції) обов'язки системного адміністратора з точки зору адміністрування комп'ютерних мереж:

– планування мережі;

– встановлення та налаштування мережевих вузлів;

– встановлення та налаштування мережевих протоколів;

– встановлення та налаштування мережевих служб.

– адміністрування служб каталогів (Novell NDS, Microsoft Active Directory);

– адміністрування служб обміну повідомленнями (системи електронної пошти);

– адміністрування служб доступу до баз даних;

– пошук несправностей;

– моніторинг мережевих вузлів та трафіку;

– забезпечення захисту даних.

1.3 Об’єкти адміністрування і моделі управління в інформаційних системах

Об’єкти адміністрування − це компоненти ІС, на які спрямовані управлінські дії адміністратора (рис. 1.1) .

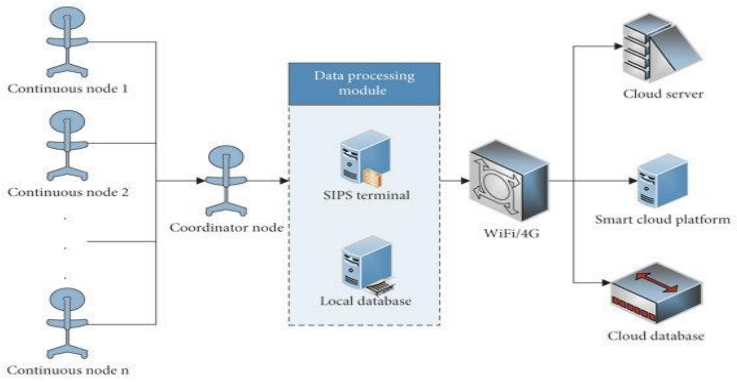


Рисунок 1.1 – Деякі компоненти ІС [4]

Таблиця 1.1 − Основні групи об’єктів адміністрування

Об’єкти	Елементи
Апаратні ресурси	– сервери; – робочі станції; – мережеве обладнання (маршрутизатори, комутатори); – системи зберігання даних (NAS, SAN)
Програмне забезпечення	– операційні системи; – серверні платформи; – прикладні системи (ERP, CRM, MES, LMS); – віртуалізація та контейнеризація
Дані та інформаційні ресурси	– бази даних; – файлові сховища; – резервні копії; – архіви
Мережна інфраструктура	– локальні та глобальні мережі; – VPN; – служби DNS, DHCP
Користувачі та ролі	– облікові записи; – права доступу; – групи користувачів; – політики безпеки
Бізнес-процеси	– процеси обробки інформації; – регламенти роботи; – інтеграційні потоки між системами.

Адміністрування серверів

Основний перелік питань, що надається під час адміністрування серверів:

– первинне налаштування і монтаж серверів;

– під’єднання до інженерних мереж, до мереж передачі даних;

– рішення інцидентів, що пов’язані з обладнанням;

– ремонт і модернізація обладнання;

– підтримка операційних систем серверів сімейства Windows Server та Linux;

– установка та базове налаштування операційних систем під час інсталяції;

– налаштування операційних систем під час експлуатації серверів;

– рішення інцидентів, що виникають під час роботи системи, служб, сервісів;

– відновлення працездатності операційної системи після збоїв;

– контроль відомих вразливостей і оновлення операційних систем;

– резервне копіювання критичних даних;

– підтримка серверних і мережевих сервісів;

– налаштування, адміністрування серверних служб і мережевих сервісів під час роботи, управління ними;

– здійснення моніторингу доступності обладнання і сервісів, що працюють.

Адміністрування інформаційних систем є ключовим фактором стабільності та ефективності ІС. Сучасні ІС потребують:

– комплексного підходу до об’єктів адміністрування;

– поєднання класичних та інтелектуальних моделей управління;

– переходу від реактивного до проактивного адміністрування.

Модель управління в інформаційних системах визначає спосіб організації контролю, ухвалення рішень і взаємодії між компонентами системи (табл. 1.2).

Таблиця 1.2 − Моделі управління в інформаційних системах

Модель	Характеристика	Переваги	Недоліки	Застосування
1	2	3	4	5
Модель управління в інформаційних системах	Єдиний центр управління. Всі рішення ухвалюються центральним адміністратором	Простота контролю та єдина політика безпеки	Низька масштабованість та єдина точка відмови	Невеликі та середні організації
Децентралізована модель управління	Автономні підсистеми, локальні адміністратори	Висока гнучкість, підвищена відмовостійкість	Складність координації та ризик несумісності політик	Великі розподілені системи

Продовження таблиці 1.2

1	2	3	4	5
Ієрархічна модель управління	Багаторівнева структура, поєднання централізованого і локального управління	Баланс контролю та автономності, зручна масштабованість	Складність реалізації	Корпоративні інформаційні системи
Проактивна модель управління	Прогнозування збоїв, використання аналітики та ШІ, автоматизоване ухвалення рішень	Прогнозування збоїв, використання аналітики та ШІ, автоматизоване ухвалення рішень	Складність впровадження та висока вартість	Industry 4.0/5.0, Smart City, сучасні ІІС
Сервісна модель управління (ITSM, ITIL)	Орієнтація на сервіси, управління SLA та процесний підхід	Прозорість якості послуг, орієнтація на користувача	–	Державні та комерційні IT-служби

Контрольні запитання

1. Опишіть поняття «системність».
2. Дайте визначення поняттю «інформаційні системи» (ІС).
3. Розкрийте переваги ІІС.
4. Які цілі адміністрування інформаційних систем?
5. Назвіть етапи адміністрування.
6. Опишіть об’єкти адміністрування в інформаційних системах.
7. Назвіть моделі управління в інформаційних системах

ТЕМА 2 Адміністрування операційної системи

2.1 Характеристика та функції сучасних операційних систем

Операційна система – це програма, яка завантажується під час увімкнення комп’ютера. Вона проводить діалог із користувачем, здійснює управління комп’ютером, його ресурсами (оперативною пам’яттю, місцем на дисках тощо), запускає інші (прикладні) програми на виконання. Операційна система забезпечує користувачу і прикладним програмам зручний спосіб спілкування (інтерфейс) із пристроями комп’ютера [3].

Операційна система – це низькорівневе програмне забезпечення, яке керує апаратними та програмними ресурсами комп’ютера і забезпечує виконання основних функцій комп’ютера, таких як планування завдань, управління ресурсами, управління пам’яттю, управління периферійними пристроями, мережева взаємодія тощо. Операційна система це − базовий комплекс програмного забезпечення, що виконує **управління** апаратним

забезпеченням комп'ютера або віртуальної машини; забезпечує керування обчислювальним процесом і організує взаємодію з користувачем.

Операційні системи (ОС) посідають важливіше місце в сукупності сучасних системних програмних засобів, які складають програмне забезпечення електронно-обчислювальних машин. Усі її програми можна поділити на **дві** групи: керуюча програма та системні оброблювальні програми.

Керуюча програма складається з низки компонентів, серед яких необхідно виділити основні:

– управління статичними ресурсами (управління завданнями);

– управління динамічними ресурсами (управління задачами);

– управління даними;

– управління поновленням.

Операційна система − це сукупність програм, які призначені для **керування ресурсами** комп'ютера й обчислювальними процесами, а також для організації взаємодії користувача з апаратурою.

Інша функція ОС – **керування обчислювальними процесами**.

Найпоширеніші операційні системи, якими ми користуємося в повсякденній роботі: Dos, Windows 3.+, Windows 95, Windows NT, Windows 98, Windows XP.

Керування функціями операційної системи здійснюється за допомогою **параметрів ядра** ОС та **спеціальних засобів** (утиліт), що входять до її складу.

Параметри ядра ОС задаються **адміністратором систем**и (АС) під час інсталяції ОС.

Після встановлення ОС **адміністратор** системи задає атрибути користувачу системи та здійснює **оперативне керування** ОС.

У процесі авторизації користувачів АС може встановити **низку параметрів** їхньої роботи: права доступу, максимальний об'єм дискового простору, пароль користувача тощо. Засоби обліку ресурсів ОС дозволяють **адміністратору** системи накопичувати для подальшого аналізу інформацію про використання окремими користувачами таких ресурсів, як кількість блоків, зчитаних / записаних із диска сервера, кількість блоків, записаних за день, тривалість роботи програми тощо.

Утиліти роботи з консоллю сервера дозволяють **адміністратору** системи контролювати функціонування робочих станцій, зупиняти або запускати принтер, керувати чергами завдань до принтерів, надсилати повідомлення користувачам. Всі операційні системи мають схожі, але дещо відмінні засоби оперативного керування.

Встановлення операційної системи − відповідальний для АС процес. Він містить підготовку майданчика та обладнання, інсталяцію файл-сервера та інсталяцію програмного забезпечення робочих станцій, планування структур каталогів (директорій), планування користувачів та груп користувачів, планування захисту, планування процедур реєстрації, налаштування параметрів.

У разі некоректної початкової інсталяції ОС та неправильно заданих параметрів подальша експлуатація системи може бути неефективною, а в деяких випадках неможливою.

Процесу інсталяції має передувати низка підготовчих дій.

Насамперед адміністратор системи повинен перевірити умови експлуатації та виконання вимог щодо електроживлення обладнання. Всі апаратні засоби потрібно під'єднати до спеціалізованих ліній живлення, виділених лише для роботи комп'ютерного обладнання. Всі розетки повинні бути заземленими трипровідними, з'єднаними безпосередньо з землею, обладнання повинно бути правильно під'єднане до сигнальних і силових ліній.

АС повинен вирішити, чи він планує оновлення існуючої версії ОС або виконувати первинну інсталяцію.

Процес встановлення ОС відбувається так: системні файли поміщаються на диск в спеціальну область. Завантажуються дискові, мережеві драйвери та драйвери периферійних пристроїв. Визначаються параметри їхньої роботи. Це може виконуватися або адміністратором системи, наприклад, окремою командою Load, або автоматично самою ОС.

Після цього адміністратор системи завантажує ядро ОС за допомогою виклику команди, запропонованої виробником, наприклад, Server.exe, і задає основні параметри роботи ядра. До цих параметрів належать:

– ім'я сервера;

– ім'я адміністратора та його пароль;

– список мережних протоколів та їхні налаштування (наприклад, TCP/IP);

– параметр блокування консолі сервера;

– опція шифрування паролів у системі;

– номери черг друку;

– команди трасування дій ядра (наприклад, Track On) тощо.

Після цього **адміністратору** системи необхідно встановити ОС на робочих станціях аналогічно до встановлення сервера.

Після встановлення адміністратор системи повинен **спланувати** додаткові директорії, наприклад, прикладні директорії для програм додатків системи або директорії загального користування для проміжного копіювання файлів, групи користувачів з їхніми правами доступу (можливе виділення для групи своєї директорії або тому) та створити користувачів у системі, приписавши їх до певних груп. Для користувачів та груп необхідно спланувати права доступу.

Далі **АС** має спланувати процедуру реєстрації користувача на сервері. Фактично виконуються завжди дві процедури – **спочатку системна** (для налаштування робочого середовища всіх користувачів), а потім **користувальницька** (для налаштування середовища конкретного користувача). До системної процедури можуть входити загальні вітання всіх користувачів, призначення імен (літери англійського алфавіту) цільовим дискам, підключення груп користувачів до різних серверів. У процедурах реєстрації користувача ініціюються параметри середовища кожного

користувача, наприклад, доступ досервера тільки цього користувача. Конкретні повноваження процедур реєстрації залежать від реалізації ОС.

Деякі ОС дозволяють підтримувати **кілька** файлових систем, під кожну з них виділяється свій том. АС перед зверненням до файлової системи повинен змонтувати те, на якому томі розташовуватиметься. Під час цієї операції адміністратор проводить перевірку типу файлової системи тому та її цілісності, зчитування системних структур даних (зміст тому), ініціалізація відповідного модуля ОС, долучення файлової системи до загального простору імен.

АС повинен пам'ятати, що складні та розвинені методи доступу зазвичай використовуються під час реалізації **не** універсальних ОС, а СУБД, як спеціалізованих ОС для роботи з даними.

При цьому **АС** повинен врахувати, що транзакції СУБД та транзакції ОС можуть не відповідати один одному, а методи відновлення даних СУБД перевершувати існуючі ОС. З іншого боку, ОС, підтримуючи файлові системи, **не** займаються питаннями цілісності даних. Це реалізується лише СУБД. Завдання АС правильно комбінувати наявні системні засоби та уникати протиріч.

2.2 Підготовка дискової підсистеми, технологія RAID

Дискова підсистема – це сукупність накопичувачів, контролерів та програмних засобів, що забезпечують зберігання й обробку даних у комп'ютерній системі. Основні компоненти підсистеми:

1. HDD (жорсткі диски).
2. SSD (твердотільні накопичувачі).
3. RAID-контролери.
4. Інтерфейси SATA, NVMe.
5. Файлові системи (NTFS, ReFS).

RAID (Redundant Array of Independent Disks) – технологія об'єднання кількох фізичних дисків у єдиний логічний масив для підвищення: продуктивності, надійності, відмовостійкості, швидкості читання / запису.

Основні рівні RAID подано у таблиці 2.1.

Таблиця 2.1 − Класифікація RAID

Рівень	Мін. дисків	Продуктивність	Надійність	Об'єм	Характеристика
1	2	3	4	5	6
RAID 0	2	Висока	Немає	100 %	Мінімум 2 диски. Дані розподіляються блоками між дисками. Відмовостійкості немає. Максимальна швидкість

Продовження таблиці 2.1

1	2	3	4	5	6
RAID 1	2	Середня	Висока	50 %	Мінімум 2 диски. Дані повністю дублюються. Висока відмовостійкість. Зменшення доступного об'єму у 2 рази
RAID 5	3	Висока	Висока	n-1	Мінімум 3 диски. Дані + паритет. Баланс між швидкістю і надійністю. Витримує відмову 1-го диска

2.3 Адміністрування файлових систем

Адміністрування файлових систем (ФС) − це процес керування логічною та фізичною структурою зберігання даних на носіях [2]. Популярні файлові системи:

1. Windows: NTFS (основна для ОС), FAT32/exFAT (для флеш-накопичувачів).
2. Linux: ext4, XFS (для великих серверів), Btrfs (із підтримкою знімків стану).
3. Мережеві: NFS, SMB/CIFS (для спільного доступу в мережі).

Адміністрування файлових систем охоплює створення, управління файлами, контроль доступу та налаштування серверів.

Основні завдання адміністратора:

– підготовка носіїв: розбиття диска на розділи (partitioning) та створення логічних томів (LVM);

– форматування: створення конкретної структури ФС (наприклад, NTFS, ext4, XFS) на підготовлених розділах;

– монтування: під'єднання файлової системи до загального дерева каталогів ОС (через команду `mount` або файл `/etc/fstab` у Linux);

– керування доступом: налаштування прав (читання, запис, виконання) для користувачів та груп (команди `chmod, chown`).

Адміністрування файлових систем передбачає обслуговування та моніторинг:

– перевірка цілісності та виправлення помилок (утиліта `fsck`);

– контроль вільного місця та управління дисковими квотами;

– зменшення фрагментації даних для підвищення швидкодії;

– резервне копіювання: створення та перевірка актуальності бекапів для запобігання втраті даних.

2.4 Протоколи передачі файлів і файлові системи Інтернет. FTP, SUN NFS та ISO FTAM

Протоколи передачі файлів забезпечують обмін даними між комп'ютерами в мережі [4]. Вони визначають:

– правила встановлення з'єднання;

– формат передавання файлів;

– механізми автентифікації;

– управління доступом;

– контроль цілісності даних.

У розвитку мережевих технологій особливу роль відіграли такі рішення (табл. 2.2):

1. **FTP** − класичний протокол передачі файлів у TCP/IP.
2. **NFS** – мережева файлова система.
3. **FTAM** – стандарт ISO для доступу та передачі файлів.

Таблиця 2.2 – Характеристика протоколів

Протокол	Основні характеристики	Переваги	Недоліки
File Transfer Protocol (FTP**)**	Працює поверх TCP. Використовує два канали: – керуючий (порт 21); – канал передачі даних (порт 20 або динамічний). Підтримує анонімний доступ. Має активний та пасивний режими роботи.	Простота реалізації. Широка підтримка. Підтримка великих файлів.	Передача логінів і паролів у відкритому вигляді. Брак шифрування (у класичній версії)
SUN NFS (Network File System)	Архітектура «клієнт –сервер». Використання RPC (Remote Procedure Call). Монтування віддалених каталогів. Працює поверх TCP або UDP.	Прозорий доступ до файлів. Централізоване зберігання. Зручність адміністрування.	Залежність від мережі. Питання безпеки у ранніх версіях.
ISO FTAM	Розроблений у рамках моделі OSI. Підтримує: – створення; – читання; – модифікацію; – видалення файлів. Забезпечує складні механізми управління доступом.		Не поширений, оскільки має високу складність реалізації та велику кількість рівнів OSI.

File Transfer Protocol (FTP) − це прикладний протокол, що використовується для передачі файлів між клієнтом і сервером у мережах TCP/IP. Сучасні варіації: FTPS, SFTP (на базі SSH).

Network File System (NFS) − розподілена мережева файлова система, розроблена компанією **Sun Microsystems**. Надає користувачам доступ до віддалених файлів так, ніби вони знаходяться на локальному диску.

File Transfer Access and Management (FTAM) − стандарт ISO для передачі, доступу та управління файлами в моделі OSI (табл. 2.3).

Таблиця 2.3 – Порівняльна характеристика протоколів

Характеристика	**FTP**	**NFS**	**FTAM**
Модель	TCP/IP	Розподілена ФС	OSI
Тип	Передача файлів	Мережева ФС	Управління файлами
Рік створення	1971	1984	1988
Безпека	Низька (класична версія)	Середня	Вища (стандарт ISO)
Складність	Невелика	Середня	Висока

Таблиця 2.3 уможливлює порівняння протоколів за типом, безпекою, складністю тощо.

Контрольні запитання

1. Опишіть дії адміністратора для сучасних операційних систем.
2. Яку роль відіграють дискові підсистеми і технологія RAID в адмініструванні операційних систем?
3. Які особливості адміністрування файлових систем?
4. Назвіть протоколи передачі файлів та їхнє призначення.

ТЕМА 3 Адміністрування мережевих систем

3.1 Адресація в мережах

Кожна людина має прізвище, ім’я, паспорт, ідентифікаційний код. Їх складають за певними правилами та вони є унікальними для кожної людини. Так само чинять з адресою ресурсів мережі «Інтернет». Кожний ресурс Інтернету (апаратний, програмний чи інформаційний) має свою адресу [3−4]. Види адресацій − **IP-адреси, доменні адреси, URL-адреси**.

IP-адреса − це ідентифікаційний номер комп’ютера в мережі. Як і в локальній мережі, IP-адреса комп’ютера в Інтернеті створюється за протоколом IPv4 та складається з чотирьох десяткових чисел від 0 до 255, розділених крапками, наприклад 78.111.176.233.

Виявилося, що кількості комбінацій чотирьох чисел (4,2 млрд) для потреб адресації недостатньо, тому з 2008 року запроваджено протокол IPv6. За ним IP-адреса записується вісьмома шістнадцятковими числами, розділеними двокрапками, наприклад 011:0db2:11d3:087f:07a0:345e:8a2e:32c2.

Це еквівалентно 16 десятковим числам від 0 до 255 і збільшує кількість можливих адрес до 340 трлн.

Комп'ютер у мережі може мати постійну (**статичну**) або тимчасову (**динамічну**) IP-адресу. Статичну адресу мають **усі** сервери, щоб комп'ютери мережі «знали», де шукати інформацію. Динамічну IP-адресу комп'ютер отримує щоразу в процесі встановлення тимчасового з'єднання.

Проте з погляду людини існування однієї тільки чисельної адресації виявилося незручним – з таким же успіхом ми могли б пронумерувати всі міста на планеті, але вони мають назви. Тому в Інтернеті були впроваджені домени з іменами, крапками, що розділяються (наприклад, URL: http://www.cnet.com).

Домен – це **група** комп'ютерів, що обслуговуються спільним сервером, який керує розподілом прав доступу користувачів до ресурсів мережі. Такий сервер називають контролером домену.

Доменне ім'я складається з **кількох частин** (імен доменів), розділених крапками. Рівень домену рахується з кінця, тобто справа наліво. Домен, ім'я якого зазначено праворуч, називають доменом першого (або верхнього) рівня (рис. 3.1).

Рисунок 3.1 − Домени першого рівня [3]

Щоб отримати інформаційні матеріали з Інтернету, адреси сервера **недостатньо**. Потрібна також адреса із зазначенням протоколу і унікального шляху до певного ресурсу. Таку адресу називають уніфікованим покажчиком ресурсу – URL (Uniform Resource Locator).

URL-адреса – це шлях до інтернет-ресурсу (документа, відео, вебсторінки, зображення тощо).

URL-адреса зазвичай містить **три** частини:

– назву протоколу, який використовується для доступу до ресурсу (http, ftp, news тощо);

– доменне ім'я або IP-адресу сервера, де зберігається файл;

– шлях до файла на сервері.

Протоколи передавання даних

Інтернет об'єднує комп'ютери в багатьох точках земної кулі. Усі ці комп'ютери мають різне апаратне забезпечення, на них встановлені різні операційні системи та програмне забезпечення. Але всі вони мають узгоджено й швидко приймати і передавати дані. Для цього у 70-х роках минулого століття почали розроблятися правила, згідно з якими відбувався обмін даними між комп'ютерами в мережі. Збірки таких правил одержали назву протоколи.

Мережеві протоколи − це **правила**, за якими здійснюється обмін даними між комп'ютерами.

Процес передавання даних від одного комп'ютера до іншого складається з декількох етапів (рівнів). Цей процес **передбачає такі операції**: отримання даних від користувача, їхнє стиснення, шифрування, формування пакетів, на які розбиваються повідомлення, встановлення сеансу зв'язку між комп'ютером, що передає дані, та тим, що їх приймає, транспортування даних по каналах зв'язку, вибір найбільш ефективного маршруту передавання даних, формування вихідного документа з пакетів даних.

На кожному з етапів **використовують окремі протоколи**, сукупність яких складає набір протоколів Інтернету, що має таку назву **TCP/IP**, що має таке тлумачення.

TCP (Transmission Control Protocol) – протокол керування передаванням, відповідає за організацію сеансу зв'язку між двома комп'ютерами у мережі.

IP (Internet Protocol) – міжмережний протокол, який відповідає за маршрутизацію, тобто за те, щоб пакет було доставлено за певною адресою. За допомогою протоколу TCP ПК перевіряє, чи всі частини отримано. Під час отримання всіх порцій TCP розміщує їх у потрібному порядку і збирає в одне ціле.

Найвідоміші протоколи, які використовують у мережі «Інтернет»

HTTP (Hyper Text Transfer Protocol) − протокол передачі гіпертексту. Використовують під час пересилання вебсторінок з одного комп'ютера на інший.

FTP (File Transfer Protocol) − протокол передачі файлів зі спеціального файлового сервера на комп'ютер користувача. Дає можливість абоненту обмінюватися двійковими і текстовими файлами з будь-яким комп'ютером мережі.

POP (Post Office Protocol) − стандартний протокол поштового з'єднання. Сервери POP опрацьовують вхідну пошту, а протокол POP призначено для опрацювання запитів на отримання пошти від клієнтських поштових програм.

SMTP (Simple Mail Transfer Protocol) − протокол, який задає набір правил для передавання пошти. Сервер SMTP повертає або підтвердження про прийом, або повідомлення про помилку, або запитує додаткову інформацію.

IRC (Unix to Unix Copy Protocol) − для забезпечення інтерактивного спілкування.

Telnet − протокол віддаленого доступу, що дає можливість працювати на будь-який ЕОМ мережі «Інтернет», як на своїй власній, тобто запускати

програми, змінювати режим роботи тощо. На практиці можливості обмежено тим рівнем доступу, який задано адміністратором віддаленої машини.

DTN − протокол, призначений для забезпечення наддалекого космічного зв'язку.

3.2 Адміністрування мережевих систем

Сучасні корпоративні інформаційні системи за своєю природою завжди є розподіленими системами. Робочі станції користувачів, сервери додатків, сервери баз даних та інші мережеві вузли розподілені по великій території. У крупній компанії офіси і промислові потужності сполучені різними видами комунікацій, що використовують різні технології і мережеві пристрої. Головне завдання мережевого адміністратора – забезпечити надійну, безперервну, продуктивну і безпечну роботу всієї цієї складної системи.

Розглядатимемо мережу як сукупність програмних, апаратних і комунікаційних засобів, що забезпечують ефективний розподіл обчислювальних ресурсів. Всі мережі можна умовно розділити на три категорії:

– локальні мережі (LAN, Local Area Network);

– глобальні мережі (WAN, Wide Area Network);

– міські мережі (MAN, Metropolitan Area Network).

Глобальні мережі дозволяють організувати взаємодію між абонентами на великих відстанях. Ці мережі працюють на дещо низьких швидкостях і можуть спричиняти значні затримки передачі інформації. Протяжність глобальних мереж може складати тисячі кілометрів. Тому вони так чи інакше інтегровані з мережами масштабу країни.

Міські мережі дозволяють взаємодіяти на територіальних утвореннях менших розмірів і працюють на швидкостях від середніх до високих. Вони менш уповільнюють передачу даних, чим глобальні, але не можуть забезпечити високошвидкісну взаємодію на великих відстанях. Протяжність міських мереж знаходиться в межах від декількох кілометрів до десятків і сотень кілометрів.

Локальні мережі забезпечують найвищу швидкість обміну інформацією між комп'ютерами. Типова локальна мережа займає простір в одну будівлю. Протяжність локальних мереж складає біля одного кілометра. Їх основне призначення полягає в об'єднанні користувачів (зазвичай однієї компанії або організації) для спільної роботи.

Механізми передачі даних у локальних і глобальних мережах істотно відрізняються. Глобальні мережі орієнтовані на з'єднання – до початку передачі даних між абонентами встановлюється з'єднання (сеанс). У локальних мережах використовуються методи, що не вимагають попередньої установки з'єднання, пакет із даними посилається без підтвердження готовності одержувача до обміну.

Окрім різниці в швидкості передачі даних, між цими категоріями мереж існують й інші відмінності. У локальних мережах кожен комп'ютер має мережевий адаптер, який сполучає його з середовищем передачі. Міські мережі містять активні комутуючі пристрої, а глобальні мережі зазвичай складаються з

груп могутніх маршрутизаторів пакетів, об'єднаних каналами зв'язку. Крім того, мережі можуть бути приватними або мережами загального користування.

Мережева інфраструктура будується з різних компонентів, які умовно можна рознести за такими рівнями:

– кабельна система і засоби комунікацій;

– активне мережеве устаткування;

– мережеві протоколи;

– мережеві служби;

– мережеве програмне забезпечення.

Кожен із цих рівнів може складатися з різних підрівнів і компонент. Наприклад, кабельні системи можуть бути побудовані на основі коаксіального кабелю («товстого» або «тонкого»), витої пари (екранованої і неекранованої), оптоволокна. Активне мережеве устаткування містить такі види пристроїв, як повторювачі (репітери), мости, концентратори, комутатори, маршрутизатори. У корпоративній мережі може бути використаний багатий набір мережевих протоколів: TCP/IP, SPX/IPX, NETBEUI, AppleTalk тощо.

Мережа дозволяє легко взаємодіяти один з одним самим різним видам комп'ютерних систем завдяки стандартизованим методам передачі даних, які дозволяють приховати від користувача все різноманіття мереж і машин.

Всі пристрої, що працюють в одній мережі, повинні спілкуватися на одній мові – передавати дані відповідно до загальновідомого алгоритму у форматі, який зрозумілий іншим пристроям. Стандарти – ключовий чинник під час об'єднання мереж.

Для більш строго опису роботи мережі розроблені спеціальні моделі. На сьогодні загальноприйнятими моделями є модель OSI (Open System Interconnection) і модель TCP/IP (або модель DARPA).

Перш ніж визначити завдання мережевого адміністрування в складній розподіленій корпоративній мережі, сформулюємо визначення терміну «корпоративна мережа» (КМ). Слово «корпорація» означає об'єднання підприємств, що працюють під централізованим управлінням і вирішують загальні завдання. Корпорація є складною, багатопрофільною структурою і внаслідок цього має розподілену ієрархічну систему управління. Крім того, підприємства, відділення і адміністративні офіси, що входять в корпорацію зазвичай розташовані на достатньому віддаленні один від одного. Для централізованого управління таким об'єднанням підприємств використовується корпоративна мережа.

Основне завдання КМ полягає в забезпеченні передачі інформації між різними додатками, використовуваними в організації. Під додатком розуміється програмне забезпечення, яке безпосередньо потрібне користувачеві, наприклад, бухгалтерська програма, програма обробки текстів, електронна пошта тощо. Корпоративна мережа дозволяє взаємодіяти додаткам, часто розташованим у географічно різних областях, і забезпечує доступ до них віддалених користувачів.

3.3 Моделі міжмережевої взаємодії (модель OSI, модель TCP/IP)

Моделі міжмережевої взаємодії призначені для формального і водночас наочного опису взаємодії мережевих вузлів між собою. На сьогодні найбільшого поширення набули і є стандартами для опису міжмережевої взаємодії **дві мережеві моделі**: **модель OSI і модель TCP/IP**. Обидві моделі розбивають процес взаємодії мережевих вузлів на декілька рівнів, кожен конкретний рівень одного вузла обмінюється інформацією з відповідним рівнем іншого вузла. Кожну з цих моделей можна подавати як об'єднання **двох** моделів:

– горизонтальної моделі (на базі протоколів, що забезпечує обмін даними одного типу між програмами і процесами, що працюють на одному і тому самому рівні на різних мережевих вузлах);

– вертикальної моделі (на основі послуг, що надаються сусідніми рівнями один одному на одному мережевому вузлі).

У горизонтальній моделі двом програмам, що працюють на різних мережевих вузлах, потрібний **загальний протокол для обміну даними**. У вертикальній − сусідні рівні обмінюються даними, виконуючи необхідні перетворення з використанням відповідних програмних інтерфейсів.

Модель OSI. У 1983 році з метою впорядкування опису принципів взаємодії пристроїв у мережах Міжнародна організація по стандартизації (International Organization for Standardization, ISO) запропонувала **семирівневу** еталонну комунікаційну модель «Взаємодія відкритих систем», модель OSI (Open System Interconnection).

Еталонна модель OSI зводить передачу інформації в мережі до семи відносно простих підзадач.

Модель OSI стала основою для розробки стандартів на взаємодію систем. Вона визначає **тільки схему** виконання необхідних завдань, і **не** дає конкретного опису їх виконання. Це описується конкретними протоколами або правилами, розробленими для певної технології з урахуванням моделі OSI. Рівні OSI можуть реалізовуватися **як** апаратно, так і програмно.

Основна ідея моделі OSI в тому, що одні й ті самі рівні на різних системах, не маючи можливості зв'язуватися безпосередньо, повинні працювати абсолютно **однаково.** Однаковим повинен бути і сервіс між **відповідним**и рівнями різних систем. Порушення цього принципу може привести до того, що інформація, надіслана від однієї системи до іншої, після всіх перетворень **не** буде ідентична початковій.

Модель OSI описує шлях інформації через мережеве середовище від однієї прикладної програми на одному комп'ютері до іншої програми на іншому комп'ютері. При цьому інформація, що пересилається, проходить вниз через всі рівні системи.

Рівні на різних системах **не** можуть спілкуватися між собою безпосередньо. Це вміє тільки фізичний рівень.

У міру проходження інформації вниз усередині системи вона набуде

вигляду, зручного для передачі по фізичних каналах зв’язку.

Для ідентифікації адресата до цієї перетвореної інформації додається заголовок з адресою. Після отримання адресатом цієї інформації, вона проходить через всі рівні вгору. У міру проходження інформація набуває первинного вигляду.

Кожен рівень системи повинен покладатися на послуги, надані йому суміжними рівнями:

1. Фізичний рівень. На цьому рівні виконується передача бітів по фізичних каналах (коаксіальний кабель, вита пара, оптоволокно).
2. Канальний рівень. Цей рівень визначає методи доступу до середовища передачі даних і забезпечує передачу кадру даних між будь-якими вузлами в мережах із типовою топологією за фізичною адресою мережевого пристрою. Адреси, які використовуються на канальному рівні в локальних мережах, часто називають MAC-адресами (MAC – «media access control»), що керують доступом до середовища передачі даних.
3. Мережевий рівень. Забезпечує доставку даних між будь-якими двома вузлами в мережі з довільною топологією, при цьому не гарантується надійна доставка даних від вузла-відправника до вузла-одержувача. На цьому рівні виконуються такі функції як маршрутизація логічних адрес мережевих вузлів, створення і ведення таблиць маршрутизації, фрагментація і збирання даних.
4. Транспортний рівень. Забезпечує передачу даних між будь-якими вузлами мережі з необхідним рівнем надійності. Для виконання цього завдання на транспортному рівні є механізми встановлення з’єднання між мережевими вузлами, нумерації, буферизації і впорядковування пакетів, переданих між вузлами мережі.
5. Сеансовий рівень. Реалізує засоби управління сесією, діалогом, а також надає засоби синхронізації в рамках процедури обміну повідомленнями, контролю над помилками, обробки транзакцій, підтримки виклику видалених процедур RPC.
6. Рівень уявлення. На цьому рівні можуть виконуватися різні види перетворення даних, такі як компресія і декомпресія, шифровка і дешифровка даних.
7. Прикладний рівень. Набір мережевих сервісів, що надаються кінцевим користувачам і додаткам. Приклади таких сервісів – обмін повідомленнями електронної пошти, передача файлів між вузлами мережі, додатки управління мережевими вузлами.

Функціонування перших трьох рівнів, фізичного, канального і мережевого, забезпечується переважно активним мережевим устаткуванням і зазвичай **реалізується** такими компонентами: мережевими адаптерами, репітерами, мостами, концентраторами, комутаторами, маршрутизаторами.

Модель TCP/IP

Модель TCP/IP називають також моделлю DARPA (скорочення від Defense Advanced Research Projects Agency – організація, в якій свого часу розроблялися мережеві проєкти, зокрема протокол TCP/IP, і яка стояла у витоків мережі «Інтернет») або моделлю Міністерства оборони США (модель DOD, Department of Defense, проєкт DARPA працював на замовлення цього відомства).

Модель TCP/IP розроблялася для опису стека (набору) протоколів TCP/IP (Transmission Control Protocol/Internet Protocol). Вона була розроблена значно раніше, ніж модель OSI.

Визначена послідовність і формат повідомлень на одному рівні, називаються протоколами. Ієрархічно організована сукупність протоколів називається стеком комунікаційних протоколів. Модель складається з **чотирьох** рівнів.

Переваги стека протоколів TCP/IP

Основна перевага стека протоколів TCP/IP в тому, що він забезпечує надійний зв'язок між мережевим устаткуванням від різних виробників.

Незалежність від мережевої технології: стек тільки визначає елемент передачі, дейтаграму, і описує спосіб її руху по мережі.

Загальна зв'язаність: стек дозволяє будь-якій парі комп'ютерів, які його підтримують, взаємодіяти один з одним. Кожному комп'ютеру призначається логічна адреса, а кожна дейтаграма містить логічні адреси відправника і одержувача. Проміжні маршрутизатори використовують адресу одержувача для ухвалення рішення про маршрутизацію.

Підтвердження. Протоколи стека забезпечують підтвердження правильності проходження інформації під час обміну між відправником і одержувачем.

Стандартні прикладні протоколи. Протоколи стека TCP/IP містять засоби підтримки основних задач комунікацій, таких як електронна пошта, передача файлів, віддалений доступ тощо.

Рівні моделі TCP/IP такі:

1. Рівень мережевого інтерфейсу не регламентований специфікаціями стека TCP/IP, і фактично до стека TCP/IP належать рівні з 1-го по 3-ій моделі TCP/IP. Цей рівень відповідає фізичному і канальному рівням моделі OSI.

2. Рівень міжмережевої взаємодії. На цьому рівні функціонує ціле сімейство протоколів. Основне завдання цього рівня – доставка пакетів від одного вузла-відправника до вузла-одержувача

Це завдання виконує протокол IP (Internet Protocol, протокол міжмережевої взаємодії). Протокол IP – базовий протокол стека TCP/IP і основний протокол мережевого рівня. Відповідає за передачу інформації по мережі. У його основі закладений дейтаграмний метод, який не гарантує доставку пакета.

Протокол ARP (Address Resolution Protocol, протокол розпізнання фізичних адрес) слугує сполучною ланкою між рівнем міжмережевої взаємодії і рівнем мережевого інтерфейсу. Він перетворить IP-адреса мережевих вузлів на фізичні MAC-адреса відповідних мережевих адаптерів. Протокол ARP припускає, що кожен пристрій знає як свою IP-адресу, так і свою фізичну адресу. ARP динамічно зв'язує їх і заносить у спеціальну таблицю, де зберігаються пари «IP-адрес – фізична адреса» (зазвичай кожен запис в ARP-таблиці має час життя 10 хвилин).

Протокол ICMP (Internet Control Message Protocol, протокол міжмережевих повідомлень) слугує для обміну інформацією про помилки. За допомогою спеціальних пакетів ICMP повідомляє мережевим вузлам інформацію про неможливість доставки пакета, про перевищення часу життя пакета тощо.

Протоколи RIP (Routing Internet Protocol) і OSPF (Open Shortest Path First) слугують для побудови таблиць маршрутизації і обчислення маршрутів під час відправлення пакетів між різними IP-мережами.

3. Транспортний рівень. Протокол TCP (Transmission Control Protocol, тобто протокол управління передачею) забезпечує, базуючись на протоколі IP, надійну передачу повідомлень між мережевими вузлами за допомогою утворення з'єднань (сеансів) між даними вузлами. Такі протоколи прикладного рівня, як HTTP і FTP, передають протоколу TCP свої дані для транспортування. Тому швидкісні характеристики TCP безпосередньо впливають на продуктивність додатків. Крім того, протокол TCP використовується для обробки запитів на вхід до мережі, розділення ресурсів тощо. На протокол TCP, зокрема, покладено завдання управління потоками і перевантаженнями. Він відповідає за узгодження швидкості передачі даних із технічними можливостями робочої станції-одержувача і проміжних пристроїв у мережі.

Протокол UDP (User Datagram Protocol, протокол дейтаграм користувача) забезпечує передачу прикладних пакетів дейтаграмним методом (що не гарантує доставку пакетів). Робота цього протоколу аналогічна IP, але основним його завданням є зв'язок мережевого протоколу і різних програмних додатків.

Мережні адреси (IPv4)

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

Розмір IPv 4-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 розв'язала проблему нераціонального використання адресного простору. Згідно з цією моделлю, увесь простір IP-адрес ділиться на 5 класів залежно від значення перших чотирьох біт IPv4-

адреси. Класам привласнені імена від A до E (табл. 3.1).

Перші 3 класи A, B і C використовуються для індивідуальної (unicast) адресації мереж і вузлів, клас D – для багатоадресного або групового (multicast) розсилання, клас E зарезервований для експериментів. Класи A, B і C мають різну довжину мережної частини адреси.

Для мереж класу A 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.

Оскільки **перший біт** ідентифікатора мережі завжди дорівнює нулю, то7 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 A рівно 126 ($2^7 – 2$). У кожній мережі класу A 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 – широкомовна адреса).

Таблиця 3.1 – Класи IP-адрес

Клас	Перші біти	Найменший номер мережі	Найбільший номер мережі	Максимальна кількість вузлів у мережі
A	0	1.0.0.0	126.0.0.0	2^{24} (поле 3 байти)
B	10	128.0.0.0	191.255.0.0	2^{16} (поле 2 байти)
C	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 A рівно 126 ($2^7 – 2$). У кожній мережі класу A можна адресувати до 16 777 214 ($2^{24} – 2$) вузлів. Дві адреси віднімаються внаслідок того, що вони використовуються в спеціальних цілях (рис. 3.2) і не можуть бути призначені пристрою (перший – адреса мережі, останній – широкомовна адреса).

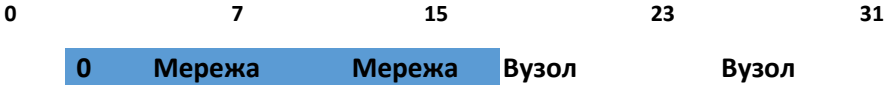


Рисунок 3.2 – Формат IPv 4-адреси класу A [3]

Мережі класу B (рис. 3.3) визначаються значеннями 10 у двох старших бітах адреси. Перші 2 байти в адресі використовуються для ідентифікатора

мережі, 2 байти, що залишилися – для ідентифікатора вузла. У результаті кількість доступних мереж класу B становить 16 384 (2^{14}) із кількістю вузлів у кожній мережі рівним 65 534 ($2^{16} – 2$).

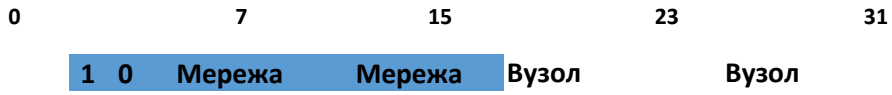


Рисунок 3.3 – Формат IPv 4-адреси класу B [3]

Для мереж 3-го класу (рис. 3.3) під ідентифікатор мережі приділяється три байти, в той час як під ідентифікатор вузла тільки 1 байт. Три старші біти першого октету **завжди рівні** 110, дозволяючи визначити, що адреса ставиться саме до класу C. Таким чином, одержуємо 2 097 152 (2^{21}) мереж, у кожній з яких перебуває 254 ($2^8 – 2$) вузла.

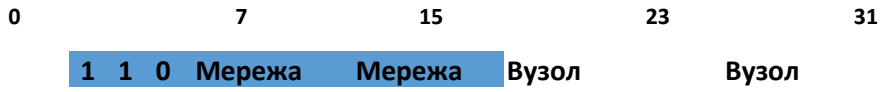


Рисунок 3.4 – Формат IPv 4-адреси класу C [3]

Мережі класу D (рис. 3.5) 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 3.5 – Формат IPv4-адреси класу D [3]

Мережі класу E (рис. 3.6) є експериментальними й у цей час не використовуються. Адреси в цьому класі визначаються значеннями 1 111 у перших чотирьох бітах.

Рисунок 3.6 – Формат IPv 4-адреси класу E [3]

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

Поняття маски IP-адреси

Із появою трирівневої ієрархії IPv4-адрес потрібні були додаткові методи, які дозволяли б визначити, яка частина адреси вказує на **ідентифікатор підмереж**і, 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. 3.7) – 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 3.7 – Формування маски підмережі [3]

Щоб одержати адресу мережі, знаючи IPv4-адресу й маску підмережі, необхідно застосувати до них операцію *логічне «І»* (рис. 3.8). Інакше кажучи, у тих позиціях IPv4-адреси, в яких у масці підмережі є двійкові 1, перебуває ідентифікатор мережі, а де двійкові 0 – ідентифікатор вузла.

Рисунок 3.8 – Одержання адреси мережі з IP-адреси й маски підмережі [3]

Для мереж класу A, B і C 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 $\mathbf{2}^{s}$, де s – кількість біт, зайнятих під ідентифікатор мережі з частини, відведеної під ідентифікатор вузла.

Кількість вузлів у кожної підмережі обчислюється за формулою $2^{n} - 2$, де n – кількість біт, що **залишилися** в частини, що ідентифікує вузол, а дві адреси – адреса підмережі й широкомовна адреса – у кожної отриманої підмережі зарезервовані.

Адресування на основі підмереж

Для більш ефективного використання адресного простору були внесені зміни в існуючу класову систему адресації. УВ RFC 950 була описана процедура розбивки мереж на підмережі, і в структуру IPv4-адреси був доданий ще один рівень ієрархії – *підмереж* (subnetwork). Поява ще одного рівня ієрархії (рис. 3.9) не змінило самої IPv4-адреси, вона залишилася 32-розрядною, а частина адреси, що раніше відводилася під ідентифікатор вузла, була розділена на 2 частини – ідентифікатор підмережі й ідентифікатор вузла.

Рисунок 3.9 – Трирівнева ієрархія IP-адреси [3]

Розбивка однієї великої мережі на більш дрібні (рис. 3.10) дозволяє:

– раціонально використовувати адресний простір (тобто виділити для сегмента мережі блок адрес не цілком класу A, B або C, а тільки частину класової мережі);

– підвищити безпеку й керованість мережі (за рахунок зменшення розмірів сегментів та ізоляції трафіка сегментів один одного).

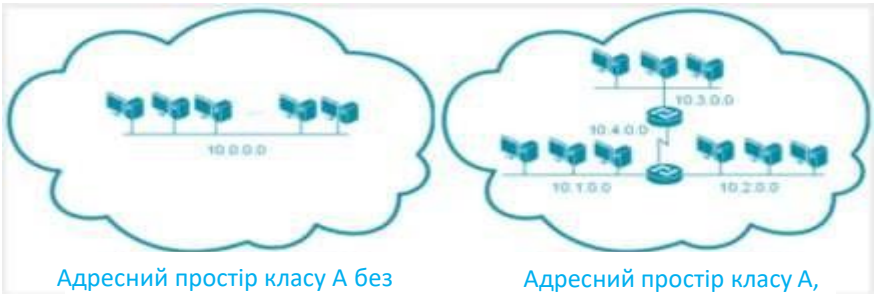


Рисунок 3.10 – Розбивка мережі на підмережі [3]

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

Контрольні запитання

1. Назвіть види адресацій.
2. Опишіть відмінність між доменною та IP-адресою комп'ютера?
3. Що називається мережевим протоколом?
4. Надайте характеристику мережевої моделі.
5. Опишіть модель OSI.
6. Назвіть рівні міжмережевої системи.
7. Які функції виконує модель TCP/IP?
8. Які переваги має стек протоколів TCP/IP?
9. Для чого потрібна фізична локальна адреса (MAC-адреса)?
10. Що таке мережні адреси (IPv4)?
11. Дайте визначення маски IP-адреси.

ТЕМА 4 Інтеграція та адміністрування баз даних

4.1 Адміністрування даних і адміністрування баз даних

Адміністрування даними передбачає виконання функцій адміністратора даних. Адміністратор даних відповідає за достовірність та повноту даних, що знаходяться у БД, їх узгодженість, а також виконання регламенту робіт з актуалізації БД [5].

Адміністрація бази даних позначає весь набір заходів, що виконуються адміністратором бази даних, щоб гарантувати, що база даних завжди доступна за потребою. Інші тісно пов'язані завдання та ролі – це безпека баз даних, моніторинг баз даних та усунення несправностей та планування подальшого зростання. Адміністрація бази даних є важливою функцією в будь-якій організації, яка залежить від однієї або декількох баз даних.

Адміністратор бази даних (DBA) зазвичай виділяє роль відділу IT для великих організацій. Однак багато менших компаній, які не можуть дозволити собі DBA на повний робочий день, зазвичай передають або контрактують роль спеціалізованому постачальнику, або об'єднують його з іншим у відділі ІКТ, щоб обидві виконувались однією особою.

Основна роль адміністрування баз даних полягає в забезпеченні максимального часу роботи бази даних, щоб вона завжди була доступна за необхідності. Зазвичай це передбачає активний періодичний моніторинг та усунення несправностей.

Адміністрація бази даних позначає весь набір заходів, що виконуються адміністратором бази даних, щоб гарантувати, що база даних завжди доступна за потреби. Інші тісно пов'язані завдання та ролі – це безпека баз даних,

моніторинг баз даних та усунення несправностей та планування подальшого зростання.

Адміністрування даними передбачає виконання функцій адміністратора даних. Адміністратор даних відповідає за достовірність та повноту даних, що знаходяться у БД, їхню узгодженість, а також виконання регламенту робіт з актуалізації БД.

Адміністрування бази даних передбачає виконання функцій адміністратора БД та інших адміністративних функцій, які забезпечують життєдіяльність системи бази даних.

Функції адміністратора баз даних:

1. Проєктування й розробка описів даних на різних рівнях та їхнє узгодження.

2. Розробка структур зберігання і стратегій доступу до даних згідно з вимогами ефективного (чи економічного) зберігання, швидкої обробки та виведення даних за бажанням користувача.

3. Реструктуризація та реорганізація БД у випадку зміни вимог до характеристик зберігання й обробки даних.

4. Проєктування та застосування механізмів захисту даних.

5. Реєстрація користувачів (імен, паролів), визначення їхніх прав доступу та повноважень.

6. Розробка й використання механізмів резервного копіювання даних та їхнє відновлення під час перебоїв.

7. Налаштування роботи БД (підвищення продуктивності механізмів обробки даних, підтримання планованої надлишковості, забезпечення ефективності їхнього зберігання).

8. Систематичне наглядання за використанням бази даних.

Адміністратор БД відповідає за забезпечення необхідного рівня продуктивності системи. Ці задачі вирішуються шляхом використання ефективних методів доступу, раціональною стратегією розміщення даних на носіях і оптимальною збитковістю даних. Адміністратор БД вирішує задачі, що пов’язані з вибором розміщення файлів на диску, визначенням необхідного об’єму дискової пам’яті, розподілом інформації на диску.

Адміністратор БД відповідає також за збір і обробку статистики функціонування системи, забезпечення ефективного використання ресурсів, за надійність функціонування системи, оцінку необхідності переналагодження середовища зберігання БД та її виконання, відновлення стану БД у разі порушення логічної і фізичної цілісності.

Реорганізація бази даних − діяльність адміністратора БД, яка забезпечує збір сміття, необхідні зміни для розміщення даних і, можливо, їхні структури згідно з проведеною модифікацією схеми зберігання. Реорганізація БД дозволяє підвищити продуктивність системи БД та/або більш ефективно використовувати ресурси простору пам’яті середовища зберігання.

Резервне копіювання (*backup*) виконується для запобігання можливого руйнування БД і, у разі потреби, відновлення даних. Виконується копіювання спеціальними утилітами. Резервуються не тільки дані, але і службова

інформація (журнали транзакцій, словники тощо). Резервна копія може бути точною копією БД, або архівною копією. Резервне копіювання може виконуватися під час роботи з БД або в інший час.

Тестування – процес виконання прикладних програм із метою пошуку помилок.

Експлуатація і супроводження полягають у промисловому використанні створеної системи, яке постійно супроводжується перевірками її поточних покажчиків функціонування, а також необхідною підтримкою.

Відновлення бази даних − усунення порушень логічної або фізичної цілісності БД із метою забезпечення її подальшого використання.

Цілісність бази даних логічна – це властивість стану БД, яка характеризується браком порушень всіх обмежень цілісності, які явно визначені в логічній схемі.

Цілісність бази даних фізична – це властивість стану БД, яка характеризується браком порушень специфікацій схеми зберігання, а також фізичних руйнувань даних на носіях інформації.

Головною одиницею відновлення в системах із БД є транзакції. Транзакція у разі успішного завершення виконує операцію Commit, а у разі невдачі − операцію Rollback. Під час виконання роботи з БД СУБД використовує буфер. *Буфер БД* − це ділянка основної пам'яті для тимчасового зберігання даних, яка призначена для прискорення операцій із диском.

Причиною необхідності відновлення бази даних є аварійне завершення роботи, відмова носіїв інформації, стихійні лиха, помилки прикладних програм, недбале поводження з базою даних.

Функції відновлення бази даних − резервне копіювання, засоби ведення журналу, створення контрольних точок, може відбуватися з використанням відкладеного оновлення або з використанням негайного оновлення.

У відновленні БД ключову роль відіграє журнал, в якому зберігаються зміни, внесені у БД, а також стан кожної транзакції.

Журнал − функціональний компонент СУБД, який забезпечує реєстрацію в процесі функціонування системи БД відомостей про виконання транзакцій, про працюючих користувачів, про застосування, що використовуються, про доступ до різних структур даних тощо.

Журнал забезпечує фіксацію всіх дій транзакцій: старт транзакції, зміну значень елементів БД, результатів або припинення роботи. Інформація журналу і зміни, які вносяться в елементи даних повинні виконуватися синхронно, але конкретний спосіб синхронізації залежить від обраного режиму протоколювання. Існують два основні методи відновлення БД: із відкладеним оновленням і негайним оновленням.

Відновлення з використанням *відкладеного оновлення* передбачає, що оновлення не заносяться у БД доти, поки транзакція не дасть команду фіксації зроблених змін. Якщо виконання транзакції було закінчено до досягнення цієї точки, то ніяких змін у БД виконано не буде, тому їхнє скасування не потрібне.

4.2 Поняття сутності, модель «сутність – зв’язок»

На початку проєктування баз даних зазвичай створюється модель предметної області, для якої створюється ця БД.

У ній вказуються типи об’єктів, що увійдуть до бази даних, і зв’язки між ними. Для наочності таку модель можна подати у графічному вигляді.

Сутність предметної області – це тип реального або уявного об’єкта предметної області.

Тип об’єкта предметної області називають сутністю.

У разі подальшої формалізації моделі словесний опис зв’язків між сутностями замінюють на їхні умовні позначення відповідного типу. За множинністю виділяють такі типи зв’язків:

1. Один до одного (позначається як 1 : 1), коли одному екземпляру однієї сутності відповідає один екземпляр іншої сутності. Для наведеного прикладу такий зв’язок є між сутностями «учитель» і «підручник з інформатики», якщо кожен учитель використовує тільки один підручник, який не використовують інші вчителі.

2. Один до багатьох (позначається як $1 : \infty$ або 1 : M, де M – від англ. *«many»* – «багато»), коли одному екземпляру однієї сутності відповідає кілька екземплярів іншої сутності.

3. Багато до одного (позначається як $\infty : 1$ або M : 1), коли кільком екземплярам однієї сутності відповідає один екземпляр іншої сутності. Цей тип зв’язку є протилежним до зв’язку один до багатьох.

4. Багато до багатьох (позначається як $\infty : \infty$ або M : M), коли кільком екземплярам однієї сутності можуть відповідати кілька екземплярів іншої сутності.

Зв’язки між сутностями класифікують також *за повнотою*. За цією класифікацією виділяють зв’язки, в яких:

1. Кожний екземпляр однієї сутності *обов’язково пов’язаний* з одним чи кількома екземплярами іншої сутності. Наприклад, зв’язок між сутностями «клас» і «учень» передбачає, що кожен учень належить до певного класу і кожен клас складається з певної групи учнів.

2. Кожен екземпляр однієї сутності *не обов’язково пов’язаний* хоча б із одним екземпляром іншої сутності. Наприклад, між сутностями «учень» і «комп’ютер» може бути встановлений зв’язок, який передбачає, що не кожен учень має власний комп’ютер.

Розглянуту модель предметної області називають моделлю «сутність – зв’язок» або ER-моделлю, чи ER-діаграмою (англ. *«Entity-Relationship»* – «сутність (об’єкт) – зв’язок (відношення)»). Під час створення ER-моделі використовуються спеціальні позначення типів сутності, властивостей екземплярів сутностей, зв’язків тощо. Набір таких умовних позначень називають нотацією (англ. *«notation»* – «позначення», «зображення умовними знаками»).

4.3 Адміністрування СУБД

СУБД (система управління баз даних) «бачить» адміністратора як користувача, який володіє певним набором привілеїв. Привілеї адміністратора дають йому можливість використовувати такі команди і утиліти СУБД і мати доступ до таких системних таблиць, які недоступні рядовим користувачам. Зазвичай СУБД надають у розпорядження адміністратора ще і спеціальний інструментарій.

У всіх СУБД розрізняються (хоча і називаються по-різному) **два рівні адміністрування**: системний адміністратор (адміністратор СУБД) і адміністратори бази даних (БД). Одна копія програмного продукту СУБД може підтримувати одночасне існування багатьох БД.

Різні БД можуть бути пов'язані з різними проєктами і навіть із різними організаціями, тому у кожної БД повинен **бути** свій адміністратор. Функції системного адміністратора належать до всієї системи загалом, його права і привілеї поширюються на всі об'єкти і на всіх суб'єктів у системі. Функції адміністратора БД належать не тільки до підмножини системних ресурсів, виділених конкретної БД, його права і привілеї поширюються на об'єкти, що належать до цієї БД, і на суб'єктів, що мають до них доступ.

Функції адміністратора:

– інсталяція СУБД;

– управління пам'яттю;

– управління розподілом даних між користувачами;

– копіювання і відновлення БД;

– управління безпекою у системі;

– передача даних між СУБД та іншими системами;

– управління продуктивністю.

Інсталяція СУБД є функцією **тільки** системного адміністратора. Розробники СУБД прагнуть максимально автоматизувати процес інсталяції і звести дії користувача в цьому процесі до мінімуму.

Управління пам'яттю. Дані в СУБД зберігаються на зовнішній пам'яті. Адміністратор повинен забезпечити таке виділення пам'яті, щоб з одного боку, її було досить для зберігання і ефективного доступу до даних, а з іншого – мінімальна кількість виділеної пам'яті залишалася невикористаною.

Управління розподілом даних між користувачами. Поділ даних між користувачами в разі їхньої паралельної роботи забезпечується автоматично засобами СУБД і підтримується засобами мови SQL. Однак за одночасної роботи незалежних додатків (іноді – в рамках однієї програми) можуть виникати конфлікти одночасного доступу. Адміністратор, маючи вичерпне уявлення про дисципліни поділу застосовуваних СУБД, виступає в ролі консультанта прикладних програмістів, зводячи до мінімуму взаємне блокування додатками один одного.

Копіювання і відновлення БД. Копіювання і відновлення є необхідними для гарантування збереження даних навіть у разі повного краху системи. Ця

частина функцій адміністратора містить роботу з відповідними утилітами СУБД і з протоколами транзакцій.

Управління безпекою даних захищає їх від несанкціонованих користувачів. Воно полягає в реєстрації користувачів у системі, виділення користувачам привілеїв і бюджетів.

Передача даних між СУБД та іншими системами. Дані, що зберігаються у БД, можуть знадобитися для використання в інших БД, що працюють в іншій інсталяції, або в додатках, що не залежать від СУБД. Для цілей перенесення даних у розпорядженні адміністрації є утиліти вивантаження даних у форматі, придатному для перенесення і, відповідно, завантаження даних, що надійшли з іншої системи.

Управління продуктивністю містить **три** аспекти: налаштування параметрів функціонування самої СУБД, окремих БД і окремих додатків. Перше забезпечується конфігурацією системи і використанням системних утиліт. Друге – складом і структурою компонентів БД (таблиць, індексів, тригерів тощо). Третє – вибором засобів розробки і оптимізацією формулювань запитів, тобто залежить переважно від прикладного програміста.

До функцій, які виконує адміністратор БД, належать також збір і аналіз статистики продуктивності роботи СУБД, управління цією продуктивністю.

4.4 Захист інформації у базах даних

Рисунок 4.1 − Засоби захисту інформації в базах даних [5]

Щойно користувач отримає право доступу до СУБД, йому автоматично надаються різні привілеї, пов’язані з його ідентифікатором. Ці привілеї можуть містити дозвіл на доступ до певних таблиць, подань, індексів, а також на певні операції над цими об’єктами: перегляд (SELECT), додавання (INSERT), оновлення (UPDATE), вилучення.

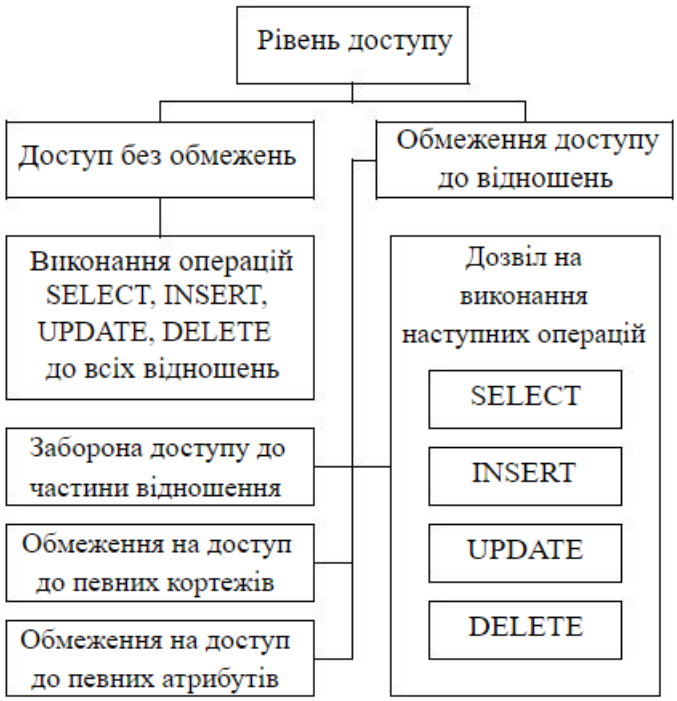


Рисунок 4.2 – Рівні доступу до бази даних [5]

У сучасних СУБД підтримуються два найбільш загальні підходи до забезпечення безпеки даних: вибірковий підхід і обов'язковий підхід.

Вибірковий підхід передбачає, що кожен користувач має різні права під час роботи з цими об'єктами. Різні користувачі можуть мати різні права доступу до одного й того ж об'єкта. Такий підхід – це однорівнева модель безпеки.

Обов'язковий підхід передбачає, що кожному об'єкту даних надається деякий кваліфікаційний рівень, а кожен користувач має деякий рівень допуску. У разі такого підходу правом доступу до певного об'єкта даних володіють тільки користувачі з відповідним рівнем доступу. Такий підхід – це багаторівнева модель безпеки.

Шифрування даних – метод забезпечення таємності даних шляхом генерації нового їх подання, яке допускає однозначне відновлення вихідного подання.

Система шифрування містить ключ шифрування, алгоритм шифрування, ключ дешифрування і алгоритм дешифрування.

Контрольні запитання

1. Які функції адміністратор бази даних?
2. У чому необхідність реорганізації баз даних?
3. Які причини відновлення бази даних?
4. У чому суть предметної області?
5. Опишіть рівні адміністрування СУБД.
6. Як захистити інформацію в базах даних?

ТЕМА 5 Засоби інтегрування інформаційних систем

5.1 Поняття інтегрування інформаційних систем, вимоги до процесу інтегрування інформаційних систем

Термін інтеграція має широке значення. Під ним можна розуміти об’єднання інформаційних систем, додатків, різних компаній або людей.

Інтеграція поділяється на зовнішню й внутрішню. Внутрішня інтеграція має на увазі об’єднання корпоративних додатків в одній організації (Enterprise Application Integration), а зовнішня – інтеграцію інформаційних систем організацій (Business-to-Business Application).

Існують чотири основні типові інтеграційні підходи:

– інтеграція на рівні даних;

– інтеграція на рівні бізнес-функцій та бізнесів-об'єктів;

– інтеграція на рівні бізнес-процесів;

– портали.

Інтеграція на рівні даних (Information-Oriented Integration) має на увазі наявність у системах баз даних, для роботи з якими необхідно розробити єдиний програмний інтерфейс [1].

До основних технологічних рішень цього підходу належать:

– системи реплікації даних;

– федеративні бази даних.

Реплікація є процесом синхронізації даних між різними джерелами. Необхідність у цьому виникає у момент зміни блоку інформації в розподілених системах зберігання, щоб гарантувати коректність і несуперечність даних, які використовуються у всіх модулях або додатках інформаційної системи. Звичайно функції реплікації покладають на проміжне програмне забезпечення.

Федеративні бази даних (Federated Database Systems) надають єдиний інтерфейс до розподілених даних. Це забезпечує інтеграцію безлічі автономних даних, які можуть бути фізично розташовані на різних пристроях у мережі. Такі бази даних прийнято називати віртуальними.

Інтеграція на рівні бізнес-функцій і бізнес об’єктів має на увазі реалізацію спільно використовуваних служб (сервісів). Служба може бути набором функцій, що використовується у декількох додатках. Набір служб і буде бізнесами-функціями.

Під час використання сервісно-орієнтованої архітектури, бізнес-функції можна розглядати як бізнес-сервіси, а в разі компонентного підходу – бізнес-об’єктами (бізнесами-компонентами).

Інтеграція на рівні бізнес-процесів розрізняється залежно від рівня інтеграції. Під час внутрішньої інтеграції взаємодіє велика кількість сервісів, а під час зовнішньої інтеграції переважно два. Самі бізнес-процеси функціонують над виділеними службами, для керування якими існує спеціальна інтерпретована мова.

Портали можна вважати графічними інтерфейсами бізнес-процесів, оскільки вони призначені для персоналізованого доступу до інформації й консолідації даних із декількох джерел.

Головне призначення процесу інтеграції – об'єднання функцій додатків або модулів для надання нової функціональності.

Під час інтеграції додатків можна виділити два основні типи завдань:

– завдання інтеграції корпоративних додатків;

– завдання інтеграції додатків із різних інформаційних систем.

Наступним кроком у розробці інтеграційних архітектур можна вважати появу корпоративної **сервісної шини** (Enterprise Service Bus – ESB).

Інтеграція − це процес об'єднання і спільної роботи інформаційних систем і програмних додатків (рис. 5.1). Серед основних підходів до ведення інтеграції інформаційних систем необхідно зазначити такі:

1. «Клаптева» інтеграція − «стихійна» інтеграція систем за браком єдиної інфраструктури. Цей підхід призводить до низької надійності й високої взаємодії систем, тому що:

– немає можливості формалізації та єдиного середовища виконання бізнес-процесів;

– здійснюється прив'язка до конкретних типів інтерфейсів і джерел даних.

2. На основі системи електронного документообігу використовується єдина інформаційна система для вирішення всіх завдань організації, що тягне за собою такі проблеми:

– нестабільна робота системи в разі серйозного навантаження;

– обмежені механізми автоматизації процесів;

– працює з даними певного і часто закритого формату;

– часто немає відкритих інтерфейсів і протоколів взаємодії з системою;

– накладає серйозні обмеження на розвиток інформаційної інфраструктури.

3. За допомогою єдиної інтеграційної шини ESB і сервера бізнес-процесів замовник може уникнути всіх перерахованих вище проблем і отримати в своє розпорядження такі переваги:

– високий ступінь стандартизації і необмежена розширюваність;

– готова інфраструктура інтеграції;

– вбудовані засоби розробки і виконання процесів;

– можливість моделювання і оцінки ключових параметрів продуктивності;

– підтримка версій процесу і можливість змінити логіку його роботи «на льоту» без зупинки середовища виконання.

Першим етапом інтеграції систем є створення вимог до результатів інтеграції та формулювання бізнес-вимог і правил інтеграції систем. Дотримання експлуатаційних вимог:

– відкриті стандарти взаємодії інформаційних систем;

– наскрізне управління і моніторинг;

– єдиний підхід до розвитку інфраструктури;

– широкий набір засобів для роботи з інтеграційними компонентами, що входять у технологічну платформу.

Способи інтеграції розрізняють за кількома ознаками.

За ступенем відособленості взаємозв'язків підсистем в інтегрованій системі (інакше за структурою інтеграції) розрізняють варіанти «точка – точка» і «зірка» (інтегруюча середа).

У варіанті «точка – точка» взаємодія підсистем здійснюється за схемою повного графа, тобто для кожної пари взаємодіючих підсистем створюється специфічний для них інтерфейсний зв'язок, наприклад, у вигляді конверторів даних із мови однієї підсистеми на мову іншої (таку схему стосовно до мереж зазвичай позначають *peer-to-peer* або p2p). Оскільки кількість таких дуплексних зв'язків може доходити до N (N – 1) / 2, де N – кількість підсистем, то варіант «точка – точка» виявляється прийнятним тільки для малих N. Підключення до системи кожної нової підсистеми виявляється досить трудомістким.

Нові підходи до інтегрування інформаційних систем враховують сучасні тенденції цифрової трансформації, індустрії 4.0 та інтелектуальних інформаційних систем, що є релевантним і для освітніх, і для промислових середовищ [1]. Інтеграція інформаційних систем забезпечує узгоджену роботу різнорідних програмних рішень, підвищує ефективність управління даними та підтримує цифрову трансформацію організацій.

Інтегрування інформаційних систем (ІС) є ключовим чинником цифрової трансформації організацій. Сучасні підходи до інтеграції спрямовані на підвищення гнучкості, масштабованості та сумісності різнорідних програмних і апаратних компонентів (табл. 5.1).

Зростання кількості цифрових сервісів, використання хмарних технологій, інтернету речей та штучного інтелекту вимагає нових підходів до інтегрування інформаційних систем.

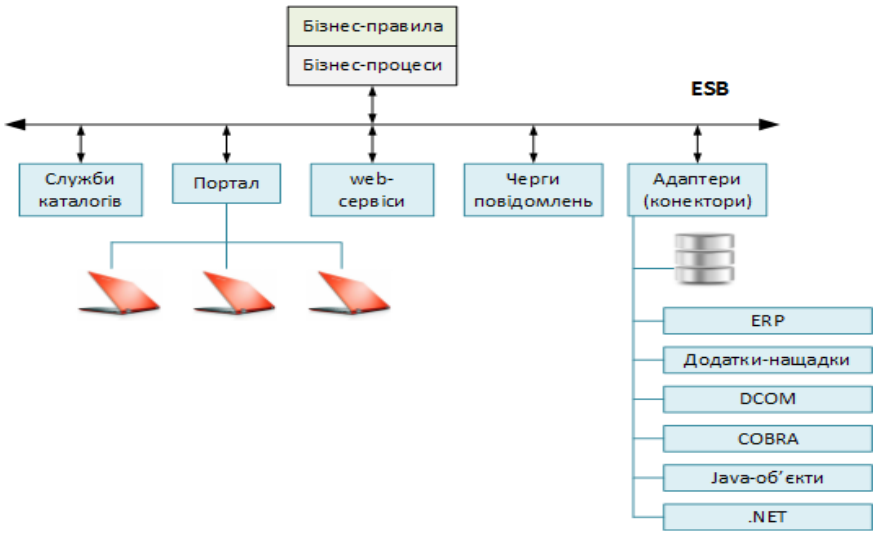


Рисунок 5.1 – Типова структура інтеграційної системи

Таблиця 5.1 – Передумови впровадження нових підходів

Проблеми традиційної інтеграції	**Сучасні потреби**
Жорстка зв’язаність систем	Гнучка та масштабована архітектура
Складність модифікації	Швидка адаптація до змін
Висока вартість підтримки	Оптимізація ресурсів

Нові підходи до інтегрування інформаційних систем орієнтовані на сервісність, подієвість та використання стандартних протоколів обміну даними (табл. 5.2).

Таблиця 5.2 – Характеристика сучасних підходів до інтегрування інформаційних систем

Підхід	**Коротка характеристика**	**Переваги**
SOA	Сервісно-орієнтована архітектура	Повторне використання сервісів
	Інтеграція через вебінтерфейси	Простота та універсальність
Мікросервіси	Декомпозиція системи на незалежні сервіси	Масштабованість
ESB	Інтеграційна шина даних	Централізоване управління
Event-driven	Подієва взаємодія систем	Реактивність

Хмарні та гібридні інтеграційні платформи (iPaaS) забезпечують швидке з’єднання локальних, хмарних та мобільних систем.

Приклад хмарної інтеграції: користувачі → вебдодатки, або мобільні додатки → API Gateway → хмарні сервіси → бази даних.

Інтелектуальна інтеграція інформаційних систем із використанням штучного інтелекту та машинного навчання дозволяє автоматизувати маршрутизацію даних, виявляти аномалії та оптимізувати інтеграційні процеси (табл. 5.3).

Таблиця 5.3 – Використання ШІ в інтеграції

Технологія ШІ	Призначення
Машинне навчання	Оптимізація потоків даних
NLP	Інтеграція неструктурованих даних
Інтелектуальні агенти	Автономна взаємодія систем

Ключовим є використання сервісно-орієнтованої та мікросервісної інтеграції SOA (Service-Oriented Architecture). Інтеграція відбувається через стандартизовані сервіси (SOAP/REST), що забезпечує повторне використання функціоналу і застосовується у великих корпоративних ІС (ERP, CRM, MES). Мікросервісна архітектура декомпозує системи на незалежні сервіси, асинхронно взаємодіє через API та брокери повідомлень. Переваги цього інтеграційного підходу у масштабованості, гнучкості, швидкому оновленні компонентів.

API-орієнтований підхід (API-First) застосовується у проєктуванні системи, він починається з контрактів API, використовує REST, GraphQL, gRPC, інтегрує внутрішні та зовнішні системи (B2B, B2C), поширений на вебплатформах, SaaS-рішеннях, освітніх платформах.

Подієво-орієнтована інтеграція (Event-Driven Architecture) виконує обмін даними через події в реальному часі, використовує message broker: Kafka, RabbitMQ, MQTT, забезпечує слабку зв'язаність компонентів, ефективна для проактивного управління, систем моніторингу, IoT-інтеграцій.

Інтеграція на основі даних (Data-Centric Integration) − об'єднання систем через спільні сховища: Data Warehouse, Data Lake, Lakehouse, ETL/ ELT-процеси. Використовується для аналітики, BI, ШІ та машинного навчання.

Інтелектуальна інтеграція з використанням ШІ застосовується для машинного навчання, семантичних моделей, онтологій, автоматично узгоджує формати даних, виявляє аномалії інтеграційних процесів, проактивно адаптує взаємодії між системами. Використання штучного інтелекту та машинного навчання дозволяє автоматизувати маршрутизацію даних, виявляти аномалії та оптимізувати інтеграційні процеси.

Low-Code / No-Code інтеграція використовує iPaaS-платформи: MuleSoft, Zapier, Power Automate, Make. Також вона мінімізує програмування, дозволяє швидке створення інтеграційних сценаріїв.

Хмарна та гібридна інтеграція – це інтеграція on-premise та cloud-систем із використанням контейнерів (Docker, Kubernetes), підтримкою мультихмарних середовищ, із забезпеченням безперервності бізнес-процесів. Хмарні інтеграційні платформи (iPaaS) забезпечують швидке з'єднання локальних, хмарних та мобільних систем. *Приклад хмарної інтеграції:* користувачі → вебдодатки або мобільні додатки → API Gateway → хмарні сервіси → бази даних.

Семантична та онтологічна інтеграція використовує RDF, OWL, knowledge graph, має єдине розуміння предметної області. Актуальна для освітніх систем, наукових платформ, експертних систем.

Інтеграція з урахуванням кібербезпеки (Secure-by-Design) – це Zero Trust Architecture, IAM, OAuth 2.0, OpenID Connect. Має захищені API-шлюзи, контроль доступу на рівні сервісів і даних.

Проактивна інтеграція − інтеграція не лише реактивна, а прогнозна, у ній системи ініціюють обмін даними до виникнення проблем, поєднують IoT, ШІ, аналітики в реальному часі (табл. 5.4).

Таблиця 5.4 − Нові підходи до інтегрування інформаційних систем

Підхід інтеграції	**Суть підходу**	**Переваги**	**Недоліки**	**Типові технології**	**Сфера застосування**
1	2	3	4	5	6
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

Продовження таблиці 5.4

1	2	3	4	5	6
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
Мікросервісна 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

5.2 Методи і технології реінжинірингу інформаційних систем

Реінжиніринг інформаційних систем (ІС) є комплексним підходом до радикального переосмислення та перебудови інформаційних процесів, архітектури програмного забезпечення і організаційних процедур із метою підвищення ефективності функціонування підприємства або організації. Основна ідея реінжинірингу полягає не у поступовому вдосконаленні існуючої системи, а у її суттєвому переосмисленні та створенні нової, більш ефективної структури інформаційної підтримки діяльності [6].

Реінжиніринг ІС широко використовується у випадках, коли наявні інформаційні системи морально застаріли, мають складну структуру, не підтримують сучасні технології або не відповідають потребам бізнесу. У таких ситуаціях проводиться глибокий аналіз існуючих процесів, їхня оптимізація та подальша реалізація у вигляді нової або модернізованої інформаційної системи.

Основні методи реінжинірингу інформаційних систем

У практиці розробки та модернізації інформаційних систем застосовується низка методів реінжинірингу, що дозволяють здійснювати аналіз, реконструкцію та оптимізацію систем:

1. Метод зворотного інжинірингу (Reverse Engineering). Цей метод передбачає дослідження існуючої інформаційної системи з метою відновлення її структури, функціональних залежностей і логіки роботи. Зворотний інжиніринг дозволяє отримати модель системи, навіть якщо документації немає або вона застаріла. У процесі аналізу визначаються модулі

програмного забезпечення, взаємозв'язки між ними, структури баз даних і алгоритми обробки даних.

2. Метод прямого інжинірингу (Forward Engineering). Після проведення аналізу і моделювання існуючої системи здійснюється створення нової інформаційної системи або її компонентів на основі сучасних технологій. Прямий інжиніринг містить проєктування архітектури, розробку програмного забезпечення, створення баз даних та інтеграцію системи з іншими інформаційними ресурсами.

3. Метод реструктуризації програмного забезпечення. Цей метод передбачає зміну внутрішньої структури програмної системи без зміни її функціональності. Основною метою є підвищення зрозумілості програмного коду, його модульності, а також полегшення подальшого супроводу та розвитку системи.

4. Метод рефакторингу. Рефакторинг є спеціалізованою формою реструктуризації програмного коду, що передбачає його покращення без зміни зовнішньої поведінки системи. У результаті підвищується якість програмного забезпечення, зменшується складність алгоритмів і покращується підтримка системи.

5. Метод повторного використання програмних компонентів (Reuse). У сучасних інформаційних системах значна увага приділяється повторному використанню програмних модулів, бібліотек та сервісів. Це дозволяє скоротити час розробки, знизити витрати та підвищити надійність системи.

6. Метод міграції системи. Міграція передбачає перенесення інформаційної системи або її компонентів на нову технологічну платформу, іншу операційну систему чи сучасні серверні середовища. Наприклад, перенесення локальної системи на хмарну платформу або модернізацію бази даних.

Технології реінжинірингу інформаційних систем

Окрім методів, у процесі реінжинірингу використовуються різноманітні технології, що забезпечують автоматизацію аналізу та проєктування інформаційних систем.

CASE-технології (Computer-Aided Software Engineering). CASE-засоби призначені для підтримки процесів аналізу, моделювання та проєктування інформаційних систем. Вони дозволяють створювати діаграми, моделі даних, описувати бізнес-процеси та автоматично генерувати програмний код. До популярних CASE-засобів належать Enterprise Architect, Rational Rose, Visual Paradigm.

Технології моделювання бізнес-процесів. Для аналізу та оптимізації діяльності організації застосовуються нотації моделювання, такі як BPMN, UML, IDEF0, DFD. Ці технології дозволяють формалізувати процеси, визначити вузькі місця системи та знайти шляхи їхньої оптимізації.

Технології сервісно-орієнтованої архітектури (SOA). SOA передбачає побудову інформаційної системи як набору незалежних сервісів, що взаємодіють між собою через стандартні інтерфейси. Такий підхід підвищує гнучкість системи та спрощує її модернізацію.

Хмарні технології. Сучасний реінжиніринг ІС часто пов’язаний із переходом до хмарної інфраструктури. Використання хмарних платформ (AWS, Microsoft Azure, Google Cloud) дозволяє масштабувати систему, підвищити її доступність та знизити витрати на обслуговування.

Технології контейнеризації та мікросервісної архітектури. Контейнеризація (Docker, Kubernetes) дозволяє розгортати програмні компоненти у стандартизованому середовищі, що значно спрощує процес модернізації та масштабування інформаційних систем.

Етапи застосування методів реінжинірингу

Застосування методів і технологій реінжинірингу зазвичай здійснюється у кілька послідовних етапів:

1. Аналіз існуючої інформаційної системи.
2. Моделювання бізнес-процесів та інформаційних потоків.
3. Виявлення недоліків та вузьких місць системи.
4. Проєктування нової архітектури інформаційної системи.
5. Реалізація та впровадження нових технологічних рішень.
6. Тестування та оптимізація системи.

Методи і технології реінжинірингу інформаційних систем відіграють важливу роль у модернізації інформаційної інфраструктури організацій. Їх застосування дозволяє підвищити ефективність бізнес-процесів, покращити якість обробки даних, зменшити витрати на підтримку програмного забезпечення та забезпечити адаптацію системи до сучасних вимог цифрової економіки.

Методи і технології **реінжинірингу інформаційних систем (ІС)** застосовуються у випадках, коли існуюча система вже не відповідає сучасним вимогам підприємства, має застарілу архітектуру або не забезпечує необхідної ефективності управління даними і бізнес-процесами. Реінжиніринг передбачає комплексне переосмислення структури, функцій і технологій інформаційної системи з метою підвищення її продуктивності, гнучкості та інтегрованості з новими цифровими рішеннями.

У сучасних організаціях інформаційні системи часто створювались протягом тривалого часу і містили різнорідні програмні модулі, різні платформи та бази даних. У таких умовах модернізація системи шляхом простих оновлень не завжди ефективна. Тому застосовується реінжиніринг, який передбачає глибокий аналіз існуючої системи, її функціональних можливостей і бізнес-процесів, а також розробку нової архітектури на основі сучасних інформаційних технологій.

Суть реінжинірингу інформаційних систем

Реінжиніринг інформаційних систем − це процес **аналізу, перепроєктування та модернізації існуючих програмних та інформаційних компонентів** із метою підвищення ефективності їхнього функціонування та відповідності новим вимогам бізнесу або організації.

Основними завданнями реінжинірингу є:

− підвищення продуктивності та надійності ІС;

− спрощення структури програмного забезпечення;

− усунення дублювання функцій і даних;

− підвищення гнучкості та масштабованості системи;

− інтеграція з новими інформаційними технологіями.

Реінжиніринг дозволяє зменшити витрати на підтримку системи та забезпечити її подальший розвиток без повного створення нового програмного продукту.

Типовий процес реінжинірингу інформаційної системи містить кілька послідовних етапів:

1. **Аналіз існуючої системи** − дослідження структури програмного забезпечення, архітектури та баз даних.

2. **Зворотна інженерія** − відновлення логіки роботи системи та створення її моделей.

3. **Реструктуризація та оптимізація** − вдосконалення програмного коду та структури даних.

4. **Перепроєктування системи** − створення нової архітектури та технологічної платформи.

5. **Реалізація та тестування** − упровадження модернізованої системи.

Реінжиніринг ІС тісно пов'язаний із **реінжинірингом бізнес-процесів (Business Process Reengineering, BPR)**. У цьому підході аналізуються існуючі процеси підприємства (модель **AS-IS**) і формується нова оптимізована модель (**TO-BE**). Основні етапи BPR:

– аналіз поточних бізнес-процесів;

– виявлення проблем та неефективних операцій;

– розробка нових процесів;

– інтеграція нових процесів у інформаційну систему.

Схема модернізації застарілої інформаційної системи наведена на рисунку 5.2.

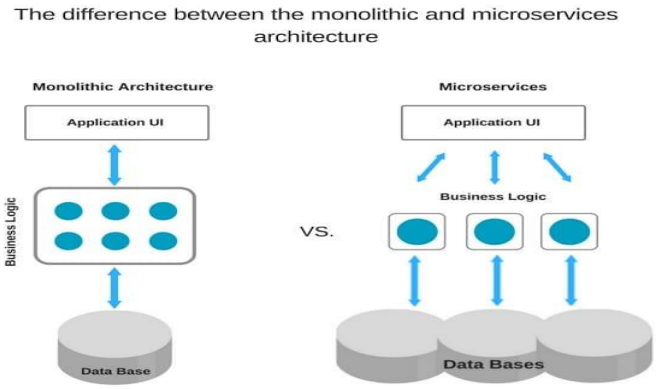


Рисунок 5.2 − Схема реінженерії [6]

Одним із поширених варіантів реінжинірингу є **модернізація застарілих систем (Legacy Systems)**. У цьому випадку стара монолітна система поступово трансформується у сучасну архітектуру, наприклад мікросервісну або хмарну.

Основні напрями модернізації: міграція баз даних; розподіл монолітної

архітектури на модулі або сервіси; використання API для інтеграції систем; перенесення системи у хмарну інфраструктуру.

Контрольні запитання

1. Опишіть вимоги до процесу інтегрування.
2. Назвіть типи інтеграційних процесів.
3. За якими ознаками розрізняють способи інтеграції ІС?
4. У чому суть нових підходів до інтеграції?
5. Як використовується ШІ у процесах інтеграції?
6. Назвіть основні методи реінжинірингу.
7. У чому полягає суть реінжинірингу ІС?

ТЕМА 6 Технологія віртуалізації

6.1 Поняття віртуалізації, основні типи віртуалізації

Віртуалізація – риса інформаційного суспільства. Глобальний інформаційний простір створює окремий світ зі своїми власними параметрами та законами у додаток до фізичного простору.

Цей світ динамічно розвивається, оскільки глобальність, мережевість, цифровізованість дають йому безпрецедентні можливості. Індивід і спільнота за такими умовами все частіше використовують його для реалізації своїх цілей, потреб, намірів, фактично «переселяючись» **із фізичного у віртуальний контекст**. Віртуалізацію спричинює розвиток технологій обробки інформації, здатних створити образ реальності, що майже **не** відрізняється від самої реальності та стає **гіперреальністю** (Жан Бодрійяр). Інформація в умовах віртуалізації легко **заміщує матерію** та стає середовищем життя людини [7].

Засобами віртуалізації є комп'ютерні технології, іміджмейкінг, мас-медіа, соціальний інжиніринг тощо. Її породженнями є кіберпротезування, створення віртуальних я-копій, ескейпізм, кіберпанк як життєва стратегія, шоу-політика, медіаподієвість тощо.

Засади віртуалізації .Технологія віртуалізації призначена для підтримання можливості одночасного запуску на одному комп'ютері декількох (зокрема різних) операційних систем (далі – ОС). Реалізація цієї можливості здійснюється за допомогою спеціальних програм – віртуальних машин (далі – ВМ). Віртуальна машина – це повністю ізольований програмний контейнер, здатний виконувати операційну систему і додатки як фізичний комп'ютер. Операційна система, що виконується на віртуальній машині, називаються гостьовою операційною системою (англ. «guest OS»), а платформа, що підтримує віртуальну машину, – хостом (англ. «host»). Операційна система хосту (англ. «host OS») – це операційна система фізичного комп'ютера, у якій виконується ВМ. Для успішного виконання віртуальної машини на хост-компьютері він завжди «ділиться» своїми реальними ресурсами, такими як пам'ять (оперативна, дискова та відео-) та процесор (виділяючи час роботи та

кількість ядер) для ВМ.

Системи віртуальних машин використовуються:

1) як полігон для тестування нових програм та операційних систем – перебої та віруси в них ніяк **не** відбиваються на працездатності хоста;

2) як оболонка для **серфін**гу по Інтернету – у ролі гостьової ОС запускається Linux зі своїм браузером; при цьому ймовірність проникнення шкідливих програм на хост-компьютер зводиться до мінімуму; перегляд сайтів; відвідування вебсайтів, пошук інформації в інтернеті;

3) для підвищення надійності, розподілу навантаження та масштабування серверів корпоративних інформаційних систем;

4) для тренування навичок і підвищення професійного рівня системних адміністраторів;

5) інших аналогічних цілей.

Віртуалізація – це процес, що використовується для створення віртуального середовища. Це дозволяє користувачеві одночасно запускати кілька операційних систем на одному комп'ютері. Це створення віртуальної (а не фактичної) версії чогось, наприклад, операційної системи, сервера або мережевих ресурсів тощо. Для багатьох компаній віртуалізацію можна розглядати як частину загальної тенденції в IT-середовищах, які здатні керувати собою на основі сприйнятої діяльності та корисних обчислень. Найважливіша мета віртуалізації – зменшити адміністративні завдання, покращивши масштабованість та навантаження. Коротше кажучи, віртуалізація абстрагує обчислювальну функціональність пристрою від його фізичного обладнання.

Переваги віртуалізації:

– ефективне використання обчислювальних ресурсів;

– скорочення витрат на інфраструктуру;

– скорочення витрат на програмне забезпечення;

– підвищення гнучкості та швидкості реагування системи;

– новий метод управління IT-інфраструктурою, що допомагає IT-адміністраторам витрачати менше часу на виконання повторюваних завдань – наприклад, на ініціацію, налаштування, відстежування і технічне обслуговування;

– можливість несумісних додатків працювати на одному комп'ютері;

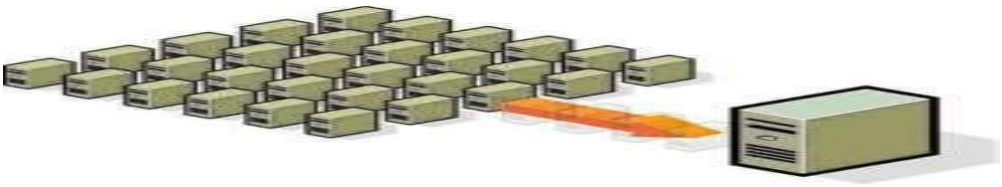
– підвищення доступності додатків і забезпечення безперервності роботи підприємства;

– можливості легкої архівації.

Підвищення керованості інфраструктури – використання централізованого управління віртуальною інфраструктурою.

На рисунку 6.1 показані відмінності класичної архітектури комп'ютера від архітектури, що містить віртуальні машини (ВМ).

Гостьові системи і хостові ОС працюють одночасно, обмінюються даними і беруть участь у мережевій взаємодії не тільки з хостовою ОС, але і з зовнішньою щодо фізичного комп'ютера мережею.



а) б)

Рисунок 6.1 – Порівняння віртуального і фізичного комп'ютера:
а – віртуальна машина; б – класична архітектура

Основні особливості віртуальних машин

Сумісність. Віртуальні машини зазвичай сумісні з усіма стандартними комп'ютерами. Як і фізичний комп'ютер, віртуальна машина працює під управлінням **власної** гостьової операційної системи і **виконує** власні додатки. Вона містить **усі** компоненти, стандартні для фізичного комп'ютера (материнську плату, відеокарту, мережевий контролер тощо). Віртуальні машини повністю сумісні з усіма стандартними операційними системами, програмами та драйверами пристроїв.

Ізольованість. Віртуальні машини повністю ізольовані одна від одної так, як і фізичні комп'ютери.

Інкапсуляція. Віртуальні машини повністю інкапсулюють обчислювальне середовище. Віртуальна машина – це програмний контейнер, що зв'язує «інкапсулюючий» повний комплект віртуальних апаратних ресурсів, а також ОС і всі її додатки у програмному пакеті.

Незалежність від устаткування. Віртуальні машини повністю незалежні від базового фізичного обладнання, на якому вони працюють.

Гіпервізор. Віртуалізація – **ілюзія** присутності кількох окремих комп'ютерів, тобто **віртуальних машин**, на тому самому фізичному обладнанні.

Віртуальна машина (VM) – програмна реалізація машини (комп'ютера), яка виконує програми подібно до справжньої машини.

Створюється ця ілюзія за допомогою **гіпервізору** (рис. 6.2).

Гіпервізори прийнято поділяти на **два** типи. Але є ще й гібридний гіпервізор, який поєднує властивості обох типів:

Рисунок 6.2 – Типи гіпервізорів

До **першого типу** гіпервізорів належать Xen, VMware ESXi, Hyper-V тощо. Гіпервізори **першого типу**, які іноді ще називають **нативними** або металевими, працюють прямо **на** апаратному забезпеченні хоста, щоб контролювати його та керувати гостьовими машинами.

Цей гіпервізор **не** потребує додаткового програмного забезпечення для процесу інсталяції і безпосередньо працює на апаратних ресурсах хост-машини.

Популярні гіпервізори **другого типу** − Oracle VirtualBox, VMware Workstation, KVM. Oracle VM VirtualBox – модульний кросплатформний гіпервізор для операційних систем Linux, MacOS, Microsoft Windows, FreeBSD, Solaris / OpenSolaris, ReactOS, DOS тощо від корпорації Oracle.

Гіпервізор другого типу працює **як** один із процесів, що виконуються основною ОС, найчастіше – Linux. Повноваження гіпервізору другого типу значно скромніше: він керує **гостьовими операційними** системами, а емуляцію і управління фізичними ресурсами здійснює хостова ОС.

6.2 Огляд платформ віртуалізації

Поняття віртуалізації умовно можна розділити на **дві** категорії: – **віртуалізація платформ та віртуалізація ресурсів.**

Продуктом **віртуалізації платформи** є **віртуальні машини** – деякі програмні абстракції, що запускаються на **платформі** реальних апаратно-програмних систем.

Є кілька видів віртуалізації платформ, у кожному з яких здійснюється свій підхід до поняття «віртуалізація». Види віртуалізації платформ залежать від того, наскільки повно здійснюється симуляція апаратного забезпечення. Повна емуляція (симуляція).

За такого вигляду віртуалізації віртуальна машина повністю віртуалізує все апаратне забезпечення зі збереженням гостьової операційної системи **в незмінному** вигляді. Такий підхід дозволяє **емулювати** різні апаратні архітектури. Довгий час такий вид віртуалізації використовувався, щоб розробляти програмне забезпечення для нових процесорів ще до того, як вони були фізично доступними. Такі емулятори також застосовують для низькорівневого налагодження операційних систем. Основний мінус цього підходу полягає в тому, що емульоване апаратне забезпечення істотно уповільнює швидкодію гостьової системи, що робить роботу з нею дуже незручною, тому, крім як для розробки системного програмного забезпечення, а також освітніх цілей, такий підхід мало де використовується.

Часткова емуляція (нативна віртуалізація). У цьому випадку віртуальна машина віртуалізує лише необхідну кількість апаратного забезпечення, щоб вона могла бути запущена ізольовано. Такий підхід дозволяє запускати гостьові операційні системи, розроблені тільки для тієї ж архітектури, що і у хоста. Таким чином, кілька примірників гостьових систем можуть бути запущені одночасно. Цей вид віртуалізації дозволяє істотно збільшити швидкодію гостьових систем порівняно з повною емуляцією і на сьогодні широко використовується. Також, із метою підвищення швидкодії, у платформах віртуалізації, що використовують цей підхід, застосовується

спеціальний «прошарок» між гостьовою операційною системою і устаткуванням (гіпервізор), що дозволяє гостьовий системі безпосередньо звертатися до ресурсів апаратного забезпечення.

Віртуалізація ресурсів. Цей вид віртуалізації має на меті **комбінування або спрощення подання** апаратних ресурсів для користувача і здобуття користувальницьких абстракцій обладнання, просторів імен, мереж тощо.

Система, що надає апаратні ресурси і програмне забезпечення, називається хостовою *(host*), а системи, що симулюються нею – **гостьовими** *(guest).* Щоб гостьові системи могли стабільно функціонувати на платформі хостової системи, необхідно, щоб програмне і апаратне забезпечення хосту було досить **надійним і надавало** необхідний набір інтерфейсів для доступу до його ресурсів.

Є кілька **видів** віртуалізації платформ. Види віртуалізації платформ залежать від того, наскільки повно здійснюється симуляція апаратного забезпечення.

Віртуалізація ресурсів. Під час опису віртуалізації платформ ми розглядали поняття віртуалізації у вузькому сенсі, переважно застосовуючи його до процесу створення віртуальних машин. Однак, якщо розглядати віртуалізацію в широкому сенсі, можна прийти до поняття віртуалізації ресурсів, що узагальнює підходи до створення віртуальних систем. **Віртуалізація ресурсів дозволяє концентрувати, абстрагувати і спрощувати управління групами ресурсів, таких як** мережі, сховища даних і простору імен.

Види віртуалізації ресурсів: об’єднання, агрегація і концентрація компонентів. Під таким видом віртуалізації ресурсів розуміється організація декількох фізичних або логічних об’єктів у пули ресурсів (групи), що надають зручні інтерфейси користувачеві. Приклади такого виду віртуалізації: багатопроцесорні системи, що подаються як одна потужна система, RAID-масиви та засоби керування томами, що комбінують декілька фізичних дисків в один логічний, віртуалізація систем зберігання, використовувана під час побудови мереж зберігання даних SAN (Storage Area Network), віртуальні приватні мережі (VPN) і трансляція мережевих адрес (NAT), що дозволяють створювати віртуальні простори мережевих адрес та імен.

Кластеризація комп’ютерів та розподілені обчислення (grid computing). Цей вид віртуалізації містить технології, які застосовуються в разі об’єднання безлічі окремих комп’ютерів у глобальні системи (метакомп’ютери) для спільного вирішення загальної задачі.

Поділ ресурсів (partitioning). Під час поділу ресурсів у процесі віртуалізації відбувається поділ одного великого ресурсу на кілька однотипних об’єктів, зручних для використання. У мережах зберігання даних це називається зонуванням ресурсів («zoning»). Інкапсуляція. Багатьом це слово відомо як приховування об’єктом всередині себе своєї реалізації. Можна сказати, що віртуалізація – це процес створення системи, яка надає користувачеві зручний інтерфейс для роботи з нею і приховує подробиці складності своєї реалізації**.** **Наприклад**, використання центральним

процесором кешу для прискорення обчислень не відбивається на його зовнішніх інтерфейсах.

Віртуалізація ресурсів, на відміну від **віртуалізації платформ**, має більш широкий і розпливчастий сенс і має багато різних підходів, спрямованих на підвищення зручності звернення користувачів із системами загалом. Тому далі ми будемо спиратися переважно на поняття віртуалізації платформ, оскільки технології, пов'язані саме з цим поняттям, є на сьогодні найбільш динамічно розвиненими й ефективними. На рисунках 6.1, а і 6.1, б показані відмінності класичної архітектури комп'ютера від архітектури, що містить віртуальні машини (ВМ). Гостьові системи й хостові ОС працюють одночасно, обмінюються даними і беруть участь у мережевій взаємодії не тільки з хостовою ОС, але і з зовнішньою щодо фізичного комп'ютера мережею.

Гіпервізор – це програма, яка керує фізичними ресурсами обчислювальної машини та розподіляє ці ресурси між декількома різними операційними системами, дозволяючи запускати їх одночасно.

Інакше кажучи, **гіпервізор створює з одного фізичного комп'ютера кілька копій, клонів апаратних ресурсів**, і кожен клон видно з боку користувача як окремий пристрій. На кожну віртуальну машину можна встановити гостьову операційну систему користувача, яка не прив'язана до «заліза» хосту.

Гіпервізор ізолює запущені ОС одну від одної так, щоб кожна з них монопольно використовувала виділені їй ресурси. Але за необхідності гіпервізор дозволяє операційним системам віртуальних машин взаємодіяти між собою. Механізмом зв'язку між ОС може бути загальний **доступ** до певних файлів та **обмін** даними по локальній мережі.

6.3 Методи, технології, інструменти, переваги віртуалізації

За особливостями реалізації можна виділити апаратну і програмну види віртуалізації.

У кожному окремому випадку неможливо обійтися без гіпервізору. Ця функція застосовується за різних видів віртуалізації, вона спрямована на створення та управління безліччю віртуальних машин (ВМ) на сервері. У якості гіпервізору може виступати як звичайна ОС, так і спеціальна.

Особливості апаратної віртуалізації

Під час реалізації цієї технології на сервер-хості необхідно встановити звичайну ОС та створити ізольовані віртуальні машини, кожна з яких мала б повноцінну операційну систему й використовувала б у роботі її ядро. Зі свого боку, апаратна віртуалізація ділиться на підтипи:

1. Повна. Так називається методика, за якої повною мірою імітується базове устаткування, а гостьове ПЗ не потребує змін.

2. Емуляція. У цьому підтипі віртуальна машина імітує потрібне для її роботи обладнання, завдяки чому стає незалежною. Гостьова ОС також не вимагає змін.

3. Паравіртуалізація. У цьому випадку гостьове ПО запускає власні автономні домени, а апаратне не моделюється.

Серед переваг особливо варто відзначити повноцінний поділ енергоресурсів сховища. Одна віртуальна машина не впливає на швидкість і споживану потужність інших.

Але водночас у апаратної віртуалізації є також мінуси. Працює вона повільніше, ніж програмна. Це зумовлено тим, що для її реалізації потрібно емулювати функції всієї апаратури на сервер-хості й контролювати роботу ОС на кожній віртуальній машині.

Вирішити проблему з прискоренням процесу можна шляхом застосування паравіртуалізації. Її принцип дії заснований на тому, що ОС «знає», що вона працює у віртуальній машині, може залучити деякі функції, які забезпечує гіпервізор. Із цієї причини відпадає необхідність в емулюванні роботи ВМ. Однак паравіртуалізація з боку гостьових ОС підтримується лише в системах, у яких відкрито вихідний код. До таких, наприклад, належить Linux.

Особливості програмної віртуалізації

Ця технологія може бути застосована на Windows та Android. Поділ ресурсів на сервері виконується засобами ОС, а для забезпечення роботи віртуальних машин використовується загальне програмне ядро. Серед важливих вимог такої технології віртуалізації варто відзначити те, що гостьова операційка повинна бути ідентична з тією, яка застосовується на хост-системі.

Класифікується ця технологія за такими видами:

1. Віртуалізація ОС. За такої технології можна розміщувати декілька операційних систем на власній.

2. Віртуалізація додатків. У цьому випадку окремі додатки розташовуються ізольовано від власної операційки.

3. Віртуалізація послуг. Це рішення передбачає надання процесам і сервісам, які пов’язані з певним додатком, окремого хостингу.

Переваги технології полягають у легкості налаштувань і високій швидкості створення віртуальних машин. Але на тлі цього є недолік. Він виражений у тому, що за такої технології немає можливості реалізувати повноцінний поділ ресурсів. Через це навантаження на віртуальні машини розподіляється нерівномірно.

Крім класифікації за способом реалізації (на апаратну і програмну), віртуалізація також може застосовуватися щодо окремих процесів: пам’яті, мережі, робочого столу, сховища та даних. Використовують технології також для оптимізації роботи системи й розширення фізичного простору.

Віртуалізація – це зручна технологія, за допомогою якої вдається переносити обладнання для забезпечення роботи системи на зовнішній підряд. Завдяки цьому можна легко усунути енерговитрати й оптимізувати роботу.

Контрольні запитання

1. Надайте визначення поняттю віртуалізація.
2. Для чого призначено віртуалізацію?
3. Які переваги має віртуалізація?
4. Що таке віртуальна машина?

5. Як працює віртуальна машина?
6. Дайте визначення гіпервізору, назвіть типи гіпервізорів.
7. Опишіть апаратну віртуалізацію.
8. Опишіть програмну віртуалізацію.

ТЕМА 7 Адміністрування процесу конфігурації інформаційних систем

7.1 Поняття та значення конфігурації інформаційних систем

Сучасні інформаційні системи (ІС) є складними програмно-апаратними комплексами, які містять програмне забезпечення, апаратні ресурси, мережеву інфраструктуру, бази даних та різні сервіси [1]. Для забезпечення стабільної роботи таких систем необхідно правильно організувати **процес конфігурації**.

Конфігурація інформаційної системи – це процес визначення, налаштування та управління параметрами програмних, апаратних і мережевих компонентів системи з метою забезпечення її коректної роботи, безпеки та ефективності.

Процес конфігурації інформаційних систем є важливою складовою управління ІТ-інфраструктурою. Правильна конфігурація дозволяє забезпечити стабільну роботу системи, підвищити її продуктивність та безпеку. Використання сучасних технологій автоматизації, таких як Infrastructure as Code та системи управління конфігурацією, значно спрощує процес адміністрування та підвищує ефективність експлуатації інформаційних систем.

Процес конфігурації є складовою **керування життєвим циклом інформаційної системи** та тісно пов’язаний із адмініструванням, підтримкою та модернізацією ІТ-інфраструктури.

Основні цілі конфігурації ІС:

– забезпечення стабільної роботи системи;
– оптимізація продуктивності;
– підвищення рівня інформаційної безпеки;
– забезпечення масштабованості системи;
– підтримка відповідності бізнес-процесам організації.

У сучасних організаціях конфігурація ІС здійснюється не лише на етапі впровадження системи, а й постійно в процесі її експлуатації, що пов’язано з оновленнями програмного забезпечення, змінами інфраструктури та вимог користувачів.

Етапи процесу конфігурації інформаційної системи

Процес конфігурації ІС містить кілька послідовних етапів.

1. Аналіз вимог до системи. На цьому етапі визначаються:

– функціональні вимоги;
– технічні параметри системи;
– вимоги до продуктивності;
– вимоги до безпеки та надійності.

Результатом етапу є формування **специфікації конфігурації системи**.

2. Визначення компонентів конфігурації. До основних елементів конфігурації належать:

– серверне обладнання;

– операційні системи;

– мережеві компоненти;

– програмне забезпечення;

– бази даних;

– системи безпеки.

Кожен компонент описується у вигляді **конфігураційних елементів (Configuration Items – CI)**.

3. Налаштування параметрів системи. На цьому етапі здійснюється:

– встановлення програмного забезпечення;

– налаштування серверів;

– конфігурація мережі;

– налаштування доступу користувачів;

– оптимізація продуктивності.

4. Тестування конфігурації. Після налаштування система проходить тестування для перевірки:

– стабільності роботи;

– відповідності вимогам;

– продуктивності;

– безпеки.

5. Документування конфігурації. Документування дозволяє:

– швидко відновити систему у разі збою;

– забезпечити підтримку системи;

– спростити модернізацію ІС.

Основні завдання конфігурації ІС

Процес конфігурації інформаційних систем передбачає вирішення низки важливих завдань:

1. Управління конфігураційними елементами. Необхідно вести облік усіх компонентів системи:

– програмних модулів;

– серверів;

– мережевого обладнання;

– версій програмного забезпечення.

2. Управління змінами. Будь-які зміни в конфігурації повинні контролюватися, щоб уникнути порушення роботи системи.

3. Забезпечення сумісності компонентів. Важливо забезпечити коректну взаємодію між:

– операційними системами;

– прикладними програмами;

– базами даних;

– мережевими сервісами.

4. Забезпечення безпеки системи. Конфігурація повинна містити:

– управління правами доступу;

– налаштування політик безпеки;

– контроль мережевого доступу.

5. Підтримка продуктивності системи. Конфігурація повинна враховувати:

– балансування навантаження;

– оптимізацію ресурсів;

– масштабованість системи.

7.2 Проблеми конфігурації інформаційних систем

У процесі конфігурації ІС можуть виникати різні проблеми:

1. Складність ІТ-інфраструктури. Сучасні системи можуть містити:

– десятки серверів;

– хмарні сервіси;

– мікросервісну архітектуру;

– розподілені бази даних.

Це значно ускладнює процес управління конфігурацією.

2. Часті зміни програмного забезпечення. Оновлення програм та систем безпеки можуть призводити до конфліктів конфігурацій.

3. Людський фактор. Помилки адміністраторів часто є причиною:

– неправильних налаштувань;

– втрати даних;

– збоїв системи.

4. Брак централізованого управління. Якщо конфігурація здійснюється вручну на кожному сервері, це призводить до:

– неузгодженості параметрів;

– складності адміністрування;

– високого ризику помилок.

7.3 Технології конфігурації інформаційних систем

Для автоматизації конфігурації ІС застосовуються спеціальні технології:

1. Configuration Management (CM). Це підхід до управління конфігураціями, який передбачає:

– контроль версій;

– автоматизацію налаштувань;

– документування змін.

Він широко використовується у DevOps-середовищах.

2. Infrastructure as Code (IaC). IaC передбачає опис інфраструктури у вигляді програмного коду. Переваги:

– автоматизація налаштувань;

– повторюваність конфігурацій;

– швидке розгортання систем.

Популярні інструменти: **Ansible, Puppet, Chef, Terraform.**

3. Контейнеризація. Контейнерні технології дозволяють запускати програмні компоненти у стандартизованому середовищі.

Приклади технологій: Docker, Kubernetes.

Це значно спрощує конфігурацію та масштабування систем.

4. Системи управління конфігурацією. До них належать:

– системи контролю версій (Git);

– системи автоматизації розгортання;

– системи моніторингу.

7.4 Оцінка ефективності конфігурації ІС

Ефективність конфігурації інформаційної системи оцінюється за кількома показниками:

1. Продуктивність системи визначається швидкістю обробки даних та реакції системи.

2. Надійність характеризується стабільністю роботи та кількістю відмов.

3. Безпека оцінюється рівнем захисту від несанкціонованого доступу та атак.

4. Масштабованість – можливість збільшення ресурсів системи без значних змін конфігурації.

5. Керованість системи визначає, наскільки легко адмініструвати систему.

Практичні рекомендації щодо конфігурації ІС

Для ефективної конфігурації інформаційних систем рекомендується:

1. Використовувати автоматизацію. Автоматизація дозволяє мінімізувати людські помилки та пришвидшити розгортання систем.

2. Документувати конфігурації. Документація повинна містити:

– опис архітектури системи;

– параметри налаштування;

– процедури відновлення.

3. Використовувати системи контролю версій. Це дозволяє відстежувати всі зміни конфігурації.

4. Впроваджувати моніторинг системи. Моніторинг дозволяє:

– виявляти проблеми на ранніх етапах;

– оптимізувати ресурси;

– забезпечувати стабільність системи.

5. Використовувати стандарти управління ІТ, зокрема ITIL, ISO/IEC 20000, COBIT.

6. Забезпечення безпеки. Конфігурація повинна містити:

– контроль доступу;

– політики безпеки;

– захист від несанкціонованого доступу.

7. Інтеграція з іншими системами. Сучасні ІС працюють у складному цифровому середовищі та повинні інтегруватися з:

– ERP-системами;
– CRM-системами;
– вебсервісами;
– хмарними платформами.

8. Автоматизована конфігурація. Використання спеціальних інструментів для автоматичного налаштування систем. Приклади технологій: Ansible, Puppet, Chef, Terraform. Переваги:

– швидкість налаштування;
– зменшення помилок;
– повторюваність конфігурації.

7.5 Практичні рекомендації щодо конфігурації інформаційних систем

Для ефективної конфігурації ІС рекомендується дотримуватися таких принципів;

1. Використання автоматизації. Автоматизація зменшує кількість помилок та прискорює процес налаштування.

2. Документування конфігурації. Усі параметри системи повинні бути задокументовані.

3. Використання систем контролю версій. Конфігураційні файли бажано зберігати у системах контролю версій (Git).

4. Використання тестового середовища. Перед впровадженням конфігурації необхідно тестувати систему.

5. Моніторинг системи. Після конфігурації необхідно здійснювати моніторинг роботи системи. Для цього використовуються системи: Prometheus, Grafana, Zabbix.

Контрольні запитання

1. Дайте визначення конфігурації ІС.
2. Опишіть технології конфігурації.
3. Як оцінити ефективність конфігурації?
4. Назвіть практичні рекомендації щодо конфігурації інформаційних систем.

ТЕМА 8 Базові інструменти адміністрування інформаційних систем

8.1 Служби мережевої інфраструктури

Мережева інфраструктура – це сукупність різного устаткування, а також програмного забезпечення, яка формує особливе середовище для ефективного процесу обміну даними, а також для роботи додатків [2].

Системи зберігання і обробки даних − одна з найважливіших складових IT-інфраструктури. Ефективна робота сучасних додатків неможлива без надійного фундаменту, на якому будується центр обробки даних (ЦОД). Одним із будівельних блоків цього фундаменту є мережева інфраструктура.

Базовий набір мережевих служб будь-якої корпоративної мережі

складається з таких служб:

1) служби мережевої інфраструктури: DNS, DHCP, WINS;

2) служби файлів і друку;

3) служби каталогів (наприклад, Novell NDS, MS Active Directory);

4) служби обміну повідомленнями;

5) служби доступу до баз даних.

Верхній рівень функціонування мережі − мережеві додатки.

Мережа дозволяє легко взаємодіяти один з одним різним видам комп'ютерних систем завдяки стандартизованим методам передачі даних, які дозволяють приховати від користувача все різноманіття мереж і машин.

Всі пристрої, що працюють в одній мережі, повинні спілкуватися однією мовою − передавати дані за загальновідомим алгоритмом у форматі, який буде зрозумілий іншими пристроями. Стандарти − ключовий фактор під час об'єднання мереж.

Службою DNS – називається служба, що виконує перетворення символічних даних доменних імен в IP- адреси, у відповідь на запити клієнтів.

Служба DNS ставить відповідно ось такому символічному (букви це у цьому випадку символічні дані) доменному імені, наприклад, як «moycomp.rk.com», відповідна йому IP-адреса: 10.10.10.10.

DNS-сервером називається комп'ютер, на якому запушена служба DNS.

DNS-клієнтом називається комп'ютер, який звертається до DNS сервера з запитом на дозвіл імені. Клієнт DNS функціонує «прозоро» для користувача. Що значить «прозоро»? Користувач не бачить, і не чує, і не відчуває як це відбувається.

У відповідь на свій запит із вирішення імені клієнт DNS повинен отримати або IP-адресу, або повідомлення про неможливість доступу імені, надіслану сервером DNS. Далі цей клієнт передає отриману IP-адресу додатку, якому потрібна ця IP-адреса.

DNS-клієнт (від англ. «Domain Name System-client» − «доменних імен система – клієнт») – програма або модуль у програмі, що забезпечує з'єднання із DNS-сервером для визначення IP-адреси за його доменним іменем.

Авторитетність (англ. «authoritative») – ознака розміщення зони на DNS-сервері. Відповіді DNS-сервери можуть бути двох типів: авторитетні (коли сервер заявляє, що сам відповідає за зону) і неавторитетні (англ. «Non-authoritative»), коли сервер обробляє запит, і повертає відповідь інших серверів. У деяких випадках замість передачі запиту далі DNS-сервер може повернути вже відоме йому (за запитами раніше) значення (режим кешування).

Основними компонентами простору імен DNS є домени. Домени треба розуміти як групу мережевих хостів (вузлів), що об'єднані за деяким логічним принципом. Домени взаємодіють один з одним за допомогою відносин «батько – нащадок», утворюючи тим самим певну ієрархію. Тобто, коли йдеться про ієрархію доменів треба розуміти ставлення «батько-нащадок». Положення домену в ієрархії визначає рівень домену (домен другого рівня, домен третього рівня тощо, зважаючи на його положення від батьківського домену).

В основі ієрархічної побудови простору імен DNS перебуває домен, який

називається кореневим доменом («rootdomain»). Кореневий домен формальний або чисто символічний, але водночас він є прабатьком всіх інших нині існуючих в інтернеті доменів. Або батьком доменів першого рівня таких як *com, edu, org …* тощо, які належать різним організаціям. Так само домени належать країнам: *ua, ro, jp*... тощо.

Домени виступають в ролі контейнерів. І якщо розглядати це як ієрархію, то в якості рівнів виступають відомості про ресурси цих доменів. Будь-який об'єкт мережі, який з огляду на стандарти визначає службу DNS називається хост (host) або хостами.

Будь-який об'єкт простору імен DNS, чи то домен або хост, має унікальне ім'я в межах батьківського контейнера. Повне доменне ім'я хосту утворюється з імені об'єкта і суфікса DNS. Єдиним об'єктом простору імен DNS, що не має імені, є кореневий домен. Для посилання на нього використовується крапка.

Сучасні DNS так само є, по суті, базою даних з тією ж основною інформацією, яку містив попередник DNS, файл HOSTS. Але на відміну від історичного файлу HOSTS, який тримав всю інформацію про всі хости в єдиному файлі, сучасні DNS є розподіленими системами зберігання інформації. DNS це – розподілена база даних.

DHCP *(від англ. «Dynamic Host Configuration Protocol», тобто «протокол динамічної конфігурації вузла»)* – це мережевий сервіс, який дозволяє комп'ютерам у мережі автоматично отримувати налаштування мережі з сервера замість того, щоб налаштовувати кожен мережевий хост вручну. Комп'ютери, налаштовані бути клієнтами DHCP, не керують тим, які налаштування вони отримають від DHCP сервера, і це налаштування зовсім непомітне для користувача комп'ютера. Стандарт протоколу DHCP був прийнятий у жовтні 1993 року

У випадку налаштування дані, передані DHCP сервером клієнтам, містять:

– *IP-адресу і мережеву маску;*

– *IP-адреса шлюзу;*

– *IP-адресу DNS серверів;*

Однак DHCP сервер може також надати такі параметри налаштування, як:

– *ім'я хосту;*

– *ім'я домену;*

– *адреса сервера часу;*

– *адреса сервера друку.*

Перевага використання DHCP полягає в нестабільності налаштувань мережі, наприклад, зміна адреси DNS сервера потребує змін тільки на DHCP-сервері, а всі мережеві хости будуть переналаштовані в момент наступного запиту їхнього DHCP-клієнта до DHCP-сервера. Додаткова перевага полягає в простому підключенні нових комп'ютерів до мережі, оскільки не потрібно перевіряти доступність IP адрес. Конфлікти за виділеними IP адресами також мінімальні.

DHCP сервер може надавати налаштування, використовуючи такі

методи.

Виділення вручну (за MAC-адресою). Цей метод передбачає використання DHCP для визначення унікальної апаратної адреси кожної мережевої карти, під'єднаної до мережі, і потім тривалого надання незмінної конфігурації кожен раз, коли DHCP-клієнт робить запит на DHCP-сервер, використовуючи свій мережевий пристрій. Це гарантує, що певну адресу буде автоматично надано цій мережевій картці на основі її MAC-адреси.

Динамічне виділення (пул адрес). У разі цього методу DHCP-сервер виділятиме IP-адреси з пулу адрес (діапазоном або областю) на період часу (або в оренду), який налаштовується на сервері, або поки клієнт не проінформує сервер, що більше взагалі не потребує адреси. Таким чином, клієнти отримують свої налаштування динамічно за принципом «перший прийшов – перший обслуговований». Коли DHCP-клієнт відсутній в мережі певний час, адреса вважається простроченою і повертається в пул адрес для використання іншими DHCP-клієнтами. Це означає, що адреса орендується або видається на певний період часу. Після закінчення цього періоду клієнт повинен повторно домовлятися про використання адреси з сервером.

Автоматичне виділення. Використовуючи цей метод, DHCP автоматично присвоює постійну IP-адресу пристрою, вибрану з пулу доступних адрес. Зазвичай DHCP використовується для видачі тимчасового адреси, але DHCP-сервер може використовувати нескінченний час оренди.

Два останні методи можна розглядати як автоматичні, оскільки DHCP-сервер видає адреси без додаткового втручання. Єдина різниця полягає в тому, наскільки часу орендується адреса, інакше кажучи, коли закінчиться час використання адреси клієнтом.

8.2 Файлові служби і служби друку

Файлові служби і служби друку надають технології, здатні допомогти в управлінні сховищем і загальними папками, виконанні реплікації і швидкого пошуку файлів, а також підтримці й забезпеченні доступу клієнтських UNIX-комп'ютерів для вирішень завдань друку рівня підприємства. До складу файлових служб Windows Server 2008 R2 входить:

1. Управління загальними ресурсами і зберіганням.
2. Розподілена файлова система (DFS).
3. Диспетчер ресурсів файлового сервера.
4. Служби для NFS.
5. Служба пошуку Windows.
6. Файлові служби Windows Server 2003.
7. Служба BranchCache для мережевих файлів.

Управління загальними ресурсами і зберіганням надає інтегроване і спрощене управління загальними папками і ресурсами зберігання. Цю консоль можна використовувати для надання загального доступу до вмісту папок і управління використанням загальних папок.

Розподілена файлова система (DFS) складається із двох технологій, які можуть бути використані разом або окремо для подання відмовостійких і

гнучких служб надання загального доступу до файлів і реплікації в мережах на основі операційних систем Windows.

Диспетчер ресурсів файлового сервера (FSRM) є комплектом інструментальних засобів, що дозволяють адміністраторам контролювати об'єми і типи даних, що зберігаються на серверах, а також керувати ними. За допомогою диспетчера ресурсів файл-сервера адміністратори можуть задавати квоти для папок і томів, активно відстежувати та автоматично класифікувати файли, застосовувати заснований на класифікації термін дії файлів і використовувати особливі завдання, а також генерувати докладні звіти сховища.

Служби для NFS надають вирішення спільного доступу до файлів для підприємств, що працюють у змішаному середовищі Windows і UNIX. Служби для NFS дозволяють користувачам переміщувати файли між операційною системою Windows Server 2008 або Windows Server 2008 R2 і операційними системами UNIX за допомогою протоколу NFS.

Служба пошуку Windows дозволяє виконувати швидкий пошук файлів на сервері з клієнтських комп'ютерів, сумісних із пошуком Windows.

Файлові служби Windows Server 2003 містять служби індексування, сумісні з Windows Server 2003. Служба індексування розподіляє вміст по каталогу і властивості файлів на локальних і віддалених комп'ютерах. Також вона дозволяє швидко знайти файли за допомогою гнучкої мови запитів.

На одному комп'ютері не можна встановити разом службу пошуку Windows і службу індексування. Обидва рішення індексування займають ресурси системи в разі активного індексування томів і папок, тому одночасна робота обох служб може істотно зменшити продуктивність системи.

Основні терміни і поняття Active Directory

Каталог (directory) – сукупність інформації про об'єкти, які тим або іншим способом пов'язані один з одним. Тобто це **певний інформаційний** ресурс, використовуваний для зберігання інформації про який-небудь об'єкт. Наприклад, найпростіший історичний каталог, це такий зошит, в який деякий громадянин записує своїх боржників. Більш сучасний приклад – телефонний довідник, що містить інформацію про абонентів телефонної мережі. У файловій системі каталоги (директорії, теки) зберігають інформацію про файли, пов'язані між собою тим або іншим способом, наприклад, усі файли з теки такої-то – це деякі оповідання.

У мережі в каталозі може зберігатися інформація про об'єкти нашої мережі. Це можуть бути принтери, комп'ютери, користувачі, програмні продукти та різні бази даних, які є у нас. Головна «фішка» або завдання тут в тому, щоб надати користувачам можливість виявити (знайти) ці об'єкти і використовувати їх, але під пильним оком адміністратора, а не так, як хочуть користувачі.

Об'єкт – самостійна одиниця, яка є деяким ресурсом, існуючим у мережі (ПК, користувачі, ПО тощо) з усіма його атрибутами.

Служба каталогів (directory service) за допомогою допоміжних служб забезпечує **зберігання усієї** необхідної для застосування об'єктів і управління

ними інформації в одному місці (централізовано), і завдяки цьому процес виявлення і адміністрування ресурсів мережі спрощується.

На відміну від каталогу, служба каталогів має одночасно дві ролі: джерела інформації та механізму, за допомогою якого ця інформація готується для доступу з боку користувачів.

Служба каталогів функціонує на зразок основної панелі керування мережевої операційної системи (центрального пульта). Служба каталогів – це таке програмне забезпечення (така програма) у складі мережевої операційної системи, яка зберігає усю інформацію про об'єкти мережі, яка дозволяє виявляти ці об'єкти (різні комп'ютери, принтери, користувачів, які є в мережі) надавати їх в розпорядження користувачам, і управляти ними (адмініструват**и**).

Служба каталогів – це таке ПЗ (програмне забезпечення), яке «зберігає», «виявляє», «робить доступним користувачам» і «керує ними», тобто об'єктами.

Контролер домену – це комп'ютер, на якому встановлено таке ПЗ як Active Directory.

Служба каталогів Active Directory – це програмне забезпечення написане (розроблене) Microsoft. Microsoft назвав «службу каталогів» ім'ям «Active Directory», Макінтош назвав свою службу каталогів – Open Directory, а Novell – NDS.

Виходить, що Active Directory містить каталог, в якому зберігається інформація про наші мережеві ресурси і служби, що надають доступ до цієї інформації. Це як бази даних з інформацією про мережеві ресурси.

«Служба каталогів» – це, з одного боку, інструмент адміністрування, а з іншого, – засіб взаємодії кінцевого користувача з системою.

Чим сильніше збільшується мережа, тим більше в ній об'єктів, якими необхідно керувати, і тут без впровадження служби каталогів (Active Directory) не обійтися.

Обліковий запис користувача (акаунт) – складається з імені користувача (логін) і пароля (наприклад, «pupkin» і пароль «d345rtHfa»). Без цих **даних, не** можна увійти до мережі або працювати а комп'ютері. Перший раз під час входу в мережу пароль повідомляє **адміністрат**ор і залежно від того, як він налаштував роботу з паролем в Active Directory, можна відразу його змінити на свій пароль, який ніхто не знає. Системний адміністратор може змінити будь-які налаштування облікового запису.

Контейнер аналогічний об'єкту в тому сенсі, що він також має атрибути і належить простору імен. Проте, на відміну від об'єкта, контейнер не означає нічого конкретного: він може містити групу об'єктів або інші контейнери.

Active Directory **передбачено** низку компонентів, що допомагають збудувати структуру каталогу відповідно до наших потреб, і діляться ці компоненти на **логічні і фізичні** компоненти. Логічне ділення передбачає такі компоненти як **домени, підрозділи** (OU, organizational unit), **дерева і ліси**. Фізичне – на **контролери домену і сайти**. Логічні і фізичні компоненти розділені.

Служби обміну повідомленнями – це **безкоштовна** програма обміну текстовими повідомленнями для надсилання та отримання повідомлень. Для

роботи програм для обміну повідомленнями не потрібне підключення до інтернету.

Миттєві (миттьові́) повідомлення або система обміну миттєвими повідомленнями, скорочено *ІМ* − телекомунікаційна служба для обміну текстовими повідомленнями **між** комп'ютерами або іншими пристроями користувачів **через** комп'ютерні мережі (зазвичай через інтернет).

Для користування цим видом комунікації **необхідна клієнтська програма**. Клієнтську програму системи миттєвих повідомлень часто називають *інтернет-пейджером* або *месенджером*.

Відмінність миттєвих повідомлень від, наприклад, електронної пошти тут в тому, що обмін повідомленнями відбувається в реальному часі. Під час відправлення повідомлення електронною поштою повідомлення **зберігаєтьс**я у поштовій скриньці на сервері. Для того, щоб отримати повідомлення, отримувач повинен сам перевірити свою поштову скриньку і забрати їх. У інтернет-пейджерах зв'язок між користувачами **утримується** постійно і надіслане повідомлення одразу передається користувачу.

Система миттєвих повідомлень працює **за деяким протоколом**. Протоколи бувають серверні або безсерверні. Найпоширенішими **є серверні протоколи**, коли **месенджери не** працюють самостійно, а під'єднуються до центрального комп'ютера мережі обміну повідомленнями, який називають сервером. Тому месенджери й називають клієнтами (клієнтськими програмами).

У безсерверних протоколах (FChat, NASSI, UChat) повідомлення передаються безпосередньо від одного співрозмовника до іншого.

Служби доступу до баз даних (БД)

Доступ користувачів до БД здійснюється зазвичай за допомогою оглядачів (browser). Інформаційні системи в локальних комп'ютерних мережах будуються за такими варіантами: файл-сервер, клієнт-сервер. Способом доступу до даних БД, вони поділяються на бази даних із локальним доступом і бази даних із віддаленим (мережевим) доступом. Відповідно БД із мережевим доступом проєктують згідно з архітектурою – файл-сервер або клієнт-сервер.

Клієнт-серверна архітектура більше поширена, на ній функціонує інтернет. Суть концепції клієнт-сервера полягає в тому, що **окрім зберігання** централізованої бази даних, центральна машина (сервер бази даних) повинна забезпечувати виконання основного обсягу обробки даних. Запит на дані, який видається клієнтом (робочою станцією), **породжу**є пошук і **вилучення** даних на сервері. Дані транспортуються мережею від сервера до клієнта. Нові дані записуються / перезаписуються до БД вебдодатка. Специфікою архітектури клієнт-сервер **є** використання мови запитів SOL.

Контрольні запитання

1. Дайте визначення мережевих структур.
2. Назвіть мережеві служби.
3. Опишіть файлові служби і служби друку.
4. Назвіть основні терміни і поняття Active Directory.

ТЕМА 9 Проблеми безпеки інформаційних систем

9.1 Види та захист від загроз безпеки

Згідно з загальноприйнятим визначенням, **безпечна комп'ютерна інформаційна система** − це ідеальна система, яка коректно і у повному обсязі реалізує *лише ті* цілі, що відповідають намірам її користувача [7].

На практиці побудувати складну систему, що задовольняє цьому принципові, неможливо. У зв'язку з цим забезпечення безпеки зводиться найчастіше до *управління ризиком:* визначення потенційних загроз, оцінка ймовірності їхнього настання та оцінка потенційної шкоди з наступним вжиттям запобіжних заходів в обсязі, що враховує технічні можливості й економічні обставини.

Для кращого розуміння методів захисту комп'ютерної системи необхідно спочатку ознайомитися з типами атак, які можуть бути здійснені проти неї. Загалом, ці загрози відносяться до однієї з таких категорій:

1. Чорний хід, або бекдор у комп'ютерній системі, криптосистемі чи алгоритмі − це метод обходу звичайного процесу аутентифікації, забезпечення віддаленого доступу до комп'ютера, одержання доступу до незашифрованої інформації тощо. Бекдори можуть відбуватися у формі встановлення програми (наприклад, Back Orifice) або змін у роботі існуючої програми чи фізичного пристрою.
2. DoS, DDoS, 4DoS − атаки на відмову в обслуговуванні.
3. Атаки на CPU (центральний процесор).
4. Атаки на RAM (оперативна пам'ять).
5. Атаки на дисковий простір (комп'ютерна пам'ять).
6. Проникнення всередину периметра (SQL ін'єкція, метод «грубої сили» тощо).
7. Фішинг (email, formjacking, XSS, etc.).
8. Вимагання (ransomware – програма-шантажист).

На відміну від інших атак, DoS-атаки застосовуються **не** для одержання несанкціонованого доступу чи керування системою, а для того, щоб **унеможливити** роботу останньої. У **результа**ті атаки акаунт окремої жертви може виявитися заблокованим унаслідок умисного багаторазового введення невірного пароля, або ж унаслідок перевантаження мережі буде заблоковано усіх її користувачів. На практиці цьому виду атак **дуже складно перешкодити**, оскільки для цього необхідно проаналізувати поведінку цілих мереж, а не лише поведінку невеличкої частини коду.

Користувач, який одержав несанкціонований доступ до комп'ютера (чи його частини), може встановлювати на ньому різні типи програмного (зокрема модифікації операційних систем, віруси, програмні кілогери) та апаратного (апаратні кілогери, пристрої для прослуховування) забезпечення, внаслідок чого безпека системи опиниться під загрозою. Кілогери (або кейлогери, від англ. «keylogger») − це шпигунські програми або апаратні пристрої, що непомітно фіксують натискання клавіш на клавіатурі для викрадення паролів, логінів та особистих даних. Вони можуть бути програмними (віруси) або

апаратними (вбудованими в клавіатуру), працюючи невидимо для користувача.

Такі порушники можуть легко скачати великі об'єми даних на зовнішні носії. Ще одним видом атак безпосереднього доступу є завантаження операційної системи з зовнішнього носія з наступним зчитуванням даних із жорсткого диску (дисків). Цей різновид атак є зазвичай єдиним методом атаки комп'ютерів, що не підключені до інтернету.

Зростання кількості складних кібератак, поява zero-day, APT, вимоги до автоматизації захисту потребують формулювання проблеми і вирішення завдання, оскільки традиційні методи (сигнатурні, правило-орієнтовані) дедалі гірше долають нові загрози − тут і потрібен ШІ як адаптивний компонент.

Традиційні методи захисту, такі як системи контролю доступу, брандмауери, антивірусні рішення чи системи виявлення вторгнень більше не здатні ефективно протидіяти динамічним і складним загрозам. Це зумовлює необхідність інтеграції **штучного інтелекту (ШІ)** у системи комп'ютерної безпеки.

Традиційні методи (сигнатурні, правилові) добре працюють проти відомих загроз, вони прості в розгортанні, зрозумілі. Методи на базі ШІ – ефективні проти невідомих / поведінкових атак, адаптуються, але потребують великого об'єму даних, обчислень і контролю (етика, explainability, захист від атак на моделі).

Системи ШІ у комп'ютерній безпеці: виявлення вторгнень (IDS/IPS), класифікація подій, аномальна поведінка користувачів (UEBA), кореляція логів, прогнозування ризику тощо.

Джерелами даних в інтеграційних із ШІ системах є мережеві потоки (NetFlow/IPFIX), мережеві / системні логи (syslog, Windows Event), журнали аутентифікацій, журнал доступу до файлів, телеметрія кінцевих точок (EDR), події SIEM, дані OT/SCADA (якщо є).

Децентралізована архітектура на базі агентів

Інтелектуальні агенти встановлюються у вузлах мережі та обмінюються даними.

Забезпечується самонавчання та локалізація загроз у межах сегмента.

Гібридні архітектури містять поєднання класичних засобів (брандмауерів, IDS) із моделями машинного навчання, а також передбачають використання експертних систем і нечіткої логіки для аналізу нестандартних загроз.

Архітектурні рішення залежать від масштабу системи, рівня загроз і доступних ресурсів.

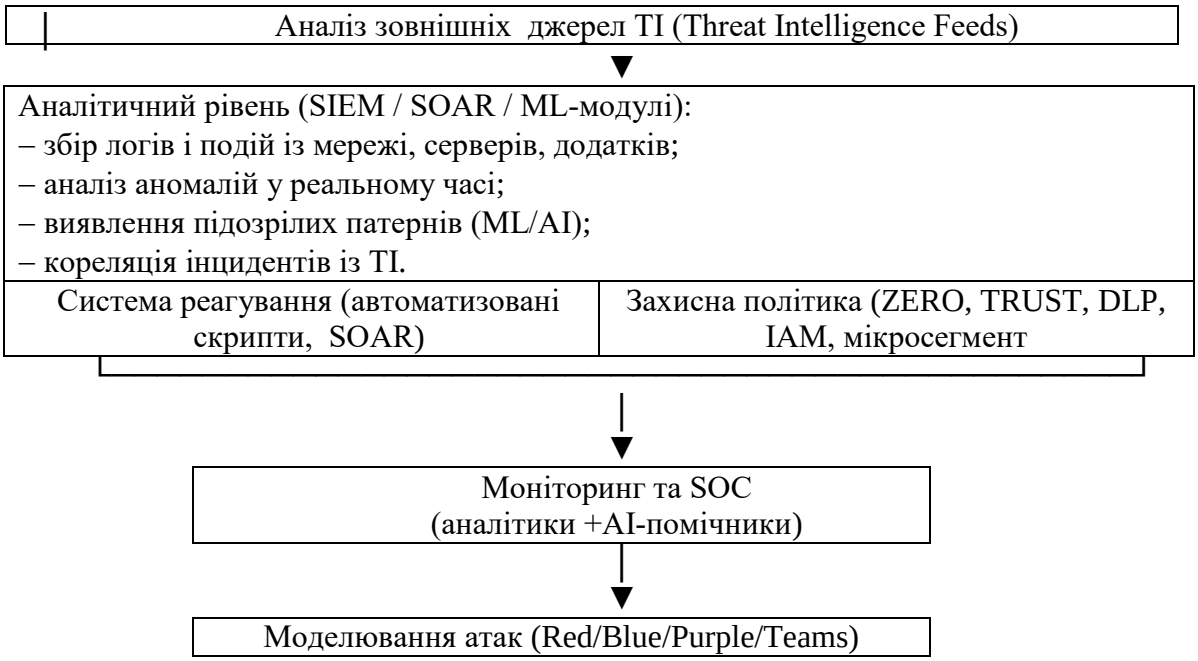


Рисунок 9.1 – Схема кіберзахисної інфраструктури

Проактивна кіберзахисна інфраструктура поєднує **автоматизовані системи (SIEM, SOAR, AI/ML) та експертні дії аналітиків**, що дозволяє виявляти й запобігати загрозам **ще до** їхнього реалізації. Такий підхід підвищує стійкість організації до кіберінцидентів, знижує ризики та забезпечує безперервність бізнес-процесів.

9.3 Засоби, заходи та норми забезпечення безпеки

Сучасний розвиток інформаційних технологій супроводжується зростанням кількості та складності кіберзагроз, що зумовлює необхідність створення ефективних систем комп'ютерної безпеки. Для забезпечення належного рівня захисту інформації у корпоративних, державних та критичних інфраструктурах необхідним є систематичний аналіз їхнього стану, дослідження методів захисту та формування інформаційної бази, яка слугує основою для ухвалення рішень.

Основні методи дослідження систем комп'ютерної безпеки, підходи до оцінювання їх ефективності, а також структурована інформаційна база, що підтримує процеси моніторингу, аналізу й управління безпекою охоплюють широкий спектр підходів – від класичних аналітичних методик до сучасних інструментів на основі штучного інтелекту. Використання комбінації цих методів дозволяє комплексно оцінювати рівень захисту інформаційних систем, виявляти вразливості та розробляти ефективні стратегії протидії кіберзагрозам.

Методи дослідження у сфері комп'ютерної безпеки можна поділити на **аналітичні**, **емпіричні** та **комбіновані** (табл. 9.1). Вони забезпечують різні рівні розуміння проблеми: від виявлення потенційних вразливостей до моделювання поведінки системи під час атак. Аналітичні методи передбачають:

1. Аналіз ризиків (Risk Assessment) – визначення ймовірності виникнення загроз та їхнього впливу на систему.

2. Моделювання загроз (Threat Modeling) – побудова моделей можливих атак і шляхів їхньої реалізації.

3. Формальна верифікація політик безпеки – перевірка правильності та узгодженості правил доступу й контролю.

До емпіричних методів належать:

1. Пен-тестинг (penetration testing) – імітація реальних атак для перевірки захищеності.

2. Аудит інформаційної безпеки – перевірка відповідності системи стандартам (ISO/IEC 27001, NIST, GDPR).

3. Моніторинг та лог-аналіз – вивчення подій у журналах безпеки (SIEM-системи).

Комбіновані методи:

1. Машинне навчання та ШІ – виявлення аномалій і прогнозування атак.

2. Моделювання сценаріїв інцидентів – поєднання статистичних і симуляційних підходів для оцінки стійкості системи.

3. Red Team та Blue Team – інтерактивні тренування, де одна команда атакує систему, а інша захищає її.

Таблиця 9.1 − Методи дослідження систем комп'ютерної безпеки

Група методів	Конкретний метод	Призначення	Приклад застосування
Аналітичні	Аналіз ризиків	Визначення рівня загроз і вразливостей	ISO/IEC 27005
Аналітичні	Моделювання загроз	Виявлення потенційних шляхів атак	STRIDE, DREAD
Емпіричні	Пен-тестинг	Перевірка стійкості системи до реальних атак	White/Grey/Black-box-тестування
Емпіричні	Аудит ІБ	Оцінка відповідності стандартам безпеки	ISO/IEC 27001 аудит
Емпіричні	Моніторинг і лог-аналіз	Виявлення атак у реальному часі	SIEM (Splunk, QRadar)
Комбіновані	ML/AI аналіз	Виявлення аномалій, прогноз атак	IDS/IPS з ML
Комбіновані	Red/Blue Team	Навчання і тестування стратегії захисту	Кіберполігони

Розвиток кіберзагроз відбувається паралельно з розвитком технологій. Порівняльний аналіз підтверджує переваги ШІ-орієнтованих методів у швидкості виявлення атак, точності класифікації загроз і можливості адаптації до нових умов (табл. 9.2).

Таблиця 9.2 − Порівняння традиційних методів захисту з методами на базі штучного інтелекту (ШІ)

Критерій	Традиційні методи	Методи на базі ШІ
Виявлення загроз	Сигнатурний підхід, ручне оновлення баз	Аномалії, поведінкові моделі, самонавчання
Адаптація до нових атак	Низька (нові загрози вимагають оновлень)	Висока (навчання на нових даних)
Швидкість реакції	Залежить від людини-адміністратора	Автоматизована, у реальному часі
Прогнозування атак	Немає	Можливе завдяки прогнозним моделям
Надійність у складних середовищах	Обмежена	Висока, за рахунок багатовимірного аналізу
Вартість впровадження	Нижча	Вища на початковому етапі, але більш економічна у довгостроковій перспективі

Комплексне застосування різних підходів створює **багаторівневу модель захисту**, де кожен метод компенсує недоліки іншого (табл. 9.3)

Таблиця 9.3 – Характеристики багаторівневої моделі

Підходи	Використання	Методи	Приклади
Машинне навчання	Для класифікації шкідливого трафіку	Дерева рішень, метод опорних векторів (SVM), ансамблеві моделі (Random Forest, XGBoost)	Фільтрація спаму, виявлення аномалій у логах
Глибинне навчання	Багатошарові нейронні мережі (CNN, RNN, LSTM, Autoencoders)	Ефективне у виявленні zero-day атак та аналізі складних шаблонів трафіку	Deep Packet Inspection із використанням CNN
Експертні системи	На базі правил і знань	Визначення рівня критичності атаки, вибір політики реагування	
Нечітка логіка	Для ухвалення рішення в умовах невизначеності	Багатофакторна автентифікація	Оцінка ризику доступу користувача (низький, середній, високий)

9.3 Криптографія і гамування в захисті інформації

Методом забезпечення безпеки інформаційних систем є криптографія. Криптографія містить шифрування, дешифрування та автентифікацію.

Шифрування − це процес перетворення відкритого тексту, або читабельних даних, у незрозумілу інформацію, відому як шифротекст. Це досягається за допомогою використання алгоритму та конкретного ключа. Алгоритм виконує складні математичні операції над відкритим текстом, а ключ впливає на результат алгоритму. Отриманий шифротекст може бути дешифрований і зрозумілий лише тому, хто має відповідний ключ

дешифрування.

Шифрування виступає важливим захистом конфіденційних даних під час їх передачі або зберігання. Навіть якщо несанкціонована особа перехопить дані, вона не зможе зрозуміти їх зміст. Шифрування широко використовується у різних додатках, включно з безпечними повідомленнями, онлайн-банкінгом та електронною комерцією.

Дешифрування − це зворотний процес шифрування. Воно містить перетворення шифротексту назад у його оригінальний формат відкритого тексту. Дешифрування використовує ключ дешифрування, відповідний ключу шифрування, що використовувався в процесі шифрування.

Тільки авторизовані особи, які мають правильний ключ дешифрування, можуть отримати доступ та зрозуміти зашифровані дані. Алгоритми дешифрування використовують математичні операції для зворотного процесу шифрування та отримання оригінального відкритого тексту. Це дозволяє безпечно передавати чутливу інформацію за умови, що ключ шифрування залишається конфіденційним.

Автентифікація відіграє важливу роль у забезпеченні цілісності та походження повідомлення або даних. Вона підтверджує особу відправника та гарантує, що дані не були змінені під час передачі або зберігання. Техніки автентифікації використовують криптографічні алгоритми для створення цифрових підписів або кодів автентифікації повідомлень (MAC).

Гамування є широко застосовуваним криптографічним перетворенням. Під гамуванням розуміють процес накладення за визначеним законом гами шифру на відкриті дані. Гама шифру − це псевдо випадкова послідовність, вироблена по заданому алгоритмі для шифровки відкритих даних і дешифрування зашифрованих даних. Процес шифрування полягає в генерації гами шифру за допомогою сенсора псевдовипадкових чисел і накладення отриманої гами на вихідний відкритий текст, наприклад, із використанням операції додавання по модулю. Результатом додавання двох цілих чисел по модулю є залишок від ділення (наприклад: 5 + 10 mod 4 = 15 mod 4 = 3). Шифри гамування (адитивні шифри) є найефективнішими з погляду стійкості та швидкості перетворень (процедур шифрування і дешифрування). У літературі шифри цього класу часто називають потоковими, хоча до потокових належать й інші різновиди шифрів. У шифрах гамування може використовуватися додавання по модулю N (загальний випадок) і по модулю 2 (окремий випадок).

Контрольні запитання

1. Назвіть методи дослідження систем комп’ютерної безпеки.
2. Як працює багаторівнева модель захисту інформації?
3. Опишіть процеси застосування криптографії і гамування в захисті інформації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Бойко О. В. 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: 02.03.2026). – 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: 09.03.2026). – Назва з екрана.

3. Комп'ютерні мережі : навч. посіб. [Електрон. ресурс] / 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: 10.04.2026). – Назва з екрана.

4. Голь В. Д. Телекомунікаційні та інформаційні мережі [Електрон. ресурс] : навч. посіб. – Електрон. текст. дані. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 250 с. – Режим доступу: https://ela.kpi.ua/items/c550fb2c-13a8-43a8-b0f3-348d3ab439df, вільний (дата звернення: 18.03.2026). – Назва з екрана.

5. Каштан В. Ю. Конспект лекцій з дисципліни «Бази даних в інформаційних системах» для студентів галузі знань 12 «Інформаційні технології» спеціальності 126 – Інформаційні системи та технології» [Електрон. ресурс] / В. Ю. Каштан, Д. В. Іванов. – Дніпро : НТУ «ДП», 2021. – 58 с. – Режим доступу: https://it.nmu.org.ua/ua/scientific_method_materials/lecture_notes/%D0%9A%D0%BE%D0%BD%D1%81%D0%BF%D0%B5%D0%BA%D1%82_%D0%BB%D0%B5%D0%BA%D1%86%D1%96%D0%B9_%D0%91%D0%94_%D1%87%D0%B0%D1%81%D1%82%D0%B8%D0%BD%D0%B01_2021.pdf, вільний (дата звернення: 20.03.2026). – Назва з екрана.

6. Pressman R. Software Engineering [Electronic resource]: A Practitioner's Approach / R. Pressman, B. Maxim. – Electronic text data. – McGraw-Hill, 2020. – P. 671. – Regime of access: https://books.google.com.ua/books/about/Software_Engineering.html?id=Enn2zQEACAAJ&redir_esc=y, free (date of the application: 12.03.2026). – Header from the screen.

7. Костюченко А. О. Віртуалізація операційних систем : навч.-метод. посіб. / А. О. Костюченко, Ю. В. Горошко. – Чернігів : ФОП Баликіна С. М., 2021. – 56 с.

8. Технології забезпечення безпеки мережевої інфраструктури [Електрон. ресурс] : підручник / В. Л. Бурячок, А. О. Аносов, В. В. Семко [та ін.]. – Електрон. текст. дані. – Київ : КУБГ, 2019. – 218 с. – Режим доступу: https://spadok.org.ua/books/Buryachok-Osnovy-info-ta-ciberbezpeky.pdf, вільний (дата звернення: 15.03.2026). – Назва з екрана.

Електронне навчальне видання

СІЗОВА Наталія Дмитріївна

СИСТЕМНА ІНТЕГРАЦІЯ ТА АДМІНІСТРУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

КОНСПЕКТ ЛЕКЦІЙ

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

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

План 2026, поз. 89Л

Підп. до друку 17.06.2026. Формат 60 × 84/16.
Ум. друк. арк. 4,5

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.