

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ АГРАРНИЙ УНІВЕРСИТЕТ**

**ФАКУЛЬТЕТ МЕНЕДЖМЕНТУ
КАФЕДРА ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ, КОМП'ЮТЕРНИХ НАУК ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

конспект лекцій для здобувачів першого (бакалаврського) рівня
вищої освіти ОПП «Комп'ютерні науки» спеціальності F3 (122)
«Комп'ютерні науки» денної форми здобуття вищої освіти



Миколаїв
2025

Друкується за рішенням науково-методичної комісії факультету менеджменту Миколаївського національного аграрного університету від 24 квітня 2025 року, протокол №8.

Укладачі:

- С. І. Тищенко - канд. пед. наук, доцент, доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій, Миколаївський національний аграрний університет;
- Т. С. Кучмійова - канд. екон. наук, доцент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій Миколаївського національного аграрного університету
- В. В. Співак - асистент кафедри економічної кібернетики, комп'ютерних наук та інформаційних технологій, Миколаївський національний аграрний університет;

Рецензенти:

- О. С. Садовий - канд. техн. наук, доцент, завідувач кафедри агроінженерії Миколаївського національного аграрного університету
- Ю. В. Грицук - канд. техн. наук, доцент кафедри загальної інженерної підготовки Донбаської національної академії будівництва і архітектури

Інформаційні системи та технології : конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти ОПП «Комп'ютерні науки» спеціальності F3 (122) «Комп'ютерні науки» денної форми здобуття вищої освіти/ уклад. С. І. Тищенко, Т. С. Кучмійова, В. В. Співак. Миколаїв : МНАУ, 2025. 38 с.

Конспект лекцій призначений для вивчення теоретичних та практичних основ інформатики, архітектури обчислювальних систем та комп'ютерних наук, ознайомлення з фундаментальними поняттями, принципами побудови алгоритмів та інструментами розробки програмного забезпечення, забезпечення глибокого засвоєння специфіки застосування інформаційних технологій у професійній діяльності, опанування методів системного аналізу, обробки великих масивів даних та захисту інформації, формування розуміння архітектури інформаційних систем у сучасному світі. Містить навчальні матеріали з основних тем курсу, що передбачені освітньо-професійною програмою «Комп'ютерні науки»

УДК 004.9:004.4

© Миколаївський національний аграрний університет, 2025

ЗМІСТ

ПЕРЕДМОВА.....	4
ЗМІСТОВИЙ МОДУЛЬ 1. ОСНОВИ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ.	5
ТЕМА 1.1. ВВЕДЕННЯ В ІНФОРМАЦІЙНІ СИСТЕМИ (ІС) І ТЕХНОЛОГІЇ.....	5
ТЕМА 1.2. СУТНІСТЬ ІНФОРМАЦІЙНИХ СИСТЕМ	9
ТЕМА 1.3. КОРПОРАТИВНІ ІНФОРМАЦІЙНІ СИСТЕМИ.....	13
ТЕМА 1.4. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В СУЧАСНОМУ БІЗНЕСІ	16
ТЕМА 1.5. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ НА ПІДПРИЄМСТВІ....	19
ЗМІСТОВИЙ МОДУЛЬ 2. АПАРАТНІ, ПРОГРАМНІ І СИСТЕМНІ ЗАСОБИ ІНФОРМАЦІЙНИХ СИСТЕМ.....	23
ТЕМА 2.1. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	23
ТЕМА 2.2. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.....	28
ТЕМА 2.3. ОСНОВИ ОПЕРАЦІЙНИХ СИСТЕМ. WINDOWS	31
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	36

Передмова

Курс дисципліни «Інформаційні системи та технології» призначений для забезпечення здобувачів вищої освіти спеціальності «Комп'ютерні науки» фундаментальними теоретичними знаннями та практичними навичками у сфері проектування, розробки, впровадження та супроводу сучасних інформаційних систем (ІС). Курс має на меті ознайомити майбутніх фахівців з архітектурою ІС, методологіями їх створення, базовими технологіями обробки даних, а також сформувати глибоке розуміння ролі інформаційних технологій в автоматизації бізнес-процесів та цифровій трансформації суспільства.

Предметом вивчення дисципліни є принципи побудови та функціонування інформаційних систем, методи системного аналізу, інформаційного моделювання, технології організації баз даних, апаратні та програмні платформи, мережеві архітектури, а також сучасні методології розробки та інтеграції програмного забезпечення.

Об'єктом вивчення дисципліни є архітектура інформаційних систем, життєвий цикл розробки програмного забезпечення (SDLC), системи управління базами даних (СУБД), хмарні та розподілені обчислення, апаратно-програмні комплекси, а також інструментальні засоби проектування інформаційних технологій.

Викладання дисципліни ставить за мету сформувати у здобувачів вищої освіти інженерне та системне мислення, розвинути навички проектування архітектури ІС, вибору оптимальних технологічних стеків для вирішення комплексних завдань, а також виробити розуміння стандартів якості, надійності та кібербезпеки в ІТ-індустрії.

Основними завданнями, що мають бути вирішені у процесі викладання дисципліни, є:

- навчити розуміти базові концепції, класифікацію та багаторівневу архітектуру сучасних інформаційних систем;
- надати практичні навички системного аналізу, моделювання предметної області та проектування ІС;
- ознайомити з принципами організації баз даних, сховищ даних та алгоритмами обробки інформації;
- сформувати вміння обґрунтовано обирати та застосовувати релевантні апаратні, програмні та мережеві технології для реалізації конкретних ІТ-проектів;
- навчити оцінювати ефективність інформаційних систем та впроваджувати базові заходи з інформаційної безпеки та захисту даних;
- сформувати практичні навички роботи з сучасними інструментальними середовищами розробки та підтримки життєвого циклу ІС.

ЗМІСТОВИЙ МОДУЛЬ 1. ОСНОВИ СУЧАСНИХ ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ.

ТЕМА 1.1. ВВЕДЕННЯ В ІНФОРМАЦІЙНІ СИСТЕМИ (ІС) І ТЕХНОЛОГІЇ

Мета: Ознайомити здобувачів вищої освіти з теоретичними основами комп'ютерних наук, природою даних та інформації, а також базовими принципами побудови та еволюцією обчислювальної техніки.

План лекції:

1. Фундаментальні поняття дисципліни: Computer Science, дані, інформація, знання.
2. Теорія інформації: синтаксичний, семантичний та прагматичний виміри.
3. Еволюція та етапи розвитку обчислювальної техніки (The stages of computer development).
4. Сутність інформаційних систем (ІС) у контексті загальної теорії систем та кібернетики.
5. Інформаційні технології (ІТ) як інструментальний базис обробки даних.
6. Інформаційне суспільство: критерії, концепції та роль Computer Science у розвитку соціуму.
7. Інформаційна культура, цифрова етика та кібергігієна сучасного фахівця.
8. Стратегічні перспективи розвитку інформаційних технологій у перспективі 2026–2035 років.

1. Фундаментальні поняття дисципліни та епістемологія даних

Основою підготовки фахівців галузі інформаційних технологій є глибоке розуміння концептуального апарату. Базовою науковою дисципліною виступає **Computer Science (Комп'ютерні науки)** — фундаментальна та прикладна наука, що вивчає закономірності, процеси та методи алгоритмічної обробки інформації, архітектуру обчислювальних систем та теорію створення програмного забезпечення.

Для побудови будь-якої інформаційної системи (ІС) необхідно розрізняти епістемологічні рівні абстракції (модель DIKW):

- **Дані (Data):** Дискретні, об'єктивні факти, що подані у формалізованому вигляді (числа, символи, сигнали, зображення), придатні для передачі та обробки технічними засобами. Самі по собі дані не мають внутрішнього змісту; вони є сировиною для інформаційних процесів.

- **Інформація (Information):** Дані, що пройшли процес інтерпретації, структурування та аналізу, завдяки чому вони набули змісту (контексту) для

конкретного користувача. З кібернетичної точки зору (за Норбертом Вінером), інформація — це міра зменшення невизначеності (ентропії) у системі.

- **Знання (Knowledge):** Закономірності, принципи, правила та зв'язки, виявлені в результаті практичного досвіду роботи з інформацією. Знання є активним ресурсом, який дозволяє системі або людині генерувати нову інформацію, приймати рішення та прогнозувати події. Формалізовані знання є основою для систем штучного інтелекту (Експертних систем).

2. Теорія інформації: виміри та властивості

В інформатиці інформація розглядається не лише як абстрактне поняття, але і як вимірювана фізична величина. Існує три рівні аналізу інформації:

1. **Синтаксичний (структурний) вимір:** Досліджує способи кодування та передачі сигналів каналами зв'язку, абстрагуючись від їхнього змісту (теорія Клода Шеннона). Одиницею вимірювання є **біт (bit - binary digit)** — кількість інформації, необхідна для розрізнення двох рівноімовірних станів.

2. **Семантичний (змістовний) вимір:** Досліджує зміст повідомлення та його відношення до об'єкта реального світу. Оцінюється через тезаурус (словник понять) приймача: якщо система не має алгоритму для розшифровки даних, семантична кількість інформації дорівнює нулю.

3. **Прагматичний (ціннісний) вимір:** Визначає корисність інформації для досягнення поставленої мети або вирішення задачі користувачем.

Ключові властивості якісної інформації в ІС:

- *Достовірність (Accuracy):* відсутність викривлень та помилок.
- *Повнота (Completeness):* наявність усіх необхідних атрибутів для прийняття рішення.
- *Релевантність (Relevance):* відповідність інформації поточним інформаційним потребам (запиту).
- *Актуальність (Timeliness):* своєчасність надходження інформації (застаріла інформація перетворюється на марні дані).
- *Толерантність (Ергономічність):* зручність форми подання для сприйняття користувачем.

3. Еволюція та етапи розвитку обчислювальної техніки (The stages of computer development)

Розвиток апаратно-програмних засобів (hardware and software) прийнято ділити на технологічні покоління. Кожне нове покоління характеризується зміною елементної бази, збільшенням швидкодії, зменшенням габаритів та зміною парадигми програмування.

1. **I покоління (середина 1940-х – кінець 1950-х рр.):** Елементна база — електронно-вакуумні лампи. Величезні габарити (розміром із будинок), високе енергоспоживання, низька надійність. Програмування здійснювалося в машинних кодах (на рівні нулів та одиниць) або за допомогою перфокарт. Представники: ENIAC, МЕСМ (створена в Києві під керівництвом С.О. Лебедєва).

2. **II покоління (кінець 1950-х – середина 1960-х рр.):** Елементна база — напівпровідникові транзистори. З'явилися зовнішні носії на магнітних стрічках. Відбувся революційний перехід до алгоритмічних мов програмування високого рівня (Fortran, COBOL, ALGOL), що відокремило процес програмування від знання апаратної архітектури.

3. **III покоління (друга половина 1960-х – кінець 1970-х рр.):** Елементна база — малі інтегральні схеми (мікросхеми). Поява серійних ЕОМ (наприклад, IBM System/360), які мали сумісну архітектуру. Виникнення повноцінних операційних систем, багатозадачності та баз даних.

4. **IV покоління (з початку 1980-х рр. до нашого часу):** Елементна база — великі та надвеликі інтегральні схеми (ВІС/НВІС), поява мікропроцесорів. Цей етап ознаменувався створенням персональних комп'ютерів (ПК), графічних інтерфейсів (GUI), розвитком локальних та глобальних комп'ютерних мереж (Internet).

5. **V покоління та майбутнє (The Future of Computing):** Сучасний етап переходу від класичної архітектури фон Неймана до багатоядерних і нейроморфних процесорів, хмарних обчислень (Cloud Computing), квантових комп'ютерів (Quantum Computing) та систем зі штучним інтелектом (AI).

4. Сутність інформаційних систем (ІС) та загальна теорія систем

Система — це сукупність взаємопов'язаних елементів, які утворюють певну цілісність і спільно функціонують для досягнення спільної мети.

Інформаційна система (ІС) — це організаційно-впорядкована сукупність інформаційних ресурсів, технологій, технічних засобів, методів та персоналу, що забезпечують процеси збирання, зберігання, обробки та розповсюдження інформації.

Згідно з системним підходом, ІС володіє фундаментальними властивостями:

- **Емерджентність:** система в цілому має властивості, яких не має жоден з її елементів окремо (комп'ютер без ПЗ і даних не здатний видавати звіти).
- **Гомеостаз:** здатність системи підтримувати стабільність функціонування в умовах змінного зовнішнього середовища (наприклад, резервне копіювання даних при збоях).

Важливо: ІС — це ширше поняття, ніж комп'ютерна мережа. ІС включає в себе людей (користувачів, адміністраторів, аналітиків), регламенти та бізнес-процеси організації.

5. Інформаційні технології (ІТ) як інструментальний базис

Якщо інформаційна система — це середовище, то **Інформаційна технологія (ІТ)** — це сукупність методів, виробничих та програмно-технологічних засобів, об'єднаних у технологічний ланцюжок, що забезпечує цілеспрямований процес переробки даних.

Мета ІТ — зниження трудомісткості використання інформаційних ресурсів та підвищення їхньої надійності. Технології поділяються на:

- *Забезпечувальні ІТ* (технології баз даних, мережеві протоколи, технології захисту).
- *Функціональні ІТ* (спрямовані на вирішення конкретних задач — текстові процесори, системи комп'ютерного моделювання (CAD), бухгалтерські системи).

6. Інформаційне суспільство та роль Computer Science

Розвиток обчислювальної техніки спричинив цивілізаційний зсув, описаний соціологами (Е. Тоффлер, Д. Белл) як перехід до постіндустріального, а згодом — до **інформаційного суспільства**.

Інформаційне суспільство — це форма соціально-економічної організації, за якої виробництво, зберігання, поширення та використання інформації і знань стають головним джерелом багатства, зайнятості та конкурентоспроможності.

Критерії інформаційного суспільства:

1. *Технологічний*: широке впровадження ІТ у всі сфери виробництва, освіти, побуту (ІоТ, Smart Cities).
2. *Економічний*: інформація стає ключовим економічним ресурсом і товаром; зростання частки сфери послуг та ІТ-сектору у ВВП.
3. *Соціальний*: зміна структури зайнятості. Формування класу «інтелектуальних працівників» (knowledge workers).
4. *Культурний*: визнання інформаційної цінності на рівні суспільної моралі, забезпечення вільного доступу до знань.

The role of computer science in the development of society полягає у створенні технологічного фундаменту для цього переходу: від глобальної комунікації (Інтернет) до автоматизації наукових відкриттів (розшифровка геному людини за допомогою алгоритмів Big Data).

7. Інформаційна культура та кібергігієна

В умовах глобальної цифровізації на перше місце виходить **інформаційна культура особистості** — система знань, навичок та морально-етичних норм, які забезпечують ефективну взаємодію людини з інформаційним середовищем.

Для фахівця з Computer Science складовими цієї культури є:

- **Алгоритмічне мислення**: здатність розбивати складні проблеми на послідовність формалізованих кроків.
- **Кібергігієна та безпека**: глибоке розуміння методів захисту даних (шифрування, двофакторна автентифікація, виявлення фішингу, Zero Trust architecture).
- **Медіаграмотність**: критичний аналіз джерел інформації, вміння виявляти дезінформацію (deepfakes).
- **Цифрова етика**: дотримання авторських прав (ліцензії на ПЗ, Open Source), повага до приватності користувачів (Privacy by Design), відповідальне використання систем штучного інтелекту.

8. Перспективи розвитку ІТ на період 2025–2035 років

Сучасний етап розвитку комп'ютерних наук визначається наступними глобальними трендами:

1. **Повсюдний штучний інтелект (Generative AI & AGI):** Перехід від систем, що класифікують дані, до автономних агентів, здатних створювати новий код, архітектури та вирішувати комплексні завдання без втручання людини.

2. **Edge Computing (Периферійні обчислення):** Обробка даних максимально близько до їхнього джерела (на датчиках або мобільних пристроях), що зменшує навантаження на мережу та забезпечує нульову затримку.

3. **Квантові комунікації та постквантова криптографія:** Створення систем зв'язку, захищених законами квантової фізики, і розробка нових алгоритмів шифрування, стійких до атак квантових комп'ютерів.

4. **Симбіоз людини і машини (Web 4.0 / Web 5.0):** Розвиток нейрокомп'ютерних інтерфейсів (BCI), просторових обчислень (Spatial Computing) та повного злиття фізичної та доповненої реальності (AR/VR/MR).

Питання для обговорення та контролю:

1. Охарактеризуйте ключові відмінності між комп'ютерами третього та четвертого поколінь. Як мікропроцесор змінив парадигму використання ЕОМ?

2. Чому інформацію не можна вважати матеріальним об'єктом, але водночас вона має матеріальну цінність? Обґрунтуйте з економічної та фізичної точок зору.

3. Проаналізуйте концепцію «інформаційного суспільства». Які негативні соціальні наслідки (наприклад, «цифрова нерівність», «кліпове мислення») виникають на цьому етапі розвитку людства?

4. Як розвиток систем генеративного штучного інтелекту у 2025-2026 роках впливає на поняття "інформаційна етика" та "авторське право"?

5. Доведіть або спростуйте твердження: «Будь-яка інформаційна технологія може існувати лише в межах певної інформаційної системи».

ТЕМА 1.2. СУТНІСТЬ ІНФОРМАЦІЙНИХ СИСТЕМ

Мета: Зрозуміти апаратно-програмну архітектуру інформаційних систем та їхню еволюцію до хмарних технологій.

План лекції:

1. Поняття автоматизованої інформаційної системи (АІС): визначення, мета, завдання.

2. Архітектура та базові принципи побудови АІС (забезпечувальні підсистеми).

3. Еволюція та покоління автоматизованих інформаційних систем (від 1950-х до 2026 року).

4. Розвиток офісних інформаційних систем (ОІС): від локальних програм до хмарних екосистем.

5. Технології електронного документообігу та автоматизації колективної роботи (Groupware).

1. Поняття автоматизованої інформаційної системи (АІС)

У сучасній теорії комп'ютерних наук критично важливо розрізняти поняття «Інформаційна система» (ІС) та «Автоматизована інформаційна система» (АІС).

Будь-яка картотека в бібліотеці, розклад руху поїздів на паперовому стенді або телефонний довідник є інформаційними системами. Однак об'єктом вивчення нашої дисципліни є саме АІС — **організаційно-технічна система, в якій реалізується технологія автоматизованої (за допомогою ЕОМ) обробки інформації з метою задоволення інформаційних потреб користувачів.**

Головна мета АІС — зниження трудовитрат на обробку даних, підвищення оперативності та якості прийняття управлінських рішень. Автоматизація означає, що рутинні операції (пошук, сортування, математичні розрахунки, формування звітів) виконує комп'ютер, тоді як за людиною залишається функція цілепокладання, аналізу нестандартних ситуацій та прийняття остаточного рішення. Система називається автоматизованою (а не автоматичною), оскільки в її контурі управління обов'язково присутня людина.

Основні завдання АІС:

- Безперервний збір та реєстрація первинної інформації.
- Передача інформації від джерел виникнення до місця обробки.
- Логічна та математична обробка даних за заданими алгоритмами.
- Накопичення, надійне зберігання та актуалізація баз даних.
- Захист даних від несанкціонованого доступу, спотворення чи знищення.
- Видача результатної інформації у зручному для сприйняття вигляді (таблиці, дашборди, графіки).

2. Архітектура та забезпечувальні підсистеми АІС

Для того щоб АІС функціонувала як єдиний організм, вона проектується на основі модульного принципу і складається з **забезпечувальних підсистем:**

1. **Технічне забезпечення (Hardware):** комплекс технічних засобів (КТЗ), що включає сервери, персональні комп'ютери, периферійні пристрої, засоби зв'язку та мережеве обладнання. У сьогоденні сюди також відносять сенсори Інтернету речей (IoT) та мобільні термінали.

2. **Програмне забезпечення (Software):** поділяється на загальносистемне (операційні системи, утиліти) та спеціальне/прикладне (програми, створені для вирішення конкретних задач даної АІС).

3. **Інформаційне забезпечення (Data/Information):** єдина система класифікації та кодування інформації, уніфіковані системи документації, схеми інформаційних потоків, а також самі бази та сховища даних (Data Warehouses).

4. **Математичне та алгоритмічне забезпечення:** сукупність математичних методів, моделей та алгоритмів, що застосовуються для розв'язання задач управління та обробки даних (наприклад, алгоритми оптимізації логістики або нейромереві моделі для прогнозування).

5. **Лінгвістичне забезпечення:** сукупність мовних засобів, що використовуються в системі (мови програмування, мови запитів до БД типу SQL, мови розмітки, інтерфейсні словники).

6. **Організаційно-правове забезпечення:** інструкції, регламенти, статuti, накази та посадові обов'язки персоналу, що регламентують порядок створення, впровадження та експлуатації АІС, а також юридичний статус електронних документів.

7. **Ергономічне забезпечення:** сукупність методів і засобів, призначених для створення оптимальних умов праці людини в системі (зручність інтерфейсів користувача — UI/UX).

3. Покоління АІС

Розвиток АІС відбувався паралельно з еволюцією обчислювальної техніки та управлінських парадигм:

- **Перше покоління (1950–1960-ті рр.) — Системи обробки транзакцій (TPS - Transaction Processing Systems).** Базувалися на мейнфреймах. Основна функція — автоматизація рутинних паперових робіт: нарахування заробітної плати, бухгалтерський облік, обробка банківських чеків. Характеризувалися пакетним режимом обробки (дані накопичувалися і оброблялися раз на день/тиждень).

- **Друге покоління (1970–1980-ті рр.) — Управлінські інформаційні системи (MIS - Management Information Systems).** З появою міні-ЕОМ та перших СУБД (наприклад, ієрархічних та мережевих) фокус змістився на забезпечення менеджерів регулярними звітами для контролю за діяльністю підприємства.

- **Третє покоління (1980–1990-ті рр.) — Системи підтримки прийняття рішень (DSS - Decision Support Systems).** Поява персональних комп'ютерів дозволила створити системи, здатні моделювати ситуації типу «Що буде, якщо...?». У цей час також з'являються Експертні системи (ES), що використовують бази знань для вирішення вузькоспеціалізованих задач.

- **Четверте покоління (2000–2010-ті рр.) — Інтегровані корпоративні системи (ERP, CRM, SCM).** Епоха Інтернету та клієнт-серверної архітектури. АІС охоплюють усі підрозділи компанії, об'єднуючи управління фінансами, виробництвом, кадрами та відносинами з клієнтами в єдиний інформаційний простір у режимі реального часу.

- **П'яте покоління (2020-ті рр. і далі) — Інтелектуальні та хмарні АІС (AI-driven & Cloud-native).** Сучасні системи характеризуються відмовою від власних фізичних серверів на користь хмарних сервісів (IaaS, PaaS, SaaS). Інтеграція алгоритмів машинного навчання дозволяє системам самостійно аналізувати Великі Дані (Big Data), знаходити аномалії, прогнозувати попит та автоматизувати складні бізнес-процеси (RPA - Robotic Process Automation).

4. Розвиток офісних інформаційних систем (ОІС)

Офісна інформаційна система (ОІС) — це різновид АІС, призначений для забезпечення комунікацій, автоматизації документообігу та підтримки колективної роботи співробітників як усередині організації, так і із зовнішнім середовищем.

Еволюція ОІС:

1. **Автономні інструменти (1980–1990-ті):** Окремі текстові редактори (WordPerfect, ранній MS Word) та електронні таблиці (VisiCalc, Lotus 1-2-3). Файли передавалися на дискетах (так званий «Sneakernet»).

2. **Локальні офісні пакети (1990–2010-ті):** Інтегровані пакети (Microsoft Office), де програми можуть обмінюватися даними (наприклад, технологія OLE). Поява локальних мереж, електронної пошти та спільних мережеских папок.

3. **Хмарні та колаборативні платформи (сучасний етап):** Перехід від концепції «файл на жорсткому диску» до «документ у хмарі». Системи типу Google Workspace та Microsoft 365 дозволяють десяткам людей одночасно редагувати один документ, бачити зміни в реальному часі та комунікувати через вбудовані відеоконференції.

Ключові технології сучасного офісу:

- **Groupware (Програмне забезпечення для групової роботи):** координація завдань, спільні календарі, корпоративні месенджери (Slack, MS Teams).

- **Системи електронного документообігу (СЕД):** автоматизація життєвого циклу документа (створення, узгодження, підписання електронним цифровим підписом (ЕЦП), архівування). Дозволяють повністю відмовитися від паперу та прискорити рух інформації в сотні разів.

- **Unified Communications (Уніфіковані комунікації):** інтеграція всіх засобів зв'язку (IP-телефонія, відеозв'язок, email) в єдиний інтерфейс.

Питання для обговорення та контролю:

1. Які підсистеми АІС є найбільш вразливими до помилок і чому? Проаналізуйте роль "людського фактора".

2. Чим принципово відрізняються системи підтримки прийняття рішень (DSS) від систем обробки транзакцій (TPS)? Наведіть приклади з реального бізнесу.

3. Опишіть переваги та недоліки переходу підприємства від використання локальних офісних пакетів до хмарних платформ (SaaS).

4. Як розвиток технологій електронного цифрового підпису (ЕЦП) та систем типу "Дія" вплинув на еволюцію систем документообігу в Україні?

5. Яким ви бачите офіс майбутнього (через 10 років)? Які рутинні процеси будуть повністю передані під управління штучного інтелекту?

ТЕМА 1.3. КОРПОРАТИВНІ ІНФОРМАЦІЙНІ СИСТЕМИ

Мета: Ознайомити здобувачів вищої освіти з принципами класифікації та архітектурною структурою інформаційних систем, а також розкрити сутність, призначення та інженерні особливості розробки, розгортання та інтеграції корпоративних інформаційних систем (КІС).

План лекції:

1. Класифікація інформаційних систем за масштабом, архітектурою та режимом обробки даних.
2. Структура ІС: логічна (багаторівнева) та фізична (топологія розгортання).
3. Корпоративні інформаційні системи (КІС): сутність, еволюція та основні класи (ERP, CRM, SCM).
4. Характерні ознаки, інженерні вимоги та методи інтеграції корпоративних систем.

1. Класифікація інформаційних систем

Для правильного вибору технологічного стеку та патернів проектування при розробці програмного забезпечення, ІТ-фахівець повинен чітко розуміти місце майбутнього продукту в загальній класифікації інформаційних систем. ІС класифікують за кількома базовими критеріями:

За масштабом (сферою охоплення):

- **Персональні ІС:** призначені для одного користувача, працюють локально, зазвичай не потребують складної мережевої взаємодії.
- **Групові ІС (Departmental IS):** автоматизують роботу окремого відділу. Використовують архітектуру клієнт-сервер у межах локальної мережі (LAN) та спільну базу даних.
- **Корпоративні ІС (Enterprise IS):** розподілені системи, що охоплюють усю організацію, об'єднуючи територіально віддалені філії та тисячі користувачів у єдиний інформаційний простір.

За режимом обробки даних:

- **Пакетна обробка (Batch Processing):** дані накопичуються і обробляються періодично великими порціями (наприклад, генерація щомісячних фінансових звітів).
- **Обробка в реальному часі (Real-time Processing / OLTP):** кожна транзакція обробляється негайно після надходження (наприклад, бронювання квитків, банківські перекази).
- **Аналітична обробка (OLAP):** системи, оптимізовані для виконання складних запитів до великих масивів історичних даних (сховищ даних) для пошуку закономірностей.

За типом архітектури програмного забезпечення:

- **Монолітні (Monolithic):** уся система (інтерфейс, бізнес-логіка, доступ до даних) скомпільована і розгортається як єдиний цілісний блок. Легше тестувати на початкових етапах, але важко масштабувати.

- **Сервіс-орієнтована архітектура (SOA):** система будується з набору слабкозв'язаних сервісів, які спілкуються між собою за чітко визначеними протоколами (наприклад, SOAP).
- **Мікросервісні (Microservices / Cloud-native):** сучасний еволюційний крок SOA. Система розбивається на десятки або сотні незалежних міні-додатків (мікросервісів), кожен з яких відповідає за одну функцію, має власну базу даних і може бути написаний на різних мовах програмування.

2. Структура інформаційної системи: логічна та фізична

При проектуванні інформаційних систем в інженерії програмного забезпечення розрізняють логічну (як код розділений на функціональні блоки) та фізичну (на якому "залізі" цей код працює) структури.

Логічна структура (N-tier architecture) Найчастіше реалізується через **трирівневу архітектуру (3-tier architecture)**, яка дозволяє незалежно оновлювати різні частини коду:

1. **Рівень представлення (Presentation Layer / UI):** Відповідає за взаємодію з користувачем. Реалізується через веб-фреймворки (React, Angular) або мобільні клієнти.

2. **Рівень бізнес-логіки (Business Logic Layer / Application Tier):** "Мозок" системи. Тут реалізуються математичні розрахунки, алгоритми перевірки даних, маршрутизація запитів. *Технології:* Java (Spring), Python (Django/FastAPI), C# (.NET Core).

3. **Рівень доступу до даних (Data Layer):** Відповідає за взаємодію із СУБД, кешування та виконання транзакцій. *Технології:* PostgreSQL, MySQL, Redis, MongoDB.

Фізична структура (Топологія розгортання) Описує, як логічні компоненти розподілені по реальних або віртуальних вузлах (серверах, контейнерах).

- У найпростішому випадку всі три логічні рівні можуть працювати на одному фізичному сервері (що неприпустимо для корпоративних систем).
- У сучасних ІС рівень представлення завантажується в браузер клієнта, бізнес-логіка розгортається у кластері контейнерів (наприклад, Kubernetes) у хмарі AWS чи Azure, а база даних розноситься на окремі потужні сервери з реплікацією.

3. Корпоративні інформаційні системи (KIC): сутність та класи

Корпоративна інформаційна система (Enterprise Information System - EIS) — це комплексна, масштабована та інтегрована програмно-апаратна платформа, яка консолідує управління всіма бізнес-процесами організації на базі єдиного сховища даних.

Головна мета впровадження KIC — ліквідація **"інформаційних колодязів" (data silos)** (ситуацій, коли відділи ізольовані і використовують несумісні локальні програми) та забезпечення **Єдиного джерела правди (Single Source of Truth)**.

Основні класи корпоративних систем:

- **ERP (Enterprise Resource Planning - Планування ресурсів підприємства):** Ядро будь-якої КІС. Об'єднує управління фінансами, виробництвом, закупівлями та збутом. Зміна даних у модулі складу миттєво відображається в модулі бухгалтерії без ручного перенесення.
- **CRM (Customer Relationship Management - Управління відносинами з клієнтами):** Системи для автоматизації воронки продажів, маркетингу та служби підтримки. Зберігають всю історію взаємодії з кожним клієнтом.
- **SCM (Supply Chain Management - Управління ланцюгами поставок):** Оптимізують логістику, управління запасами та взаємодію з постачальниками матеріалів.
- **HRM (Human Resource Management):** Системи кадрового обліку, нарахування зарплат, трекінгу робочого часу та управління талантами.

4. Характерні ознаки та інтеграція корпоративних систем

Розробка корпоративного програмного забезпечення (Enterprise Software Development) суттєво відрізняється від створення звичайних веб-сайтів. Для КІС характерні такі визначальні інженерні ознаки:

1. **Масштабованість (Scalability) та Відмовостійкість (Fault Tolerance):** КІС проектується для роботи в режимі 24/7. Вони підтримують *High Availability* (високу доступність) завдяки резервуванню серверів, балансуванню навантаження (Load Balancing) та реплікації баз даних.

2. **Управління основними даними (Master Data Management - MDM):** Механізми, що гарантують чистоту, актуальність та відсутність дублікатів у критично важливих довідниках компанії (наприклад, єдиний довідник контрагентів для всіх філій).

3. **Глибока інтеграбельність (EAI - Enterprise Application Integration):** КІС рідко існує у вакуумі. Вона повинна обмінюватися даними з державними реєстрами, банками (через відкриті API), поштовими сервісами.

Підходи до інтеграції КІС:

- *Інтеграція «точка-точка» (Point-to-Point):* пряме написання коду для зв'язку кожної системи з кожною. Призводить до "спагеті-архітектури" при великій кількості систем.

- *Корпоративна сервісна шина (Enterprise Service Bus - ESB):* централізований архітектурний компонент, який виступає посередником. Всі системи підключаються тільки до ESB, яка сама займається маршрутизацією, перетворенням форматів даних (наприклад, з XML у JSON) та гарантованою доставкою повідомлень.

- *Подійно-орієнтована архітектура (Event-Driven Architecture):* системи спілкуються через брокери повідомлень (Apache Kafka, RabbitMQ). Наприклад, модуль продажів генерує подію "Товар продано", а модулі складу та бухгалтерії автоматично реагують на цю подію.

Питання для обговорення та контролю:

1. У чому полягають архітектурні недоліки монолітних систем при їх масштабуванні порівняно з мікросервісними підходами?
2. Поясніть різницю між логічною 3-рівневою архітектурою та фізичною топологією сервера. Наведіть приклад розгортання.
3. Що таке "інформаційні колодязі" (data silos)? Як архітектура спільної бази даних у системах ERP вирішує цю проблему?
4. Яку роль відіграють брокери повідомлень (наприклад, RabbitMQ або Apache Kafka) у побудові сучасних корпоративних інформаційних систем?
5. Чому для інтеграції великої кількості внутрішніх систем компанії використання інтеграції «точка-точка» вважається антипатерном (поганою практикою)?

ТЕМА 1.4. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В СУЧАСНОМУ БІЗНЕСІ

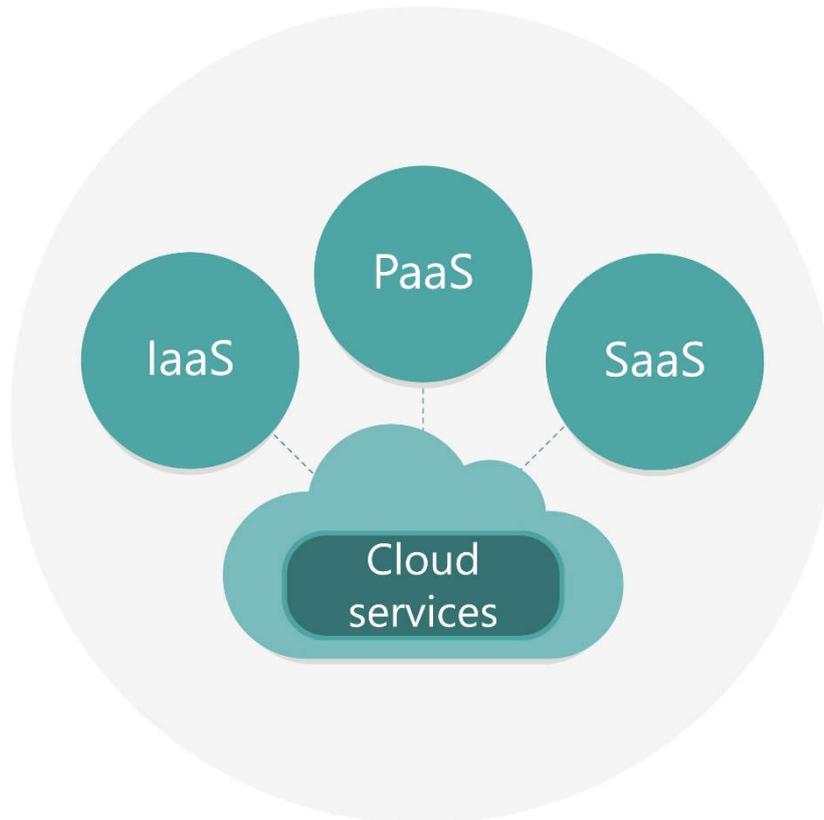
Мета: Ознайомити здобувачів вищої освіти із сучасними інформаційними технологіями, які використовує бізнес: хмарними обчисленнями, принципами взаємодії через API, основами роботи з великими даними (Big Data) та впровадженням штучного інтелекту.

План лекції:

1. Хмарні обчислення та перехід до контейнеризації програмного забезпечення.
2. API-економіка: як різні системи спілкуються між собою.
3. Технології Великих Даних (Big Data) та бізнес-аналітика (BI).
4. Штучний інтелект (AI) та роботизована автоматизація рутини (RPA).
5. Основи DevOps: як сучасний код потрапляє до користувача (CI/CD).
6. Сучасні підходи до корпоративної безпеки (Zero Trust) та блокчейн.

1. Хмарні обчислення та контейнеризація

Сучасний бізнес масово відмовляється від власних фізичних серверних кімнат на користь **Cloud Computing** — оренди обчислювальних потужностей через Інтернет (AWS, Google Cloud, Microsoft Azure).



Основні моделі хмарних послуг:

- **IaaS (Інфраструктура як послуга):** оренда віртуального "заліза". Ви отримуєте порожній сервер і самі встановлюєте операційну систему та бази даних.
- **PaaS (Платформа як послуга):** хмара надає готове середовище для запуску коду. Розробнику не треба думати про налаштування сервера — він просто завантажує туди свій застосунок.
- **SaaS (Програмне забезпечення як послуга):** повністю готовий до використання продукт (наприклад, Google Workspace або CRM-система).

Контейнеризація (на прикладі Docker): Замість того, щоб створювати "важкі" віртуальні машини з повноцінною операційною системою для кожної програми, сучасні інженери пакують код у легкі **контейнери**. Контейнер містить лише саму програму та необхідні для неї бібліотеки. Це дозволяє запускати додаток однаково стабільно як на ноутбучі розробника, так і на бойовому сервері.

2. API-економіка та інтеграція систем

Сучасні ІТ-продукти рідко створюються з нуля. Бізнес збирає свої сервіси як конструктор Lego, використовуючи сторонні рішення (наприклад, підключення платіжного шлюзу від банку або карт від Google). Це відбувається завдяки **API (Application Programming Interface)**.

API — це набір правил та інструкцій, за якими дві різні програми спілкуються між собою.

- Найпопулярнішим стандартом побудови API сьогодні є **REST**. Він використовує звичайні веб-запити (як ті, що робить ваш браузер) для передачі даних, найчастіше у легкому текстовому форматі JSON.
- Новіший підхід — **GraphQL**, який дозволяє клієнту (наприклад, мобільному додатку) запитати у сервера тільки ту інформацію, яка потрібна в даний момент, щоб не перевантажувати мережу зайвими даними.

3. Технології Великих Даних (Big Data) та Business Intelligence

Щохвилини бізнес генерує гігабайти інформації. Щоб ці дані приносили гроші, їх потрібно правильно зберігати та аналізувати.

- **Озера даних (Data Lakes):** величезні цифрові "сховища", куди звальюються всі сирі, неструктуровані дані компанії (тексти, відео, логи з сайтів) "як є", доки вони не знадобляться.
- **Сховища даних (Data Warehouses):** бази даних, куди інформація потрапляє вже очищеною, відсортованою та підготовленою до аналізу у вигляді чітких таблиць.

Business Intelligence (BI) — це системи (наприклад, Power BI або Tableau), які підключаються до цих сховищ і автоматично будують зрозумілі графіки та дашборди. Це дозволяє керівникам приймати рішення на основі реальних цифр, а не інтуїції.

4. Штучний інтелект (AI) та автоматизація (RPA)

- **RPA (Robotic Process Automation):** це створення "програмних роботів" (скриптів), які імітують дії людини за комп'ютером. Вони можуть самостійно відкривати листи, копіювати дані з PDF-рахунків і вставляти їх у бухгалтерську програму 24/7 без помилок.
- **Штучний інтелект (AI):** інтеграція нейромереж та великих мовних моделей (LLM) у бізнес. ШІ використовується для створення розумних чат-ботів у техпідтримці, прогнозування попиту на товари та автоматичного аналізу відгуків клієнтів.

5. Основи DevOps: як код потрапляє до користувача

Раніше програмісти писали код, а потім передавали його системним адміністраторам для встановлення на сервер. Це часто призводило до помилок і затримок.

Сучасний підхід називається **DevOps** — це об'єднання розробки (Development) та експлуатації (Operations). Його головний інструмент — автоматизований конвеєр **CI/CD**:

- Як тільки програміст завершує писати частину коду, система автоматично його перевіряє (тестує).

- Якщо помилок немає, код автоматично збирається в контейнер і розгортається (оновлюється) на серверах. Завдяки цьому користувачі отримують оновлення додатків непомітно і без зупинки роботи сервісу.

6. Сучасна корпоративна безпека та блокчейн

- **Zero Trust (Нульова довіра):** класична безпека будувалася як фортеця (товсті стіни-фаєрволи, але всередині всім можна довіряти). Архітектура Zero Trust означає "не довіряй нікому". Навіть якщо співробітник знаходиться в офісі компанії, система все одно вимагатиме підтвердження особи (наприклад, код на телефон) при доступі до кожного нового документа чи сервісу.

- **Блокчейн у бізнесі:** це не тільки про криптовалюти. Блокчейн — це база даних, у якій неможливо змінити або видалити минулі записи. Це ідеально підходить для логістики (відстеження походження товарів) або укладання "смарт-контрактів", які виконуються автоматично при настанні певних умов.

Питання для обговорення та контролю:

1. Поясніть на прикладах з життя різницю між моделями IaaS, PaaS та SaaS. Що з цього найбільше підходить для звичайного користувача, а що — для розробника?

2. Як ви розумієте поняття API? Наведіть приклади, як відомі сервіси (наприклад, Uber чи Glovo) використовують API сторонніх розробників.

3. Чим "Озеро даних" (Data Lake) відрізняється від "Сховища даних" (Data Warehouse)?

4. Які рутинні процеси у сучасному банку або університеті можна повністю доручити технологіям RPA (програмним роботам)?

5. У чому полягає філософія інформаційної безпеки "Zero Trust" і чому старі методи захисту мереж сьогодні не працюють?

ТЕМА 1.5. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ НА ПІДПРИЄМСТВІ

Мета: Ознайомити здобувачів вищої освіти з етапами життєвого циклу розробки інформаційних систем, методологією їх вибору, роллю стейкхолдерів у процесі проектування та сучасним ринком корпоративного програмного забезпечення.

План лекції:

1. Життєвий цикл інформаційної системи (SDLC) та методології розробки.

2. Роль замовника (стейкхолдерів) та бізнес-аналізу у створенні ІС.

3. Критерії та інженерні методи вибору корпоративної системи (Build vs. Buy).

4. Огляд сучасного ринку програмного забезпечення для управління підприємством.

1. Життєвий цикл інформаційної системи (SDLC)

Створення інформаційної системи — це не хаотичне написання коду, а суворо регламентований інженерний процес, який називається **SDLC (Software Development Life Cycle — Життєвий цикл розробки ПЗ)**. Він складається з таких базових етапів:

1. **Збір та аналіз вимог (Requirements Analysis):** визначення того, що саме має робити система.
2. **Проектування архітектури (Design):** вибір технологічного стеку, баз даних, розробка UI/UX.
3. **Розробка (Implementation/Coding):** безпосереднє написання коду програмістами.
4. **Тестування (Testing / QA):** пошук багів, перевірка безпеки та стійкості до навантажень.
5. **Впровадження (Deployment):** розгортання системи на серверах та перехід компанії на нове ПЗ.
6. **Експлуатація та супровід (Maintenance):** технічна підтримка, випуск оновлень (патчів) та моніторинг.

Методології розробки:

- **Каскадна (Waterfall):** лінійний підхід, де кожен наступний етап починається лише після повного завершення попереднього. Підходить для систем з критичними вимогами до безпеки (авіація, медицина), де помилка коштує надто дорого.
- **Гнучка (Agile / Scrum / Kanban):** розробка ведеться короткими ітераціями (спринтами по 2-4 тижні). Після кожного спринту замовник отримує робочий шматок продукту (MVP). Це сучасний стандарт для більшості IT-проектів.

2. Роль замовника в створенні інформаційної системи

Найбільша проблема в IT — це коли розробники створюють ідеальну з технічної точки зору систему, яка абсолютно не потрібна бізнесу.

Замовник (Стейкхолдер) — це ключова фігура, яка фінансує проект і визначає його бізнес-цілі. В методології Agile замовника часто представляє **Product Owner (Власник продукту)**.

- **Проблема "Труднощів перекладу":** Бізнес розмовляє мовою грошей, продажів та процесів. Програмісти розмовляють мовою API, баз даних та фреймворків.
- **Рішення:** Між замовником і розробниками завжди стоїть **Бізнес-аналітик (Business Analyst - BA)**. Його завдання — вислухати "болі" замовника і перетворити їх на чіткі технічні специфікації (Software Requirements Specification - SRS) або користувацькі історії (User Stories), які зрозуміють програмісти.

3. Критерії і методи вибору корпоративної ІС

Коли підприємство вирішує автоматизувати свої процеси, перше стратегічне питання, яке постає перед ІТ-архітектором: **Build vs. Buy** (Розробляти власне чи купити готове?).

Власна розробка (Custom Development):

- *Плюси:* Ідеально підходить під унікальні бізнес-процеси компанії (наприклад, алгоритми підбору авто для Uber).
- *Мінуси:* Дуже довго, дорого, високий ризик невдачі проекту, повна залежність від власної команди розробників.

Купівля готового рішення (наприклад, впровадження готової ERP-системи): Якщо компанія вирішує купувати готове ПЗ, вибір базується на таких інженерних та фінансових критеріях:

1. **ТСО (Total Cost of Ownership - Сукупна вартість володіння):** витрати не лише на купівлю ліцензій, але й на сервери, інтеграцію, навчання персоналу та щорічну технічну підтримку впродовж 5-10 років. Вартість ліцензії — це часто лише "верхівка айсберга".
2. **Масштабованість та Гнучкість (Scalability & Customization):** чи зможе система працювати без збоїв, якщо компанія виросте в 5 разів?
3. **Наявність відкритого API:** наскільки легко цю систему буде інтегрувати з іншими сервісами компанії.
4. **Ризик Vendor Lock-in (Залежність від постачальника):** наскільки важко буде перенести свої дані в іншу систему, якщо поточний постачальник ПЗ різко підніме ціни.

4. Огляд ринку ПЗ для управління підприємством

Ринок корпоративних систем (КІС) сьогодні глибоко трансформований переходом у хмару та впровадженням ШІ (Штучного Інтелекту). Сучасний тренд — **Composable ERP** (модульні системи, де підприємство збирає потрібний функціонал з різних блоків, як конструктор).

Світові лідери ринку (Global Enterprise):

- **SAP (Німеччина):** Абсолютний лідер серед великих корпорацій (заводи, глобальна логістика). Дуже потужна, але надзвичайно дорога і складна у впровадженні система.
- **Oracle (США):** Лідер у системах управління базами даних та фінансовому секторі.
- **Microsoft Dynamics 365:** Популярна завдяки безшовній інтеграції з екосистемою Microsoft (Office, Teams, Azure).
- **Salesforce:** Світовий стандарт №1 у сфері CRM (управління клієнтами) як хмарний сервіс (SaaS).

Український контекст: Критичною зміною останніх років в Україні стала повна відмова від застарілих російських систем (типу 1С/BAS) через загрози національній кібербезпеці та санкції. Їх місце зайняли:

- *Локальні розробки:* **IT-Enterprise** (потужна українська ERP для великого виробництва), **VJet**, **Дебет Плюс**.

- *Міжнародні гнучкі системи:* **Odoo** (популярна open-source бельгійська система для малого і середнього бізнесу, що активно впроваджується в Україні), **Creatio** (від компанії Terrasoft, лідер у Low-code автоматизації).

Питання для обговорення та контролю:

1. Чому методологія розробки Waterfall поступово витісняється підходами Agile у більшості сучасних IT-проектів? Які плюси та мінуси для замовника?
2. Поясніть принцип "Garbage in, garbage out" ("сміття на вході — сміття на виході") в контексті етапу збору вимог (Requirements Analysis).
3. Що таке сукупна вартість володіння (ТСО)? Чому купівля дешевої ліцензії на програму може обійтися компанії дорожче у перспективі 5 років?
4. Уявіть, що ви — IT-архітектор банку. Який шлях ви оберете для створення нового мобільного банкінгу: Build (писати з нуля) чи Buy (купити готове ядро)? Обґрунтуйте.
5. Як змінився ринок корпоративного ПЗ в Україні після відмови від російського софту? Назвіть сучасні альтернативи для українського бізнесу.

ЗМІСТОВИЙ МОДУЛЬ 2. АПАРАТНІ, ПРОГРАМНІ І СИСТЕМНІ ЗАСОБИ ІНФОРМАЦІЙНИХ СИСТЕМ.

ТЕМА 2.1. АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ.

Мета: Ознайомити здобувачів вищої освіти з глибинною архітектурою сучасних обчислювальних систем, апаратними принципами обробки графіки та периферійної взаємодії, детально розібрати життєвий цикл адитивного виробництва (3D-друку), а також сформуванню фундаментального розуміння арифметико-логічних основ роботи мікропроцесорів.

План лекції:

1. Коротка характеристика основних частин персонального комп'ютера (ПК).
2. Відеосистема ПК.
3. Зовнішні пристрої ПК.
4. Технології, процеси та етапи тривимірного друку.
5. Сфери застосування 3D-друку.
6. Основні функціональні характеристики ПК.
7. Арифметичні основи комп'ютерної техніки.
8. Логічні основи комп'ютерної техніки.

1. Коротка характеристика основних частин персонального комп'ютера (ПК)

Апаратне забезпечення сучасного комп'ютера базується на **магістрально-модульному принципі**, який є еволюційним розвитком класичної архітектури фон Неймана. Цей принцип дозволяє будувати систему з окремих функціональних блоків (модулів), що обмінюються даними через спільні канали зв'язку — магістралі (шини).

Системний блок є ядром ПК і містить наступні базові вузли:

- **Материнська (системна) плата (Motherboard):** Багатошарова друкована плата, що виконує роль комунікаційного центру. Її найважливішим елементом є **чипсет (набір системної логіки)**. У сучасних системах більшість контролерів (наприклад, контролер оперативної пам'яті та ліній PCI Express для відеокарти) перенесено безпосередньо в процесор, а на материнській платі залишається чип PCH (Platform Controller Hub) для керування повільнішою периферією (USB, SATA, аудіо).

- **Центральний процесор (CPU - Central Processing Unit):** «Мозок» системи, що виконує машинний код програм. Складається з:

- *Арифметико-логічного пристрою (АЛП):* виконує всі математичні та логічні перетворення даних.

- *Пристрою керування (ПК):* генерує керуючі сигнали для всіх інших вузлів, синхронізуючи їх роботу за допомогою тактового генератора.

- *Регістрів та Кеш-пам'яті (L1, L2, L3)*: надшвидкої внутрішньої пам'яті для мінімізації простоїв процесора під час очікування даних.
- **Внутрішня пам'ять:**
 - *Оперативна пам'ять (RAM)*: Енергозалежна динамічна пам'ять (сучасні стандарти DDR4, DDR5). Забезпечує процесор даними в режимі реального часу.
 - *Постійна пам'ять (ROM)*: Енергонезалежна пам'ять, що містить мікропрограми базової системи вводу-виводу (BIOS/UEFI), які ініціалізують обладнання під час увімкнення живлення.
- **Накопичувачі даних**: Сьогодні домінують **SSD (Solid-State Drives)** на базі флеш-пам'яті NAND. Найсучасніші з них використовують інтерфейс NVMe через шину PCIe, що дозволяє досягати швидкостей читання/запису у понад 7000 МБ/с, що кардинально перевищує можливості застарілих механічних жорстких дисків (HDD).

2. Відеосистема ПК

Відеосистема перейшла від ролі простого "адаптера для монітора" до статусу потужної обчислювальної підсистеми, що часто перевищує за складністю сам центральний процесор.

- **Архітектура відеокарт**: Сучасна відеокарта є, по суті, окремим комп'ютером всередині комп'ютера. Вона має власну плату, систему живлення, охолодження, відеопам'ять (VRAM стандартів GDDR6/GDDR6X) та **графічний процесор (GPU)**.
- **CPU проти GPU**: Якщо центральний процесор (CPU) має кілька (від 4 до 24) дуже складних і потужних ядер для виконання послідовних задач, то графічний процесор (GPU) містить тисячі простіших ядер (наприклад, CUDA-ядер у NVIDIA). Це дозволяє GPU реалізовувати **масовий паралелізм** — одночасно розраховувати колір мільйонів пікселів або обробляти гігантські матриці даних.
- **Технології та GPGPU**: Окрім рендерингу графіки (зокрема за допомогою апаратного трасування променів — Ray Tracing), сучасні відеосистеми активно використовуються для неграфічних обчислень (GPGPU): навчання нейромереж, майнінг криптовалют, наукове моделювання складних фізичних процесів.

3. Зовнішні пристрої ПК

Зовнішні (периферійні) пристрої підключаються через стандартизовані інтерфейси (USB-C, Thunderbolt, Bluetooth, Wi-Fi) і розширюють можливості базової системи.

Детальна класифікація:

1. **Пристрої введення інформації**: * *Текстові та координатні*: Клавіатури (механічні, мембранні), миші (з оптичними сенсорами високої роздільної здатності DPI), графічні планшети (дигітайзери) для професійного малювання.

о *Аудіовізуальні*: Сканери, веб-камери з апаратним кодуванням відео, студійні мікрофони.

о *Датчики*: Біометричні сканери відбитків пальців, системи розпізнавання обличчя (Face ID).

2. Пристрої виведення інформації:

о *Монітори*: Базуються на матрицях IPS (точна передача кольору), VA (висока контрастність) або OLED (ідеальний чорний колір завдяки самопідсвічуванню пікселів).

о *Друкуючі пристрої*: Лазерні (використовують тонер та фотобарабан), струменеві (для високоякісного фотодруку), плотери (широкоформатний інженерний друк).

3. Засоби комунікації: Мережеві карти (Ethernet), бездротові адаптери.

Важливим програмним компонентом для роботи периферії є **драйвер** — утиліта, що перекладає команди операційної системи на зрозумілу для конкретного пристрою мову апаратних сигналів.

4. Технології, процеси та етапи тривимірного друку

Адитивне виробництво (3D-друк) — це парадигма створення фізичних об'єктів шляхом пошарового додавання матеріалу, на протигагу традиційному субтрактивному виробництву (фрезерування, токарна обробка), де зайвий матеріал відсікається.

Повний життєвий цикл 3D-друку:

1. **CAD-моделювання**: Створення об'єкта в системах автоматизованого проектування (SolidWorks, Blender, AutoCAD).

2. **Експорт у формат сітки**: Перетворення точної математичної моделі на полігональну сітку з трикутників (формати .STL або .OBJ).

3. **Слайсинг (Slicing)**: Критично важливий етап. Спеціальне програмне забезпечення (слайсер) розрізає модель на сотні горизонтальних шарів товщиною від 0.05 мм до 0.3 мм. Тут користувач задає відсоток внутрішнього заповнення (infill) для економії матеріалу та генерує підтримки для нависаючих елементів. На виході формується файл з **G-кодом**.

4. **Процес друку**: Мікроконтролер принтера порядково виконує G-код, керуючи кроковими двигунами та нагрівальними елементами.

5. **Постобробка**: Видалення підтримок, хімічне згладжування (наприклад, парами ацетону для ABS-пластику), ультрафіолетове дозасвічення (для смол).

Ключові технології:

- **FDM (Fused Deposition Modeling)**: Екструзія розплавленого термопластику (PLA, ABS, PETG) через сопло.

- **SLA/DLP (Стереолітографія)**: Фотополімеризація — перетворення рідкої смоли на твердий пластик під дією ультрафіолетового світла.

- **SLS (Selective Laser Sintering)**: Лазерне спікання полімерного або металевих порошків. Дозволяє друкувати об'єкти без структур підтримки.

5. Сфери застосування 3D друку

Адитивні технології стирають межу між цифровим світом та фізичною реальністю, маючи колосальний економічний вплив:

- **Медицина та стоматологія:** Виготовлення індивідуальних титанових імплантів кісток, що ідеально підходять пацієнту за результатами КТ. Друк прозорих елайнерів (кап) для вирівнювання зубів, біо-друк каркасів для вирощування тканин.
- **Аерокосмічна галузь:** Компанії (наприклад, SpaceX або GE) друкують форсунки ракетних та авіаційних двигунів з металевих порошків. Це дозволяє створювати деталі з внутрішніми каналами охолодження, які неможливо виготовити традиційним литтям.
- **Швидке прототипування (Rapid Prototyping) в інженерії:** Інженери можуть розробити деталь зранку, надрукувати її вдень і протестувати ввечері, що скорочує цикл розробки продукту з місяців до днів.
- **Архітектура та будівництво:** Використання промислових будівельних екструдерів для пошарового зведення стін будинків зі спеціальних бетонних сумішей.

6. Основні функціональні характеристики ПК

Продуктивність комп'ютера — це комплексний показник, який не можна оцінити лише за однією характеристикою. Основними метриками є:

- **Тактова частота (f_s):** Вимірюється в гігагерцах (ГГц). 1 ГГц означає один мільярд тактів (коливань) генератора за секунду. Втім, висока частота не завжди означає вищу швидкість.
- **IPC (Instructions Per Cycle):** Показник того, скільки корисних інструкцій процесор здатний виконати за один такт. Сучасний процесор з частотою 3 ГГц може бути значно швидшим за старий процесор із частотою 4 ГГц саме завдяки вищому IPC.
- **Багатоядерність та багатопотоковість (Hyper-Threading / SMT):** Фізичні ядра можуть ділитися на логічні потоки, що дозволяє процесору ефективніше розподіляти ресурси при одночасній роботі десятків фонових програм.
- **Розрядність (32 біт vs 64 біт):** Визначає ширину регістрів процесора. 64-бітна архітектура дозволяє за один такт обробляти вдвічі більше даних і знімає бар'єр у 4 ГБ максимально підтримуваної оперативної пам'яті (дозволяючи адресувати екзабайти пам'яті).
- **Пропускна здатність системної шини:** Розраховується як добуток тактової частоти шини на її ширину в байтах. Визначає, наскільки швидко дані можуть мігрувати між оперативною пам'яттю та процесором.

7. Арифметичні основи комп'ютерної техніки

Електронно-обчислювальні машини не сприймають десяткову систему числення, звичну для людини. На апаратному рівні використовується виключно **двійкова система числення (Binary)**.

- **Фізична суть:** Інформація кодується рівнями напруги. Наприклад, напруга від 0 до 0.4 Вольт сприймається системою як логічний «0», а напруга від 2.4 до 3.3 Вольт — як логічна «1». Це робиться для максимальної завадостійкості: навіть якщо сигнал трохи спотвориться через перешкоди, система безпомилково розпізнає, був це нуль чи одиниця.

- **Одиниці виміру:** Найменший елемент — **біт** (bit — binary digit). Вони групуються у **байти** (1 байт = 8 біт) та **машинні слова** (зазвичай 32 або 64 біти).

- **Арифметика:** Додавання у двійковій системі виконується за правилами стовпчика ($1+1=10$ у двійковій, де 0 записується, а 1 переноситься у старший розряд). Для віднімання чисел процесори використовують математичний трюк — **доповняльний код**. Це дозволяє замінити операцію віднімання на операцію додавання з від'ємним числом, що дає змогу використовувати ту саму електронну схему (суматор) для обох дій.

8. Логічні основи комп'ютерної техніки

Фундаментом апаратної логіки ПК є **Булева алгебра**. Процесор складається з мільярдів транзисторів, які об'єднуються у прості електронні схеми — **логічні вентиля (gates)**. Кожен вентиль реалізує певну логічну функцію.

Базові логічні операції:

1. **Кон'юнкція (Логічне І / AND):** $F = A \ \& \ B$. Сигнал на виході буде дорівнювати одиниці тоді й тільки тоді, коли на обох входах є одиниці.

2. **Диз'юнкція (Логічне АБО / OR):** $F = A \ \vee \ B$. На виході буде одиниця, якщо хоча б на одному з входів присутня одиниця.

3. **Інверсія (Логічне НЕ / NOT):** $F = \neg A$. Цей вентиль має один вхід. Він просто перевертає значення: з нуля робить одиницю, з одиниці — нуль.

4. **Виключне АБО (XOR):** $F = A \ \oplus \ B$. Видає одиницю, тільки якщо вхідні сигнали *різні* (один 0, інший 1). Саме цей вентиль є критично важливим для побудови електронних суматорів, що виконують арифметичні дії.

Комбінуючи ці базові вентиля, інженери-архітектори створюють тригери (комірки пам'яті), шифратори, дешифратори та, зрештою, мільярдні транзисторні масиви сучасних багатоядерних процесорів.

Питання для обговорення та контролю:

1. Поясніть роль "чипсету" на материнській платі. Як змінилася його архітектура у сучасних системах порівняно з ПК 10-річної давнини?

2. У чому полягає принципова архітектурна різниця між CPU та GPU? Наведіть приклади неграфічних задач (GPGPU), де відеокарта буде значно ефективнішою.

3. Опишіть етап "слайсінгу" під час підготовки до 3D-друку. Які параметри задаються на цьому етапі і як вони впливають на кінцевий виріб?
4. Чому показник тактової частоти не є абсолютним мірилом продуктивності процесора? Поясніть термін IPC.
5. Обґрунтуйте з фізичної та інженерної точок зору, чому обчислювальна техніка використовує двійкову систему числення (0 та 1), а не десяткову.
6. Напишіть таблицю істинності для логічної операції XOR. Як саме вона використовується в комп'ютерній арифметиці?

ТЕМА 2.2. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Мета: Розглянути сутність, багаторівневу структуру та класифікацію програмного забезпечення ПК, вивчити сучасні інструменти розробки, а також опанувати фундаментальні принципи захисту даних, резервного копіювання, архівації та протидії шкідливому коду.

План лекції:

1. Основні поняття, склад та структура програмного забезпечення ПК.
2. Класифікація: системне ПЗ, прикладне ПЗ, інструментальні засоби.
3. Захист інформації та стратегії резервного копіювання.
4. Принципи роботи з архівними файлами та алгоритми стиснення.
5. Загальні відомості про шкідливе ПЗ та антивірусні програми.

1. Основні поняття, склад та структура програмного забезпечення ПК

Якщо апаратне забезпечення (Hardware) — це «тіло» комп'ютера, то програмне забезпечення (Software) — це його «інтелект».

Програмне забезпечення (ПЗ) — це сукупність програм, систем обробки інформації та програмних документів, необхідних для експлуатації обчислювальної машини. ПЗ перетворює набір електронних схем на багатофункціональний інструмент.

Сучасне ПЗ має чітку **багаторівневу (ієрархічну) структуру**. Кожен вищий рівень спирається на функції нижчого:

1. **Апаратний рівень (Hardware):** Фізичні компоненти (процесор, пам'ять).
2. **Мікропрограмний рівень (Firmware):** Базова система вводу-виводу (BIOS/UEFI), яка "вшита" в мікросхеми плати.
3. **Рівень операційної системи (OS):** Ядро ОС керує ресурсами, розподіляє пам'ять та процесорний час.
4. **Прикладний рівень (Application Software):** Програми, з якими безпосередньо взаємодіє користувач для вирішення своїх задач (браузери, офісні пакети).

5. **Рівень користувача:** Людина, яка ставить завдання системі.

2. Системне ПЗ, прикладне ПЗ, інструментальні засоби

Весь масив комп'ютерних програм прийнято ділити на три великі класи залежно від їх призначення.

Клас ПЗ	Призначення	Типові представники
Системне ПЗ	Керування ресурсами ПК, забезпечення взаємодії між залізом та користувачем. Без нього комп'ютер не може працювати.	Операційні системи (Windows, Linux, macOS), драйвери пристроїв, системні утиліти (дефрагментатори, диспетчери завдань).
Прикладне ПЗ	Вирішення конкретних завдань користувача чи бізнесу.	Текстові редактори (Word), електронні таблиці (Excel), графічні редактори (Photoshop), CAD-системи, браузері, відеоігри.
Інструментальне ПЗ	Програми для створення інших програм. Робоче середовище програмістів.	Інтегровані середовища розробки (IDE: Visual Studio, PyCharm), компілятори, інтерпретатори, зневаджувачі (Debuggers).

Окремо про інструментальні засоби:

Будь-яка програма пишеться мовою програмування (C++, Python, Java). Проте процесор розуміє лише машинні коди (нулі та одиниці). Інструментальне ПЗ (компілятори або інтерпретатори) виконує роль «перекладача» з мови, зрозумілої людині, на мову, зрозумілу процесору.

3. Захист і резервування інформації

Інформація сьогодні є найціннішим активом. Її захист базується на класичній **Тріаді інформаційної безпеки (CIA)**:

- **C (Confidentiality) - Конфіденційність:** Дані доступні лише авторизованим особам.
- **I (Integrity) - Цілісність:** Дані не можуть бути змінені або знищені без дозволу.
- **A (Availability) - Доступність:** Система та дані завжди доступні для легітимних користувачів, коли це потрібно.

Резервне копіювання (Backup):

Найкращий захист від втрати даних (через поломку диска, вірус або помилку користувача) — це бекап. У корпоративному середовищі золотим стандартом є **правило «3-2-1»**:

1. Зберігайте щонайменше **3** копії даних (одна робоча + дві резервні).
2. Використовуйте **2** різні типи носіїв (наприклад, локальний сервер + зовнішній жорсткий диск).

3. Зберігайте **1** копію поза межами офісу (offsite), наприклад, у хмарному сховищі, щоб захиститися від пожеж чи фізичного викрадення техніки.

Типи резервного копіювання:

- *Повне (Full)*: Копіюються абсолютно всі файли. Найбільш надійне, але займає багато часу та місця.
- *Диференційне (Differential)*: Копіюються лише ті файли, які змінилися з моменту останнього повного бекапу.
- *Інкрементне (Incremental)*: Копіюються лише файли, змінені з моменту будь-якого останнього бекапу. Найшвидше, але відновлення потребує цілого ланцюжка копій.

4. Робота з архівними файлами

Архівація — це процес об'єднання кількох файлів або каталогів в один єдиний файл (архів) для зручності передачі.

Стиснення (Компільована архівація) — це процес зменшення фізичного обсягу файлу за рахунок усунення надлишковості даних за допомогою математичних алгоритмів.

Принцип стиснення без втрат (Lossless compression):

Алгоритм аналізує файл і шукає послідовності бітів, що повторюються. Замість того, щоб зберігати одне й те саме слово або набір пікселів сто разів, архіватор записує його один раз і створює «словник», вказуючи координати, де це слово має повторюватися.

Ефективність стиснення виражається коефіцієнтом:

$$K_c = \frac{V_{in}}{V_{out}}$$

де V_{in} — початковий обсяг даних, а V_{out} — обсяг після стиснення. Текстові файли та бази даних стискаються дуже добре (до 10-20% від початкового розміру), тоді як формати JPEG, MP3 або MP4 вже є стисненими, тому архіватори (ZIP, RAR, 7z) майже не зменшують їх розмір.

5. Загальні відомості про антивірусні програми

Шкідливе програмне забезпечення (Malware) — це програми, створені для порушення роботи ПК, крадіжки даних або вимагання грошей.

- *Віруси*: прикріплюються до «чистих» файлів і заражають систему при їх запуску.
- *Хробаки (Worms)*: поширюються мережею самостійно, без участі людини.
- *Трояни*: маскуються під корисні програми, але відкривають зловмисникам таємний доступ до ПК.
- *Ransomware (Програми-вимагачі)*: шифрують усі файли на диску і вимагають викуп у криптовалюті за ключ розшифровки.

Як працюють антивірусні програми:

Сучасні антивіруси (Endpoint Protection Platforms) використовують кілька рівнів захисту:

1. **Сигнатурний аналіз:** Антивірус має величезну базу даних зі "зліпками" (хеш-сумами) відомих вірусів. Він сканує файли і порівнює їх з цією базою. Якщо є збіг — файл блокується.

2. **Евристичний аналіз:** Дозволяє знаходити нові (невідомі бази) віруси. Антивірус аналізує програмний код: якщо програма намагається приховано змінити системні файли або зашифрувати диск, вона визнається підозрілою.

3. **Пісочниця (Sandbox):** Антивірус запускає підозрілий файл у віртуальному, ізольованому середовищі і спостерігає за його поведінкою. Якщо файл поводить себе агресивно, він не допускається до реальної операційної системи.

Питання для обговорення та контролю:

1. Поясніть багаторівневу структуру ПЗ. Чому прикладні програми не можуть працювати безпосередньо з апаратним забезпеченням (процесором)?

2. Наведіть приклади системного та прикладного програмного забезпечення. До якої категорії належать драйвери відеокарти?

3. У чому полягає суть правила «3-2-1» при резервному копіюванні даних? Чому важливо мати хоча б одну копію поза межами будівлі?

4. У чому різниця між інкрементним та диференційним резервним копіюванням? Яке з них дозволяє швидше відновити систему при повному збої?

5. Чому текстовий документ (Word) можна стиснути в архіві у кілька разів, а відеофайл (MP4) або картинку (JPEG) майже неможливо?

6. Як працює евристичний аналіз в антивірусних програмах і чому класичного сигнатурного аналізу сьогодні вже недостатньо?

ТЕМА 2.3. ОСНОВИ ОПЕРАЦІЙНИХ СИСТЕМ. WINDOWS

Мета: Сформувати глибоке розуміння концепції операційної системи як головного програмного комплексу ПК, вивчити еволюцію, архітектурну будову та базові функції ОС, а також детально розібрати принципи організації даних у сучасних файлових системах.

План лекції:

1. Поняття та призначення операційної системи (ОС).
2. Склад та архітектурна структура ОС.
3. Еволюція та класифікація операційних систем.
4. Основні функції ОС: керування ресурсами.
5. Файлова система: логіка зберігання даних (на прикладі NTFS).

1. Поняття та призначення операційної системи (ОС)

Апаратне забезпечення комп'ютера (процесор, пам'ять, диски) — це лише набір електронних компонентів, які «не знають», як взаємодіяти між собою.

Операційна система (ОС) — це базовий комплекс системних програм, що забезпечує керування апаратними ресурсами комп'ютера, організовує роботу з файлами та створює середовище для виконання прикладних програм і взаємодії з користувачем.

ОС виконує роль **посередника (інтерфейсу)** у двох напрямках:

- *Між апаратурою та прикладним ПЗ:* програмісту (наприклад, розробнику відеогри) не потрібно знати, як фізично влаштована ваша відеокарта чи SSD; він звертається до ОС, а ОС уже дає команду «залізу».
- *Між ПК та користувачем:* перетворює складні машинні коди на зрозумілі вікна, курсори та кнопки.

2. Склад та архітектурна структура ОС

Сучасна ОС (така як Windows 11) має складну багаторівневу організацію. Її можна умовно поділити на кілька критичних блоків:

1. **Ядро (Kernel):** Фундамент системи. Це програма, яка завантажується в оперативну пам'ять першою і має найвищий рівень привілеїв (Ring 0). Ядро керує процесором, розподіляє оперативну пам'ять між програмами і обробляє переривання від обладнання. Сучасні версії Windows базуються на гібридному ядрі архітектури **Windows NT**.

2. **Драйвери пристроїв:** Спеціалізовані мікропрограми, які розробляються виробниками обладнання (наприклад, NVIDIA або Intel). Вони інтегруються в ОС і «навчають» ядро правильно спілкуватися з конкретною деталлю.

3. **Командний процесор / Оболонка (Shell):** Частина ОС, яка приймає команди від користувача.

o *GUI (Graphical User Interface):* Графічна оболонка (Робочий стіл, Провідник Windows, вікна).

o *CLI (Command Line Interface):* Командний рядок або PowerShell для текстового керування.

4. **Системні бібліотеки (API):** Набір стандартних функцій і процедур (у Windows це файли .dll), які можуть використовувати прикладні програми для створення вікон, відтворення звуку тощо.

3. Еволюція та класифікація ОС

Операційні системи пройшли довгий шлях від перфокарт до хмарних обчислень.

Коротка еволюція (на прикладі систем Microsoft):

• *Епоха пакетної обробки (1950-60-ті):* Програми виконувалися строго по черзі.

• *Епоха командного рядка (1980-ті): MS-DOS.* Однозадачна ОС. Користувач вводив команди текстом на чорному екрані.

• *Епоха графіки (1990-ті): Windows 95/98.* Поява звичного «Робочого столу», багатовіконності та миші.

- *Епоха надійності та мереж (2000-ні - наш час): Windows XP, 7, 10, 11.* Перехід на ядро Windows NT, що забезпечило стабільність, справжню багатозадачність та багаторівневу безпеку.

Класифікація сучасних операційних систем

Операційні системи проєктуються з урахуванням конкретних інженерних та бізнес-завдань. Залежно від архітектури ядра та цільового використання, їх класифікують за кількома ключовими критеріями:

1. За цільовим призначенням (Архітектурний фокус)

Різниця між цими типами полягає не стільки у зовнішньому вигляді, скільки в пріоритетах планувальника завдань (Scheduler) всередині ядра ОС.

- **Десктопні ОС (Настільні):** Орієнтовані на максимальний комфорт користувача. Планувальник ядра віддає найвищий пріоритет тій програмі, яка зараз відкрита на екрані (активне вікно), щоб інтерфейс не «гальмував». (Приклади: *Windows 11, macOS*).

- **Серверні ОС:** Спроєктовані для безперебійної роботи в режимі 24/7. Їхній пріоритет — фонові служби (сервіси) та швидка обробка тисяч одночасних мережеских запитів. Навіть якщо графічний інтерфейс зависне, база даних продовжить працювати. (Приклади: *Windows Server, Red Hat Enterprise Linux*).

- **Мобільні ОС:** Головний пріоритет — енергоефективність (збереження заряду батареї) та жорстка ізоляція програм одна від одної (Sandboxing) для безпеки. (Приклади: *Android, iOS*).

- **Вбудовані ОС та ОС реального часу (RTOS - Real-Time Operating Systems):** Використовуються в маршрутизаторах, медичному обладнанні, бортових комп'ютерах автомобілів та ракет. Їхня головна риса — **детермінованість**: система гарантує обробку сигналу (наприклад, розкриття подушки безпеки) за строго визначений час (мілісекунди) без жодних затримок. (Приклади: *QNX, FreeRTOS*).

2. За відкритістю вихідного коду (Ліцензійна політика)

Цей критерій визначає правовий та технічний статус системи, що критично впливає на вибір архітектури для бізнесу.

- **Пропріетарні ОС (Закритий код):** Код системи є комерційною таємницею. Користувач отримує готовий продукт («чорну скриньку») і не може змінювати його внутрішню логіку.

- *Переваги:* Звичний інтерфейс, гарантована комерційна підтримка виробника, сумісність із більшістю комерційного ПЗ. Бізнес обирає їх для робочих місць співробітників (офісні ПК). (Приклади: *Windows, macOS*).

- **ОС з відкритим кодом (Open Source):** Вихідний код системи є публічним. Будь-хто може його вивчити, модифікувати та створити власну збірку (дистрибутив).

- *Переваги:* Високий рівень безпеки (тисячі незалежних розробників шукають вразливості), відсутність плати за ліцензії, можливість "вирізати" з ядра все зайве для максимальної швидкодії. Саме тому 90% серверів у світі та суперкомп'ютерів працюють на Open Source. (Приклади: *Ubuntu, Debian, CentOS*).

3. За кількістю задач (Багатозадачність)

- **Однозадачні:** Історичний етап еволюції ОС. Процесор повністю віддавався одній програмі. Щоб запустити іншу, першу потрібно було закрити. (Приклад: *MS-DOS*).

- **Багатозадачні:** Сучасний стандарт. Використовується **багатозадачність із витісненням (Preemptive multitasking)**: ядро ОС примусово забирає процесорний час у однієї програми і віддає іншій сотні разів на секунду. Користувачу здається, що браузер, плеєр та антивірус працюють одночасно. (Приклади: *Всі сучасні ОС*).

4. За кількістю користувачів

- **Однокористувацькі:** Навіть якщо в системі створено кілька облікових записів (наприклад, "Батьки" і "Діти"), одночасно за комп'ютером може працювати лише одна людина. (Приклад: *старі версії Windows для домашнього використання*).

- **Багатокористувацькі (Термінальні):** Дозволяють десяткам або сотням користувачів одночасно підключатися до одного потужного сервера через мережу. ОС створює для кожного ізольовану сесію (віртуальний робочий стіл), і вони працюють паралельно, не заважаючи одне одному. (Приклади: *Linux-сервери, Windows Server з роллю Remote Desktop Services*).

4. Основні функції ОС: керування ресурсами

Щоб комп'ютер не «завис» від конфлікту програм, ОС працює як суворий диспетчер, виконуючи чотири головні функції:

1. **Керування процесором (Планування / Scheduling):** Оскільки програм запущено багато (браузер, музика, антивірус), а процесор один, ОС швидко перемикає процесор між задачами (тисячі разів на секунду). Це створює ілюзію, що все працює одночасно. Цей процес називається **багатозадачністю з витісненням**.

2. **Керування пам'яттю:** ОС виділяє кожній програмі її власний, ізольований шматок оперативної пам'яті. Якщо фізичної пам'яті (RAM) не вистачає, ОС використовує **віртуальну пам'ять** — скидає частину даних на жорсткий диск (файл `pagefile.sys` у Windows).

3. **Керування пристроями вводу-виводу:** ОС організовує черги на друк, обробляє рухи миші, малює курсор на екрані та записує дані на флешку так, щоб ці процеси не заважали один одному.

4. **Керування безпекою та доступом:** Перевірка паролів (автентифікація) та розмежування прав доступу (користувач не може видалити системний файл, якщо він не Адміністратор).

5. Файлова система сучасних ОС (NTFS)

Диск комп'ютера фізично є суцільним масивом секторів для запису нулів та одиниць. **Файлова система (ФС)** — це логічна структура (таблиця та правила), за допомогою якої ОС організовує ці сирі дані у вигляді зручних файлів та папок (каталогів).

Ключові поняття:

- **Файл:** Послідовність байтів, яка має унікальне ім'я та зберігається на носії.
- **Каталог (Папка):** Логічний контейнер, який містить посилання на файли або інші каталоги. Утворює ієрархічну (деревоподібну) структуру (наприклад, C:\Users\Student\Documents\).
- **Кластер:** Найменша одиниця місця на диску, яку ФС може виділити для файлу (зазвичай 4 КБ). Якщо файл важить 1 байт, він все одно займе на диску цілий кластер.

Еволюція файлових систем Windows:

1. **FAT32:** Стара ФС. Її головний мінус сьогодні — вона не підтримує файли, розмір яких перевищує 4 ГБ.
2. **exFAT:** Оптимізована для сучасних флеш-накопичувачів і зовнішніх дисків, знято обмеження в 4 ГБ.
3. **NTFS (New Technology File System):** Основний стандарт для внутрішніх дисків Windows.

Чому Windows використовує саме NTFS?

- **Журналювання:** NTFS записує всі свої наміри у спеціальний прихований журнал до того, як внести фізичні зміни. Якщо раптово вимкнеться світло, система не «зламається», а просто відкотить незавершену дію за журналом.
- **MFT (Master File Table):** Головна таблиця бази даних NTFS, де зберігається інформація про кожен файл (де він лежить, якого він розміру).
- **Безпека (Permissions):** Можливість призначити права (читання, запис, виконання) для конкретного файлу чи папки на рівні окремих користувачів.
- **Квотування та стиснення:** Можливість обмежити місце для конкретного користувача або стиснути файли "на льоту" на рівні файлової системи.

Питання для обговорення та контролю:

1. Поясніть метафору «ОС як перекладач та диспетчер». Що саме вона перекладає і чим керує?
2. Яку роль відіграє ядро (Kernel) операційної системи і чому воно має найвищий рівень привілеїв?
3. Чим відрізняється багатозадачна ОС від однозадачної на рівні взаємодії з процесором?
4. У чому полягає принцип роботи віртуальної пам'яті (файлу підкачки)? Які її плюси та мінуси порівняно з RAM?
5. Чому сьогодні не рекомендується формувати внутрішній жорсткий диск Windows у файлову систему FAT32?
6. Що таке журналювання в NTFS і як воно рятує файлову систему при раптовому знеструмленні ПК?

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Азарова А. О., Юрчук Н. П., Нікіфорова Л. О., Шиян А. А. Інформаційні системи і технології. Частина 2 : навчальний посібник. Вінниця : ВНТУ, 2024. 160 с. URL: https://pdf.lib.vntu.edu.ua/books/2025/Azarova_P2_2024_160.pdf
2. Басюк Т. М., Думанський Н. О., Пасічник О. В. Основи інформаційних технологій : навчальний посібник. Львів : “Новий Світ-2000”, 2024. 390 с.
3. Бутенко Т. А., Сирий В. М. Інформаційні системи та технології : навчальний посібник. Харків : ХНАУ ім. В.В. Докучаєва, 2020. 207 с. URL: https://repo.btu.kharkov.ua/bitstream/123456789/4849/1/INFO_SYSTEMS_20.pdf
4. Волосюк Ю. В., Кузьома В. В., Коваленко О. А. Інформаційні технології: навч. посібник / за ред. А. В. Нелепової. Вінниця: ВНТУ, 2025. 240 с.
5. Дискретна математика та логічні основи ЕОМ: онлайн-курс. Prometheus, 2025. URL: <https://prometheus.org.ua/courses/discrete-math>.
6. Допомога та навчання з Windows: вебсайт. Microsoft Support, 2025. URL: <https://support.microsoft.com/uk-ua/windows>.
7. Інформаційна безпека та антивірусний захист систем: конспект лекцій. Харків: НТУ «ХП», 2025. URL: http://web.kpi.kharkov.ua/sp/wp-content/uploads/sites/95/2020/12/NK_Cyber_1.pdf.
8. Інформаційні системи та технології : підруч. / за ред. В. Б. Вишні. Дніпро : Дніпроп. держ. ун-т внутр. справ, 2021. 280 с. URL: <http://ir.nmu.org.ua/handle/123456789/164375>
9. Інформаційні технології : навчальний посібник / О. І. Зачек, В. В. Сеник, Т. В. Магеровська та ін. Львів : Львівський державний університет внутрішніх справ, 2022. 432 с. URL: <https://dspace.lvduvs.edu.ua/handle/1234567890/6995>
10. Кірчук Р. В., Герасимчук О. О., Завіша В. В. Сучасні інформаційні технології : навчальний посібник. Луцьк : Технічний коледж Луцького НТУ, 2020. 134 с. URL: <https://surl.li/xupnkv>
11. Класифікація та загальна характеристика програмного забезпечення, операційна система: навч. матеріали. Київ : КУ імені Бориса Грінченка, 2025. URL: <https://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/56/56.html>.
12. Кравченко І. В., Микитенко В. І. Інформаційні технології : підручник для студ. спеціальності «Автоматизація та комп'ютерно-інтегровані технології» / КПІ ім. Ігоря Сікорського . Київ : КПІ ім. Ігоря Сікорського, 2022. 447 с. URL: <https://ela.kpi.ua/handle/123456789/51682>
13. Момот А. С., Мамута М. С. Інтелектуальні та інформаційні системи. Практикум : навчальний посібник / КПІ ім. Ігоря Сікорського. Київ : КПІ

ім. Ігоря Сікорського, 2022. 221 с. URL:

<https://ela.kpi.ua/handle/123456789/47696>

14. Програмне забезпечення ПК: вебсайт. Київ: КУ імені Бориса Грінченка, 2025. URL: <https://www.kievoit.ippo.kubg.edu.ua/kievoit/2013/57/57.html>.

Навчальне видання

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Конспект лекцій

Укладачі:

Тищенко Світлана Іванівна
Кучмійова Тетяна Сергіївна
Співак Вадим Вікторович

Формат 60x84 1/16. Ум. друк. арк. 2,3.

Наклад 50 прим. Зам. № _____

Надруковано у видавничому відділі
Миколаївського національного аграрного університету
54020, м. Миколаїв, вул. Георгія Гонгадзе, 9

Свідоцтво суб'єкта видавничої справи ДК № 4490 від 20.02.2013 р.