

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

О.С. Козлова, А.Г. Кривохата, О.В. Кудін

**МЕТОДИ ТА ЗАСОБИ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Навчальний посібник
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Інженерія програмного забезпечення»
освітньо-професійної програми «Програмна інженерія»

Затверджено
вченою радою ЗНУ
Протокол № від

Запоріжжя
2019

УДК:004.451(075.8)
К595

Козлова О.С., Кривохата А.Г., Кудін О.В. Методи та засоби комп'ютерних інформаційних технологій : навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія». Запоріжжя : ЗНУ, 2019. 145 с.

У навчальному посібнику подано теоретичний матеріал до тем курсу «Методи та засоби комп'ютерних інформаційних технологій». Основна увага приділяється засвоєнню знань про найпоширеніші команди операційних систем Windows та Linux, загальні принципи написання командних файлів, види та структуру файлових менеджерів, основні поняття графічного інтерфейсу користувача. Розглянуто основні методи та засоби обробки текстової і табличної інформації.

До кожної теми надано практичні завдання, які дають змогу застосувати отримані знання до розв'язання конкретних задач, тестові завдання для самоперевірки, контрольні запитання, відповіді на які сприятимуть засвоєнню теоретичного матеріалу.

Для здобувачів ступеня вищої освіти бакалавра спеціальності «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія».

Рецензент

О.С. Пшенична, кандидат педагогічних наук, доцент, доцент кафедри комп'ютерних наук

Відповідальний за випуск

А.О. Лісняк, кандидат фізико-математичних наук, доцент, завідувач кафедри програмної інженерії

Зміст

Вступ	5
Тема 1. Основні поняття інформатики	7
1.1 Інформатика як наука. Інформаційні технології. Комп'ютерні технології	7
1.2 Інформація. Основні етапи обробки інформації	8
1.3 Комп'ютери. Історія розвитку ЕОМ	9
1.4 Подання інформації в ЕОМ. Одиниці вимірювання інформації	13
1.5 Форми подання даних. Збереження даних в ЕОМ	15
1.6 Основні сфери застосування комп'ютерних технологій	17
Тема 2. Склад і структура комп'ютера	20
2.1 Основні принципи побудови комп'ютерів. Архітектура фон Неймана. Багатопроекторні ЕОМ	20
2.2 Основні елементи ПК та їх призначення. Схема побудови ПК	23
2.3 Технічні характеристики сучасних комп'ютерів	24
Тема 3. Програмне забезпечення ПК	29
3.1 Класифікація сучасного програмного забезпечення	29
3.2 Характеристика системного програмного забезпечення	30
3.3 Характеристика службового програмного забезпечення	31
3.4 Характеристика прикладного програмного забезпечення	32
3.5 Огляд систем програмування (інструментальних систем). Транслятори та інтерпретатори	37
3.6 Розв'язання задач за допомогою ЕОМ: основні етапи	39
3.7 Поняття алгоритму, його основні властивості та способи опису	39
Тема 4. Текстові процесори	43
4.1 Поняття про текстовий процесор	43
4.2 Класифікація програм для роботи з текстом та їх основні функції	45
4.3 Формати файлів текстових документів	47
4.4 Загальна структура текстового процесора та його основні елементи	49
Тема 5. Програми обробки електронних таблиць	53
5.1 Поняття про електронну таблицю та табличний процесор	53
5.2 Загальна технологія роботи з електронними таблицями	55
5.3 Типова структура вікна електронної таблиці	57
5.4 Основні формати даних табличного процесора	61
5.5 Форматування символічних даних у комірках	62
Тема 6. Графічний інтерфейс операційних систем	70
6.1 Визначення графічного інтерфейсу операційних систем	70
6.2 Види та загальна характеристика інтерфейсів операційних систем	71
6.3 Основні поняття графічного інтерфейсу	73
6.4 Основні елементи вікна графічної операційної системи та операції з ним. Типи вікон	74
6.5 Панель задач та головне меню. Довідкова система	78
6.6 Налаштування робочого середовища графічної операційної системи	80

Тема 7. Загальносервісні системні програми	83
7.1 Визначення основних характеристик ПК за допомогою програмних засобів.....	83
7.2 Поняття про комп'ютерні віруси. Класифікація вірусів. Ознаки зараження вірусом.....	86
7.3 Класифікація і принципи роботи антивірусних програм.....	88
7.4 Архівация файлів і каталогів.....	90
7.5 Робота з програмами-архіваторами.....	94
Тема 8. Організація файлової системи	98
8.1 Класифікація, призначення, основні функції та склад операційних систем	98
8.2 Системний диск. Логічні та фізичні диски.....	100
8.3 Поняття файлу. Типи файлів. Атрибути. Правила запису імен файлів в операційних системах. Поняття каталогу. Повне ім'я файлу. Маршрут у файлової системі. Шаблони. Деревоподібна структура збереження інформації	102
8.4 Файлова система. Логічна організація файлу	105
8.5 Фізична організація файлу	107
8.6 Загальна модель файлової системи	110
Тема 9. Файлові менеджери	113
9.1 Загальний огляд файлових менеджерів: призначення і можливості .	113
9.2 Структура панелі файлового менеджера	114
9.3 Загальні прийоми роботи файлових менеджерів	116
9.4 Запуск програм і виконання команд операційної системи	116
9.5 Зміна конфігурації файлового менеджера	119
Тема 10. Мова команд операційної системи	122
10.1 Командний рядок. Загальна структура команд операційної системи	122
10.2 Внутрішні та зовнішні команди. Основні правила використання команд операційної системи	124
10.3 Довідкова система командного рядка. Загальносистемні команди операційної системи.....	125
10.4 Зарезервовані імена пристроїв операційної системи. Введення кількох команд в одному рядку.....	127
10.5 Порівняння систем команд операційних систем Windows та GNU/Linux.....	128
Тема 11. Призначення та основні принципи написання командних файлів	134
11.1 Поняття про командний файл	134
11.2 Windows: основні команди, створення і запуск командних файлів. Приклади командних файлів.....	135
Предметний покажчик	141
Рекомендована література	144

Вступ

На сучасному етапі розвитку інформаційного суспільства для фахівців будь-якої галузі важливою є здатність аналізувати, вибирати і застосовувати методи і засоби для створення та користування програмним забезпеченням. Різноманітність технологій, які використовуються при створенні програмного забезпечення, висуває вимогу володіння базовими принципами роботи з комп'ютерною технікою та програмним забезпеченням. Цей факт підкреслює необхідність та актуальність вивчення дисципліни «Методи та засоби комп'ютерних інформаційних технологій».

Курс «Методи та засоби комп'ютерних інформаційних технологій» є фаховим, професійно-орієнтованим. Його опанування передбачає здобуття необхідних базових знань та вмінь, є основою для вивчення таких дисциплін: «Операційні системи», «Бази даних», «Основи організації та функціонування ЕОМ» та ін. Він надає майбутньому фахівцю фундамент у просторі комп'ютерних інформаційних технологій, що дозволить надалі використовувати отримані знання у практичній діяльності.

Метою вивчення навчальної дисципліни «Методи та засоби комп'ютерних інформаційних технологій» є формування теоретичних знань про сучасні інформаційні і комп'ютерні технології, набуття практичних навичок щодо встановлення, конфігурування, управління й обслуговування, що використовуються у професійній діяльності майбутніми фахівцями з розробки та тестування програмного забезпечення.

Основними **завданнями** вивчення дисципліни «Методи та засоби комп'ютерних інформаційних технологій» є: оволодіння основними принципами та методами використання інформаційних технологій, необхідними практичними знаннями для подальшого їх застосування у вивченні спеціальних дисциплін та професійній підготовці.

Структура та вміст навчального посібника відповідають змісту робочої програми дисципліни «Методи та засоби комп'ютерних інформаційних технологій».

У результаті вивчення дисципліни студенти повинні:

знати:

- основні поняття інформатики, текстового і табличного процесорів, графічного інтерфейсу користувача;
- склад та структуру комп'ютера;
- основні види програмного забезпечення ПК;
- основні види загальносервісних системних програм та файлів;
- основні поняття графічного інтерфейсу користувача;
- поняття та принципи організації файлової системи;
- види та структуру файлових менеджерів;
- основні команди операційної системи, призначення і принципи написання командних файлів;

– основні поняття та принцип роботи в Internet;

вміти:

– налагоджувати та встановлювати основні види програмного забезпечення;

– використовувати основні команди операційної системи для роботи з файловою системою;

– створювати та налагоджувати командні файли;

– використовувати загальносервісні системні програми і файли;

– використовувати файлові менеджери для роботи з файловою системою;

– вводити, редагувати та оброблювати текстову, графічну і табличну інформацію за допомогою текстового процесору;

– оброблювати табличну інформацію за допомогою програм обробки електронних таблиць;

– використовувати хмарні сховища у роботі;

– використовувати Internet для пошуку необхідної інформації.

Набуті знання та навички будуть також затребувані при вивченні спеціальних дисциплін та стануть інструментом для грамотного виконання та оформлення курсових і кваліфікаційних робіт.

Навчальний посібник складається зі вступу, 11 теоретичних тем, в яких значна увага приділяється відомостям про вільне програмне забезпечення (Libre Office, Linux, FAR). Наприкінці кожної теми наведено практичні завдання, тестові завдання для самоперевірки, контрольні запитання, відповіді на які покращать засвоєння теоретичного та практичного матеріалу.

Тема 1. Основні поняття інформатики

Мета: ознайомитись з основними поняттями інформаційних, комп'ютерних технологій, етапами розвитку ЕОМ, формами подання та збереження даних у комп'ютері.

План

1. Інформатика як наука. Інформаційні технології. Комп'ютерні технології.
2. Інформація. Основні етапи обробки інформації.
3. Комп'ютери. Історія розвитку ЕОМ.
4. Одиниці вимірювання інформації.
5. Форми подання даних. Збереження чисел в ЕОМ.
6. Основні сфери застосування комп'ютерних технологій.

1.1 Інформатика як наука. Інформаційні технології. Комп'ютерні технології

Інформатика – комплексна технічна наука, що систематизує прийоми створення, збереження, відтворення, обробки та передачі даних засобами обчислювальної техніки, а також принципи функціонування цих засобів та способи керування ними.

Термін «інформатика» походить від французького слова *Informatique* та утворюється з двох слів: інформація та автоматика. Запроваджено цей термін у Франції у середині 60-х років ХХ ст., коли розпочалося широке використання обчислювальної техніки. Тоді в англійських країнах увійшов до вжитку термін «*Computer Science*» для позначення науки про перетворення інформації, що ґрунтується на використанні обчислювальної техніки. Тепер ці терміни є синонімами.

Предмет інформатики складають:

1. Апаратне забезпечення засобів обчислювальної техніки.
2. Програмне забезпечення.
3. Засоби взаємодії апаратного та програмного забезпечення.
4. Засоби взаємодії людини між апаратним і програмним забезпеченням.

Засоби взаємодії в інформатиці прийнято називати **інтерфейсом**. Тому засоби взаємодії апаратного та програмного забезпечення інколи називають програмно-апаратним інтерфейсом, а засоби взаємодії людини з апаратними та програмними засобами – інтерфейсом користувача.

Основним завданням інформатики є систематизація прийомів та методів роботи з апаратними та програмними засобами комп'ютерної техніки.

Основними напрямками інформатики для практичного застосування є архітектура обчислювальних систем, інтерфейси обчислювальних систем,

програмування., перетворення даних, захист інформації, автоматизація, стандартизація.

Інформаційна технологія – це людино-машинна технологія збору, обробки та передачі інформації, що ґрунтується на використанні обчислювальної техніки.

Поняття інформаційної технології з'явилося з виникненням інформаційного суспільства, основою соціальної динаміки в якому є не традиційні матеріальні, а інформаційні ресурси: знання, наука, організаційні чинники, інтелектуальні здібності, ініціатива, творчість та ін.

В інформатиці поняття «система» найчастіше використовують стосовно набору технічних засобів і програм. Системою називають також апаратну частину комп'ютера. Доповнення поняття «система» словом «інформаційна» відображає мету її створення та функціонування.

Інформаційна система – взаємопов'язана сукупність засобів, методів і персоналу, що використовуються для зберігання, обробки та видачі інформації з метою вирішення конкретного завдання.

Етапи роботи інформаційної системи:

1. Зародження даних. На цьому етапі відбувається формування первинних повідомлень, що фіксують результати певних операцій, властивості об'єктів і суб'єктів управління, параметри процесів.

2. Накопичення та систематизація даних. Тут відбувається організація такого їх розміщення, яке забезпечувало б швидкий пошук і відбір потрібних відомостей, методичне оновлення даних, захист їх від спотворень, втрати, деформування цілісності та ін.

3. Обробка даних. Тут маються на увазі процеси, внаслідок яких на підставі раніше накопичених даних формуються нові види даних: узагальнюючі, аналітичні, рекомендаційні, прогнозні.

Потреби другого і третього етапів задовольняються у сучасних інформаційних системах в основному засобами **комп'ютерних технологій**. Засоби, що забезпечують доступність інформації для людини, тобто засоби відображення даних, є компонентами **комп'ютерних технологій**.

4. Відображення даних, тобто подання їх у формі, придатній для сприйняття людиною. Передусім – це друкування, тобто виготовлення документів на так званих твердих (паперових) носіях. Широко використовують побудову графічних ілюстративних матеріалів (графіків, діаграм) і формування звукових сигналів.

1.2 Інформація. Основні етапи обробки інформації

Інформація – це сукупність відомостей (даних), які сприймають із навколишнього середовища (вхідна інформація), видають у навколишнє середовище (вихідна інформація), або зберігають всередині певної системи.

Інформація існує у вигляді документів, креслень, рисунків, текстів, звукових і світлових сигналів, електричних і нервових імпульсів тощо.

Найбільш важливими властивостями інформації є: об'єктивність та суб'єктивність; повнота; достовірність; адекватність; доступність; актуальність.

Дані є складовою частиною інформації, що являють собою зареєстровані сигнали. Під час інформаційного процесу дані перетворюються з одного виду в інший за допомогою методів. Обробка даних містить у собі багато різних операцій. Основними операціями є:

- агрегування даних – накопичення інформації з метою забезпечення достатньої повноти для прийняття рішення;
- формалізація даних – приведення даних, що надходять із різних джерел до однакової форми;
- фільтрація даних – усунення зайвих даних, які не потрібні для прийняття рішень;
- сортування даних – впорядкування даних за заданою ознакою з метою зручності використання;
- архівація даних – збереження даних у зручній та доступній формі;
- захист даних – комплекс дій, які скеровано на запобігання втрат, відтворення та модифікації даних;
- транспортування даних – прийом та передача даних між віддаленими користувачами інформаційного процесу. Джерело даних прийнято називати **сервером**, а споживача – **клієнтом**;
- перетворення даних – перетворення даних з однієї форми в іншу, або з однієї структури в іншу, або зміна типу носія.

1.3 Комп'ютери. Історія розвитку ЕОМ

Історію розвитку ЕОМ варто описувати, використовуючи знання про покоління обчислювальних машин. Кожне покоління ЕОМ характеризується своїми конструктивними особливостями і можливостями. Зробимо короткий опис кожного з поколінь, однак слід зазначити, що розподіл ЕОМ на покоління є умовним, оскільки водночас випускалися машини різного рівня.

Перше покоління

Різкий стрибок у розвитку обчислювальної техніки відбувся у 40-х роках, після Другої світової війни, і пов'язаний він був із появою якісно нових електронних пристроїв – електронно-вакуумних ламп.

Електричні схеми, побудовані на цих лампах, працювали значно швидше, ніж схеми на електромеханічних реле. Застосування ЕОМ значно розширило коло розв'язуваних завдань, наприклад: розрахунки інженерних споруд, розрахунки руху планет, балістичні розрахунки тощо.

Перша ЕОМ створювалася в 1943 – 1946 рр. у США і називалася ЕНІАК (ENIAC-Electronic Numerical Integrator and Calculator – електронно-числовий інтегратор і обчислювач). Ця машина містила близько 18 тисяч електронних ламп, багато електромеханічних реле, причому щомісяця виходило з ладу близько 2 тисяч ламп. У машини ЕНІАК, а також в інших перших ЕОМ був серйозний недолік – програма, що виконувалася і зберігалася не у пам'яті

машини, а набиралася складним способом за допомогою зовнішніх перемичок (рис.1.1).

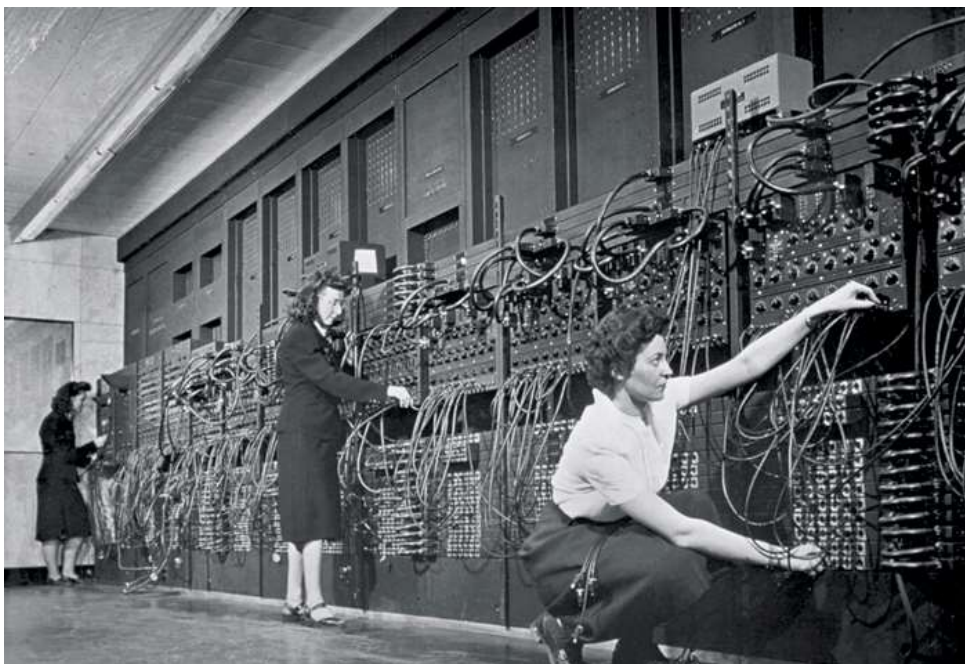


Рисунок 1.1 – Робота машини ЕНІАК

У 1945 р. відомий математик і фізик-теоретик фон Нейман сформулював загальні принципи роботи універсальних обчислювальних пристроїв.

За фон Нейманом, обчислювальна машина повинна керуватися програмою з послідовним виконанням команд, а сама програма – зберігатися в пам'яті машини. Перша подібна ЕОМ була побудована в Англії у 1949 р.

ЕОМ постійно вдосконалювалися, завдяки чому до середини 50-х років їх швидкодію вдалося підвищити від кількох сотень до кількох десятків тисяч операцій за секунду. Згодом на зміну лампам прийшли напівпровідникові прилади. Так завершився перший етап розвитку ЕОМ. Обчислювальні машини цього етапу прийнято називати ЕОМ першого покоління.

Характерними рисами ЕОМ першого покоління є:

1. Застосування електронних ламп у цифрових схемах.
2. Великі розміри.
3. Трудомісткий процес програмування.

Насправді, ЕОМ першого покоління розміщувалися у великих машинних залах, споживали багато електроенергії та вимагали охолодження за допомогою потужних вентиляторів. Програми для цих ЕОМ потрібно було складати у машинних кодах, і це могли робити тільки фахівці, що знали детально пристрій ЕОМ.

Друге покоління

Розробники ЕОМ завжди прямували за прогресом в електронній техніці. Коли у середині 50-х років на зміну електронним лампам прийшли напівпровідникові прилади, почалося переведення ЕОМ на напівпровідники.

Такі прилади (транзистори, діоди) були, по-перше, значно компактнішими, ніж їхні лампові попередники. По-друге, вони мали триваліший термін служби. По-третє, споживання енергії в ЕОМ на напівпровідниках було істотно нижчим.

З упровадженням цифрових елементів на напівпровідникових приладах почалося створення ЕОМ другого покоління.

В Україні першою малою ЕОМ стала машина «Днепр-1», серійне виробництво якої було налагоджено на заводі «Арсенал» (м. Київ). ЕОМ «Днепр-1» передувала унікальній за своєю архітектурою машині «Мир-1», розробленій в 1965 р. в Інституті кібернетики (керівник В.М. Глушков). Машина «Мир-1» та її наступна модифікація «Мир-2» передбачались для інженерних розрахунків, які виконував на ЕОМ сам користувач без допомоги оператора (рис. 1.2).



Рисунок 1.2 – Комп'ютер «Мир-2»

ЕОМ другого покоління характеризуються застосуванням напівпровідникових елементів і використанням алгоритмічних мов програмування.

За кордоном найпоширенішими машинами другого покоління були «Елліот» (Англія), «Сіменс» (ФРН), «Стретч» (США).

Третє покоління

Чергова зміна поколінь ЕОМ відбулася наприкінці 60-х років при переході від напівпровідникових приладів у пристроях ЕОМ до інтегральних схем. Застосування інтегральних схем надало можливість збільшити кількість електронних елементів в ЕОМ без зміни їхніх реальних розмірів. Швидкодія ЕОМ зростає до 10 мільйонів операцій за секунду. Крім того, складати програми для ЕОМ стало під силу простим користувачам, а не тільки фахівцям у галузі електроніки.

У третьому поколінні (рис. 1.3) з'явилися великі серії ЕОМ, таких як: ІВМ 360/370, ЄС ЕОМ (Єдина Система ЕОМ), СМ ЕОМ (Система Малих ЕОМ) і «Електроніка» (система мікро-ЕОМ).



Рисунок 1.3 – Комп'ютер третього покоління

Характерними рисами ЕОМ третього покоління є застосування інтегральних схем і можливість використання розвинутих мов програмування (мов високого рівня).

Четверте покоління

У процесі вдосконалення мікросхем збільшувалася їхня надійність і щільність розміщених у них елементів. З'явилися великі інтегральні схеми (ВІС), в яких на один квадратний сантиметр припадає декілька десятків тисяч елементів. На основі ВІС були розроблені ЕОМ наступного четвертого покоління (рис. 1.4).

ЕОМ четвертого покоління характеризуються застосуванням мікропроцесорів, побудованих на великих інтегральних схемах.

Вплив персональних комп'ютерів на уявлення людей про обчислювальну техніку виявився настільки великим, що поступово з ужитку зник термін «ЕОМ», а його місце зайняло слово «комп'ютер».



Рисунок 1.4 – Комп'ютер четвертого покоління

П'яте покоління

Нині створюються та розвиваються ЕОМ п'ятого покоління – ЕОМ на надвеликих інтегрованих схемах. Ці ЕОМ використовують нові рішення в архітектурі комп'ютерної системи та принципи штучного інтелекту, користувач може голосом подавати команди машині.

Таблиця 1.1 – Коротка характеристика поколінь ЕОМ

Покоління ЕОМ	Роки	Електронні елементи	Швидкодія
I	1950–1960	Електровакуумні лампи (ENIAC, МЕСМ)	10–20 тис. оп./с
II	1960–1965	Транзистори	100–500 тис. оп./с
III	1965–1970	Інтегральні схеми	1 млн оп./с
IV	з 1970	Мікропроцесори	сотні млн оп./с

Приклад 1.1 Скільки операцій в секунду виконує комп'ютер, який має частоту 2 ГГц?

Щоб знайти пікову продуктивність ЕОМ R , **терафлопс**, потрібно тактову частоту F , МГц, помножити на число процесорів (процесорних ядер) n , домножити на кількість інструкцій з плаваючою точкою на такт (4 для процесорів Core2 – 2 операції Float Multiple Add; 8 – для процесорів Intel з AVX) і поділити на 1000000:

$$F \times n \times 4 \times 10^{-6} = R$$

У загальному випадку, співвідношення максимальної і пікової продуктивності варіюється від 60% до 83 %, тому за вказаною формулою можна обчислювати лише пікову продуктивність супер ЕОМ. Так, наприклад, пікова продуктивність комп'ютера на базі чотирьох-ядерного процесора AMD Phenom 9500 sAM2+ з тактовою частотою 2,2 ГГц дорівнює:

$$2200 \text{ МГц} \times 4 \text{ ядра} \times 4 \cdot 10^{-6} = 35200 \text{ млн операцій за секунду} = \\ = 0,0352 \text{ терафлопс}$$

Для чотирьох-ядерного процесора Core 2 Quad Q6600:

$$2400 \text{ МГц} \times 4 \text{ ядра} \times 4 \cdot 10^{-6} = 38,4 \text{ млрд операцій в секунду} = \\ = 0,0384 \text{ терафлопс}$$

1.4 Подання інформації в ЕОМ. Одиниці вимірювання інформації

Основною одиницею інформації в обчислювальній техніці є *біт*. Що таке біт, найпростіше зрозуміти на прикладах ситуацій, коли вам потрібно зробити

вибір із двох варіантів. Скажімо, ви кидаєте монету, що може впасти одним з боків («орел» або «решка»). До підкидання монети інформація про те, яким боком упаде монета, відсутня. Після того як монету підкинуто і ви дізналися про результат, вважаєте, що ви отримали інформацію, яка дорівнює 1 біт. Така сама кількість інформації буде міститися у відповідях на запитання: «У якій руці цукерка?», «Хто першим вийде з дверей під'їзду: чоловік або жінка?».

Біт – це найменша порція інформації, яку одержують при виборі між двома подіями, наприклад, можна закодувати один символ текстової інформації.

Назву «біт» для одиниці інформації обрано не випадково. Подія, що має два виходи, може бути записана за допомогою двох цифр: 0 та 1. Число, що набуває двох значень, називається двійковим числом, або англійською мовою Binary Digit (скорочено bit – біт). У дослідях із монетою випаданню «орла» можна надати значення 0, а «решки» – 1.

Двійковими числами, тобто числами, що складаються із нулів та одиниць, в обчислювальній техніці подається будь-яка інформація

Біт – одиниця досить дрібна, її недостатньо для вимірювання обсягів інформації, якими оперують сучасні комп'ютери та інші обчислювальні пристрої. Тому використовують більші одиниці, основною із них є байт:

$$1 \text{ байт} = 2^3 = 8 \text{ біт}$$

Байт є кратним біту і є послідовністю з восьми двійкових знаків 0 та 1, наприклад, 10110100 або 00101110. Ще більшими одиницями інформації є одиниці, кратні байту (рис. 1.5):

- кілобайт (Кбайт): 1 Кбайт = 2^{10} байт = 1024 байт;
- мегабайт (Мбайт): 1 Мбайт = 2^{10} Кбайт = 1024 Кбайт;
- гігабайт (Гбайт): 1 Гбайт = 2^{10} Мбайт = 1024 Мбайт;
- терабайт (Тбайт): 1 Тбайт = 2^{10} Гбайт = 1024 Гбайт.

$$\text{Бит} \xrightarrow{\cdot 8} \text{байт} \xrightarrow{\cdot 1024} \text{Кб} \xrightarrow{\cdot 1024} \text{Мб} \xrightarrow{\cdot 1024} \text{Гб} \xrightarrow{\cdot 1024} \text{Тб}$$

$$\text{Бит} \xleftarrow{\cdot 8} \text{байт} \xleftarrow{\cdot 1024} \text{Кб} \xleftarrow{\cdot 1024} \text{Мб} \xleftarrow{\cdot 1024} \text{Гб} \xleftarrow{\cdot 1024} \text{Тб}$$

Рисунок 1.5 – Одиниці вимірювання інформації

Увага! Префікс «кіло» означає не 1000, а число 1024. Саме стільки дає піднесення двійки в десятій степінь.

Приклад 1.2 Переведіть 2 Мб у Кб, байти, біти.

1 спосіб:

$$2 \text{ Мб} \cdot 1024 \text{ Кб} = 2048 \text{ Кб}$$

$$2 \text{ Мб} \cdot 1024 \text{ Кб} \cdot 1024 \text{ байт} = 2097152 \text{ байт}$$

$$2 \text{ Мб} \cdot 1024 \text{ Кб} \cdot 1024 \text{ байт} \cdot 8 \text{ біт} = 16777216 \text{ біт}$$

2 спосіб:

$$2 \text{ Мб} = 2 \cdot 1024 \text{ Кб} = 2048 \text{ Кб}$$

$$2 \text{ Мб} = 2048 \cdot 1024 \text{ байт} = 2097152 \text{ байт}$$

$$2\text{Мб} = 2097152 \cdot 8 \text{ біт} = 16777216 \text{ біт}.$$

Якщо кількість бітів збільшити до двох, то тоді можна вже закодувати чотири поняття : 00, 01, 10, 11. Трьома бітами кодують вісім понять: 000, 001, 010, 011, 100, 101, 110, 111. Збільшуючи на одиницю кількість розрядів у системі двійкового кодування, ми збільшуємо у два рази кількість значень, які можуть бути виражені у цій системі кодування, тобто кількість значень обчислюється за формулою:

$$N = 2^k,$$

де N – кількість незалежних значень, що кодуються, k – це розрядність двійкового кодування.

1.5 Форми подання даних. Збереження даних в ЕОМ

При автоматизації роботи з даними, що відносяться до різних типів важливо уніфікувати форму їх представлення. Для цього, як правило, використовується кодування, тобто представлення даних одного типу через данні іншого типу. Найпоширенішою для подання чисел у пам'яті комп'ютера є двійкова система числення. Для зображення чисел у цій системі необхідні дві цифри: 0 та 1, тобто достатньо двох стійких станів фізичних елементів. Ця система є близькою до оптимальної за економічністю. Слід зауважити, що вся інформація, що зберігається та обробляється засобами обчислювальної техніки, незалежно від її типу (числа, текст, графіка, звук, відео), представлена у двійковому коді.

Для скорочення запису адрес та вмісту оперативної пам'яті комп'ютера використовують шістнадцяткову і вісімкову системи числення.

Таблиця 1.2 – Перші 16 чисел, що записано у різних системах числення

Системи числення			
Десяткова	Двійкова	Вісімкова	Шістнадцяткова
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C

Системи числення			
Десяткова	Двійкова	Вісімкова	Шістнадцяткова
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Для кодування символів у комп'ютері використовується таблиця ASCII (American Standard Code for Information Interchange – стандартний американський код обміну інформацією) – система кодів, у якій числа від 0 до 127 включно поставлені у відповідність літерам, цифрам і символам пунктуації. У більшості комп'ютерів код ASCII використовується для представлення тексту, що дозволяє передавати дані від одного комп'ютера до іншого. Стандартний набір символів ASCII використовує тільки 7 біт для кожного символу. Додавання 8-го розряду дозволяє збільшити кількість кодів таблиці ASCII до 255. Коди від 128 до 255 є розширенням таблиці ASCII. Ці коди використовуються для кодування символів національних алфавітів, а також символів псевдографіки, які можна використовувати, наприклад, для оформлення у тексті різних рамок і текстових таблиць.

Наприклад, 45 відповідає знаку переносу, а 65 – літері «А» великій. Перші 32 коди використовуються для керівних функцій, на зразок введення і видалення попереднього символу. Система широко використовується для зберігання тексту і передачі інформації між комп'ютерами.

Таблиця 1.3 – Таблиця ASCII кодів

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	\$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(72	48	110	H	104	68	150	h
9	9	11		41	29	51)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	
29	1D	35		61	3D	75	=	93	5D	135]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

В інформатиці для виміру даних використовується той факт, що різні типи даних мають універсальне двійкове представлення.

1.6 Основні сфери застосування комп'ютерних технологій

Інформатика – практична наука. Її досягнення повинні проходити перевірку на практиці і застосовуватися у тих випадках, коли вони відповідають критерію підвищення ефективності. У складі основної задачі сьогодні можна виділити такі основні **сфери застосування комп'ютерних технологій**:

- архітектура обчислювальних систем (прийоми та методи побудови систем, призначених для автоматичної обробки даних);
- інтерфейси обчислювальних систем (прийоми та методи керування апаратним та програмним забезпеченням);
- програмування (прийоми, методи та засоби розробки комплексних задач);
- перетворення даних (прийоми та методи перетворення структур даних);
- захист інформації (узагальнення прийомів, розробка методів і засобів захисту даних);
- автоматизація (функціонування програмно-апаратних засобів без участі людини);
- стандартизація (забезпечення сумісності між апаратними та програмними засобами, між форматами представлення даних, що відносяться до різних типів обчислювальних систем).

Найбільш затребуваними у майбутньому будуть такі професії як:

- кліматологи та екологи;
- фермери-агропоники;
- біофармакологи;
- ІТ-медики та ІТ-генетики;
- молекулярні дієтологи;
- оператори безпілотників;
- проектувальники роботів;
- тайм-менеджери;
- міські кібераналітики;
- архітектори живих систем;
- архітектори коду (ІТ-шники).

Тестові завдання

1. Засоби взаємодії між апаратним і програмним забезпеченням називають:

- а) інтерфейсом;
- б) сервером;
- в) клієнтом;

г) немає засобів взаємодії між апаратним і програмним забезпеченням.

2. Інформаційна система – це:

а) взаємопов'язана сукупність засобів методів і персоналу, що використовується для зберігання, обробки та видачі інформації з метою вирішення конкретного завдання;

б) сукупність прийомів і методів роботи з апаратними та програмними засобами обчислювальної техніки;

в) сукупність засобів взаємодії людини між апаратним і програмним забезпеченням.

3. Не є властивістю інформації:

а) результативність;

б) адекватність;

в) актуальність;

г) доступність.

4. Перше покоління ЕОМ характеризується:

а) застосуванням електронних ламп у цифрових схемах;

б) можливістю програмувати без допомоги оператора;

в) застосуванням алгоритмічних мов програмування;

г) компактними розмірами.

5. Загальні принципи роботи комп'ютера сформулював:

а) Джон фон Нейман;

б) Готфрід Вільгельм Ляйбніц;

в) Ісаак Ньютон;

г) Білл Гейтс.

6. Одиницею виміру пікової продуктивності ЕОМ є:

а) кілобайт;

б) Паскаль;

в) Герц;

г) терафлопс.

7. Не є одиницею виміру інформації:

а) кілобайт;

б) друкований аркуш;

в) біт;

г) терабайт.

8. У рівності $1 \text{ Кбайт} = * \text{ байт}$ запишіть замість * вірне число:

а) 1000;

б) 100;

в) 1024;

г) 0,001.

9. ASCII – це:

а) назва першого комп'ютера;

б) таблиця кодування друкованих символів;

в) одиниця вимірювання інформації;

г) назва системи числення.

10. Інформація зберігається у комп'ютері у вигляді:

а) друкованих символів;

б) послідовності 0 та 1;

- в) деякої послідовності деяких кодів;
- г) в якому вигляді вводиться, в такому і зберігається.

Практичні завдання

1. Перевести: 34 байти у біти, 23 Кб у байти, 50 Мб у Кб, 12 Гб у Мб, 6 Тб у Гб.
2. Перевести у двійкову систему числення такі числа: 10; 528; 1024; 42,18. Для перевірки триманих результатів зробити зворотній перевід.
3. Використовуючи таблицю ASCII, закодуйте у двійковій формі свою фамілію.
4. Використовуючи таблицю ASCII, закодуйте у десятковій формі такі слова: byte, computer, information.
5. Закодуйте коротку фразу. Обміняйтесь отриманими кодами із сусідом і декодуйте тексти один одного.

Контрольні запитання

1. Що таке інформація?
2. Якими рисами характеризуються комп'ютери кожного з п'яти поколінь?
3. Які є основні операції з даними?
4. Що називають сервером?
5. Як називається основна одиниця інформації в обчислювальній техніці?
6. Скільки біт в одному байті?
7. Які системи числення використовують у комп'ютері?
8. Що таке таблиця кодування?
9. Як розшифровується ASCII?
10. В яких сферах можуть застосовуватися комп'ютерні технології?

Тема 2. Склад і структура комп'ютера

Мета: ознайомитись з основними елементами, принципами побудови, та технічними характеристиками сучасних комп'ютерів.

План

1. Основні принципи побудови комп'ютерів. Архітектура фон Неймана. Багатопроцесорні ЕОМ.
2. Основні елементи ПК та їх призначення. Блок-схема побудови ПК.
3. Технічні характеристики сучасних комп'ютерів.

2.1 Основні принципи побудови комп'ютерів. Архітектура фон Неймана. Багатопроцесорні ЕОМ

У 1952 р. Джон фон Нейман першим сформулював основні принципи побудови ЕОМ, які були реалізовані ним у машині ІАК. Викладені ним принципи і структура ЕОМ згодом одержали назву найманівських, хоча в їхній розробці брали участь Еккерт і Мочлі.

За фон Нейманом **основними принципами побудови комп'ютера** є:

1. **Принцип програмного керування.** Цей принцип забезпечує автоматизацію керування програмою. Згідно з цим правилом, що було запропоноване англійським математиком Ч. Беббіджем у 1833 році, для розв'язання кожної задачі складається програма, що визначає послідовність дій ЕОМ. Ефективність програмного керування є високою тоді, коли задача розв'язується за тією самою програмою багато разів (хоч і за різних початкових даних). Якщо задача виконується один раз, то ефективність принципу буде низькою.

2. **Принцип програми, що зберігається в пам'яті ЕОМ.** Згідно з цією нормою, яку було сформульовано Нейманом, команди програми подаються, як і дані, у вигляді чисел й обробляються так само, як і числа. Це прискорює процес їх виконання. Така структура унеможлиблювала розрізнення команд від даних в пам'яті машини за структурою представлення, але фон-Нейманом було вказано, що таке розміщення можливе «за умови, якщо машина якимось чином зможе розрізнити їх». Для того, щоби вказати машині на те, де є команди, а де є дані, була згодом запропонована концепція лічильника команд (program counter), в якому зберігалась адреса поточної команди, а після її виконання замінювалась адресою наступної. Адреси ж самих операндів (даних) зберігались безпосередньо в команді.

3. **Універсальність алгоритмів при розв'язанні задач на ЕОМ.**

Алгоритм – це система формальних правил, які чітко й однозначно визначають процес виконання заданої роботи.

Набір операцій, виконуваних універсальною ЕОМ, є достатнім для запису будь-якого алгоритму, що реалізує чисельні методи розв'язання математичної задачі. У теорії доводиться, що універсальність відносно обчислювальних

алгоритмів є універсальністю відносно цифрової інформації взагалі. Отже, ЕОМ, що реалізує чисельні методи обчислень, є універсальним перетворювачем інформації.

Отже, **архітектурою фон Неймана** (архітектурою фон Неймана-Екерта-Маклі) називають архітектуру електронних обчислювальних машин, основною відмінністю якої від інших подібних архітектур є спільне зберігання даних і машинних команд у комірках однієї й тієї ж пам'яті, що унеможливило їх розрізнення за способом представлення або кодування. Ця структура (рис. 2.1) і по сьогодні залишається домінуючою схемою організації ЕОМ загального призначення.

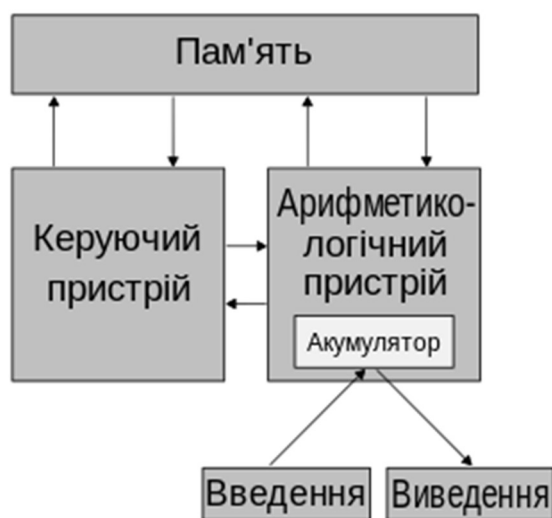


Рисунок 2.1 – Схематичне зображення схеми фон Неймана

Підсумовуючи все вищезгадане можна зробити висновок, що **обчислювальна машина є машиною з архітектурою фон Неймана**, якщо:

1. Програма та дані зберігаються в одній загальній пам'яті.
2. Кожна комірка пам'яті машини ідентифікується унікальним номером, який називається адресою.
3. Різні слова інформації (команди та дані) розрізняються за способом використання, але не за способом кодування та структурою представлення у пам'яті.
4. Кожна програма виконується послідовно, починаючи з першої команди, якщо немає спеціальних вказівок. Для зміни цієї послідовності використовуються команди передачі керування.

На підставі цих принципів можна стверджувати, що сучасний комп'ютер – це технічний пристрій, який після введення у пам'ять початкових даних у вигляді цифрових кодів і програми їх оброблення, яку також представлено цифровими кодами, здатний автоматично здійснити обчислювальний алгоритм, заданий програмою обчислень, і видати готові результати розв'язання задач.

Реальна структура комп'ютера значно складніша, ніж розглянута вище. У сучасних комп'ютерах все частіше здійснюється відхід від традиційної

архітектури фон Неймана. Однак багато в чому структура ЕОМ і принципи її побудови та функціонування зберігаються.

Основні принципи побудови сучасних комп'ютерів є такими:

- 1) наявність пристроїв пам'яті, введення та виведення, керування;
- 2) кодування та збереження програми в пам'яті;
- 3) двійкова система числення для кодування чисел та команд;
- 4) автоматичне виконання обчислень на основі збереженої програми;
- 5) наявність як арифметичних так і логічних операцій;
- 6) ієрархічний принцип побудови пам'яті;
- 7) використання чисельних методів для реалізації обчислень.

Далі розглянемо окремі архітектури, відмінні від найманівської.

Гарвардська архітектура – архітектура обчислювальних машин, головною відмінністю якої від інших подібних є те, що дані та оператори зберігаються окремо.



Рисунок 2.2 – Архітектура з роздільними шинами даних і команд

Така структура має одну важливу перевагу над фон-нейманівською архітектурою: дані можна завантажувати для обробки з запам'ятовувального пристрою одночасно з командами. У фон-нейманівській архітектурі використовується одна шина, тому необхідно спочатку завантажити у процесор команду, а вже потім – завантажити дані. Наявність у гарвардській архітектурі двох незалежних підсистем пам'яті з окремими шинами, дозволяє вести процес завантаження команд і даних практично паралельно.

Гарвардська архітектура широко застосовується у спеціалізованих обчислювачах, зокрема у мікроконтролерах та цифрових сигнальних процесорах, де є необхідним високоінтенсивний обмін даними.

Архітектура потоків даних (dataflow architecture) – архітектура обчислювальних машин, в якій процесом обчислень керує потік даних, які переміщуються між обчислювальними пристроями у машині, де обробляються, поступово «приводячи себе» до «обробленого» вигляду.

Фактично, у цій архітектурі процесом обчислень керує безпосередньо потік даних, а не спеціалізовані команди. У машинах потоків даних не має

потреби у будь-якому централізованому керуванні. Програма для машини потоків даних зазвичай представляється у вигляді графа потоків даних, на якому зображені шляхи руху даних через операційні пристрої.

Ці архітектури доволі специфічні у програмуванні, що навряд дозволяє їм конкурувати з фон-нейманівськими. Тим не менш, деякі принципи машин потоків даних застосовуються у сучасних швидкодіючих процесорах для досягнення високого рівня паралелізму і зменшення простою внаслідок виникнення різних залежностей між операндами.

Паралельна архітектура – різновид архітектури, в якій явно (для програміста) відбувається розподіл процесу обчислень між декількома обчислювальними пристроями (процесорами), які виконують свої завдання одночасно.

Так з'явилися **багатопроесорні ЕОМ**, в яких кілька процесорів працюють одночасно, а це означає, що продуктивність комп'ютера дорівнює сумі продуктивностей процесорів. Такий комп'ютер має багатопроесорну архітектуру. У потужних комп'ютерах для складних інженерних розрахунків і САПР, а також для виконання робіт, що пов'язані з комп'ютерною графікою, часто встановлюють два або чотири процесори. В надпотужних ЕОМ (такі машини можуть, наприклад, моделювати ядерні реакції зі швидкістю природного процесу, передбачати погоду в глобальному масштабі) кількість процесорів досягає кількох десятків.

2.2 Основні елементи ПК та їх призначення. Схема побудови ПК

Комп'ютер – це пристрій, що виконує операції введення інформації, оброблення її за певною програмою, виведення одержаних результатів у формі, придатній для сприйняття людиною. Блок-схему побудови комп'ютера представлено на рисунку 2.3

Структура комп'ютера – це модель, що визначає склад, порядок та принципи взаємодії компонентів комп'ютера.

Основними функціями комп'ютера є обробка та зберігання даних та обмін інформацією із зовнішніми об'єктами.

Додаткові функції – всі ті, що підвищують ефективність виконання основних функцій.

Ці функції комп'ютера реалізуються за допомогою апаратних та програмних засобів. Кожну функцію виконують окремі блоки комп'ютера, такі як: пристрій введення, центральний процесор та пристрій виведення інформації. Всі ці блоки складаються з окремих пристроїв. Номенклатура блоків може варіюватися, але мінімальний комплект складають: системний блок, клавіатура, монітор, маніпулятор (миша). У числі додаткових пристроїв можуть бути: принтер, додатковий накопичувач та ін.



Рисунок 2.3 – Блок-схема побудови комп'ютера

Основні блоки комп'ютера: мікропроцесор, материнська плата або системна плата, накопичувач, постійний запам'ятовувальний пристрій, оперативний запам'ятовувальний пристрій, блок живлення, адаптери, блок живлення.

Материнська (системна) плата – це велика друкована плата, яка містить головні компоненти комп'ютерної системи: центральний процесор, оперативну пам'ять, центральну шину, чіпсет, кілька слотів. На материнській платі є перемички та перемикачі. Також на системній платі встановлюються мікросхема BIOS.

2.3 Технічні характеристики сучасних комп'ютерів

Мікропроцесор – це центральний пристрій комп'ютера, який виконує арифметичні і логічні операції і керує роботою всіх блоків; до нього входять такі компоненти: керуючий пристрій (КП), АЛП (арифметико-логічний пристрій) та мікропроцесорна пам'ять (МПП) (рис. 2.4).

Основні характеристики процесора:

Розрядність – кількість двійкових розрядів, що одночасно обробляються при виконанні однієї команди. Більшість сучасних процесорів – це 32-розрядні процесори, але випускаються і 64-розрядні процесори.

Тактова частота – кількість циклів роботи пристрою за одиницю часу. Чим вище тактова частота, тим вище продуктивність.

Важливими характеристиками мікропроцесорів також є ємність і швидкість функціонування кеш-пам'яті, наявність вбудованого математичного співпроцесора.

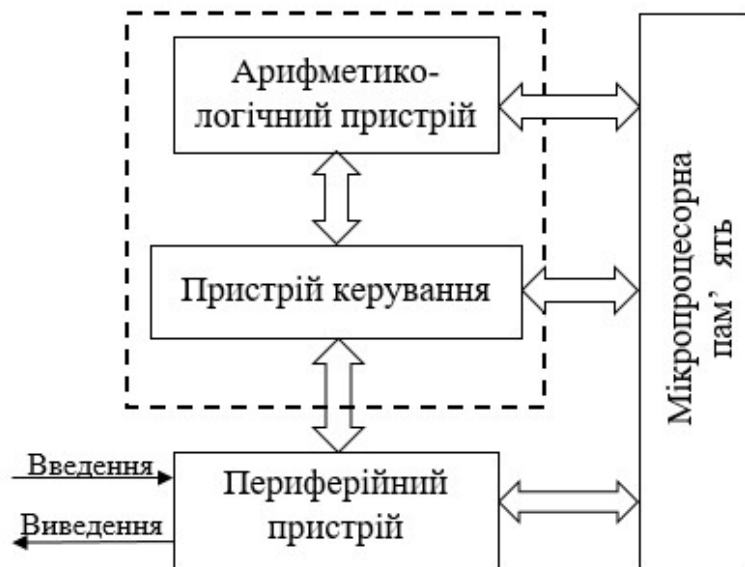


Рисунок 2.4 – Функціональна схема мікропроцесорного пристрою

Кеш-пам'ять буває двох рівнів: першого рівня – це невелика надшвидка пам'ять, що призначено для зберігання проміжних результатів; кеш-пам'ять другого рівня трохи повільніша, але більша, вона може бути інтегрована на самому кристалі або у вигляді додаткового кристалу.

Накопичувач – пристрій для довготривалого зберігання даних (операційної системи, призначених для користувача програм, створених документів і ін.), які є необхідними при роботі з комп'ютером. Накопичувач можна розглядати як сукупність носія та відповідного приводу. Розрізняють накопичувачі зі **змінними і незмінними носіями**. Накопичувач на жорстких магнітних дисках (скорочено НЖМД) – це пристрій з незмінним носієм. На швидкість впливають частота обертання, ємність кеш-пам'яті, час пошуку (час доступу).

Адаптери – пристрої які підтримують взаємодію інших складових і перетворюють інформацію з однієї форми в іншу.

Монтажна плата – це друкована плата, яка має стандартне з'єднання для сполучення з шиною та специфічним з'єднанням для зв'язку з відповідним пристроєм (плотер, дигітайзер, модем, монітор, мишка та ін.).

Постійний запам'ятовувальний пристрій (ПЗП) – пристрій, що використовується для зберігання програм та іншої незмінюваної інформації. Важлива перевага ПЗП у порівнянні з ОЗП – зберігання інформації при виключеному живленні.

Оперативна пам'ять (ОЗП – оперативний запам'ятовувальний пристрій) енергозалежна, яку призначено для зберігання програм, початкових і проміжних даних, з якими працює процесор, та вихідних результатів. Вона дозволяє не тільки зчитувати дані, які у ній зберігаються, але і записувати їх. Від зовнішніх запам'ятовувальних пристроїв оперативна пам'ять відрізняється високою швидкістю обробки даних, завдяки малому часу доступу до них (2-10 наносекунд і менше). Недоліком оперативної пам'яті є те, що при відключенні живлення вона цілком очищується, і всі дані, які в ній зберігаються, втрачаються назавжди.

Із внутрішніх типів пам'яті лише на ОЗП користувач може реально впливати (замінювати модулі ОЗП, додавати нові модулі з метою збільшити загальний обсяг пам'яті), тому розглянемо її більш детально.

Обсяг оперативної пам'яті на теперішній час вимірюється від сотень Мб до декількох Гб і визначається кількістю та ємністю модулів ОЗП, які підключено до системної плати.

Крім обсягу пам'яті для запам'ятовуючих пристроїв також важлива така характеристика як **швидкість читання інформації** з цього пристрою та запису на нього. Інакше цей параметр зветься швидкістю передачі даних або пропускну здатністю запам'ятовуючого пристрою.

Пропускна здатність запам'ятовувального пристрою – це максимальна кількість даних, яку пристрій може передати (чи отримати) за одиницю часу (за секунду).

Стосовно запам'ятовувальних пристроїв пропускна здатність складається з таких параметрів як ефективна частота і розрядність каналу (шини), по якому передаються дані. У свою чергу, ефективна частота складається із реальної частоти (тобто фізичної частоти, з якою працює пристрій), що помножено на кількість разів передачі (отримування) даних за один реальний (фізичний) такт. Розрядність каналів (шин) передачі даних для ОЗП звичайно становить 64 біт (8 байт).

Приклад 2.1 Нехай реальна частота, з якою працює ОЗП становить 133 МГц.

Якщо за 1 такт пам'ять може двічі передати дані, то ефективна частота буде дорівнювати:

$$133 \text{ МГц} \times 2 = 266 \text{ МГц}$$

При можливості 4 рази передати дані ефективна частота буде:

$$133 \text{ МГц} \times 4 = 532 \text{ МГц}$$

Приклад 2.2 Визначити пропускну здатність запам'ятовувального пристрою, якщо ефективна частота дорівнює 800 МГц, розрядність шини даних становить 64 біт (8 Байт).

$$\text{Пропускна здатність} = 800 \text{ МГц} \times 8 \text{ Байт} = 6400 \text{ МБ/с або } 6,4 \text{ ГБ/с} .$$

Приклад 2.3 Визначити пропускну здатність запам'ятовувального пристрою, якщо реальна частота дорівнює 500 МГц, кількість передач даних за такт – 2, розрядність шини даних становить 64 біт (8 Байт).

$$\text{Пропускна здатність} = 500 \text{ МГц} \times 2 \times 8 \text{ Байт} = 8000 \text{ МБ / с або } 8 \text{ ГБ / с}$$

Тестові завдання

1. Для архітектури фон Неймана характерне:
 - а) програма та дані зберігаються в одній загальній пам'яті;
 - б) програма та дані зберігаються окремо;
 - в) дані не зберігаються;
 - г) команди не зберігаються.
2. За фон-Нейманом одним із принципів побудови комп'ютера є:
 - а) принцип програми, що зберігається в пам'яті ЕОМ;
 - б) принцип паралельних обчислень;
 - в) принцип наявності арифметичних і логічних операцій;
 - г) принцип окремого збереження даних і операторів.
3. Відмінністю Гарвардської архітектури від Нейманівської є:
 - а) дані та оператори зберігаються окремо
 - б) процесом обчислень керує потік даних
 - в) кодування програми для збереження в пам'яті
 - г) наявність пристроїв введення-виведення
4. Архітектура потоків даних – це:
 - а) архітектура, основною відмінністю якої від інших є спільне зберігання даних і машинних команд у комірках однієї й тієї ж пам'яті;
 - б) архітектура, головною відмінністю якої від інших подібних є те, що дані та оператори зберігаються окремо;
 - в) архітектура обчислювальних машин, в якій процесом обчислень керує потік даних;
 - г) різновид архітектури, в якій явно відбувається розподіл процесу обчислень між декількома обчислювальними пристроями, які виконують свої завдання одночасно.
5. Який елемент відсутній на материнській платі?
 - а) центральний процесор;
 - б) мікросхема BIOS;
 - в) система охолодження;
 - г) CD-ROM.
6. Основними характеристиками процесора є:
 - а) масовість;
 - б) розрядність і тактова частота;
 - в) дискретність;
 - г) універсальність.
7. Обсяг оперативної пам'яті вимірюється у:
 - а) байтах;
 - б) метрах;
 - в) бітах;
 - г) Герцах.
8. Пристрої, які підтримують взаємодію інших складових і перетворюють інформацію з однієї форми в іншу називаються:
 - а) драйверами;

- б) адаптерами;
- в) утилітами;
- г) НЖМД.

Практичні завдання

1. Скільки секунд потрібно модему, який передає повідомлення зі швидкістю 28800біт/с, для того, щоб передати 100 сторінок тексту на 30 сторінок, у кожній з яких міститься 60 символів, за умови, що кожен символ кодується 1 байтом?

2. Пристрій А передає інформацію пристрою С через пристрій В у рамках таких правил:

а. Інформація передається пакетами по 200 байт.

б. Пристрій В може одночасно приймати інформацію від пристрою А і передавати раніше отриману інформацію пристрою С.

в. Пристрій В може передавати черговий пакет пристрою С тільки після того, як повністю отримає цей пакет від пристрою А.

г. Пристрій В володіє необмеженим за об'ємом буфером, в якому може зберігати отримані від пристрою А, але ще не відправлені пристрою С пакети.

Пропускна здатність каналу між А і В – 100 байт на секунду.

Пропускна здатність каналу між В і С – 50 байт на секунду.

Було відправлено три пакети інформації. Через скільки секунд С завершить прийом всієї інформації від А?

3. Швидкість передачі даних через ADSL-з'єднання дорівнює 128000 біт/с. Через це з'єднання передають файл розміром 625 Кбайт. Визначте час передачі файлу у секундах.

Контрольні запитання

1. Як називається архітектура з послідовним виконанням операцій?
2. Які компоненти входять до процесора?
3. У чому полягає відмінність гарвардської архітектури від фон-нейманівської?
4. Що таке архітектура потоків даних?
5. У чому полягає концепція паралельної архітектури?
6. З яких основних блоків складається комп'ютер?
7. Що таке тактова частота?
8. Що таке розрядність?
9. Яке призначення монтажної плати?
10. Яким чином визначити пропускну здатність запам'ятовувального пристрою?

Тема 3. Програмне забезпечення ПК

Мета: ознайомитись з класифікацією сучасного програмного забезпечення, поняттями транслятора, інтерпретатора, властивостями алгоритму.

План

1. Класифікація сучасного програмного забезпечення.
2. Характеристика системного програмного забезпечення.
3. Характеристика службового програмного забезпечення.
4. Характеристика прикладного програмного забезпечення.
5. Огляд систем програмування (інструментальних систем).
Транслятори та інтерпретатори.
6. Розв'язання задач за допомогою ЕОМ: основні етапи.
7. Поняття алгоритму, його основні властивості та способи опису.

3.1 Класифікація сучасного програмного забезпечення

Програмне забезпечення можна поділити на три класи: системне, прикладне та інструментальне. Наведена класифікація є досить умовною. Інтеграція програмного забезпечення призвела до того, що практично будь-яка програма має риси кожного класу.

Стосовно роботи на комп'ютері часто використовують термін **програмне забезпечення** (software), під яким розуміють сукупність програм, процедур і правил, а також документації, що стосуються функціонування системи оброблення даних.

Програмне та апаратне забезпечення у комп'ютері працюють у нерозривному зв'язку та взаємодії.

Склад програмного забезпечення обчислювальної системи називається **програмною конфігурацією**.

Міжпрограмний інтерфейс – це розподіл програмного забезпечення на декілька, пов'язаних між собою, рівнів. Рівні програмного забезпечення складають піраміду (рис. 3.1), де кожен вищий рівень базується на програмному забезпеченні попередніх рівнів. Виділяють такі рівні: **базовий, системний, службовий та прикладний**.

Базовий рівень є найнижчим і відповідає за взаємодію з базовими апаратними засобами.

Базове програмне забезпечення міститься у складі базового апаратного забезпечення і зберігається у спеціальних мікросхемах постійного запам'ятовувального пристрою (ПЗП), утворюючи базову систему введення/виведення **BIOS** (BIOS (Basic Input-Output System)). Саме тут зберігаються основні налаштування комп'ютера. За допомогою BIOS можна змінити швидкість роботи процесора, параметри роботи для інших внутрішніх і

деяких зовнішніх пристроїв комп'ютера). Програми та дані записуються у ПЗП на етапі виробництва і не можуть бути змінені у процесі експлуатації.



Рисунок 3.1 – Класифікація програмного забезпечення за рівнями

Системний рівень є перехідним, забезпечує взаємодію інших програм комп'ютера з програмами базового рівня і, відповідно, з програмного забезпечення. І від програм цього рівня залежить експлуатаційні показники всієї обчислювальної системи.

Сукупність програмного забезпечення системного рівня утворює **ядро** операційної системи. Наявність ядра є першою умовою для роботи користувача з обчислювальною системою.

Ядро операційної системи виконує такі **функції**:

- 1) керування пам'яттю;
- 2) процеси введення/виведення;
- 3) керування файловою системою;
- 4) організація взаємодії процесів;
- 5) облік використання ресурсів та ін.

Службовий рівень також взаємодіє з програмами базового та системного рівня, але призначення службових програм полягає в автоматизації робіт, перевірці та налаштування комп'ютерної системи.

3.2 Характеристика системного програмного забезпечення

Системний рівень є перехідним. Програми цього рівня забезпечують взаємодію інших програм комп'ютера з програмами базового рівня і безпосередньо з апаратним забезпеченням. Від програм цього рівня залежать експлуатаційні показники всієї обчислювальної системи. При під'єднанні до комп'ютера нового обладнання, на системному рівні повинна бути встановлена програма, що забезпечує для решти програм взаємозв'язок із цим пристроєм. Конкретні програми, які призначено для взаємодії з конкретними пристроями, називають **драйверами** (цей підрівень ближче розташований до базового рівня).

Інший клас програм системного рівня відповідає за взаємодію з користувачем (цей підрівень знаходиться ближче до службового рівня). Завдяки

йому є можливість введення даних у обчислювальну систему, керувати її роботою й отримувати результат у зручній формі. Це засоби забезпечення користувацького інтерфейсу, від них залежить зручність та продуктивність роботи з комп'ютером.

Системне програмне забезпечення (ПЗ) – це комплекс програм, багато з яких постачаються разом з комп'ютером та документацією до нього і який призначено для управління роботою комп'ютера, розподілу його ресурсів, підтримки діалогу з користувачами, надання їм допомоги в обслуговуванні комп'ютера, а також для часткової автоматизації розробки нових програм.

Зазвичай, системні програми забезпечують взаємодію інших програм з апаратними складовими, організацію інтерфейсу користувача. **Системне ПЗ можна розділити на три частини:**

1. **Операційна система (ОС)** – це базовий комплекс програм, що виконує управління апаратною складовою комп'ютера або віртуальної машини; забезпечує керування обчислювальним процесом та організовує взаємодію з користувачем.

2. **Системи програмування.** Призначені для полегшення та часткової автоматизації процесу розробки програм. Основними компонентами є: транслятори, компілятори, інтерпретатори та ін.

3. **Сервісні програми** що розширюють можливості ОС. Наприклад: засоби комунікації, засоби перегляду та відтворення, засоби комп'ютерної безпеки, утиліти для форматування та дефрагментації диска та ін.

Тобто **системний рівень:** драйвер, операційна система, деякі службові програми або сервісні програми (утиліти). Як раз саме у частині утиліт і відбувається перетікання на службовий рівень.

3.3 Характеристика службового програмного забезпечення

Програми цього рівня взаємодіють як із програмами базового рівня, так і з програмами системного рівня.

Службові програми (утиліти) – це програми для автоматизації робіт по перевірці та налаштуванню комп'ютерної системи, а також для покращення функцій системних програм.

Деякі службові програми (програми обслуговування) відразу додають до дистрибутиву операційної системи, але більшість є зовнішніми програмами і розширюють функції операційної системи. Тобто, у **розробці службових програм відслідковуються два напрямки:**

- інтеграція з операційною системою (і в цій частині змивається границя між службовим і системним рівнем);
- автономне функціонування.

До службового ПЗ відносять такі програми:

1. **Диспетчери файлів** (файлові менеджери). За їх допомогою виконується більшість операцій з обслуговування файлової структури: копіювання, переміщення, перейменування файлів, створення каталогів, папок, знищення об'єктів, пошук файлів та навігація у файловій структурі.

2. **Засоби стиснення даних** (архіватори). Призначені для створення архівів. Архівні файли мають підвищену щільність запису інформації, внаслідок чого, ефективніше використовуються носії інформації.

3. **Засоби діагностики**. Призначені для автоматизації процесів діагностування програмного та апаратного забезпечення. Їх використовують для виправлення помилок і для оптимізації роботи комп'ютерної системи. Наприклад, POST – Power-On Self Test – процедура самоперевірки при увімкненні, виконується при кожному вмиканні комп'ютера, діагностичні програми виробників. Якщо комп'ютер справний, то при його увімкненні можна почути один короткий звуковий сигнал. Характер звукових кодів залежить від версії BIOS та компанії-розробника. Більшість відомих виробників комп'ютерів випускають для своїх систем спеціалізоване діагностичне програмне забезпечення, яке містить набір тестів що дозволяють ретельно перевірити всі компоненти, наприклад, AMIDiag Suite, MicroScope.

4. **Програми інсталяції** (встановлення). Призначені для контролю за доданням у поточну програмну конфігурацію нового програмного забезпечення. Вони слідкують за станом і зміною оточуючого програмного середовища, відслідковують та протоколюють утворення нових зв'язків, які було загублено під час знищення певних програм. Прості засоби управління встановленням та знищенням програм містяться у складі операційної системи, але можуть використовуватись і додаткові службові програми.

5. **Засоби комунікації**. Дозволяють встановлювати з'єднання з віддаленими комп'ютерами, передають повідомлення електронної пошти, пересилають факсимільні повідомлення тощо.

6. **Засоби перегляду та відтворення**. Використовуються переважно для роботи з файлами, їх необхідно завантажити у «рідну» прикладну систему і внести необхідні виправлення. Але, якщо редагування не потрібне, існують універсальні засоби для перегляду (тексту) або відтворення (звук або відео) даних.

7. **Засоби комп'ютерної безпеки**. До них відносяться засоби пасивного та активного захисту даних від пошкодження, несанкціонованого доступу, перегляду та зміни даних. **Засоби пасивного захисту** – це службові програми, які призначено для резервного копіювання. Засоби активного захисту застосовують антивірусне програмне забезпечення. Для захисту даних від несанкціонованого доступу, їх перегляду та зміни використовують спеціальні системи, що базуються на криптографії.

3.4 Характеристика прикладного програмного забезпечення

Прикладний рівень – комплекс прикладних програм, за допомогою яких виконуються конкретні завдання. Прикладне програмне забезпечення призначено для розв'язання завдань за фаховою діяльності людини. Спектр таких програм є надзвичайно широким: від виробничих та наукових до навчальних та розважальних. Сюди відносяться розрахункові, навчальні, програми для моделювання, комп'ютерні ігри, тощо.

У структурі прикладного програмного забезпечення можна виділити:

1. Прикладні програми загального призначення – це комплекс програм, який одержав широке використання серед різних груп користувачів. Найбільш відомі:

– **текстові редактори або текстові процесори.** Дозволяють формувати, тобто оформлювати текст. Основними засобами текстових процесорів є засоби забезпечення взаємодії тексту, графіків, таблиць та інших об'єктів, що складають документ, а також засоби автоматизації процесів редагування та форматування;

– **графічні редактори.** Широкий клас програм, що призначено для створення та обробки графічних зображень. **Розрізняють три категорії: растрові** редактори (графічний об'єкт представлено у вигляді комбінації точок, які мають свою **яскравість** та колір); **векторні** редактори (відрізняються способом призначення даних, об'єктом є не точка, а лінія, яка розглядається як математична крива третього порядку і представлена математичною формулою, таке представлення більш компактне за растрове але вимагає більш продуктивних ресурсів обчислювальної системи); **3D** редактори (тривимірна графіка);

– **системи управління базами даних (СУБД).** Базою даних називають великі масиви даних, які організовано у табличні структури. **Основні функції СУБД:** створення порожньої структури бази даних; наявність засобів її заповнення або імпорту даних із таблиць іншої бази; можливість доступу до даних, наявність засобів пошуку і фільтрації. У зв'язку з поширенням мережевих технологій від сучасних СУБД вимагається можливість роботи з віддаленими та розподіленими ресурсами, що знаходяться на серверах Інтернету;

– **електронні таблиці.** Надають комплексні засоби для збереження різних типів даних та їх обробки. Основний акцент зміщено на перетворення даних, надано широкий спектр методів для роботи із числовими даними. Основна особливість електронних таблиць полягає в автоматичній зміні вмісту всіх комірок при зміні відношень, які визначено математичними або логічними формулами. Широке застосування знаходять у бухгалтерському обліку, аналізі фінансових та торговельних ринків, засобах обробки результатів експериментів, тобто у автоматизації регулярно повторюваних обчислень великих об'ємів числових даних;

– **системи автоматизованого проектування (САД-системи).** Призначено для автоматизації проектно-конструкторських робіт. Застосовуються у машинобудуванні, приладобудуванні, архітектурі. Окрім графічних робіт дозволяють проводити прості розрахунки та вибір готових конструктивних елементів з існуючої бази даних. Особливість САД-систем полягає в автоматичному забезпеченні на всіх етапах проектування технічних умов, норм та правил. САПР є необхідним компонентом для гнучких виробничих систем (ГВС) та автоматизованих систем управління технологічними процесами (АСУ ТП);

– **редактори HTML.** Особливий клас редакторів, що об'єднують у собі можливості текстових та графічних редакторів. Призначено для створення і редагування Web-сторінок Інтернету. Програми цього класу можна використовувати для підготовки електронних документів і мультимедійних видань;

– **настільні видавничі системи.** Автоматизують процес верстання поліграфічних видань. Займає проміжне становище між текстовими процесорами та САПР. Видавничі системи відрізняються розширеними засобами управління взаємодії тексту з параметрами сторінки і графічними об'єктами, але мають слабші можливості для автоматизації введення та редагування тексту. Їх доцільно застосовувати до документів, які було оброблено у текстових процесорах і графічних редакторах;

– **браузери** (засоби перегляду Web-документів). Призначено для перегляду електронних документів, створених у форматі HTML. Відтворюють окрім тексту та графіки, також музику, людську мову, радіопередачі, відеоконференції і дозволяють працювати з електронною поштою;

– **системи автоматизованого перекладу.** Розрізняють електронні словники та програми перекладу мови. Електронні словники – це засоби для перекладу окремих слів у документі. Потрібні для професійних перекладачів, які самостійно перекладають текст. Програми автоматичного перекладу отримують текст на одній мові і видають текст на іншій, тобто автоматизують переклад;

– **інтегровані системи діловодства.** Засоби для автоматизації робочого місця керівника;

– **бухгалтерські системи.** Містять у собі функції текстових, табличних редакторів та СУБД. Призначено для автоматизації підготовки початкових бухгалтерських документів підприємства та їх обліку, регулярних звітів з виробничої, господарської та фінансової діяльності у формі прийнятної для податкових органів, позабюджетних фондів та органів статистичного обліку;

– **фінансові аналітичні системи.** Використовують у банківських та біржових структурах. Дозволяють контролювати та прогнозувати ситуацію на фінансових, торгівельних та ринків сировини, виконувати аналіз поточних подій, генерувати звіти;

– **геоінформаційні системи (ГІС).** Призначено для автоматизації картографічних та геодезичних робіт на основі інформації, яку отримано топографічними або аерографічними методами;

– **системи відеомонтажу.** Призначено для цифрової обробки відеоматеріалів, монтажу, створення відео-ефектів, виправлення дефектів, додавання звуку, титрів та субтитрів. Окремі категорії представляють навчальні, довідкові та розважальні системи й програми. Характерною особливістю є підвищені вимоги до мультимедійної складової.

2. Прикладні програми спеціального призначення.

Прикладні програми спеціального призначення використовують у специфічній діяльності користувачів, функції специфічних систем залежать від їх призначення.

3. Пакети прикладних програм. Широко використовуються, наприклад, для статистичної обробки даних, бухгалтерського обліку, розрахунку будівельних конструкцій та ін. Наявність у комп'ютері різноманітних ППП дозволяє розв'язувати значну частину простих прикладних задач, майже без програмування. В цьому випадку завдання на розв'язування тієї чи іншої задачі записується у вигляді директиви спеціальною проблемно-орієнтованою мовою та повідомляється комп'ютеру.

Окремим блоком у програмному забезпеченні стоять **хмарні технології**.

Хмара – це деякий дата-центр або сервер або їх мережа, де зберігаються дані та програми, що з'єднуються з користувачами через Інтернет.

Хмарні технології – це технології для забезпечення повсюдного та зручного доступу на вимогу через мережу до спільного пулу обчислювальних ресурсів, що підлягають налаштуванню (наприклад, до комунікаційних мереж, серверів, засобів збереження даних, прикладних програм та сервісів), і які можуть бути оперативно надані та звільнені з мінімальними управлінськими затратами та зверненнями до провайдера.

Архітектура «хмарних» обчислень включає безліч «хмарних» складових, що взаємодіють один з одним через веб-сервіси – інтерфейси програмних додатків (анг.: Application Program Interface, скорочено API). Два найбільш важливих компоненти архітектури «хмарних» обчислень відомі як front end і back end. **Front end** – це частина архітектури, видима клієнтом, тобто користувачем ПК. Вона включає мережу клієнта (або окремий термінал) і додатки, що використовуються для доступу у «хмару» через інтерфейс користувача (веб-браузер). **Back end** «хмарної» архітектури обчислень – це власне обчислювальна «хмара», що складається з комп'ютерів, серверів та пристроїв зберігання даних (рис. 3.2).

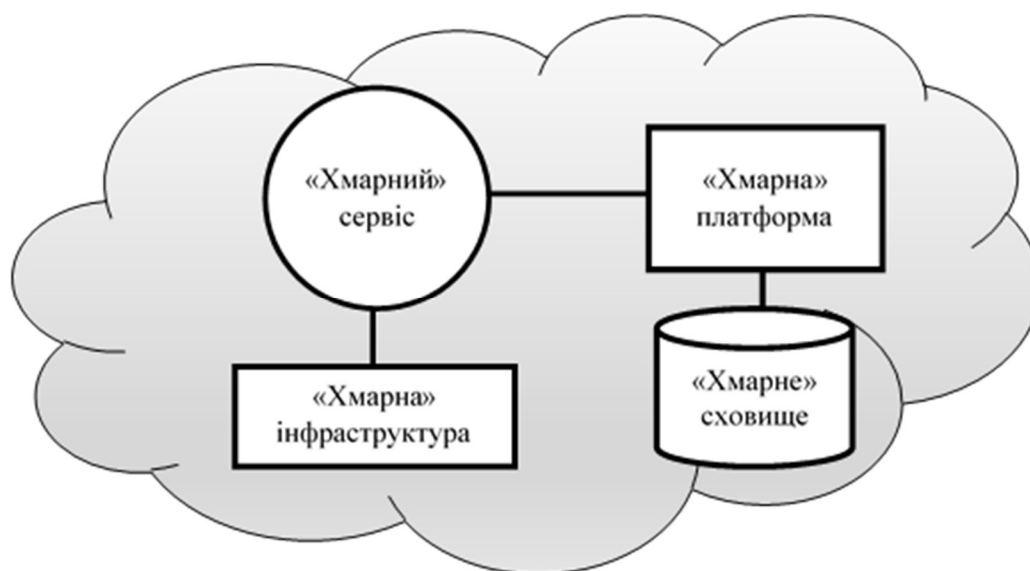


Рисунок 3.2 – Приклад архітектури «хмарних» обчислень

Хмарні технології являють собою загальний термін для всього, що включає у себе послуги хостингу через Інтернет. Основні моделі представлення «хмарних» послуг наведено у таблиці 3.1.

Таблиця 3.1 – Основні категорії «хмарних» послуг

Назва послуги	Опис
Platform as a service, PaaS («Платформа як сервіс»)	Надання інтегрованої ІТ-платформи для створення, тестування, розгортання та підтримки додатків. Клієнт отримує дистанційний доступ до програмної інфраструктури та засобам розробки
Software as a service, SaaS («Програмне забезпечення як сервіс»)	Провайдер розміщує у себе додаток, надаючи можливість платного доступу через Інтернет. Користувачі сплачують тільки за користування додатком, а не володіння ним.
Data as a service, DaaS («Дані як сервіс»)	Надання даних на вимогу користувача незалежно від його або провайдера географічного положення або організаційної приналежності.
Infrastructure as a service, IaaS («Інфраструктура як сервіс»)	Являє собою віртуальний сервер instance API для запуску, зупинки, доступу, налаштування своїх віртуальних серверів та систем збереження. IaaS дозволяє компанії платити саме за стільки потужностей, скільки їй необхідно. Дану модель іноді називають «комунальні обчислення».
Communication as a service, CaaS («Комунікації як сервіс»)	Як сервіси надаються послуги зв'язку – IP-телефонія, поштові послуги, Unified Communications та ін.
Everything as a service, EaaS («Все як сервіс»)	Комплекс всіляких «хмарних» сервісів, які задовольняють будь-які ІТ-потреби.

На сьогодні хмарний сервіс має **три основні характеристики**, які відрізняють його від звичайного сервісу:

- режим «ресурси за запитом»;
- еластичність;
- незалежність від елементів керування інфраструктурою.

Переваги хмарних обчислень:

- доступність;
- низька вартість;
- сплата фактичного використання ресурсів;
- гнучкість;
- надійність;

- безпека.

Недоліки:

- постійне з'єднання з мережею;
- програмне забезпечення та його кастомізація;
- конфіденційність;
- надійність;
- висока вартість обладнання.

3.5 Огляд систем програмування (інструментальних систем). Транслятори та інтерпретатори

Система програмування – система автоматичного програмування, що складається з мови програмування, компілятора або інтерпретатора програм, які написані на цій мові, відповідної документації, а також допоміжних засобів для підготовки програм до виконання.

До складу засобів розробки програмного забезпечення можуть входити такі програми:

- **асемблери** – комп'ютерні програми, що здійснюють перетворення програми у формі вихідного тексту на мові асемблера у машинні команди у вигляді об'єктного коду;
- **транслятори** – програми, які призначено для перекладу програми з однієї мови програмування на іншу;

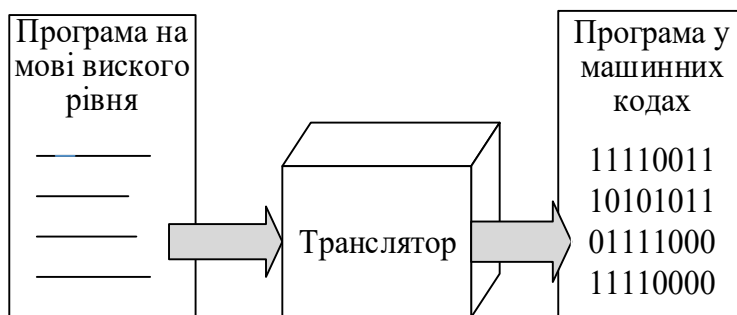


Рисунок 3.3 – Схема роботи транслятора

- **компоновщики** (редактори зв'язків) – програми, які виробляють компоновку – приймають на вхід один або кілька об'єктних модулів і збирають по ним здійснений модуль;
- **препроцесори вихідних текстів** – це комп'ютерні програми, що приймають дані на вході, і видають дані, призначені для входу іншої програми, наприклад такий, як компілятор;
- **налагоджувачі** (debugger) – програми, що є модулем середовища розробки або окремим додатком, призначеним для пошуку помилок у програмі;
- **спеціалізовані редактори вихідних текстів** – програми, необхідні для створення і редагування вихідного коду програм. Спеціалізований редактор вихідних текстів може бути окремим додатком або вбудованим в інтегроване середовище розробки та ін.

Існує два великі класи трансляторів: компілятори й інтерпретатори.

Компілятори – програми, що переводять текст програми на мові високого рівня в еквівалентну програму на машинній мові (рис. 3.4).

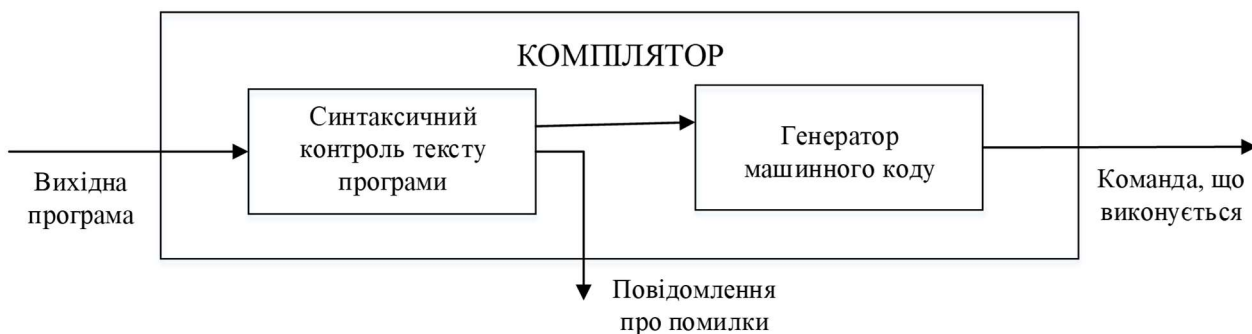


Рисунок 3.4 – Схема роботи компілятора

Інтерпретатори – програми, що аналізують команди або оператори програми і тут же виконують їх.

При використанні компіляторів весь початковий текст програми перетворюється у машинні коди, і саме ці коди записуються у пам'ять мікропроцесора. При використанні інтерпретатора у пам'ять мікропроцесора записується початковий текст програми, а трансляція проводиться при виконанні чергового оператора. Природно, що швидкодія інтерпретаторів набагато нижча порівняно з компіляторами, оскільки при використанні оператора в циклі він транслюється багато разів.

До інструментального ПЗ також відносять деякі **системи управління базами даних (СУБД)**.

СУБД – це спеціалізований комплекс програм, який призначено для організації та ведення баз даних. Оскільки системи управління базами даних не є обов'язковим компонентом обчислювальної системи, їх не відносять до системного програмного забезпечення. А так як окремі СУБД здійснюють лише службову функцію при роботі інших видів програм (веб-сервери, сервери додатків), їх не завжди можна віднести до прикладного програмного забезпечення. З цих причин їх часто відносять до інструментального програмного забезпечення.

Основні функції таких СУБД:

- управління даними у зовнішній пам'яті (на дисках);
- керування даними в оперативній пам'яті з використанням дискового кеша;
- фіксація змін у спеціальних журналах, резервне копіювання і відновлення бази даних після збоїв;
- підтримка мов БД (мова визначення даних, мова маніпулювання даними).

Інструментальне програмне забезпечення призначено для розробки всіх видів інформаційно-програмного забезпечення. При цьому під інформаційним забезпеченням розуміють сукупність попередньо підготовлених даних, необхідних для роботи програмного забезпечення. Наприклад, будь-яка сучасна

програма має вбудовану довідку для роботи з цією програмою. Файл довідки являє собою інформаційне забезпечення. До **інструментального програмного забезпечення** відносять:

- редактори (текстові, графічні, музичні тощо);
- системи табличної обробки даних (табличні процесори);
- системи управління базами даних;
- транслятори мов програмування;
- інтегровані системи для виробництва.

3.6 Розв'язання задач за допомогою ЕОМ: основні етапи

Робота з розв'язання прикладної задачі з використанням комп'ютера проходить у кілька етапів, кожен із яких важливий по-своєму, і тільки правильне виконання кожного з етапів призводить до розв'язання задачі у цілому.

Перед виконанням будь-якого проекту доцільно змодельовати хід виконання і спрогнозувати результати роботи. У цьому може допомогти побудова інформаційної моделі об'єкта-оригіналу або явища (процесу), що досліджується. У більшості випадків це дозволяє швидко і з невеликими витратами розрахувати результати виконання масштабної реальної задачі.

Основними етапами є:

1. **Постановка задачі** передбачає визначення умов задачі (що дано, що необхідно, які є припущення, які результати та в якому вигляді повинні бути отримані), проведення змістовного аналізу з уточненням мети розв'язання задачі. У межах цього етапу визначаються умови отримання результатів і те, які результати вважатимуться вірними.

2. **Побудова математичної моделі**, на цьому етапі змістовний опис задачі замінюється математичною моделлю та обґрунтовано обирається метод розв'язання.

3. **Складання алгоритмів** на основі методу, що обрано.

4. **Опис алгоритму** мовою програмування.

5. **Тестування** та налагодження програм.

6. **Остаточне виконання програми**, аналіз результатів та експлуатація програми.

Перед виконанням будь-якого проекту доцільним є моделювання ходу виконання і прогноз результатів роботи. У цьому може допомогти побудова інформаційної моделі об'єкта або процесу, що досліджується.

3.7 Поняття алгоритму, його основні властивості та способи опису

Алгоритм – це послідовність точно визначених дій, які спрямовано на досягнення зазначеної мети чи на розв'язання поставленої задачі.

Поняття алгоритму в інформатиці є фундаментальним, тобто таким, котре не визначається через інші ще більш прості поняття (для порівняння у фізиці – поняття простору і часу, у математиці – точка).

Будь-який виконавець (і комп'ютер зокрема) може виконувати тільки обмежений набір операцій. Тому **алгоритми** повинні мати наступні **властивості**:

– **зрозумілість**. Щоб виконавець міг досягти поставленої перед ним мети, використовуючи даний алгоритм, він повинен уміти виконувати кожен його вказівку, тобто розуміти кожен з команд, що входять до алгоритму;

– **визначеність (однозначність)**. Зрозумілий алгоритм не повинен містити вказівки, зміст яких може сприйматися неоднозначно. Наприклад, вказівка «посоли за смаком» є неоднозначними, тому що у різних випадках можуть призвести до різних результатів. Крім того, в алгоритмах є неприпустимими такі ситуації, коли після виконання чергового розпорядження алгоритму виконавцю не зрозуміло, що потрібно робити на наступному кроці;

– **дискретність**. Як було згадано вище, алгоритм задає повну послідовність дій, які необхідно виконувати для розв'язання задачі. При цьому, для виконання цих дій їх розбивають у визначеній послідовності на прості кроки. Виконати дії наступного етапу можна лише виконавши дії попереднього. Ця розбивка алгоритму на окремі елементарні команди, що легко виконуються даним виконавцем, і називається дискретністю;

– **масовість**. Дуже важливо, щоб алгоритм забезпечував розв'язання не однієї окремої задачі, а міг виконувати розв'язання широкого класу задач даного типу. Отож, під масовістю алгоритму мається на увазі можливість його застосування для вирішення великої кількості однотипних завдань;

– **результативність**. Взагалі кажучи, очевидно, що виконання будь-якого алгоритму повинно завершуватися одержанням кінцевих результатів. Тобто ситуації, що в деяких випадках можуть призвести до так званого «зацікнення», повинно бути виключено при створенні алгоритму.

Далі з'ясуємо, яким чином можна подати алгоритм виконавцю. Існує декілька методів запису алгоритмів, вибір яких залежить від його реалізації.

Способи опису алгоритмів:

– словесний;

– подача алгоритму у вигляді таблиць, формул, схем, малюнків;

– запис алгоритмів за допомогою блок-схеми. Цей метод було запропоновано в інформатиці для наочності представлення алгоритму за допомогою набору спеціальних блоків;

– за допомогою навчальних алгоритмічних мов (псевдокоди). Ці мови мають певний синтаксис і є максимально наближеними до машинної мови (мови програмування);

– за допомогою мови програмування.

Тестові завдання

1. За взаємодію з базовими апаратними засобами відповідає такий рівень ПЗ:

- а) системний;
- б) службовий;

- в) прикладний;
 - г) базовий.
2. Мікросхему BIOS встановлено на:
- а) материнській платі;
 - б) на моніторі;
 - в) на задній панелі системного блоку;
 - г) в USB порту.
3. Операційна система відноситься до такого рівня ПЗ:
- а) базового;
 - б) системного;
 - в) прикладного;
 - г) інструментального.
4. Є компонентом системи програмування:
- а) транслятор;
 - б) монітор;
 - в) принтер;
 - г) USB-порт.
5. Не є функцією ядра операційної системи:
- а) керування файловою системою;
 - б) керування пам'яттю;
 - в) організація взаємодії процесів;
 - г) виконання JavaScript сценаріїв.
6. Драйвер – це:
- а) програма, яку призначено для взаємодії з конкретними пристроями;
 - б) синонім слова «принтер»;
 - в) назва фірми-виробника моніторів;
 - г) тип операційної системи.
7. Система програмування відноситься до такого рівня ПЗ:
- а) базового;
 - б) системного;
 - в) прикладного;
 - г) службового.
8. Утиліта – це:
- а) комп'ютерна програма, що розширює стандартні можливості операційної системи;
 - б) драйвер для принтера;
 - в) комп'ютерна програма, що надає користувачу інтерфейс для роботи з файловою системою та файлами;
 - г) важлива частина операційної системи Windows, що займає майже увесь простір екрану монітора і знаходиться у полі зору користувача значну частину часу.
9. Текстові редактори відносяться до:
- а) прикладного програмного забезпечення;
 - б) базового ПЗ;

- в) службового ПЗ;
 - г) не є програмним забезпеченням.
10. Не є моделлю представлення хмарних послуг:
- а) PaaS;
 - б) SaaS;
 - в) SaaS;
 - г) IMDB.
11. Не є властивістю алгоритму:
- а) багатозначність;
 - б) зрозумілість;
 - в) масовість;
 - г) дискретність.

Практичні завдання

1. Складіть словесний алгоритм розв'язання рівняння

$$4x^2 + 4x + 1 = 0$$
2. За допомогою блок-схеми створіть алгоритм:
 - а. завантаження звіту з лабораторної роботи до СЕЗН Moodle;
 - б. розв'язання квадратного рівняння $x^2 + 2x + 1 = 0$;
 - в. визначення, чи належить точка із заданими координатами до однієї з координатних осей;
 - г. переведення числа у двійкову систему числення.
3. За допомогою псевдокоду складіть алгоритм обчислення:
 - а. площі поверхні кулі;
 - б. об'єму чотирикутної призми;
 - в. площі поверхні чотирикутної піраміди.
4. Мовою програмування C складіть алгоритм запиту у користувача будь-якої інформації (на власний вибір) і подальшого виведення її на екран.

Контрольні запитання

1. Які рівні програмного забезпечення можна виділити?
2. Як розшифровується BIOS?
3. Які програми можна віднести до службового ПЗ? Прикладного?
4. Що таке хмара?
5. У чому полягають недоліки та переваги хмарних обчислень?
6. У чому полягає відмінність між компілятором і інтерпретатором?
7. Що таке алгоритм? Які є способи опису алгоритмів?

Тема 4. Текстові процесори

Мета: вивчити поняття про текстовий процесор, ознайомитись з форматами файлів текстових документів, загальною структурою текстового процесора та його основними елементами.

План

1. Поняття про текстовий процесор.
2. Класифікація програм для роботи з текстом та їх основні функції.
3. Формати файлів текстових документів.
4. Загальна структура текстового процесора та його основні елементи.

4.1 Поняття про текстовий процесор

Значна частина інформації, що сприймається людиною, передається у вигляді тексту. Сьогодні робота з текстовими документами найбільш ефективна за допомогою комп'ютерів.

Текстовий документ – це інформаційний блок, що містить в якості основної інформації текст, зображення, схеми, таблиці тощо, також може знаходитися додаткова інформація (зміст, посилання, заголовки, різні види графічної інформації та ін.).

Програми для створення та обробки текстової інформації називаються, **системами обробки текстової інформації**.

Такі програми поділяються на три класи: текстові редактори, текстові процесори та видавничі системи.

Текстовий редактор – програма, яку призначено для введення, редагування та збереження у файлах неформатованого тексту. Наприклад, Notepad, EmEditor (рис. 4.2), AkelPad (рис. 4.3), Geany та ін.

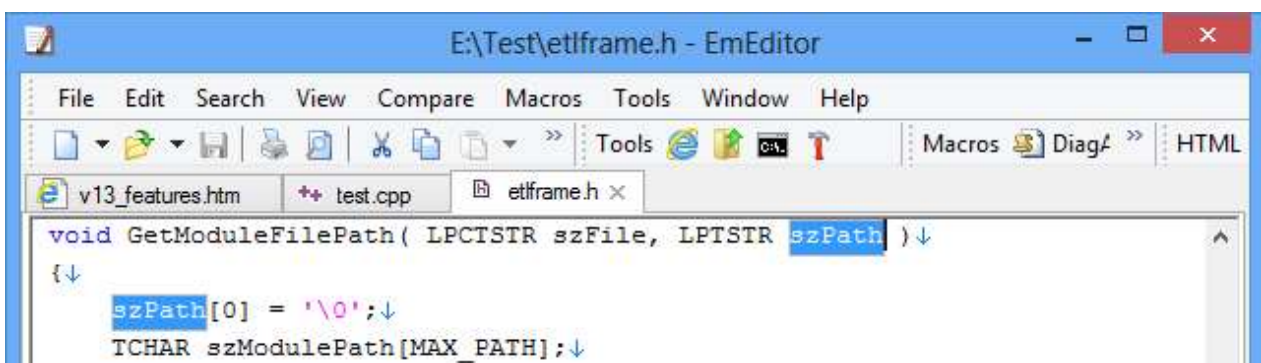


Рисунок 4.1 – Зовнішній вигляд текстового редактора EmEditor



Рисунок 4.2 – Зовнішній вигляд текстового редактора AkelPad

Текстовий процесор – програма, що дозволяє виконувати всі три групи операцій: введення, редагування та форматування тексту, крім того вони дозволяють виконувати багато додаткових операцій, таких як: додання у текстовий документ таблиць, графічних зображень, перевірка орфографії, автоматичне створення змісту, виконувати перенос слів та ін. Наприклад, Word, LibreOfficeWriter, PolyEdit (рис. 4.3), LaTeX (рис. 4.4).

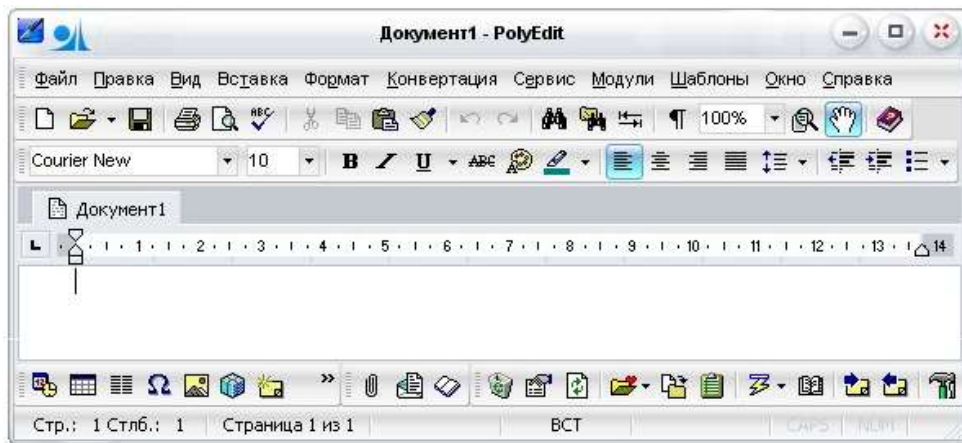


Рисунок 4.3 – Зовнішній вигляд текстового процесора PolyEdit

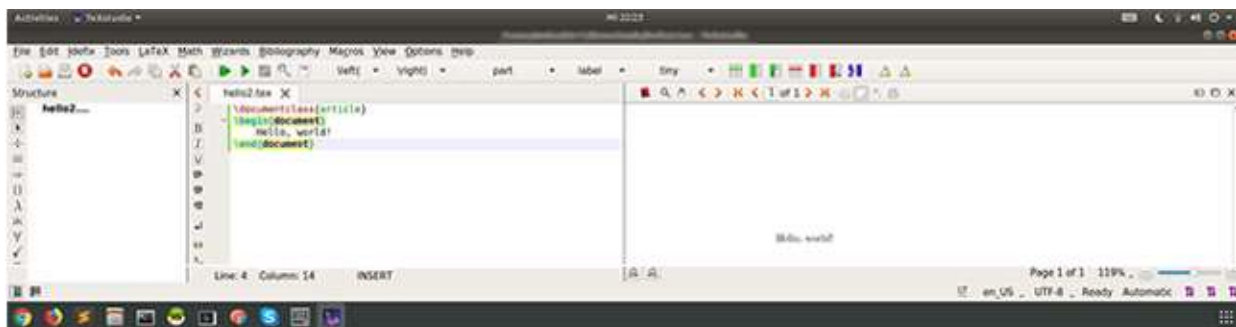


Рисунок 4.4 – Зовнішній вигляд текстового процесора LaTeX

Деякі текстові процесори є так званими WYSIWYG редакторами. Назву отримано за першими буквами фрази What You See What You Get – те, що ти бачиш – є те, що ти отримаєш. При такому підході гарантується повна відповідність зовнішнього виду документа на екрані комп'ютера і його

друкованій копії. До редакторів такого типу відносяться Word, LibreOfficeWriter, OpenOffice Writer.

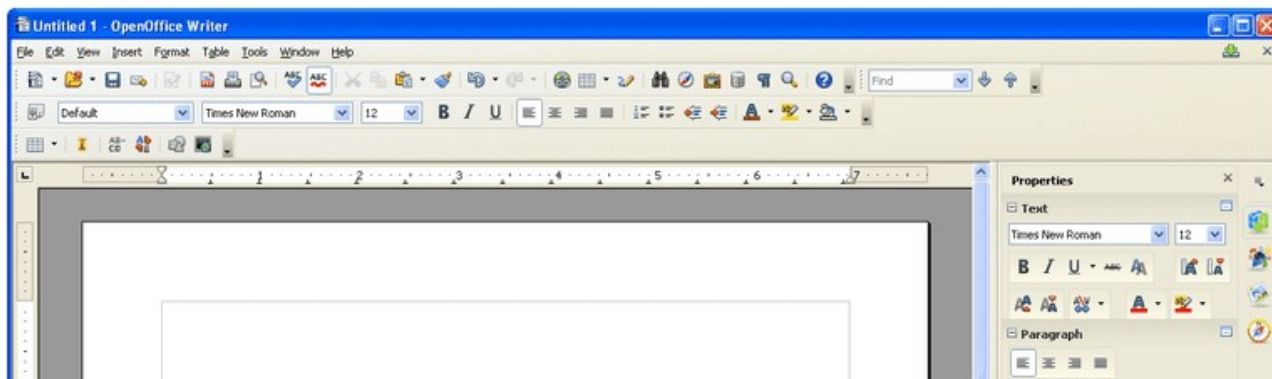


Рисунок 4.5 – Документ у OpenOffice Writer

Видавнича система – це комплекс, що складається з персональних комп'ютерів, пристроїв програмного та мережевого забезпечення і використовується для введення і редагування тексту, створення і обробки зображень, верстки і виготовлення оригінал-макетів, коректурних листів, друкарських форм та ін. Наприклад, Adobe InDesign, Open Conference System, QuarkXPress та ін.

4.2 Класифікація програм для роботи з текстом та їх основні функції

Класифікація текстових процесорів:

1. **За способом розповсюдження:**
 - а) комерційні (MS Word, Accent 2, Breeze, StratusPad 1.0);
 - б) вільно розповсюджені (AbiWord, LibreOffice Writer, OpenOffice Writer).
2. **За способом використання:**
 - а) автономні (використовує кожен користувач окремо);
 - б) хмарні або мережеві (використовують при роботі у локальній мережі або онлайн з різними правами доступу користувачів).
3. **За формою тексту:**
 - а) лінійні (текст подається символами алфавіту та синтаксичними знаками);
 - б) нелінійні (крім алфавіту, синтаксичних знаків і символів псевдографіки використовують широкий набір спеціальних знаків).
4. **За спеціалізацією:**
 - а) процесори загального призначення (MS Word);
 - б) редактори наукових документів (TEX);
 - в) редактори вихідних текстів програм (Multi-Edit, вбудовані редактори систем програмування Pascal, C, Java).

Звісно, за допомогою Word можна підготувати і текст програми, а за допомогою Multi-Edit документ загального призначення. «Спеціалізація»

редактора полягає у тому, що у ньому додано (або оптимізовано) функції, які є необхідними для обслуговування документів певного типу.

На рисунку 4.6. представлено основні функції текстових редакторів.

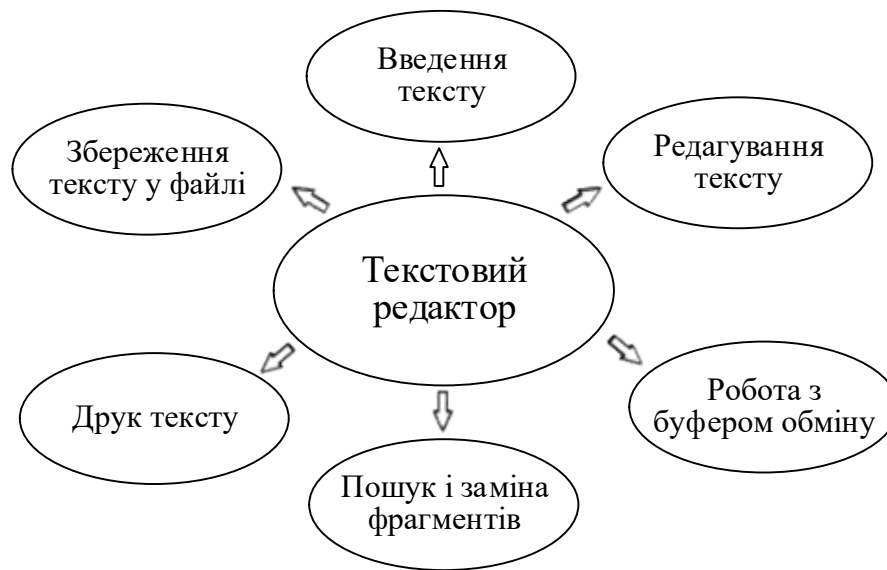


Рисунок 4.6 – Основні можливості текстових редакторів

Базові можливості текстових процесорів:

- набирання тексту (введення інформації у пам'ять комп'ютера);
- можливість введення тексту кількома мовами;
- редагування (зміна, переміщення, видалення, копіювання) фрагментів тексту;
- форматування (зміна параметрів шрифтів, абзаців, сторінок) фрагментів тексту;
- опрацювання декількох текстів одночасно;
- пошук потрібних фрагментів тексту;
- можливість додання у текст різних об'єктів (рисуноків, схем, таблиць, графіків, формул, спеціальних знаків, які немає на клавіатурі);
- перевірка правопису та автоматичне виправлення помилок при введенні тексту;
- друкування всього тексту або його фрагментів;
- створення стандартних документів (листів, резюме, записок та ін.);
- збереження тексту на зовнішньому носію.

Розширені функції текстових процесорів:

- використання колонок;
- вставлення виносок;
- генерація автоматичного змісту;
- додання складних математичних формул;
- сортування;
- засоби автоматизації роботи користувача;
- налаштування інтерфейсу.

Основні можливості видавничих систем:

- зручне маніпулювання блоками тексту;
- широкі можливості для роботи з ілюстративними об'єктами;
- швидка зміна формату документа;
- забезпечення поліграфічної якості продукції.

4.3 Формати файлів текстових документів

Існує багато **форматів файлів текстових документів**. Причому кожна із систем використовує певний набір форматів, для роботи з файлами інших форматів у кожній з них міститься конвертор. Далі наводиться перелік форматів текстових файлів та відповідних ним текстових редакторів:

- ***.DOC** (Microsoft Word Document) – формат файлів, який розроблено корпорацією Microsoft. Підтримується MS Word та іншими текстовими процесорами. Існує формат для MS Word 6.0/95 та відмінний від нього формат MS Word97/2000, що використовує Unicode-кодування;

- ***.DOCX** (Microsoft Office Word 2007 Document) – текст із форматуванням і вставленими об'єктами, основний формат Microsoft Word 2007/2010;

- ***.RTF** (Rich Text Format) – з англійської «формат збагаченого тексту», пропріетарний міжплатформений формат збереження розмічених текстових документів, який розроблено корпорацією Microsoft. Підтримується програмами для MSDOS та більшістю текстових процесорів. Він зберігає вихідне форматування документів, а також стилі зображення символів. Не підтримує стиснення. Не містить макросів, чим запобігає переносу відповідних вірусів.

- ***.ODT** (Open Document Text) – текст з форматуванням і вставленими об'єктами, основний формат LibreOffice Writer;

- ***.PDF** (Adobe Acrobat Document) – формат для перегляду та друкування з незмінним вихідним форматуванням для різних платформ. У випадку використання формату PDF, можна бути впевненим, що користувачі побачать публікацію саме такою, якою вона була створена. Формат PDF незамінний, якщо потрібно отримати точну копію документа;

- ***.FB2** (FictionBook) – формат представлення електронних версій книжок у вигляді XML-документів, де кожен елемент описується своїми тегами. Стандарт має забезпечити сумісність з будь-якими пристроями та форматами. Документи, що мають розширення fb2, можуть містити структурну розмітку основних елементів тексту, деяку кількість інформації про книжку, а також вкладення з двійковими файлами, в яких можуть зберігатися ілюстрації, наприклад, обкладинка. У форматі немає складної верстки тексту, відсутня підтримка нумерованих і маркованих списків, засобів формування «зауважень на полях», авторського форматування віршів і векторної графіки;

- ***.EPUB** (Electronic Publication) – відкритий формат електронних книг з розширенням .epub. Формат дозволяє видавцям здійснювати та розповсюджувати цифрову публікацію в одному файлі, забезпечуючи сумісність

між програмним і апаратним забезпеченням, що є необхідним для відтворення цифрових книжок та інших публікацій з плаваючою версткою;

- ***.PPTX** (Microsoft PowerPoint 2007 Presentation) – це файл презентації Microsoft PowerPoint, програми, що використовується для створення презентацій слайд-шоу; містить набір слайдів, у тому числі текст, зображення, форматування, анімації, рисунки та інші мультимедійні об'єкти;

- ***.CSV** (Comma Separated Values File) – текстовий формат, який призначено для представлення табличних даних. Програми для редагування csv-файлів: LibreOffice Calc, Gnumeric, Emacs, Microsoft Excel, Numbers, ТаблицяПро, CSVed, KSpread, Google Docs, Блокнот;

- ***.XLS** (Microsoft Excel Worksheet) – стандартний формат робочих книг Excel;

- ***.PDF** Portable Document Format (PDF/A) – міжплатформенний відкритий формат електронних документів, який призначено для представлення поліграфічної продукції в електронному вигляді. Для перегляду існує багато програм, а також офіційна безкоштовна програма Adobe Reader. Формат PDF дозволяє запроваджувати необхідні шрифти, векторні та растрові зображення, форми, мультимедіа-вставки та ін. У цьому форматі розповсюджується більшість супровідної документації;

- ***.DJVU** (DjVu Document) – технологія стиснення з втратами зображень, яку розроблено спеціально для збереження сканованих документів – книжок, журналів, рукописів та ін., де являються формули, схеми, малюнки і рукописні символи робить надзвичайно трудомістким їхнє повноцінне розпізнавання;

- ***.TXT** (Text Document) – текст з розбиттям на абзаци без форматування. Підтримується всіма операційними системами і всіма програмами. Являє собою текстовий файл у DOS-кодуванні;

- ***.HTML, *.HTM** (HTML Document) – (англійською HyperText Markup Language – мова розмітки гіпертекстових документів) стандартна мова розмітки веб-сторінок в Інтернеті і відповідне розширення файлу. HTML-документи являють собою ASCII-файли, доступні для перегляду та редагування у будь-якому текстовому редакторі. Відмінністю від звичайного текстового файлу є те, що в HTML-документах присутні спеціальні команди – теги, які визначають правила форматування документа. Додаючи теги до звичайного тексту, можна примусити програму перегляду (браузер) відображати цей текст певним чином і розміщувати на сторінці зображення;

- **XML** (англійською Extensible Markup Language – розширювана мова розмітки) – запропонований консорціумом World Wide Web (W3C) стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет і відповідне розширення файлу.

Вміст файлів останніх трьох типів можна редагувати у середовищі будь-якого текстового редактора. У цьому випадку тип вказує лише на програму, якою потрібно відкривати файл.

4.4 Загальна структура текстового процесора та його основні елементи

Типова структура інтерфейсу текстового процесора містить у собі ряд елементів (рис. 4.7).

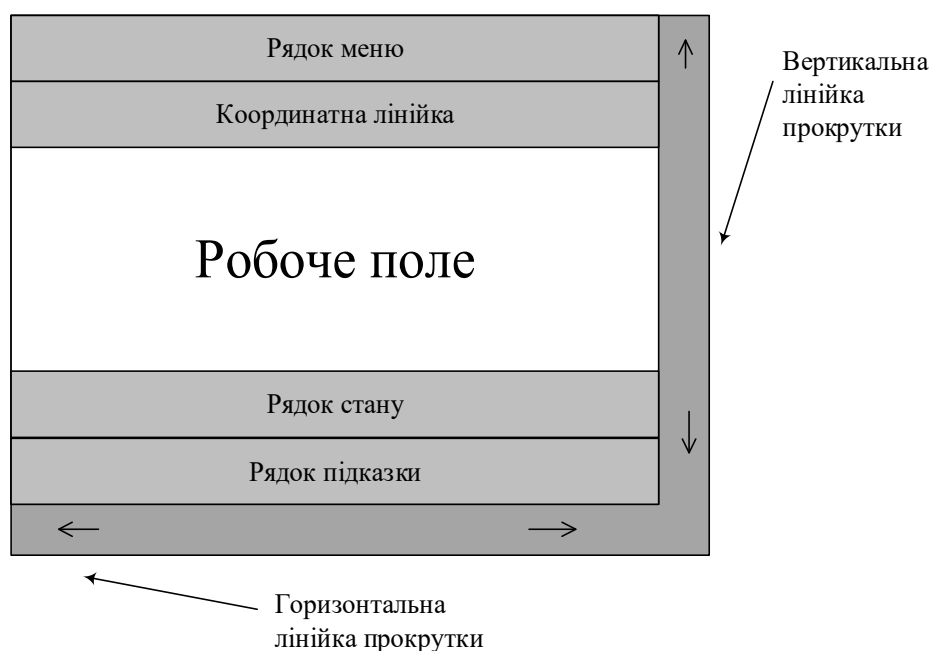


Рисунок 4.7 – Типовий інтерфейс текстового процесора

Рядок меню містить імена груп команд, що об'єднані за функціональною ознакою. Вибір режиму з рядка меню відкриває відповідне підменю, а вибір певної опції в ньому забезпечує доступ до меню більш низького рівня. Така система вкладених меню складає основу інтерфейсу текстового процесора. Команди меню обираються за допомогою миші, клавіш керування курсором або комбінацією клавіш.

Рядок стану містить ім'я документу, що редагується, і визначає положення курсору у цьому документі. У рядку вводиться додаткова інформація.

Рядок підказки містить інформацію про можливі дії користувача у поточний момент.

Робоче поле – це простір на екрані дисплея для створення документа і роботи з ним. Максимальний розмір поля визначається стандартними параметрами монітора.

Координатна лінійка – визначає межі документа і позиції табуляції. Розрізняють вертикальну і горизонтальну лінійки.

Лінійка прокрутки – служить для переміщення тексту документа у робочому полі вікна.

Курсор – миготлива лінія, що показує позицію робочого поля, де буде розміщено видимий символ або елемент тексту. Керування інтерфейсом здійснюється за допомогою клавіатури та миші.

Індикатори – знаки або символи, що відображають відповідні режими програми або комп'ютера.

Перемикач – елемент екранного інтерфейсу або команда, що використовується для вмикання або вимикання того чи іншого режиму. Індикатор може бути і перемикачем, якщо на ньому натиснути мишею.

Отже текстовий процесор – це прикладне програмне забезпечення, що використовується для створення текстових документів.

Корисні посилання

1. Путівник з текстових процесорів. URL : <http://www.ixbt.com/soft/textprocessors.shtml>

2. Текстовий процесор. URL : https://uk.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D0%B8%D0%B9_%D0%BF%D1%80%D0%BE%D1%86%D0%B5%D1%81%D0%BE%D1%80

3. Методичні рекомендації до виконання лабораторних робіт. URL : https://moodle.znu.edu.ua/pluginfile.php?file=/337580/mod_resource/content/2/Metod_rekomend_MSKiT.pdf

Тестові завдання

1. Який тип програм не відноситься до систем обробки інформації?
 - а) текстовий редактор;
 - б) текстовий процесор;
 - в) текстовий документ;
 - г) видавнича система.
2. Яка з наведених програм не є текстовим процесором?
 - а) PolyEdit;
 - б) Notepad;
 - в) LibreOfficeWriter;
 - г) LaTeX.
3. Не є редактором типу WYSIWYG:
 - а) Word;
 - б) OpenOffice Writer;
 - в) LibreOfficeWriter;
 - г) LaTeX.
4. Який з текстових редакторів є вільно розповсюджуваним?
 - а) MS Word;
 - б) Breeze;
 - в) OpenOffice Writer;
 - г) Accent 2.
5. Який з редакторів є комерційним?
 - а) AbiWord;
 - б) LibreOffice Writer;
 - в) Breeze;
 - г) OpenOffice Writer.
6. Основним форматом Microsoft Word 2007/2010 є:

- а) *.RTF;
 - б) *.DOCX;
 - в) *.PDF;
 - г) *.FB2.
7. Основним форматом LibreOfficeWriter є :
- а) *.PDF;
 - б) *.FB2;
 - в) *.RTF;
 - г) *.ODT.
8. Файл презентації Microsoft PowerPoint має формат:
- а) *.EPUB;
 - б) *.DOC;
 - в) *.PPTX;
 - г) *.XLS.
9. Відмінністю HTML-документа від звичайного текстового файлу є:
- а) наявність тегів;
 - б) розбиття тексту на абзаци без форматування;
 - в) відсутність засобів форматування тексту;
 - г) відсутність формування «зауважень на полях».
10. Не є моделлю представлення хмарних послуг:
- а) PaaS;
 - б) SaaS;
 - в) SaaS;
 - г) IMDB.

Практичні завдання

1. Порівняйте можливості, вкажіть переваги та недоліки текстових процесорів: KWord, MS Office Word, AbiWord, Libre Office, Open Office.
2. Самостійно знайти інформацію про будь-які 2 текстові процесори окрім вищезазначених.
3. Знайдіть та прочитайте статтю з енциклопедії Britannica про історію видавничої системи (<http://www.britannica.com/EBchecked/topic/159237/desktop-publishing>).
4. Створіть багаторівневий список на тему «Класифікація програмного забезпечення ЕОМ» (тема №3).
5. Використавши панелі інструментів або вказівки меню, створіть таблицю 6×8 такого вигляду:

Вирівняти по центру						
				Вирівняти знизу по правому краю		

Збережіть документ як шаблон у папку з номером групи.

Контрольні запитання

1. Що таке система опрацювання текстів?
2. Які вам відомі текстові редактори? Чим вони відрізняються між собою?
3. Що таке видавнича система і для чого вона призначена?
4. Що таке WYSIWYG? Наведіть приклади.
5. Для чого використовується формат RTF?
6. Які формати дозволяють отримати точну копію документа?
7. В яких форматах зберігаються цифрові книжки?
8. Які основні елементи інтерфейсу текстового процесора?
9. Для чого існують вертикальні та горизонтальні лінійки у вікні текстового редактора?
10. Для чого потрібен рядок стану та які елементи він містить?

Тема 5. Програми обробки електронних таблиць

Мета: вивчити поняття про електронну таблицю, ознайомитись з принципами роботи електронних таблиць, правилами створення макросів.

План

1. Поняття про електронну таблицю та табличний процесор. Основні функції табличних процесорів.
2. Загальна технологія роботи з електронними таблицями.
3. Типова структура вікна електронної таблиці.
4. Основні формати даних табличного процесора.
5. Форматування символічних даних у комірках.
6. Створення макросів.

5.1 Поняття про електронну таблицю та табличний процесор

Досить часто інформацію, що обробляється, треба представляти у вигляді таблиць. При цьому частина комірок таблиці містить вихідну або **первинну** інформацію, а частина – **похідну**. Така інформація є результатом різних арифметичних та інших операцій, що здійснюються над первинними даними.

Для вирішення завдань, які можна подати у вигляді таблиць, розроблено спеціальні пакети програм, що називаються **електронними таблицями** або **табличними процесорами**.

Електронна таблиця – це комп'ютерний еквівалент звичайної таблиці, що складається із рядків і стовпців, на перетині яких розташовуються комірки, в яких міститься числова інформація, формули, текст. Рядкам і стовпцям можна надавати назви. Екран монітора трактується як вікно, через яке можна розглядати таблицю цілком і частинами. Електронна таблиця – найбільш розповсюджена й потужна інформаційна технологія для професійної роботи з даними. Для керування електронною таблицею створено спеціальні програмні продукти – табличні процесори.

Табличний процесор – це комплекс взаємопов'язаних програм, які призначено для обробки електронних таблиць.

Табличні процесори являють собою зручний засіб для проведення бухгалтерських і статистичних розрахунків. У кожному пакеті є сотні вбудованих математичних функцій і алгоритмів статистичної обробки даних. Крім того є потужні засоби зв'язку таблиць між собою, створення і редагування баз даних. Табличні процесори мають вбудовану довідкову систему, що надає користувачу інформацію по конкретним командам меню та інші довідкові дані. Багатомірні таблиці дозволяють швидко робити вибірки у базі даних за будь-яким критерієм.

Сучасні табличні процесори: Microsoft Office Excel, SuperCalc (рис. 5.1), LibreOffice Calc, GNumeric (рис. 5.2) з пакета GNOME Office тощо.

The screenshot shows the main screen of SuperCalc 3 with a spreadsheet titled 'SAMPLE BUDGET'. The spreadsheet has columns A, B, and C, and rows 1 through 12. The data is as follows:

DESCRIPTION	AMOUNT	BALANCE
STARTING BALANCE		250.00
PAYDAY	550.00	800.00
RENT	-350.00	450.00
UTILITIES	-75.00	375.00
FOOD	-200.00	175.00
AUTO	-100.00	75.00
FINAL BALANCE		75.00

At the bottom of the screen, there is a status bar with the following text: 'Width: 25 Memory:464 Last Col/Row:C12 ? for HELP' and a keyboard shortcut legend: 'F1 = Help; F2 = Erase Line/Return to Spreadsheet; F9 = Plot; F10 = View'.

Рисунок 5.1 – Основний екран SuperCalc 3

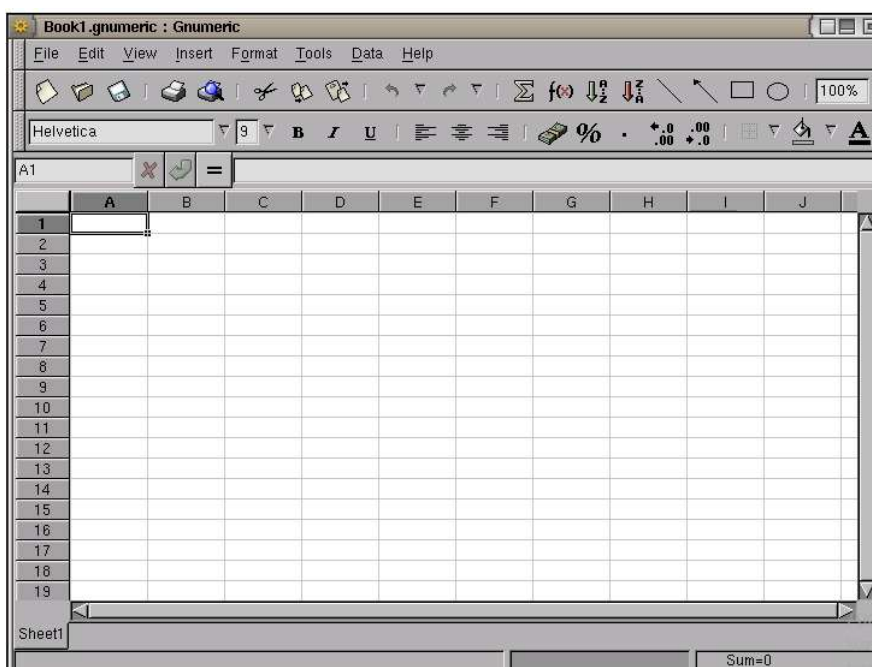


Рисунок 5.2 – Початковий екран GNumeric

Функції табличних процесорів:

- створення та редагування електронних таблиць;
- створення документів з великою кількістю таблиць, в тому числі таких, що об'єднано формулами;
- оформлення та друк електронних таблиць;
- побудова діаграм, їх модифікація;
- розв'язання економічних задач графічними методами;
- робота з електронними таблицями як з базами даних: сортування таблиць, вибірка даних за запитом;
- створення підсумкових і зведених таблиць;

- використання при побудові таблиць інформації із зовнішніх баз даних;
- створення слайд-шоу;
- розв'язання задач оптимізації;
- розв'язання економічних задач типу «що – якщо» методом підбору параметрів;
- розробка макрокоманд, налаштування середовища під потреби користувача.

Спеціальні засоби дозволяють автоматично отримувати і роздруковувати звіти з використанням десятків різних типів таблиць, графіків, діаграм, додавати до них коментарі та графічні ілюстрації.

5.2 Загальна технологія роботи з електронними таблицями

Табличний процесор може мати кілька режимів роботи, найбільш важливі з них: режим готовності, режим введення даних, командний режим, режим редагування.

Режим готовності – режим, у якому відбувається вибір комірки або блоку для корегування чи виконання якоїсь операції. У цьому режимі текстового курсора немає, а є виділення активної комірки кольором.

Режим введення даних – режим, коли починають вводити дані у певну комірку, з якого після введення даних знову переходять до режиму готовності або до командного режиму.

Командний режим – при переході до цього режиму (за допомогою клавіші F10) можна обрати і виконати певну команду (пункт) головного меню. Після виконання команди (збереження, друк чи ін.) відбувається повернення до режиму готовності.

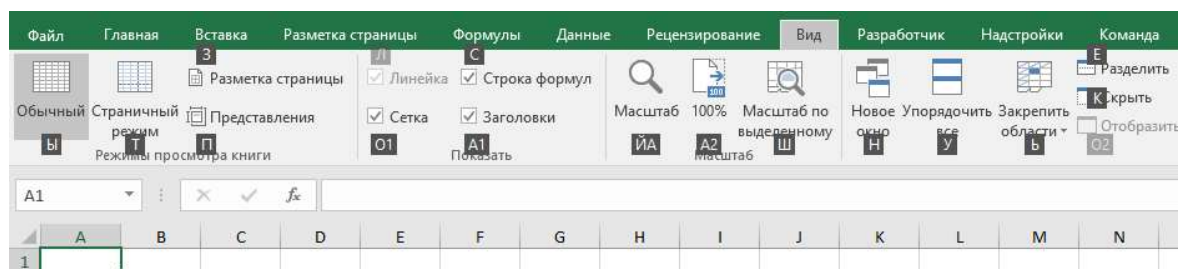


Рисунок 5.3 – Командний режим MS Excel

Режим редагування. При переході у режим введення даних попередній вміст поточної комірки втрачається. Щоб цього не відбувалося використовується режим редагування, що ініціюється певною комбінацією клавіш. У режимі редагування вміст активної комірки з'являється на контрольній панелі, стає доступним для внесення змін.

Табличний процесор має велику кількість команд, кожна з яких має різноманітні параметри (опції). Команди разом із додатковими опціями утворюють систему ієрархічного меню, яка у кожного типу табличних процесорів має свої особливості. На верхньому рівні знаходиться головне меню.

Вибір команди з меню робиться одним з двох способів: за допомогою клавіш керування курсором з натисненням клавіші введення або введенням з клавіатури спеціально виділеного символу команди.

Додаткову інформацію про команди, що складають меню електронної таблиці, і їх використання можна отримати, викликавши довідку.

Сукупність команд, що надається у розпорядження користувача деякої середньої електронної таблиці, можна розбити на такі **типові групи**:

- команди для роботи з файлами (збереження, завантаження, відкриття, пошук);
- команди редагування (переміщення, копіювання, видалення, вставка, пошук та заміна, відміна);
- команди форматування (вирівнювання тексту, вибору шрифтів, товщини та розташування ліній, висоти рядка, ширину стовпця, колір фону та ін.);
- команди для роботи з вікнами (відкриття, закриття, вибір режиму, перехід між відкритими вікнами та ін.);
- команди для роботи з таблицею як з базою даних (сортування, пошук потрібного рядка, стовпця чи комірки та ін.);
- друк;
- сервіс (команди для створення макросів, об'єднання електронних таблиць чи їх частин, встановлення захисту, експорту та імпорту, підключення додаткових математичних інструментів та ін.);
- отримання довідкової інформації;
- використання графіки.

Робота у кожній **електронній таблиці** має свої особливості, проте можна говорити про деяку **узагальнену технологію** роботи з нею.

1. На першому етапі формується структура таблиці, яка включає: визначення заголовку, назв рядків і стовпців, а також введення у комірки вихідних даних, формул і функцій.

2. На другому етапі відбувається робота з даними, яка має на увазі дослідження сформованої таблиці, яке пов'язане з використанням певних математичних моделей (моделюванням), методів одночасної роботи з кількома таблицями і методів роботи з базами даних.

3. Третій етап технології дозволяє у графічному виді представити результати, які отримано на першому і другому етапах, і найбільш яскраво їх інтерпретувати.

4. Четвертий етап забезпечує вивід результуючих даних на друк. При цьому результати можна роздрукувати у табличному вигляді або у виді графічних діаграм.

Етапи формування структури електронної таблиці представлено на рисунку 5.4.

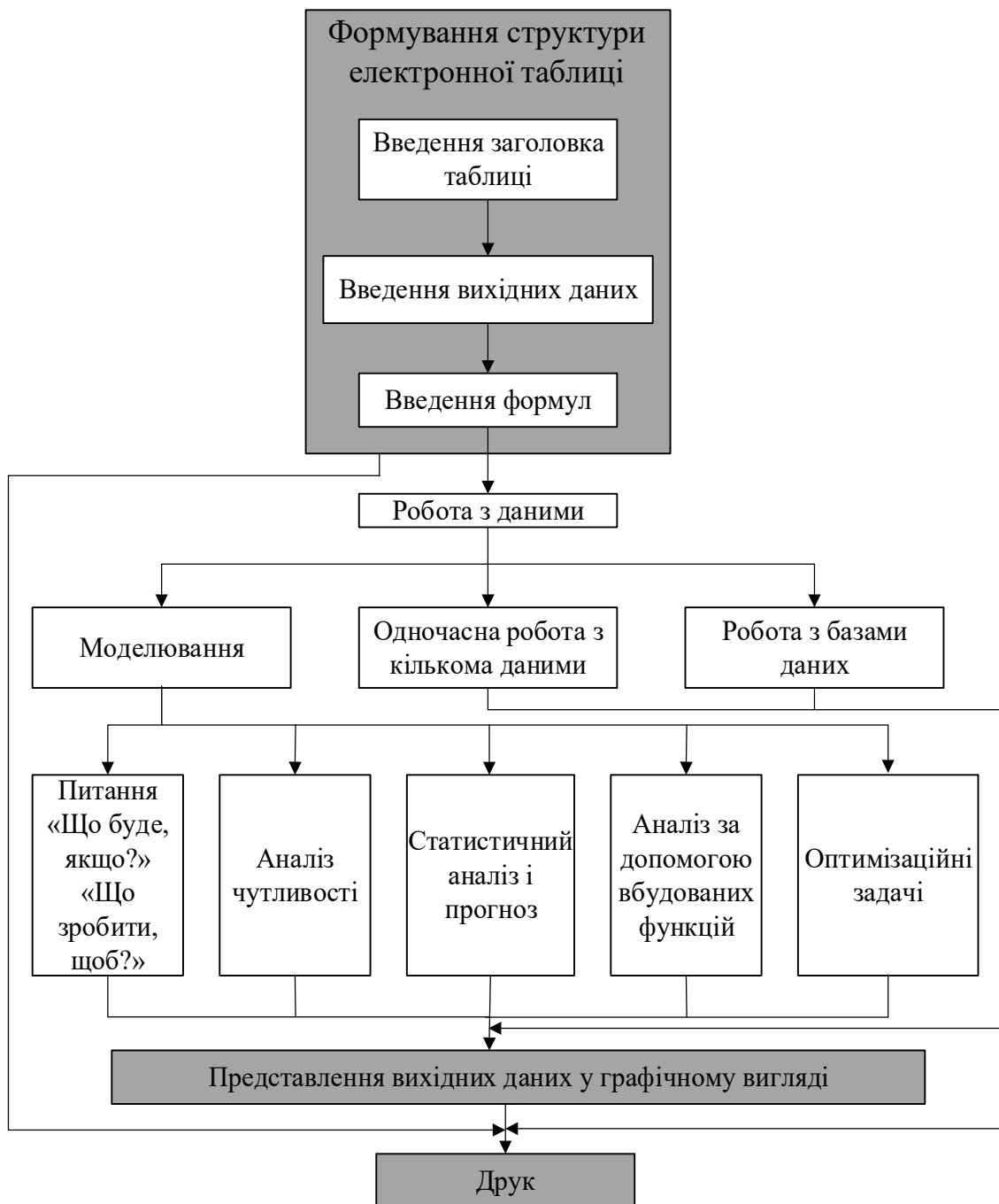


Рисунок 5.4 – Формування структури електронної таблиці

5.3 Типова структура вікна електронної таблиці

Усі відомі табличні процесори мають типову структуру інтерфейсу (рис. 5.5). При роботі з електронною таблицею на екран виводяться робоче поле таблиці і панель керування, яка містить у собі: головне меню, допоміжну область керування, рядок введення, рядок підказки. Розташування цих областей на екрані може бути мимовільним і залежить від особливостей конкретного табличного процесора.



Рисунок 5.5 – Типовий інтерфейс електронної таблиці

Рядок головного меню містить імена меню основних режимів програми. Обравши один із них, користувач отримує доступ до меню, що містить перелік команд, які в нього входять. Після вибору деяких команд меню з'являються додаткові підменю.

Додаткова область керування включає у себе рядок стану, панелі інструментів, вертикальну і горизонтальну лінійки прокрутки.

У **рядку стану** (статусному рядку) користувач знайде відомості про поточний режим роботи програми, імені файлу поточної електронної таблиці, номер поточного вікна та ін.

Панель інструментів (піктографічне меню) містить певну кількість кнопок (піктограм), які призначено для швидкої активізації виконання певних команд меню і функцій програми. Щоб викликати на екран ті області таблиці, які на ньому на даний момент не відображено, використовуються **вертикальна і горизонтальна лінійки прокрутки**. Бігунки (движки) лінійок показують відносну позицію активної комірки у таблиці і використовуються для швидкого переміщення по ній. У деяких табличних процесорах на екрані створюються спеціальні зони швидкого виклику. При натисканні миші у такій зоні викликається відповідна функція. Наприклад, при натисканні мишею на координатній лінійці викликається діалог параметрів сторінки.

Рядок введення відображає дані, що вводяться у комірку. У ній користувач може переглядати чи редагувати вміст поточної комірки. Особливість рядку введення – можливість бачити формулу чи функцію, що міститься у комірці, а не її результат. Рядок введення зручно використовувати для перегляду чи редагування текстових даних.

Рядок підказки призначено для надавання повідомлень користувачу відносно його можливих дій у даний момент.

Робоче поле – простір електронної таблиці, що складається з комірок, назв стовпців і рядків.

Панель керування – частина екрана, що дає користувачу інформацію про активну комірку і її вміст, меню і режими роботи.

Електронні таблиці надають користувачу комп'ютерний варіант так званої робочої книги для проведення розрахунків. Книга складається з окремих поіменованих **аркушів** (таблиць), які у свою чергу формуються з комірок, які виходять з перетину рядків та стовпців. Стовпці таблиці нумеруються прописними латинськими літерами (A, B, ..., Z, AA, ..., AZ, ...), а рядки – числами. Кожна комірка має унікальну адресу, що складається з імені стовпця і рядка, наприклад B2 (рис. 5.6).

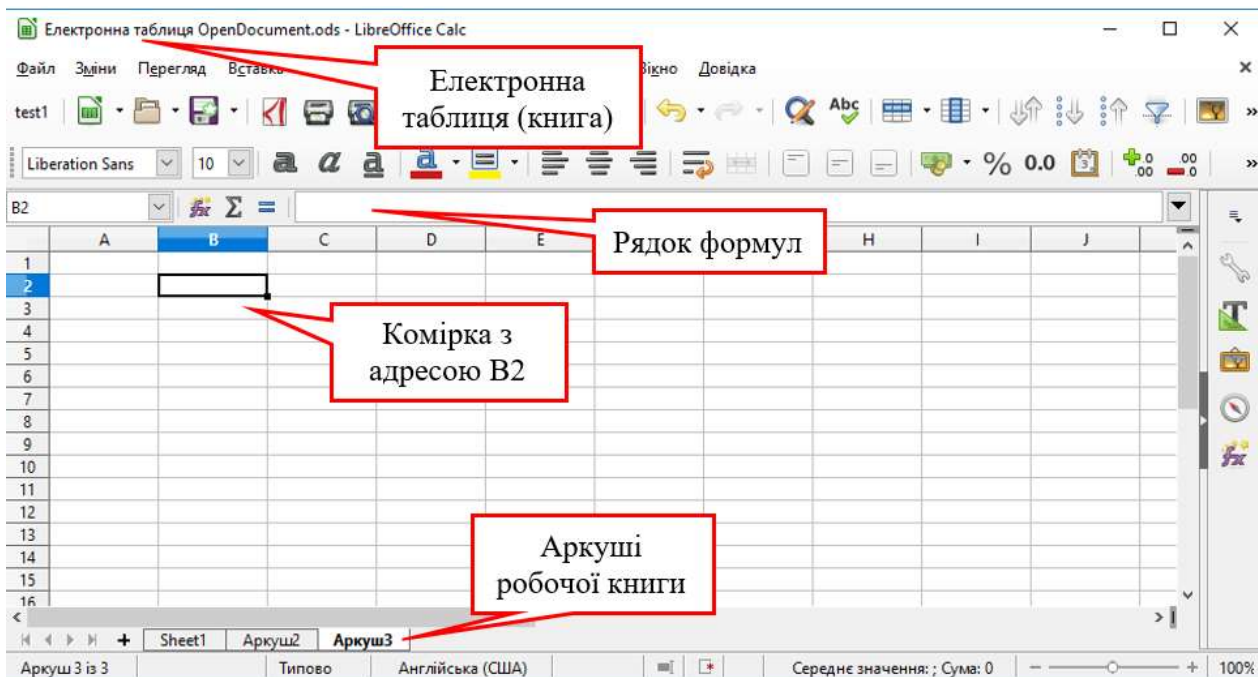
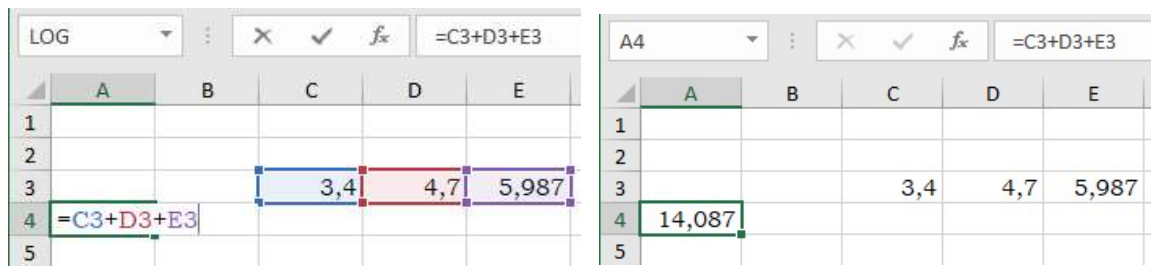


Рисунок 5.6 – Структура електронної таблиці LibreOffice Calc

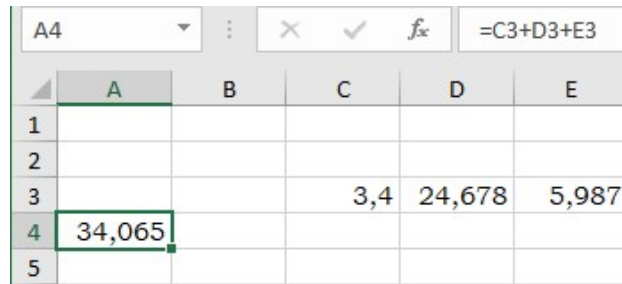
Перед тим, як помістити дані в ту чи іншу комірку таблиці, її слід виділити (натиском лівої кнопки миші). Введення даних можливе як в саму комірку, так і у поле, що відображає вміст комірки (рядок формул). Виділити групу комірок можна за допомогою миші або клавіатури. Окремі групи комірок можна формувати, встановлювати різні шрифти, їх накреслення і розмір та ін.

Комірки можуть містити текстову або числову інформацію, а також формулу для розрахунків. Запис формули починається із символу «=».

Як правило, формули, що використовуються в електронних таблицях, містять посилання на інформацію, що розміщена в інших комірках. Наприклад, якщо комірка містить формулу $C3+D3+E3$, то у ній буде відображатися сума значень, розташованих у комірках C3, D3, E3, при чому при зміні даних у цих комірках сума автоматично перераховується (рис. 5.7).



а) введення формули для обчислення суми значень



б) автоматичний перерахунок суми

Рисунок 5.7 – Формула суми значень у комірці А4

Електронні таблиці дозволяють використовувати більшість математичних, статистичних і фінансових функцій. Копіювання формул з однієї комірки в інші за допомогою миші дозволяє значно прискорити процес підготовки таблиці, що містить однотипні формули.

При вказуванні адреси тої чи іншої комірки використовують або **відносну**, або **абсолютну** адресацію. Якщо у формулі задано **відносну** адресу комірки, то при копіюванні формулу буде автоматично змінено, при цьому адреси комірок замінюються на нові (рис. 5.8).

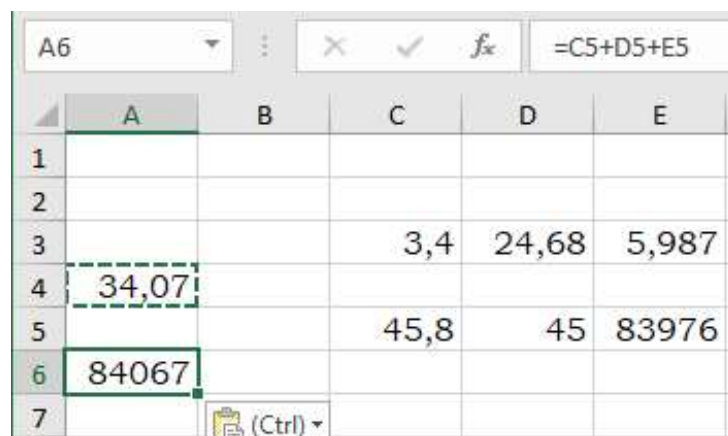


Рисунок 5.8 – Формулу з комірки А4 скопійовано в А6

При **абсолютній** адресації відповідну частину адреси буде «заморожено», тобто не буде змінено при копіюванні. Ознакою **абсолютної** адресації є символ \$ перед номером стовпця чи (і) рядка (рис. 5.9).

A6		= \$C5+D5+E5			
A	B	C	D	E	
1					
2					
3		3,4	24,68	5,987	
4	34,07				
5		45,8	45	83976	
6	84067				
7					

A6		= C5+D\$3+E5			
A	B	C	D	E	
1					
2					
3		3,4	24,68	5,987	
4	34,07				
5		45,8	45	83976	
6	84046				
7					

а) фіксація стовпчика С

б) фіксація третього рядка у стовпчику D

A6		= C5+D5+\$E\$3			
A	B	C	D	E	
1					
2					
3		3,4	24,68	5,987	
4	34,07				
5		45,8	45	83976	
6	96,79				
7					

в) абсолютна фіксація комірки E3

Рисунок 5.9 – Абсолютна адресація комірок

Електронні таблиці дозволяють перейменовувати, додавати і видаляти аркуші у робочій книзі.

5.4 Основні формати даних табличного процесора

Форматування даних – вибір форми представлення числових або символічних даних у комірці.

В електронних таблицях можна використовувати різні формати представлення числових даних у рамках однієї і тієї ж електронної таблиці. Найбільш розповсюджені **формати представлення числових даних**:

1) **основний (числовий)** формат використовується за замовчуванням, забезпечуючи запис числових даних у комірках у тому ж вигляді, в якому їх було введено або обчислено;

2) **формат із фіксованою кількістю десяткових знаків**. Наприклад, якщо встановлено режим форматування, що містить два десяткових знаки, то число 12345 буде відображено у комірці як 12345,00, а число 0,12345 – як 0,12;

3) **відсотковий формат** забезпечує представлення даних у форматі відсотків із знаком %. Наприклад, на екрані число 0,123 буде відображено як 12,3%, а 123 – як 12300,0%;

4) **грошовий формат** забезпечує таке представлення чисел, де кожні три розряди розділено комою або пробілом (рис. 5.10);

	A	B
1		
2	12 346 789,00 грн.	
3		

Рисунок 5.10 – Грошовий формат даних у комірці

5) **науковий формат**, що використовується для представлення великих чи малих величин, забезпечує представлення чисел, що вводяться у вигляді двох компонентів: мантиси, що має один десятковий розряд зліва від десяткової крапки, та деякої кількості (задається користувачем) десяткових знаків справа від неї; порядку числа. Наприклад, число 12345 буде записано в комірці як 1,2345E+04 (якщо встановлено точність 4 розряди) і як 1,234E+04 (при точності у 2 розряди), що показано на рисунку . Число 0,0000012 у науковому форматі буде мати вигляд 1,2E-06 (рис. 5.11).

	A	B
1		
2	1,23E+04	
3		

Рисунок 5.11 – Науковий формат даних у комірці

5.5 Форматування символічних даних у комірках

За замовчуванням символічні дані вирівнюються по лівому краю комірки. Можна змінити формат представлення символічних даних в електронній таблиці. Для цього існують такі можливості: вирівнювання по лівому краю комірки, вирівнювання по правому краю комірки, вирівнювання по центру комірки.

Формула являє собою сукупність математичних операторів, чисел, посилань і функцій. При обчисленнях за допомогою формул дотримуються прийнятого у математиці порядку виконання арифметичних операцій.

Формули складаються з операторів і операндів, що розташовано у певному порядку. Як операнди використовуються дані, а також посилання комірок або блоків комірок. Оператори у формулах позначають дію, що відбувається з операндами. Залежно від операторів, що використовуються, розрізняють **арифметичні** (алгебраїчні) (рис. 5.12) чи **логічні** формули.

В **арифметичних** формулах використовуються такі оператори арифметичних дій:

- + додавання;
- віднімання;
- * множення;
- / ділення;
- ^ піднесення до степеня.

B5		: X ✓ fx		=MDETERM(A1:C3)+SUM(A1:C1)+SUM(B1:B3)				
	A	B	C	D	E	F	G	H
1	0,78	0,95	1,29					
2	12,9	0,4	0,24					
3	1,7	1,9	1,6					
4								
5	формула	17,93382						
6								

Рисунок 5.12 – Арифметична формула у комірці B5

Кожна формула в електронній таблиці містить кілька арифметичних дій з її компонентами. Нормальний порядок виконання операцій змінюють введенням дужок. Операції у дужках виконуються першими. Арифметичні формули можуть також містити оператори порівняння: дорівнює (=), не дорівнює (<>), більше (>), менше (<), не більше (<=), не менше (>=). Результатом обчислення арифметичної формули є число.

Логічні формули можуть містити вказані оператори порівняння, а також спеціальні логічні оператори (рис. 5.13):

NOT – логічне заперечування «НІ»;

AND – логічне «І»;

OR – логічне «АБО»

Логічні формули визначають, вираз «істина» чи «помилка».

A3		: X ✓ fx		=AND(A1=B1; C1=A1)			
	A	B	C	D	E	F	
1	1,3	2,256	2,256				
2							
3	FALSE						
4							

Рисунок 5.13 – Логічна формула

Функції вводять у таблицю у склад формул або окремо. В електронних таблицях можуть бути представлені такі **види функцій**:

1) математичні функції виконують різні математичні операції, наприклад, обчислення логарифмів, тригонометричних функцій та ін.;

2) статистичні функції виконують операції з обчислення параметрів випадкових величин або їх розподілень, які представлено множиною чисел, наприклад, стандартного відхилення, середнього значення, медіани та ін.;

3) текстові функції виконують операції з текстовими рядками або послідовністю символів, обчислюючи довжину рядка, змінюють регістри та ін.;

4) логічні функції використовуються для побудови логічних виразів, результат яких залежить від того, чи є істинною умова, що перевіряється;

5) фінансові функції використовуються у складних фінансових розрахунках;

6) функції дати і часу.

Всі функції мають однаковий формат запису і включають ім'я функції та список аргументів, що розділено крапкою з комою.

Увага! Знак «;» – роздільник аргументів функції, «:» – діапазону аргументів функції.

Приклади функцій:

SUM (список аргументів) – статистична функція, що обчислює суму всіх числових значень у списку, який може складатися з адрес комірок і блоків, а також числових значень (рис. 5.14).

	A	B	C	D
1	0,25	0,456	1,5	
2	4,5	5,9	4,7	
3	0,7	0,6	0,45	
4				
5	2,206			
6				

Рисунок 5.14 – Сумування даних з комірок A1, B1, C1

MDETERM (масив) – функція для роботи з масивами, що повертає визначник матриці (рис. 5.15).

	A	B	C	D
1	0,25	0,456	1,5	
2	4,5	5,9	4,7	
3	0,7	0,6	0,45	
4				
5	-1,60941			
6				

Рисунок 5.15 – Обчислення визначника матриці

Приклад 5.1 Розв’язати систему лінійних алгебраїчних рівнянь методом Крамера.

Система рівнянь має такий вигляд:

$$\begin{cases} 0,4x_1 + 0,2x_2 + 0,1x_3 + 0,1x_4 = 0,5 \\ 0,1x_1 + 15x_2 + 0,3x_3 + 0,2x_4 = 0,04 \\ 0,3x_1 + 0,1x_2 + 0,2x_3 + 0,1x_4 = 0,8 \\ 0,1x_1 + 0,1x_2 + 0,1x_3 + 0,3x_4 = 0,15 \end{cases}$$

На рисунку 5.16 показано введення коефіцієнтів системи.

	A	B	C	D	E	F	G	H
1	Matrix of system	0,40	0,20	0,10	0,10		free koef	0,50
2		0,10	15,00	0,30	0,20			0,04
3		0,30	0,10	0,20	0,10			0,80
4		0,10	0,10	0,10	0,30			0,15

Рисунок 5.16 – Введення початкових даних

Далі формуємо матрицю A1: у матриці A замінюємо перший стовпчик стовпчиком вільних коефіцієнтів. Аналогічно для матриць A2-A4.

У меню функцій потрібно обрати Масив-MDETERM (визначник матриці), виділити відповідні діапазони та знайти визначники detA1-detA4 (рис. 5.17), а потім знайти корені системи за формулами Крамера ($x_1 = \text{detA1}/\text{detA}$; $x_2 = \text{detA2}/\text{detA}$, $x_3 = \text{detA3}/\text{detA}$, $x_4 = \text{detA4}/\text{detA}$) (рис. 5.17).

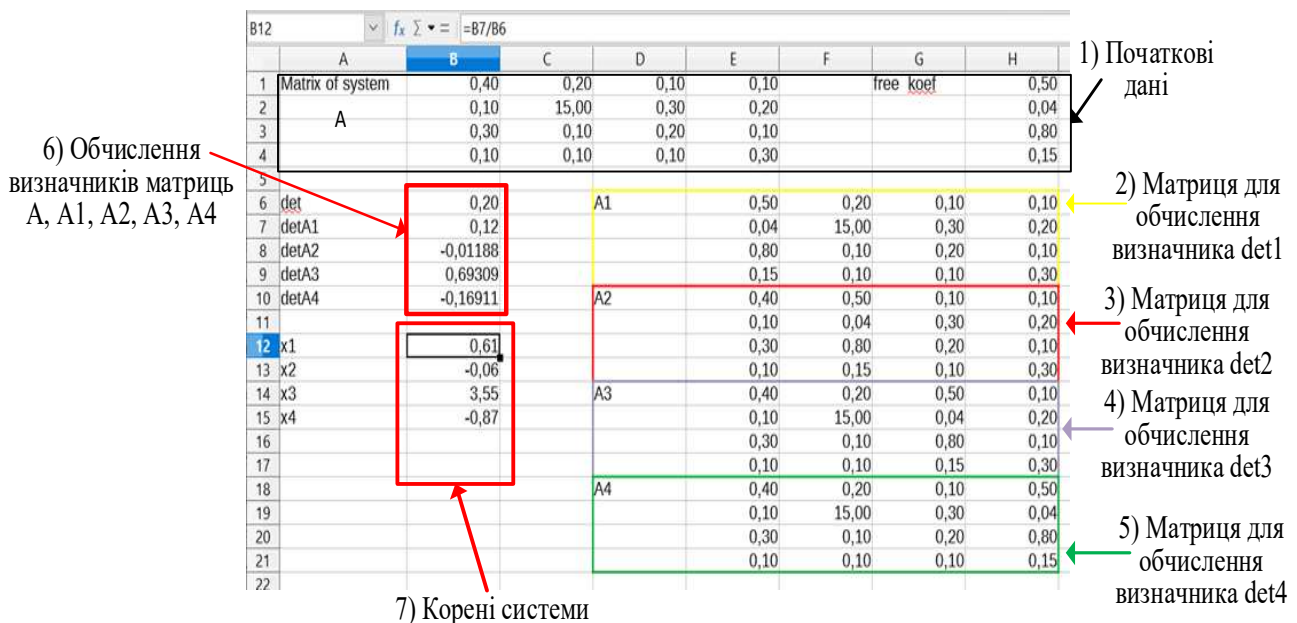


Рисунок 5.17 – Розв’язання системи СЛАР методом Крамера

5.6 Створення макросів

Макрос – це набір інструкцій, які програма виконує при команді запуску. Інструкції можуть відповідати простим натисненням клавіш чи складним наборам команд меню. Макроси зазвичай створюються при необхідності постійного виконання одного й того ж набору операцій. Вони записуються мовою програмування **VBA (Visual Basic for Applications)** для додатків.

Visual Basic for Applications (VBA) – це візуальна об’єктно-орієнтована мова програмування високого рівня, який вбудовано в офісні пакети.

Виділяють три основні види макросів:

1) **командні макроси**, які складаються з операторів, які відповідають тим чи іншим командам або параграфам діалогових вікон, які записано у визначеному порядку. Основне призначення таких макросів – зміна зовнішнього вигляду вікна або об’єкта;

2) **функції користувача**, які працюють аналогічним чином як і вбудовані функції табличного процесора, на відміну від командних макросів вони не змінюють середовище програми;

3) **макрофункції**, які є поєднанням командних макросів і користувацьких функцій.

Створити макроси можна такими способами:

1) Запустити вбудований засіб для автоматичного запису макросів – макрорекордер і виконати послідовність дій, які необхідно включити у макрос. Далі зупинити макрорекордер. Зазвичай це стосується простих макросів.

2) Ввести коди макроса та оператори VBA у вікні редактора Visual Basic.

При створенні складних макросів об'єднують ці два способи в один.

Макроси можна зберігати в:

1) **Особиста книга макросів** – це книга, яка автоматично завантажується при запуску табличного процесора, вона зазвичай є невидимою, але макроси з неї можна викликати у будь-якій робочій книзі, такі макроси називаються **глобальними**.

2) У поточному документі – **локальні**.

У VBA існують два види процедур: **підпрограми та функції**. Підпрограми можна віднести до таких категорій: ті, які можуть бути макросами, й ті, які не можуть ними бути.

Об'єкти, з якими працюють процедури і функції VBA, представляють собою засіб програмного керування програмами Office і створеними за їх допомогою документами. Найчастіше використовуються: робочі масиви, меню, діапазон комірок.

Об'єкти VBA мають властивості, які можуть приймати різні значення. Фактично властивостями є атрибути об'єктів. Об'єкт VBA вказан за допомогою посилання, яка може складатися з декількох компонентів, розділених точкою.

При роботі з комірками використовуються такі об'єкти:

1) **Range (“A1”)**

2) **Cells (M,N)** – для комірок, які змінюються у процесі роботи програми (M – стовпчик, N – рядок).

Row – рядок.

Column – стовпчик.

ActiveCell – активний рядок.

Приклад 5.2 Row(3).Select – виділиться третій рядок.

Корисні посилання

1. Табличний процесор Calc. URL : <https://libreoffice.readthedocs.io/ru/latest/calc.html>

2. Робота в Excel з таблицями. URL : <https://exceltable.com/sozdat-tablicu/kak-rabotat-v-excel-s-tablicami>

Тестові завдання

1. Адреса комірки – це:

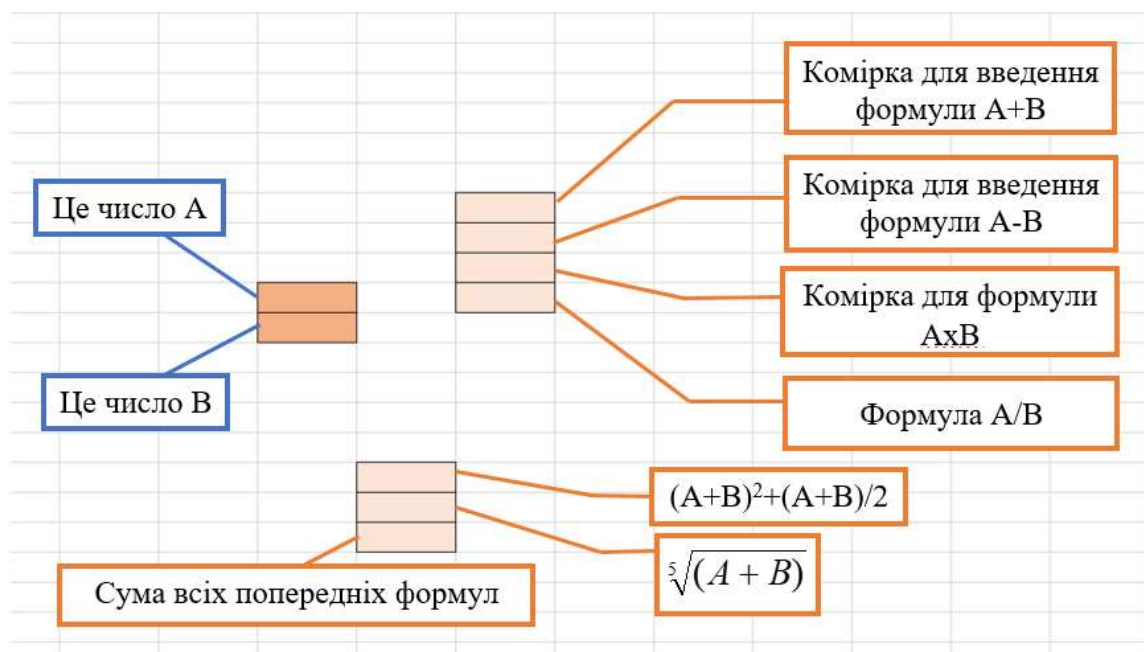
- а) позначення комірки по вертикалі й по горизонталі;
- б) кількість символів, що міститься у ній;
- в) специфічне форматування комірки;
- г) немає такого поняття.

2. Ознакою абсолютної адресації комірки є знак:
 - а) &;
 - б) *;
 - в) \$;
 - г) #.
3. З якого знаку починається введення формули до комірки?
 - а) =;
 - б) %;
 - в) *;
 - г) (.
4. Як зміниться формула $A8=\$C3+D\$5+E3$, якщо її скопіювати до комірки A10?
 - а) $A10=\$C5+D\$5+E5$;
 - б) $A10=\$C7+D\$7+F5$;
 - в) $A10=\$D5+D\$5+\$F5$;
 - г) $A10=\$E5+F\$5+G5$.
5. Який з табличних процесорів є вільно розповсюджуваним?
 - а) MS Excel;
 - б) LibreOffice Calc;
 - в) OpenOffice Writer;
 - г) Accent 2.
6. За допомогою якої функції можна обчислити визначник матриці?
 - а) MINVERSE;
 - б) MDETERM;
 - в) AVERAGE;
 - г) SUM.
7. Яким чином можна змінити порядок виконання арифметичних дій у формулі?
 - а) за допомогою введення дужок;
 - б) за допомогою відносної адресації;
 - в) за допомогою абсолютної адресації;
 - г) це неможливо.
8. Яким чином перейти у командний режим роботи табличного процесора?
 - а) натиснути F10;
 - б) натиснути ESC;
 - в) натиснути Alt+Ctrl+Del;
 - г) натиснути Alt+Shift.
9. Файл таблиці Microsoft Excel має формат:
 - а) *.EPUB;
 - б) *.DOC;
 - в) *.PPTX;
 - г) *.XLS.
10. Редактор VBA викликається такою комбінацією клавіш:
 - а) Alt+F8;

- б) Alt+Shift;
- в) F10;
- г) ESC.

Практичні завдання

1. Створіть табличний документ. У довільні комірки введіть два будь-яких числа (із дробовою частиною). Здійсніть з ними дії, які показано на рисунку.



2. Розв'яжіть систему лінійних алгебраїчних рівнянь з прикладу 5.1 за допомогою оберненої матриці.

3. За допомогою інструмента *Підбір параметра* розв'яжіть такі рівняння:

- а. $\sqrt[4]{10 + 8\sin^2 x} - \sqrt[4]{8\cos^2 x - 1} = 1$;
- б. $|\log_2(3x - 1) - \log_2 x| = |\log_2(502x) - 1|$;
- в. $8,4 \sqrt[12]{x^{-7}} - 0,2 \sqrt[4]{x^{-1}} \sqrt[3]{x^2} = \sqrt[12]{x^{11}}$.

4. Використовуючи меню *Діаграми* побудуйте графіки таких функцій:

- а. $y = |x^2 + 5|x| + 6|$;
- б. $y = 2^x \cdot 2^{|x|}$;
- в. $y = 2^{1/x}$.

Інтервал для побудови оберіть самостійно. Кількість точок – 30.

5. Знайти максимум функції F при заданій системі обмежень:

$$а. F = 2x_1 + 3x_2 \begin{cases} x_1 + 3x_2 \leq 9 \\ x_1 + x_2 \leq 7 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases};$$

$$\begin{array}{l}
 \text{б. } F = 10x_1 + 3x_2 \\
 \text{в. } F = 25x_1 + 13x_2
 \end{array}
 \left\{ \begin{array}{l}
 2x_1 + 3x_2 \leq 33 \\
 x_1 - 2x_2 \leq 6 \\
 x_1 \geq 0 \\
 x_2 \geq 0 \\
 x_1 + 3x_2 \leq 9 \\
 2x_1 + x_2 \leq 7 \\
 x_1 \geq 0 \\
 x_2 \geq 0
 \end{array} \right. ;$$

Контрольні запитання

1. Що таке електронна таблиця?
2. Які функції і призначення електронних таблиць?
3. Які є спеціальні елементи вікна Excel? Яке їх призначення?
4. Що називається робочим полем електронної таблиці?
5. Які режими роботи може мати табличний процесор?
6. Які формати даних існують у LibreOffice Calc?
7. Як вводиться формула в електронній книзі?
8. Що таке макрос?
9. Як створити макрос за допомогою макрорекордера?
10. Як відкрити редактор макросів?

Тема 6. Графічний інтерфейс операційних систем

Мета: вивчити поняття про графічний інтерфейс, ознайомитись з основними елементами графічного інтерфейсу.

План

1. Визначення графічного інтерфейсу операційних систем.
2. Види та загальна характеристика інтерфейсів операційних систем.
3. Основні поняття графічного інтерфейсу.
4. Основні елементи вікна графічної операційної системи та операції з ним. Типи вікон.
5. Панель задач та головне меню. Довідкова система.
6. Налаштування робочого середовища графічної операційної системи.

6.1 Визначення графічного інтерфейсу операційних систем

На ранніх етапах розвитку обчислювальної техніки користувальницький інтерфейс розглядався як засіб спілкування людини з операційною системою та був досить примітивним. В основному він дозволяв запустити завдання на виконання, зв'язати з ним конкретні дані та виконати деякі процедури обслуговування обчислювальної установки.

Згодом у міру вдосконалювання апаратних засобів з'явилася можливість створення інтерактивного програмного забезпечення, що використовує спеціальні користувальницькі інтерфейси. У цей час основною проблемою є розробка інтерактивних інтерфейсів до складних програмних продуктів, розрахованих на використання непрофесійними користувачами. В останні роки були сформульовані основні концепції побудови таких користувальницьких інтерфейсів і запропоновано кілька методик їхнього створення.

Інтерфейс (interface) – це сукупність логічних і фізичних принципів взаємодії компонентів технічних засобів обчислювальної системи (ОС), тобто сукупність правил алгоритмів і тимчасових угод з обміну даними між компонентами ОС (**логічний інтерфейс**), а також сукупність фізичних, механічних і функціональних характеристик засобів підключення, що реалізують таку взаємодію (**фізичний інтерфейс**).

Інтерфейсом нерідко називають також технічні і програмні засоби, що реалізують сполучення між пристроями та вузлами ОС.

Інтерфейс поширюється на всі логічні і фізичні засоби взаємодії обчислювальної системи із зовнішнім середовищем, наприклад з операційною системою, з оператором та ін.

Графічний інтерфейс користувача (Graphical user interface) GUI – інтерфейс, в якому елементи інтерфейсу (меню, кнопки, значки та ін.), що бачить користувач на екрані, реалізовано у вигляді графічних зображень. На відміну від інтерфейсу командної строки у графічному інтерфейсі користувач має довільний доступ (за допомогою засобів введення інформації) до всіх видимих об'єктів

екрану (елементів інтерфейсу) і виконує керування ними. Найчастіше елементи інтерфейсу GUI реалізуються на основі метафор і відображає їх розуміння та властивості. Це полегшує опанування програм невідготовленими користувачами.

Вперше концепція графічного інтерфейсу була запропонована вченими із лабораторії Ксерокс (Xerox laboratory) у 1973 році. У 81-му році виходить пропозиція про комп'ютер STAR на основі GUI, але далі комерційну реалізацію отримала в комп'ютерах Apple. У наш час графічний інтерфейс є стандартним компонентом більшості доступних операційних систем та додатків.

6.2 Види та загальна характеристика інтерфейсів операційних систем

Інтерфейси розрізняють за такими характеристиками, як структура зв'язків, спосіб підключення і передачі даних, принципи управління і синхронізації.

Існують такі **види інтерфейсів**:

1. **Внутрішньомашинний інтерфейс** – система зв'язку і засобів сполучення вузлів та блоків ЕОМ між собою. Внутрішньомашинний інтерфейс являє собою сукупність електричних ліній зв'язку (проводів), схем сполучення з компонентами комп'ютера, протоколів (алгоритмів) передачі й перетворення сигналів.

2. **Зовнішній інтерфейс** – система зв'язку системного блоку з периферійними пристроями ЕОМ чи з іншими ЕОМ.

3. **Інтерфейс «людино-машинний»** (інтерфейс «людина-комп'ютер» або користувальницький інтерфейс) – це спосіб, яким ви виконуєте якусь завдання за допомогою будь-яких засобів (якої-небудь програми), а саме: чинені вами дії і те, що ви отримуєте у відповідь. Інтерфейс є орієнтованим на людину, якщо він відповідає потребам людини і враховує його слабкості. Машинна частина інтерфейсу – частина інтерфейсу, реалізована в машині (апаратно-програмної її частини) з використанням можливостей обчислювальної техніки. Людська частина інтерфейсу – це частина інтерфейсу, реалізована людиною з урахуванням його можливостей, слабкостей, звичок, здібності до навчання та інших факторів.

Можна виділити такі **види користувацького графічного інтерфейсу**:

1. Простий (текстовий) в типовій екранній формі і стандартні елементи інтерфейсу забезпечуються самою підсистемою графічного інтерфейсу (рис. 6.1).

```
C:\>dir /x
Volume in drive C is Windows
Volume Serial Number is 40A6-7537

Directory of C:\

30.08.2011  11:44                0      AUTOEXEC.BAT
30.08.2011  11:44                0      CONFIG.SYS
30.08.2011  11:49        <DIR>          DOCUME~1    Documents and Settings
02.09.2011  15:47        <DIR>          DOWNLO~1    Downloads
12.09.2011  11:02        <DIR>          PROGRA~1    Program Files
05.09.2011  08:44        <DIR>          WINDOWS
                2 File(s)                0 bytes
                4 Dir(s)  22.237.544.448 bytes free
```

Рисунок 6.1 – Типовий вигляд екрана в операційній системі MS DOS.

2. Істинно графічний (рис. 6.2) або двовимірний, елементи якого (меню, кнопки, переліки та ін.) представлено на екрані у вигляді графічних об'єктів. Користувач має вільний доступ до будь-якого об'єкта на екрані (елементи інтерфейсу) та може безпосередньо керувати ними. Звичайно, що користувачеві зручніше працювати з графікою.

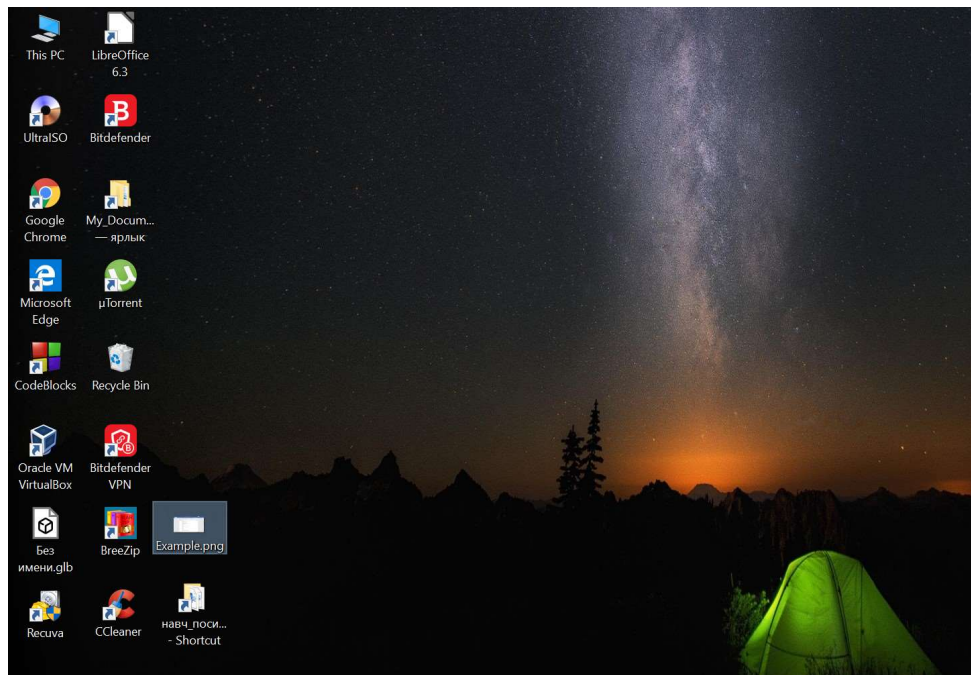


Рисунок 6.2 – Двовимірний графічний інтерфейс

3. Тривимірний.

Однією з вимог до хорошого графічного інтерфейсу програмної системи є концепція «роби те, що я маю на увазі» або **DWIM** (англ. Do What I Mean). DWIM вимагає, щоб система працювала передбачувано, щоб користувач заздалегідь інтуїтивно розумів, яку дію виконає програма після отримання його команди.

Окремою підсистемою ОС є *графічна оболонка*, що реалізує інтерфейс користувача. Наразі існує багато графічних оболонок для операційних систем, що мають дуже схожі можливості, а іноді навіть і вигляд, тобто пересічний користувач не завжди може зорієнтуватися в якій системі працює.

В ОС Windows існує одна графічна оболонка (тобто ключові елементи графічного інтерфейсу та їх реалізація залишаються постійними), але користувач може змінювати теми оформлення або налаштування деяких елементів. Натомість, ОС Linux підтримує різні графічні середовища, деякі з них вже включено до дистрибутиву, а деякі – можливо додатково інстальювати. Також існує варіант одночасного існування декількох середовищ у межах однієї системи, при цьому вибір середовища здійснюється на етапі входу до системи. Найбільш поширеними графічними середовищами для ОС Linux є конкуруючі KDE та GNOME (рис. 6.3).



Рисунок 6.3 – Графічні середовища GNOME та KDE відповідно

Окремо слід відзначити графічну оболонку Unity (підтримку та розробку якої веде компанія Canonical), що є похідною від середовища GNOME, яку було розроблено спеціально для дистрибутива Linux Ubuntu, яка є середовищем за замовчуванням.

6.3 Основні поняття графічного інтерфейсу

Основними поняттями графічного інтерфейсу Windows є: піктограма (іконка), ярлик, графічний курсор, меню, вікно.

У графічному інтерфейсі Windows кожний об'єкт відображається на екрані за допомогою деякого значка (рис. 6.5), який містить зображення (піктограма або іконка) та тексту (назви цього об'єкту). Об'єкти одного типу мають однакові піктограми, але різні назви. Базовим об'єктом у Windows є робочий стіл, який відображається одразу після завантаження операційної системи та містить значки інших об'єктів.



Рисунок 6.5 – Елементи робочого стола

Для швидкого доступу до дисків, принтера, часто використовуваних документів доцільно створити на робочому столі **ярлики** (значки з маленькими стрілками у нижньому лівому куті) – файли зв'язку, які служать покажчиком на об'єкт (наприклад, файл, який потрібно певним чином обробити), програму або команду.

Ярлик відрізняється від значка тим, що позначає об'єкт, фактично розташований не на Робочому столі, а в деякій іншій папці. Стрілка означає, що ми маємо не сам об'єкт, а посилання на нього. Ярлики створюються перетягуванням значків об'єктів на *Робочий стіл*.

Для роботи з об'єктами необхідно насамперед вказати на нього. Для цього використовується графічний курсор, керування яким здійснюється за допомогою миші.

За допомогою курсора можна виконати такі операції з об'єктом:

- вказування – підвести курсор до піктограми об'єкта, при цьому може з'явитись підказка;
- вибір (підвести курсор та натиснути ліву кнопку миші);
- відкриття (підвести курсор та натиснути двічі ліву кнопку миші);
- перетягування (підвести курсор до піктограми об'єкта та при натиснутій лівій клавіші миші перетягнути його в інше місце екрана);
- виклик контекстного меню (підвести курсор до піктограми об'єкта та натиснути праву кнопку миші);
- зміна розміру.

6.4 Основні елементи вікна графічної операційної системи та операції з ним. Типи вікон

Для кожної прикладної програми, яку запущено, створюється **вікно**, що являє собою прямокутну область та містить об'єкти для керування роботою програми, а також засоби для введення інформації користувачем.

Усі вікна мають межі та заголовок. Межі вікна показують область екрана, яку займає вікно. Заголовок використовується для відображення назви вікна та переміщення вікна на екрані. Ще однією функцією заголовка є індикація активного вікна, тобто з яким безпосередньо працює користувач. Рядок заголовка активного вікна та неактивних вікон відрізняється кольором: у активного вікна яскравіший колір заголовка. Для перемикання активного вікна використовують значки вікон на панелі задач або комбінацію клавіш **Alt+Tab**.

У залежності від задачі, що виконується, **вікна поділяються на такі типи:**

Вікна повідомлень

Наприклад, при видаленні файлу з комп'ютера з'явиться таке повідомлення (рис. 6.6):

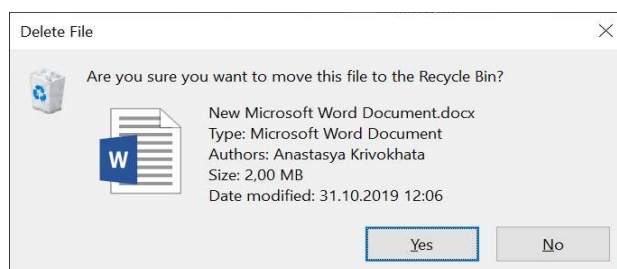


Рисунок 6.6 – Вікно повідомлення

Діалогові вікна

Зокрема, для відкриття такого вікна призначено команду **Властивості (Properties)** контекстного меню деякого об'єкта (рис. 6.6).

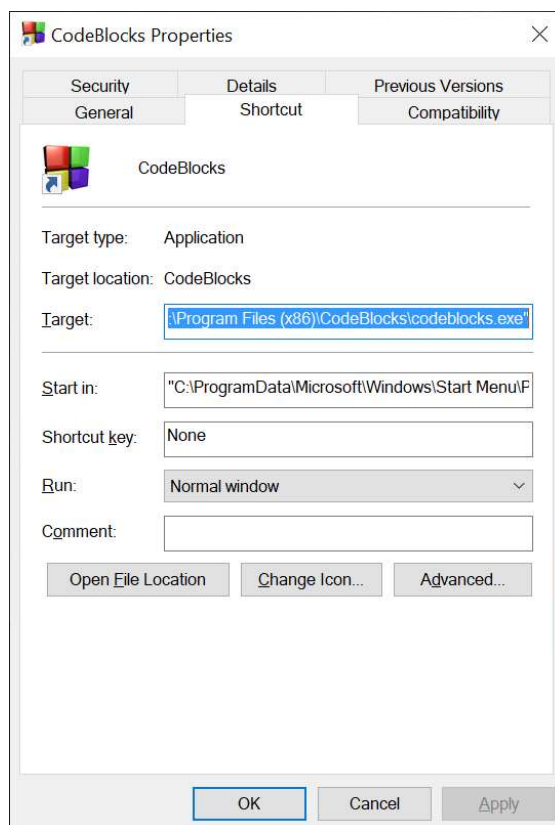


Рисунок 6.6 – Діалогове вікно

Призначення діалогових вікон полягає у тому, щоб забезпечити зручні засоби введення інформації у графічному інтерфейсі ОС. Такі вікна містять набір елементів, які дають змогу вводити інформацію. Один з елементів вікна є активним. Для зміни активного елемента використовують курсор, клавішу **Tab** або комбінацію **Shift+Tab**.

Елементи діалогових вікон такі:

1. **Вкладка** – використовується для групування елементів діалогового вікна так, щоб у разі вибору вкладки за допомогою курсора або комбінації клавіш Ctrl+Tab змінювався набір елементів діалогового вікна (рис. 6.7).

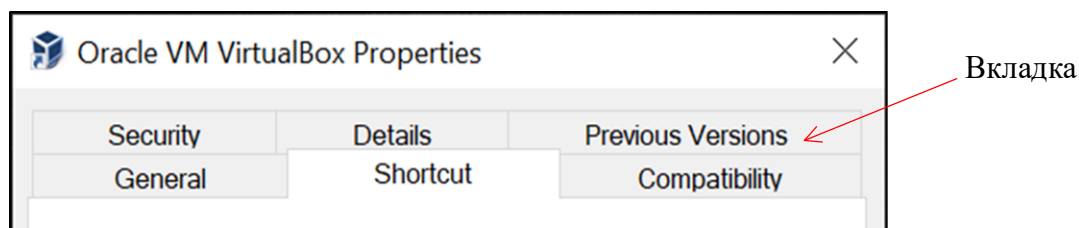


Рисунок 6.7 – Вкладки діалогового вікна Oracle VM VirtualBox Properties

2. **Кнопка** – використовується для виконання певної дії, наприклад, виклику іншого діалогового вікна (наприклад, *Change Icon* на рисунку 6.5). Для виконання дії необхідно натиснути кнопку курсором або зробити її активною та натиснути Enter.

Будь-яке діалогове вікно містить такі кнопки:

- *ОК*, що призначено для підтвердження введення інформації та закриття вікна;
- *Відмінити (Cancel)*, яку призначено для відміни введення інформації та закриття вікна.

Часто діалогові вікна містять кнопку *Застосувати (Apply)*, за допомогою якої підтверджують введення інформації без закриття вікна.

3. **Поле редагування** або **поле введення**, яке призначено для введення і редагування одного рядка символів за допомогою клавіатури. Наприклад, поле **Папка** діалогового вікна **Властивості: Робочий стіл** (рис. 6.8).



Рисунок 6.8 – Поле редагування діалогового вікна

4. **Поле зі списком** – використовується для вибору одного значення зі списку, який відображається на екрані після натискання кнопки зі стрілкою вниз праворуч від поля зі списком.

5. **Список** – вибір необхідного значення з наведеного переліку. Часто праворуч від списку розташовано смугу прокрутки, яку можна використовувати для переміщення по елементах списку. Наприклад, список *Додаткові параметри масштабування* діалогового вікна *Персоналізація* (рис. 6.9).

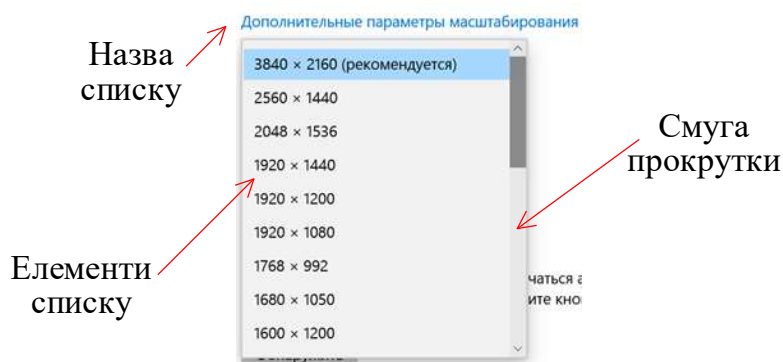


Рисунок 6.9 – Элемент *Список* діалогового вікна

6. **Вибір вмикачем (прапорцем)** – використовується для вибору кількох параметрів з переліку запропонованих. Прапорець можна встановити або зняти, вибравши його за допомогою курсора. Наприклад, на рисунку 6.10 встановлено прапорець *Закріпити панель задач* (діалогове вікно властивостей панелі задач).

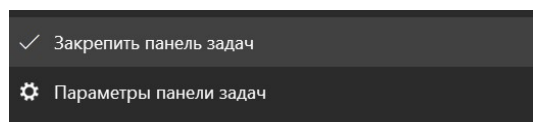


Рисунок 6.10 – Вибір прапорцем

7. **Лічильник** – являє собою поле введення, праворуч, від якого розміщені дві кнопки: збільшення та зменшення значення лічильника.

8. **Регулятор** – дає змогу змінювати числове значення пов'язаної з ним величини.

Діалогові вікна на відміну від програмних мають фіксований розмір і єдиний режим відображення, а також у рядку заголовка мають кнопки *Закрити* та *Довідка*.

Програмні вікна

Такі вікна створюються для кожної запущеної програми, за його допомогою здійснюється керування її роботою. У заголовку програмного вікна зазвичай відображається назва програми та/або назва поточного об'єкта. Крім заголовка та меж, **програмні вікна містять такі елементи** (рис. 6.11):

- **рядок меню** – використовується для виконання команд програми і завжди відображається на екрані;
- **панель інструментів** – призначено для швидкого виконання основних команд програми;
- **кнопка системного меню**, яке використовується для виконання основних операцій з вікном. Для відкриття системного меню можна також використати контекстне меню кнопки вікна на панелі задач;
- **рядок стану** – призначений для виведення додаткової інформації про роботу програми і призначення пунктів меню і кнопок;
- кнопки **Згорнути, Розгорнути, Закрити**;
- **смуга прокрутки** – використовується для перегляду вмісту вікна.

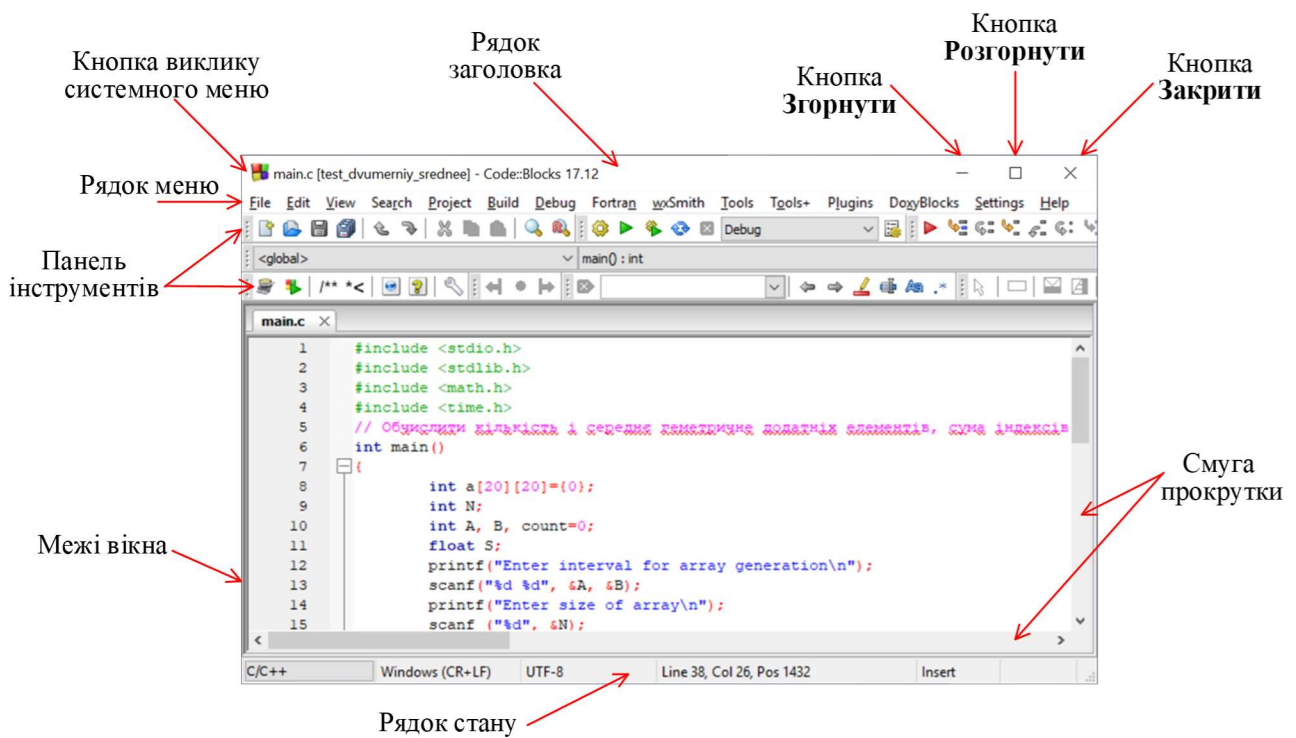


Рисунок 6.11 – Елементи програмних вікон

6.5 Панель задач та головне меню. Довідкова система

Для керування роботою об'єктів використовується меню.

Меню – це перелік команд, які можна виконати, обравши відповідний пункт меню. Головне меню Windows можна викликати, натиснувши клавішу **Win**.

Важливе значення при роботі у Windows має контекстне меню, яке викликається натисканням правої кнопки миші на значку об'єкта або клавіші **⇧**. **Контекстне меню** містить перелік команд, які можна виконати для вибраного об'єкта (рис. 6.12). Зокрема воно містить команду *Властивості*, яка дає змогу переглянути та змінити властивості об'єктів.

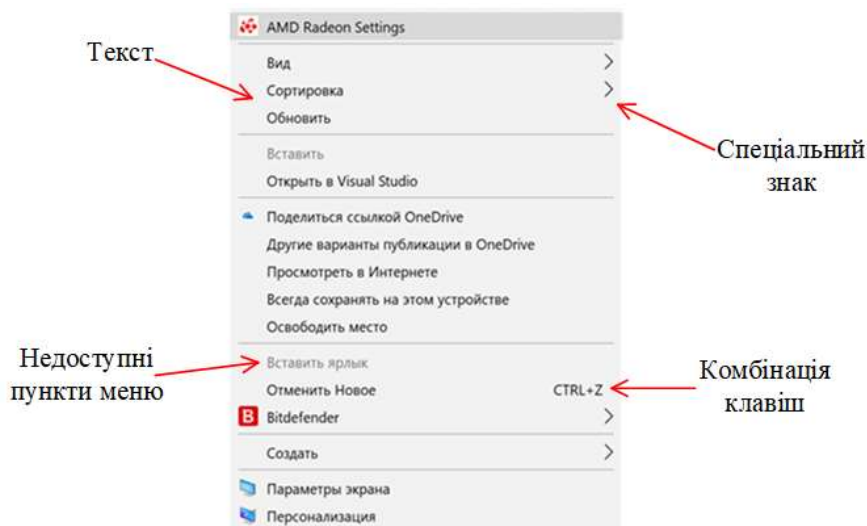


Рисунок 6.12 – Контекстне меню робочого столу

Для виконання команди необхідно вибрати відповідний пункт меню та натиснути клавішу *Enter*. Для закриття меню необхідно натиснути *Esc*. Довільний пункт меню може бути в певний момент часу доступним або недоступним.

Пункти меню містять таке:

- **текст** – короткий опис команди;
- **спеціальні знаки**, які вказують на тип пункту меню;
- **опис комбінації клавіш**, що вказує як саме можна виконати команди за допомогою клавіатури не активізуючи меню;
- **піктограму** – графічне позначення, що відображається ліворуч від тексту.

Є такі типи пунктів меню:

- **команда**: при його виборі відбувається виконання команди (спеціальні знаки відсутні);
- **прапорець**: у разі його вибору встановлюється деякий режим роботи (ліворуч з'являється спеціальний знак) або цей режим відмінюється (ліворуч спеціальний знак зникає);
- **розширена команда**: при його виборі з'являється діалогове вікно, за допомогою якого можна визначити параметри виконання команди (спеціальний знак ...);
- **каскадне меню**: У разі його вибору відкривається нове спливаюче меню (спеціальний знак \blacktriangleright).

Головне меню використовується для такого:

- запуску встановлених програм;
- вибору документів, які користувач відкривав останнім часом;
- отримання доступу до налаштувань середовища операційної системи;
- пошуку об'єктів;
- запуску довідкової системи
- завершення роботи операційної системи.

Для отримання додаткової інформації про роботу в ОС Windows користувач може скористатись довідковою системою (натисніть кнопку Пуск, введіть Підказки (Tips або Советы), а потім оберіть бажану категорію) (рис. 6.13).

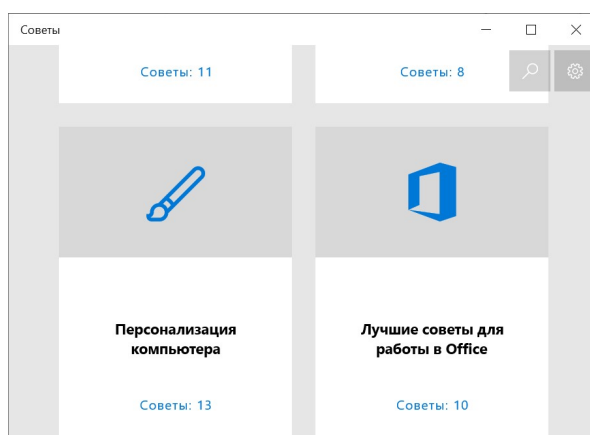


Рисунок 6.13 – Меню довідкової системи Windows

6.6 Налаштування робочого середовища графічної операційної системи

Налаштування середовища Windows здійснюється за допомогою панелі керування, вікно якої містить перелік об'єктів, наприклад, *Клавіатура, Регіональні стандарти, Пристрої та принтери, Диспетчер пристроїв* та ін.

Для налаштування панелі задач використовується діалогове вікно **Параметри панелі задач** і меню **Пуск**. Панель задач містить:

- **панель швидкого запуску** – розташована зліва від панелі задач, на ній є значки для швидкого запуску основних програм;
 - **область повідомлень** – розташована справа від панелі задач, на ній є значки для зміни розкладки клавіатури, індикатор часу та інші елементи;
- Вкладка *Панель задач* містить прапорці, які дають змогу встановити режим відображення елементів панелі задач, зокрема:
- **Закріпити панель задач** – дає можливість фіксувати положення панелі задач на екрані;
 - **Автоматично приховувати панель задач** – не відображується на екрані при роботі з іншими елементами графічного інтерфейсу Windows, але у разі вибору панелі задач вона з'являється на екрані;
 - **Відображати панель задач поверх інших вікон** – при встановленні прапорця панель задач відображається завжди, при зняттю прапорця відкриті вікна перекривають панель задач.

Корисні посилання

1. Налаштування панелі задач Windows10. URL : <http://xn--j1a5b.dp.ua/nalashtuvati-panel-zavdan-v-windows-10/>
2. Путівник по Linux. URL : <https://linuxguide.rozh2sch.org.ua/>

Тестові завдання

1. Вперше концепцію графічного інтерфейсу було запропоновано:
 - а) вченими лабораторії Херох;
 - б) фон Нейманом;
 - в) компанією Microsoft;
 - г) компанією Apple.
2. Система зв'язку і засобів сполучення вузлів та блоків ЕОМ між собою – це:
 - а) графічний інтерфейс;
 - б) людино-машинний інтерфейс;
 - в) внутрішньомашинний інтерфейс;
 - г) користувацький інтерфейс.
3. Не є видом користувацького інтерфейсу:
 - а) простий;
 - б) двовимірний;

- в) бінарний;
 - г) тривимірний.
4. Скільки графічних оболонок існує в системі Windows:
- а) 2;
 - б) 3;
 - в) 1;
 - г) безліч.
5. Графічний символ, який містить зображення деякого значка на робочому столі:
- а) піктограма;
 - б) вікно;
 - в) меню;
 - г) ярлик.
6. Файл, який служить покажчиком на об'єкт, називається:
- а) піктограма;
 - б) графічний курсор;
 - в) піктограма;
 - г) контекстне меню.
7. Для перемикання активного вікна використовують таку комбінацію клавіш:
- а) Alt+Tab;
 - б) Alt+Ctrl+Del;
 - в) Ctrl+Shift;
 - г) Esc.
8. Яка кнопка відсутня у діалоговому вікні:
- а) Відмінити;
 - б) Застосувати;
 - в) Ок;
 - г) Видалити.
9. Який тип вікон створюється для кожної запущеної програми:
- а) бінарне;
 - б) діалогове;
 - в) текстове;
 - г) програмне.
10. Головне меню Windows можна викликати, натиснувши клавішу:
- а) Win;
 - б) Alt+Shift;
 - в) F10;
 - г) ESC.

Практичні завдання

1. Розташуйте панель задач вертикально по лівій стороні екрана.
2. Додайте на панель швидкого запуску такі програми: Code Blocks, Libre Office.

3. Додайте у *Швидкий запуск* посилання на папку з лабораторними роботами з МЗКІТ.
4. Вимкніть (увімкніть) ярлик *Кошик* на робочому столі.
5. Вимкніть (увімкніть) вікно повідомлення про видалення файлу з комп'ютера.

Контрольні запитання

1. Які існують види інтерфейсів?
2. Які існують види користувацьких графічних інтерфейсів?
3. Що передбачає концепція DWIM?
4. Які вам відомі графічні середовища, які підтримує ОС Linux?
5. Як розшифровується GNOME?
6. Чим відрізняється ярлик від піктограми?
7. Як створити ярлик на робочому столі?
8. Які операції з об'єктами можна здійснити за допомогою графічного курсора?
9. Які типи вікон вам відомі?
10. Як викликати контекстне меню деякого об'єкта?
11. Які елементи програмного вікна вам відомі? Наведіть стисло опис кожного з них.
12. Для чого використовують головне меню?
13. Як отримати довідкову інформацію про роботу операційної системи?
14. Яким чином можна додати елемент до швидкого запуску?

Тема 7. Загальносервісні системні програми

Мета: ознайомитись з основними засобами діагностики комп'ютера, класифікацією вірусів, алгоритмами стиснення даних, принципами роботи програм-архіваторів.

План

1. Визначення основних характеристик ПК за допомогою програмних засобів.
2. Поняття про комп'ютерні віруси. Класифікація вірусів. Ознаки зараження вірусом.
3. Класифікація і принципи роботи антивірусних програм.
4. Архівація файлів і каталогів.
5. Робота з програмами-архіваторами.

7.1 Визначення основних характеристик ПК за допомогою програмних засобів

В операційних системах Windows та Linux вже є набір вбудованих програм і способів, щоб швидко дізнатися параметри комп'ютера, не встановлюючи додаткове програмне забезпечення.

Методи отримання інформації відрізняються рівнями складності виклику, отже користувач може підібрати для себе найбільш зручні та легкодоступні варіанти.

Стандартні способи.

Службова утиліта «Відомості про систему».

Запуск вбудованої програми-утиліти відомостей про систему відбувається у декілька етапів:

1. На робочому столі обрати значок *Цей комп'ютер* і викликати контекстне меню.
2. Далі потрібно обрати пункт *Властивості*.

У вікні, що відкрилось, відобразиться інформація про версію операційної системи, інсталювані оновлення, ліцензію (рис. 7.1). Далі відображаються відомості про характеристики процесора, пам'яті ОЗП, розрядності ОС, а також індекс продуктивності системи. Нижче – ім'я комп'ютера.

Ще скоріше доступ до цього вікна можна отримати, використовуючи гарячі комбінації клавіш: **Win+Pause/Break**. Клавіша **Pause/Break** є на стандартних клавіатурах комп'ютерів, але може бути відсутня на клавіатурах ноутбуків, виробники яких прагнуть компактності і відмовляються від деяких кнопок на догоду іншим функціям.

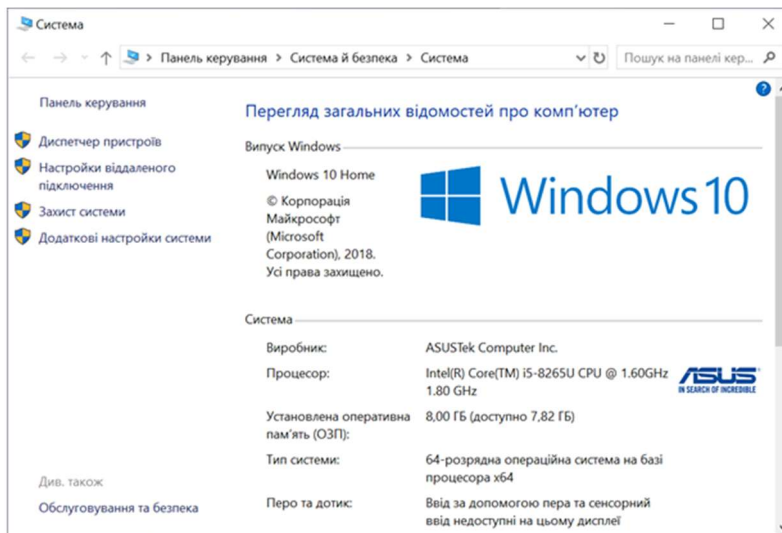


Рисунок 7.1 – Відомості про систему

Панель керування.

Не завжди буває просто запам'ятати поєднання клавіш для різних програм. Варто пам'ятати, що на будь-якому комп'ютері є головний центральний пункт контролю всіх пристроїв і параметрів. Це *Панель керування*. Достатньо запам'ятати, як її запустити і завжди буде доступ до всіх важливих функцій та інформації.

1. Увійти у меню *Пуск*, а далі обрати команду меню *Панель керування*.
2. Наступне вікно може бути представлене у двох варіантах. Це залежить від того, чи запускалось воно раніше. Стандартний вигляд вікна – за категоріями. Доступ до окремих функцій тематично згруповано під загальними заголовками, причому у пункти меню віднесено найбільш необхідні користувачу відомості. Але, як показано на рисунку 7.2, обравши іншу опцію перегляду, відкривається інше меню з більш детальною інформацією по всім пристроям.

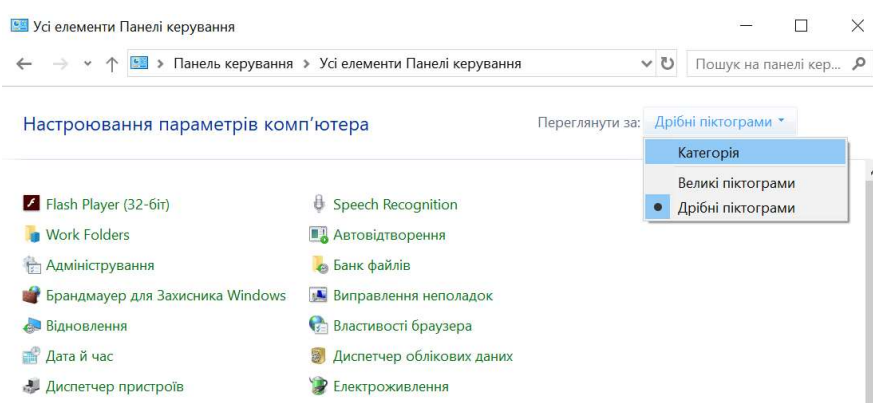


Рисунок 7.2 – Панель керування

Службова утиліта «Диспетчер пристроїв».

Дуже часто при прийнятті рішення про модернізацію комп'ютера буває недостатньо відомостей, які представлено у вікні властивостей системи. Наприклад, потрібно дізнатись точну модель процесора або дані про інші компоненти. Тому можна скористатись *Диспетчером пристроїв*, меню якого

можна відкрити, обравши відповідний пункт у панелі керування (рис. 7.2). У цьому пункті містяться відомості про всі компоненти комп'ютера, а також зовнішні пристрої, що під'єднано.

Окрім загальної інформації про склад апаратного забезпечення комп'ютера *Диспетчер пристроїв* може повідомити ще більш цінну інформацію – чи не час оновити драйвер пристрою або чи немає деяких конфліктів між пристроями. Для кожного елемента можна викликати контекстне меню та обрати пункт *Властивості*, де буде представлено детальну інформацію про компонент, драйвера та інша інформація, що буває потрібна у виключних випадках. Пристрої, що конфліктують, будуть позначені жовтим трикутником – так можна зрозуміти, що даний компонент, можливо, несумісний з деяким іншим пристроєм, тоді потрібно перевстановити програму або оновити його драйвер.

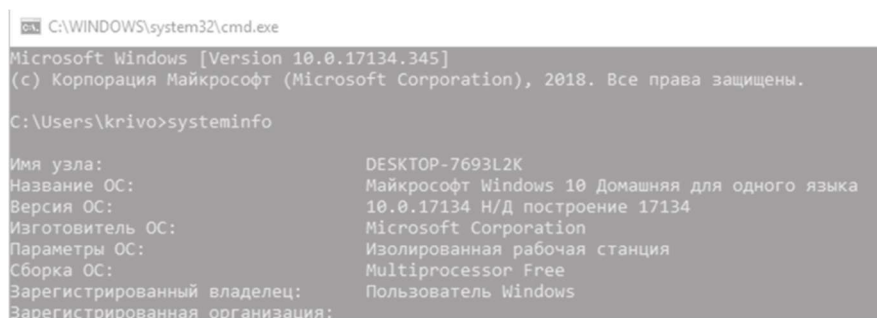
Командний рядок.

Більше інформації ніж вікно *Відомості про систему* дає запуск спеціальної утиліти через командний рядок Windows. Для цього потрібно:

1. Зайти у меню *Пуск*, або натиснути *Win+R*.

2. У вікні, що з'явилося, написати *cmd*.

3. У вікні, що з'явилося написати *systeminfo*. Через декілька секунд система завантажить дані. Окрім відомостей про версію Windows, там можна буде знайти інформацію про стан оперативної пам'яті, мережевих адаптерах, версії Bios, модель системи, доменну групу.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.345]
(c) Корпорация Майкрософт (Microsoft Corporation), 2018. Все права защищены.

C:\Users\krivo>systeminfo

Имя узла:                DESKTOP-7693L2K
Название ОС:             Майкрософт Windows 10 Домашняя для одного языка
Версия ОС:               10.0.17134 Н/Д построение 17134
Изготовитель ОС:         Microsoft Corporation
Параметры ОС:           Изолированная рабочая станция
Сборка ОС:               Multiprocessor Free
Зарегистрированный владелец: Пользователь Windows
Зарегистрированная организация:
```

Рисунок 7.3 – Фрагмент экрана після виконання команди *systeminfo*

Службова утиліта «Засіб діагностики DirectX».

Деталізовані відомості про основні пристрої можна отримати за допомогою «Засоби діагностики DirectX», це програмне забезпечення також входить у склад системи. Запуск утиліти відбувається через аплет *Виконати* та команди *dxdiag* (рис. 7.4). Зі вкладок вікна діагностики DirectX можна дізнатися відомості про версію ОС, модель комп'ютера, Bios, тип процесора, отримати дані про відео та аудіопристрої, зовнішні пристрої, які підключено.

BIOS.

Якщо вищезгаданих способів отримання інформації про систему все ще недостатньо, то можна залізти у саме «ядро» комп'ютера – в *BIOS*. На додаток до основних відомостей там можна дізнатись, наприклад, температуру жорсткого диска та процесора.

Але даний спосіб дізнатися параметри не рекомендується для недосвідчених користувачів, оскільки, по-перше, для входу в BIOS знадобиться

визначити його тип, щоб знайти потрібну клавішу входу, адже у всіх виробників вона відрізняється. По-друге потрібно буде швидко натиснути цю клавішу при завантаженні комп'ютера, оскільки на це дається всього декілька секунд. По-третє, BIOS у більшості випадків представляє собою синій або чорний екран з текстом, який є незвичним для користувачів графічного інтерфейсу. Випадкова зміна налаштувань може призвести до збою у роботі комп'ютера.

Для того, щоб **зайти в BIOS**, потрібно:

1. Почати перезавантаження комп'ютера.
2. Коли на екрані з'являться білі рядки тексту і емблема, потрібно у самому низу екрана знайти надпис «Press ... to enter SETUP».
3. Те, що буде написано після три крапки, є потрібною клавішею.

Найбільш популярними клавішами для доступу в BIOS є DEL або F2.

BIOS на всіх комп'ютерах виглядає по різному і представлення інформації може сильно відрізнятись. Головне – не потрібно змінювати налаштування без необхідності і розуміння, до чого це може призвести.

7.2 Поняття про комп'ютерні віруси. Класифікація вірусів. Ознаки зараження вірусом

Вперше про комп'ютерні віруси почали говорити ще у кінці 40-х років ХХ століття. Так відомий вчений Джон фон Нейман, читав серію лекцій «Теорія організації складних автоматів» у Нью-Йоркському університеті, у якій мова йшла про автомати чи машини, які зможуть самостійно копіюватись.

Перший вірус на комп'ютері реальних користувачів був створений для ПК компанії Apple лише у 1981 році. І зробив це 15-річний юнак, який не переслідував жодної корисливої мети. Ця програма лише виводила на екран зображення, про те, що комп'ютер заражений вірусом Ель-клоно.

Практично під усі типи операційних систем, які дають можливість встановлення додаткового програмного забезпечення, існують віруси. Отже, **вірус** – це невелика програма, яка здатна до саморозмноження й виконання різних деструктивних дій. Віруси діють тільки програмним шляхом.

Для активізації вірусу потрібно завантажити заражений файл, і тільки після цього вірус починає діяти самостійно. Деякі віруси під час запуску зараженого файлу стають резидентними (постійно знаходяться в оперативній пам'яті комп'ютера) і можуть заражати інші файли та програми, що завантажуються. Інші різновиди вірусів відразу після активізації можуть спричиняти серйозні пошкодження, наприклад, форматувати жорсткий диск.

Нині відомі десятки тисяч комп'ютерних вірусів, які поширюються через мережу Інтернет по всьому світу.

Не існує єдиної системи класифікації та іменування вірусів (хоча спроба створити стандарт була зроблена на зустрічі CARO у 1991 році). Далі на рисунку 7.4 представлено умовну класифікацію вірусів за різними ознаками.



Рисунок 7.4 – Класифікація вірусів

Зробимо декілька пояснень до рисунку 7.4:

- мережеві – поширюються по комп'ютерній мережі;
- файлові – впроваджуються у виконувани файли;
- завантажувальні – проникають у завантажувальний сектор диска (Boot-сектор) або у сектор, що містить системний завантажник вінчестера (Master Boot Record)
 - файлово-завантажувальні віруси – заражають як файли, так і завантажувальні сектори дисків. Такі віруси, як правило, мають досить складний алгоритм роботи й часто застосовують оригінальні методи проникнення у систему;
 - нешкідливі – ті, які ніяк не впливають на роботу комп'ютера (крім зменшення вільної пам'яті на диску в результаті свого поширення);
 - безпечні – вплив яких обмежується зменшенням вільної пам'яті на диску й графічними, звуковими ефектами;
 - небезпечні віруси – ті, які можуть призвести до серйозних збоїв у роботі, або до втрати інформації;
 - дуже небезпечні – ті, які можуть призвести до фізичного пошкодження обладнання (перезаписування ПЗП, виходу з ладу дискових пристроїв, пошкодження елементів материнської плати тощо);
 - резидентний вірус – при інфікуванні комп'ютера залишає в оперативній пам'яті свою резидентну частину, що потім перехоплює звернення операційної системи до об'єктів зараження й впроваджується в них (перебувають у пам'яті і є активними аж до вимикання або перезавантаження комп'ютера);
 - нерезидентні віруси – не заражають пам'ять комп'ютера і є активними обмежений час. Деякі віруси залишають в оперативній пам'яті невеликі резидентні програми, які не поширюють вірус.

Окремо розглянемо такі **типи вірусів**:

Троянські програми (Trojans) – виконують шкідливі дії замість оголошених легальних функцій або разом з ними. Вони не спроможні до самовідтворення і передаються тільки при копіюванні користувачем. Після запуску вони зазвичай знищують себе разом з іншими файлами на диску.

Руткіт (маскувальник) – це спеціальний вид троянських програм, головна мета якого – максимально глибоко інсталюватись у систему, щоб його було якомога важче знайти і видалити. Як правило, руткіти містять драйвери операційних систем і працюють на досить низькому рівні, тому їх дуже важко знайти і ще важче видалити з системи. Руткіти використовуються для маскувannya всіх інших компонентів троянця від детектування.

Поштова бомба – велика кількість електронних повідомлень (від 10 тис.), які відсилаються на електронну адресу користувача з метою виведення з ладу системи або перешкод для отримання корисних повідомлень.

Віруси-шпигуни – збирають інформацію про поведінку і дії користувача. Здебільшого їх цікавлять адреси, паролі, дані кредитних карт.

Віруси-блокувальники – блокують користувачеві доступ до операційної системи. При завантаженні комп'ютеру з'являється вікно, в якому користувача звинувачують у скачуванні неліцензійного контенту або порушенні авторських прав. І під загрозою повного видалення всіх даних з комп'ютера вимагають відіслати смс на номер телефону або поповнити його рахунок.

Ознаки зараження вірусом

- зменшення вільної пам'яті;
- уповільнення роботи комп'ютера;
- затримки при виконанні програм;
- незрозумілі зміни у файлах
- зміна дати модифікації файлів без причини;
- зміна атрибутів файлів;
- помилки при запуску ОС;
- наявність файлів невідомого походження

Ранні ознаки зараження дуже важко виявити, але коли вірус переходить в активну фазу, тоді легко помітити такі зміни:

- зникнення файлів;
- форматування жорсткого диску;
- неможливість завантаження комп'ютера, файлів;
- незрозумілі системні повідомлення
- підозріло підвищений інтернет-трафік;
- граничне завантаження процесора.

7.3 Класифікація і принципи роботи антивірусних програм

Разом з розвитком операційних систем та технологічною частиною комп'ютерів почало розвиватись антивірусне програмне забезпечення. У наш час в антивірусах з'явилося багато **можливостей**:

- автоматичний запуск зі стартом операційної системи;

- контроль роботи програм;
- контроль доступу до файлів;
- наявність фільтрів, які перевіряють веб-трафік, електронну пошту;
- перевірка файлів, які передаються по локальній мережі та ін.

Чим більше технологій зв'язку з'являється, чим більше потужностей набуває комп'ютер, тим більше різних функцій набує антивірусна програма.

Принципи роботи антивірусних програм

Сигнатурний аналіз – виявлення загрози за певною попередньо виявленою ознакою.

Сигнатура – це послідовність байт у певній програмі чи контрольна хеш сума, якогось блоку програми, яка характеризує його. Тобто сигнатура характеризує якийсь конкретний вірус і дає переваги у подальшій його обробці антивірусом.

Переваги: точне виявлення загрози.

Недоліки: спочатку загрозу має проаналізувати антивірусна лабораторія.

Евристичний аналізатор – аналізує код програми, знаходить у ньому певні закономірності, характерні для різних класів шкідливих програм і на основі цього приймає рішення щодо можливого визначення програми шкідливою чи ні.

Переваги: можливість виявлення загроз, для яких ще не створені сигнатурні записи.

Недоліки: можливе хибне спрацювання.

Поведінковий аналізатор – аналізується не код програми, а її поведінка в процесі роботи. Такий аналізатор слідкує за поведінкою програм і намагається зрозуміти чи є поведінка цих програм штатною, чи характерна для шкідливих програм. Якщо поведінковий аналізатор вважає, що дана програма поводить себе як шкідлива, він може її заблокувати і навіть «відкотити» назад певні зміни у операційній системі, які зробила шкідлива програма.

Переваги: можливість виявлення загроз, для яких ще не створені сигнатурні записи.

Недоліки: шкідлива програма працює певний час до виявлення; значне завантаження комп'ютера.

Репутаційні технології – виявлення нових загроз за рахунок оцінки від всіх користувачів антивірусу.

Переваги: точне виявлення загроз.

Недоліки: постійна потреба у підключенні до Інтернет; потреба у великій аудиторії користувачів.

Хмарні технології – сигнатурна перевірка файлів відбувається на сервері антивірусної компанії.

Переваги: точне виявлення загрози.

Недоліки: постійна потреба у підключенні до Інтернет.

Отже мінімальний базовий набір технологій для антивірусної програми:

- сигнатурний аналізатор
- евристичний аналізатор
- поведінковий аналізатор

- набір фільтрів, перехоплення подій файлової системи, електронної пошти, можливо і інших каналів.

Також набір інструментів, які дозволяють користувачу ввімкнути/вимкнути захист у режимі реального часу, провести вибіркове сканування і виконати дії над знайденими вірусами. Це є мінімальний набір функцій, який зараз включений у будь-яку безкоштовну антивірусну програму. Крім цього існують продукти, які виходять за межі функціонування безпосередньо антивірусних програм, створюючи нові класи продуктів інформаційної безпеки.

Додаткові модулі (батьківський контроль, фаєрвол, захищений браузер та ін.) дозволяють забезпечити ще більший рівень захищеності комп'ютера.

7.4 Архівація файлів і каталогів

Засновником науки про стиснення інформації вважають Клода Шенона. Його теорема про оптимальне кодування показує, до чого потрібно прагнути при кодуванні інформації та наскільки та чи інша інформація при цьому буде стискатись. Окрім того ним було проведено експерименти з емпіричної оцінки надлишковості англійського тексту. Шенон пропонував людям вгадувати наступну букву та оцінював правильність вгадування. На основі ряду дослідів він дійшов висновку, що кількість інформації в англійському тексті коливається у границях 0,6-1,3 біти на символ.

Стиснення даних – це процес, який забезпечує зменшення об'єму даних за рахунок скорочення їхньої надлишковості. Стиснення даних пов'язане із компактним розміщенням порцій даних стандартного розміру.

В залежності від того, в якому об'єкті розміщено дані, що підлягають стисненню розрізняють:

- **стиснення (архівування) файлів:** використовується для зменшення розмірів файлів при підготовці їх до передачі каналами зв'язку або до транспортування на зовнішніх носіях малої ємності;
- **стиснення (архівування) папок** для зменшення обсягу інформації або для тривалого зберігання, наприклад при резервному копіюванні;
- **стиснення (ущільнення) дисків.**

Існує багато **практичних алгоритмів стиснення файлів**, але всі вони базуються на трьох теоретичних способах зменшення надлишковості даних.

Перший полягає в зміні вмісту даних, другий – у зміні структури даних, третій – в одночасній зміні як структури, так і вмісту даних.

1. Якщо при стисненні даних відбувається зміна їх вмісту, то **метод стиснення є незворотнім**, тобто, при відновленні (розархівуванні) даних з архіву не відбувається повне відновлення інформації. Такі методи часто називають **методами стиснення з регульованими втратами інформації**.

До таких типів даних відносяться відео- та аудіодані, а також графічні дані. Методи стиснення з регульованими втратами забезпечують значно більший ступень стиснення, але їх не можна використовувати до текстових даних.

Прикладами форматів стиснення з втратами інформації можуть бути JPEG (Joint Photographic Experts Group) для графічних даних, MPG – для відеоданих; MP3 – для аудіоданих.

2. Якщо при стисненні даних відбувається тільки зміна структури даних, то **метод стиснення є зворотнім**. У цьому випадку з архіву можна відновити інформацію повністю. Зворотні методи стиснення застосовують до будь-яких типів даних, але вони дають меншу ступінь стиснення у порівнянні з незворотними методами. Приклади форматів без втрати інформації: GIF (Graphics Interchange Format), TIFF (Tagged Image File Format) – для графічних даних, AVI – для відео даних; ZIP, ARJ, RAR, CAB, LH – для довільних типів даних. Умовну класифікацію алгоритмів без втрат представлено на рисунку 7.5.

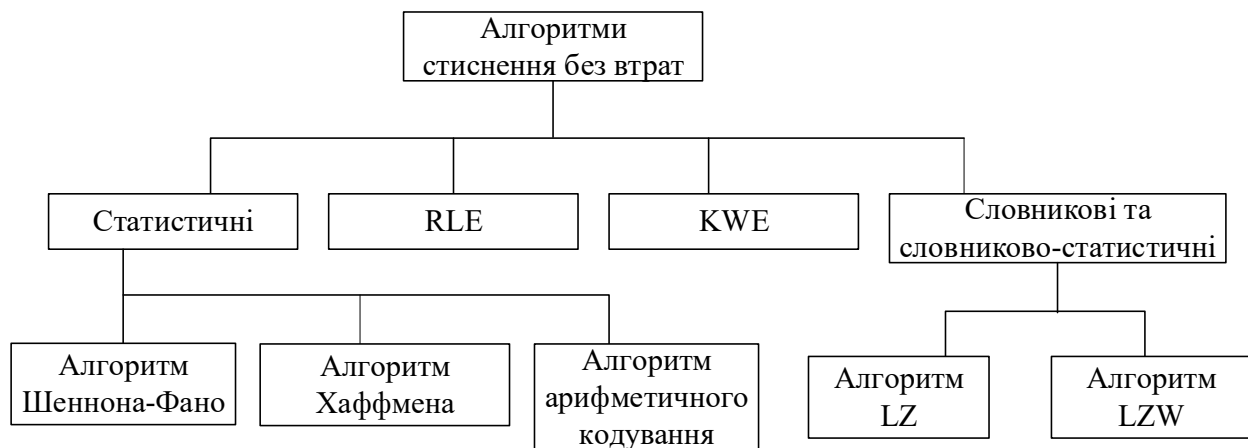


Рисунок 7.5 – Класифікація алгоритмів стиснення без втрат

Основні технічні характеристики процесів стиснення та результатів їхньої роботи:

1. Ступінь стиснення (compress rating) або відношення (ratio) об'ємів вихідного та результуючого потоків.

2. Швидкість стиснення – час, який витрачено на стиснення деякого об'єму інформації вхідного потоку, до отримання з нього еквівалентного вихідного потоку.

3. Якість стиснення – величина, яка показує на скільки сильно упаковано вихідний потік за допомогою застосування до нього повторного стиснення за тим же або іншим алгоритмом.

Існує багато різних практичних методів стиснення без втрати інформації, які мають різну ефективність, але в основі цих методів лежать **три теоретичні алгоритми**:

- алгоритм RLE (Run Length Encoding);
- алгоритми групи KWE;
- алгоритм Хаффмана.

Алгоритм RLE. Стиснення методом кодування серій

В основі лежить ідея виявлення послідовностей даних, що повторюються та заміни цих послідовностей більш простою структурою, в якій вказується код даних та коефіцієнт повторення.

Приклад 7.1 Нехай задана така послідовність даних, що підлягає стисненню.

1 1 1 1 2 2 3 4 4 4

В алгоритмі RLE пропонується замінити її такою структурою:

1 4 2 2 3 1 4 3,

де перше число кожної пари чисел – це код даних, а друге – кількість повторень. Якщо для зберігання кожного елемента даних вхідної послідовності відводиться 1 байт, то вся послідовність займатиме 10 байт пам'яті, тоді як вихідна послідовність займатиме 8 байт пам'яті. Коефіцієнт стиснення, що характеризує ступінь стиснення, можна обчислити за такою формулою:

$$k = \frac{V_x}{V_n} \cdot 100\% = \frac{8}{10} \cdot 100\% = 80\%,$$

де V_n – об'єм пам'яті для зберігання вхідної інформації, V_x – вихідної послідовності даних.

Чим менше значення коефіцієнта стиснення, тим ефективніший метод стиснення.

Даний метод є досить ефективним для стиснення растрових графічних зображень (BMP, PCX, TIF, GIF), оскільки вони містять досить багато довгих серій послідовностей байтів, що повторюються. Програмні реалізації алгоритмів RLE відрізняються простотою, високою швидкістю роботи, але забезпечують недостатнє стиснення. Для текстових даних методи RLE, як правило, неефективні.

Недоліком даного алгоритму є досить низький ступінь стиснення. До переваг можна віднести те, що він не потребує додаткової пам'яті при архівації, а також швидко працює.

Алгоритм групи KWE.

Існує багато реалізацій цього алгоритму, серед яких найбільш поширеними є алгоритм Лемпеля-Зіва (LZ), та його модифікація – алгоритм Лемпеля-Зіва-Велча (LZW).

Алгоритм починає роботу з майже пустого словника, який містить тільки один закодований рядок, коли зчитується черговий символ вхідної послідовності він додається до поточного рядка. Зчитування закінчується, коли поточний рядок не відповідає жодній фразі зі словника. У цей момент формується код, що складається з індексу збігу і наступного символу. Ця нова фраза додається у словник. Коли з'являється ця послідовність, то фразу можна використати для побудови більш довгої фрази, що підвищує ступінь стиснення інформації. Алгоритми стиснення цієї групи найефективніші для текстових даних великих обсягів і малоефективні для файлів малих розмірів.

Цей алгоритм є частиною стандарту на формат зображень GIF. У наш час алгоритм міститься у стандарті PDF.

Алгоритм Хаффмана

Основою цього алгоритму є ідея кодування бітовими групами. Спочатку встановлюється частота входження кожного символу вхідної послідовності, після цього символи сортуються за спаданням частоти. Основна ідея полягає в наступному: чим частіше зустрічається символ, тим меншою кількістю біт він

кодується. Результат кодування зводиться у словник, що необхідний для декодування.

Алгоритм Хаффмана є універсальним, але він малоефективний для файлів малих розмірів.

Приклад 7.2 Файл довжиною у 100 байт містить 6 різних символів. Підрахунок входжень кожного із символів дав такий результат:

Символ	A	B	C	D	E	F
Кількість входжень	10	20	30	5	25	10

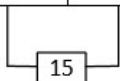
Далі сортуємо дані таблиці за спаданням частоти входження кожного символу:

Символ	C	E	B	F	A	D
Кількість входжень	30	25	20	10	10	5

Візьмемо з останньої таблиці 2 символи з найменшою частотою – це D (5) і будь-який із символів F або A(10), наприклад, A.

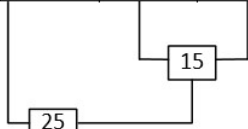
Сформуємо із «вузлів» D та A новий «вузол», частота входження якого буде дорівнювати сумі частот D і A:

Символ	C	E	B	F	A	D
Кількість входжень	30	25	20	10	10	5

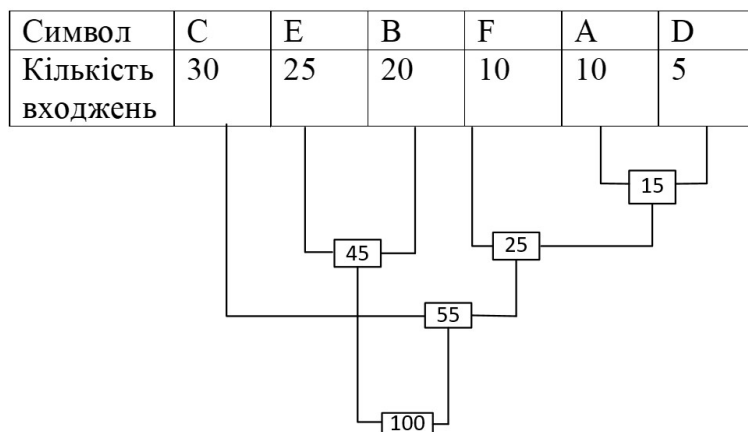


Тепер знову шукаємо два символи із самими маленькими частотами входження. Виключаючи із перегляду D і A і розглядаючи замість них новий «вузол» із сумарною частотою входження. Найнижча частота тепер у F та нового «вузла». Знову зробимо операцію злиття вузлів:

Символ	C	E	B	F	A	D
Кількість входжень	30	25	20	10	10	5



Для двох наступних символів (B і E) знову здійснюємо операцію об'єднання вузлів. Продовжуємо поки все «дерево» не сформовано, тобто поки все не зведеться до одного вузла.



Коли дерево створено, можна кодувати файл. Завжди починають з кореня (100). Кодуючи перший символ (листок C), простежуємо наверх по дереву усі повороти гілок і, якщо робимо лівий, то запам'ятовуємо 0-й біт, якщо правий – то 1-й біт. У результаті отримуємо:

C=10 (2 біти)

E=00 (2 біти)

B=01 (2 біти)

F=110 (3 біти)

A=1110 (4 біти)

D=1111 (4 біти)

При кодуванні замінюємо символи на отримані послідовності.

Початковий розмір файлу: 100 байт – 800 біт;

Розмір вихідного файлу: 30 байт – 240 біт.

Отже стиснули цей файл на 70%.

7.5 Робота з програмами-архіваторами

На практиці програмні засоби стиснення даних синтезують ці три «чисті» алгоритми стиснення файлів, оскільки їх ефективність залежить від типу і розміру файлу. Далі у таблиці 7.1 наведено найпоширеніші формати стиснення та відповідні їм програми-архіватори, що використовуються на практиці.

Таблиця 7.1 – Програми-архіватори та відповідні їм формати

Формат стиснення	Програма архівування	Програма розархівування
ARJ	WinArj.exe	WinArj.exe
RAR	WinRar.exe	WinRar.exe
ZIP	WinZip.exe	WinZip.exe

Крім того сучасні архіватори надають користувачеві повний спектр послуг для роботи з архівами, основними з яких є:

- створення нового архіву;
- додавання файлів в існуючий архів;
- розпакування файлів з архіву;
- створення саморозпаківних архівів;
- створення архівів фіксованих розмірів;
- захист архівів паролями від несанкціонованого доступу;
- перегляд вмісту файлів різних форматів без попереднього розархівування;
- пошук файлів і даних в архіві;
- перевірка на віруси в архіві до розпакування;
- вибір та налаштування коефіцієнта стиснення.

Корисні посилання

1. Які загрози очікують на користувачів у 2019 році.
URL : <https://zillya.ua/kibersvit-yaki-zagrozi-ochikuyut-na-koristuvachiv-u-2019-rotsi>
2. Онлайн курс «Основи інформаційної безпеки»
URL : https://edx.prometheus.org.ua/courses/KPI/IS101/2014_T1/info
3. Архівація у Linux. URL : https://linuxguide.rozh2sch.org.ua/#_файли_архіви
4. Як дізнатись характеристики комп'ютера Linux.
URL : <http://www.maksiv.ru/kak-uznat-karakteristiki-kompyutera-v-linux/>

Тестові завдання

1. Доступ до вікна перегляду відомостей про систему можна отримати натиснувши таку комбінацію клавіш:
 - а) Win+Pause/Break;
 - б) Alt+Ctrl+Del;
 - в) Shift+Enter;
 - г) Tab.
2. Службову утиліту «Засіб діагностики DirectX» можна викликати через аплет *Виконати* командою:
 - а) cmd;
 - б) time;
 - в) dir;
 - г) dxdiag.
3. Який тип шкідливого програмного забезпечення використовується для маскуванню інших шкідливих програм?
 - а) руткіт;
 - б) троянська програма;
 - в) хробак;
 - г) діалер.
4. Яка особливість присутності на комп'ютері завантажувальників?

- а) вони використовуються розробниками ігор;
 - б) файли, що уражені завантажувальниками потрібно лікувати;
 - в) ніяких особливостей;
 - г) свідчить про присутність і інших шкідливих програм.
5. Виявляє загрозу за попередньо виявленою ознакою:
- а) сигнатурний аналіз;
 - б) евристичний аналіз;
 - в) поведінковий аналіз;
 - г) хмарні технології.
6. Найбільш точною з перелічених антивірусних технологій є:
- а) евристичний аналіз;
 - б) поведінковий аналіз;
 - в) сигнатурний аналіз;
 - г) хмарні технології.
7. Прикладами форматів стиснення з втратами інформації можуть бути:
- а) JPEG;
 - б) TIFF;
 - в) AVI;
 - г) GIF.
8. До статистичних алгоритмів стиснення без втрат відноситься:
- а) алгоритм Хаффмана;
 - б) алгоритм LZ;
 - в) алгоритм LZW;
 - г) алгоритм KWE.
9. Ідея кодування бітовими групами лежить в основі алгоритму:
- а) KWE;
 - б) Хаффмана;
 - в) DEFLATE;
 - г) RLE.
10. Головне меню Windows можна викликати, натиснувши клавішу:
- а) Win;
 - б) Alt+Shift;
 - в) F10;
 - г) ESC.

Практичні завдання

1. Знайдіть приклади на кожен тип вірусу.
2. Створіть саморозпакований архів у Linux.
3. Закодуйте алгоритмом RLE таку послідовність та обчисліть коефіцієнт стиснення: 44 44 44 11 11 11 11 11 01 03 FF 22 22.
4. Послідовність з попереднього завдання закодуйте алгоритмом Хаффмана, обчисліть коефіцієнт стиснення та порівняйте результати.

Контрольні запитання

1. Які існують додаткові програми для діагностики системи? Наведіть приклади платних і безкоштовних утиліт.
2. Що таке вірус? Які віруси називаються резидентними? Нерезидентними? Що таке поштова бомба?
3. Які є віруси для Linux?
4. Які віруси можуть маскуватися під макроси у Word або Excel?
5. Які основні ознаки зараження вірусом?
6. Які основні принципи роботи антивірусних програм?
7. Що таке стиснення даних?
8. Чи завжди стиснення і архівація одне й те саме?
9. Які основні технічні характеристики процесу стиснення?

Тема 8. Організація файлової системи

Мета: ознайомитись з поняттями операційної системи, файлу, каталогу, засобами фізичної і логічної організації дискового простору.

План

1. Класифікація, призначення, основні функції та склад операційних систем.
2. Системний диск. Логічні та фізичні диски.
3. Поняття файлу. Типи файлів. Атрибути файлів. Правила запису імен файлів в операційних системах. Поняття каталогу. Повне ім'я файлу. Маршрут у файлової системі. Шаблони. Деревоподібна структура збереження інформації.
4. Файлова система. Логічна організація файлу.
5. Фізична організація файлу.
6. Загальна модель файлової системи.

8.1 Класифікація, призначення, основні функції та склад операційних систем

Операційна система (ОС) – це базовий комплекс програм, що виконує управління апаратною складовою комп'ютера або віртуальної машини, забезпечує керування обчислювальним процесом і організовує взаємодію з користувачем.

Основні функції операційної системи:

1. Обмін даними між комп'ютером і різними периферійними пристроями. Такий обмін називається *введення/виведення даних*.
2. Забезпечення системи організації та зберігання файлів. Також сюди відноситься перевірка, «лікування» і форматування диска, видалення та відновлення файлів, організація файлової системи та ін. Ці процеси здійснюються за допомогою спеціальних програм – *утиліт*.
3. Завантаження програм у пам'ять і забезпечення їх виконання.
4. Організація діалогу з користувачем.

Отже, операційна система виконує роль сполучною ланки між апаратною частиною комп'ютера і програмами з одного боку, а також користувачем з іншого боку.

Структуру ОС складають такі модулі:

1. **Базовий модуль (ядро ОС)** – керує роботою програм і файловою системою, забезпечуючи доступ до неї й обмін файлами між периферійними пристроями.
2. **Командний процесор** – розшифровує та виконує команди користувача.
3. **Драйвери периферійних пристроїв** – програмно забезпечують узгодженість роботи цих пристроїв з процесором.

4. **Файлова система** – структура збереження даних на зовнішніх носіях і сукупність програм, які забезпечують роботу з цією структурою. Як правило, операційна система може працювати з кількома файловими системами.

5. **Інтерфейс користувача** – сукупність засобів, які забезпечують обмін даними між операційною системою та користувачем.

Файли складових ОС зберігаються на диску, але для запуску ОС вони мають знаходитись в ОЗП (оперативній пам'яті); для проведення запису файлів ОС в ОЗП необхідно виконати програму завантаження, що полягає у послідовному завантаженні ОС в оперативну пам'ять.

Етапи завантаження ОС.

1. У системному блоці знаходиться постійна пам'ять, в якій містяться програми тестування блоків комп'ютера і першого етапу завантаження операційної системи. Вони запускаються з першим імпульсом струму при ввімкненні комп'ютера. На цьому етапі процесор звертається до диска і перевіряє наявність на певному місці (на початку диска) програми завантажувача, якщо така програма виявлена, то вона зчитується в оперативну пам'ять і їй передається керування.

2. Програма завантажувач шукає на диску базовий модуль ОС, переписує його у пам'ять і передає йому управління.

3. Після закінчення завантаження ОС управління передається командному процесору.

Підчас роботи комп'ютера в оперативній пам'яті обов'язково повинні знаходитись базовий модуль ОС і командний процесор. Отже, не має необхідності завантажувати всі файли операційної системи в оперативну пам'ять одночасно. Драйвери пристроїв і утиліти можуть довантажуватись по мірі необхідності.

Класифікація ОС.

1. За кількістю одночасно працюючих користувачів:
 - однокористувацькі;
 - багатокористувацькі.
2. За числом завдань, одночасно виконуваних під управлінням ОС:
 - однозадачні;
 - багатозадачні.
3. За кількістю підтримуваних процесів:
 - однопроцесорні;
 - багатопроцесорні.
4. За розрядністю коду ОС: 8, 16, 32, 64 розрядні.
5. За типом користувальницького інтерфейсу:
 - командні (текстові);
 - об'єктно-орієнтовані (графічні).
6. За типом доступу користувача до ресурсів комп'ютера:
 - з пакетною обробкою;
 - з розділенням часу і реального часу.
7. За типом використання ресурсів:
 - локальні;

- мережні.

Аналіз і виконання команд користувача, включаючи завантаження готових програм з файлів в оперативну пам'ять і їх запуск здійснює командний процесор ОС.

8.2 Системний диск. Логічні та фізичні диски

Основним типом пристроїв для зберігання файлів у комп'ютері є дискові накопичувачі.

Жорсткий диск HDD (фізичний диск) – запам'ятовувальний пристрій довільного доступу, в основі роботи якого є принцип магнітного запису.

Для зручності роботи жорсткий диск розбивають на логічні диски, які є віртуальними. На одному фізичному диску може бути або декілька **ЛОГІЧНИХ ТОМІВ**, або тільки один.

Інформація на жорсткому диску розміщується у кільцевих шляхах на магнітних поверхнях диску, що називаються **доріжками** (рис. 8.1).

Сукупність доріжок, що розміщені на однаковій відстані від осі диска, називають **циліндром**.

Кількість циліндрів обирається таким чином, щоб у процесі запису на кожен доріжку не руйнувалося інформація на сусідніх доріжках.

Нульова доріжка містить сервісну інформацію, параметри диска для роботи контролера. Кожна доріжка розбивається на фрагменти, які називаються **секторами** (рис. 8.2). Сектор має фіксований для конкретної системи розмір, що дорівнює деякому ступеню двійки (наприклад 512 байтів). **Сектор** – це найменша адресована одиниця обміну даними з оперативною пам'яттю. Для того, щоб контролер зміг знайти потрібний сектор, потрібно вказати номер циліндра, номер поверхні і номер сектора.

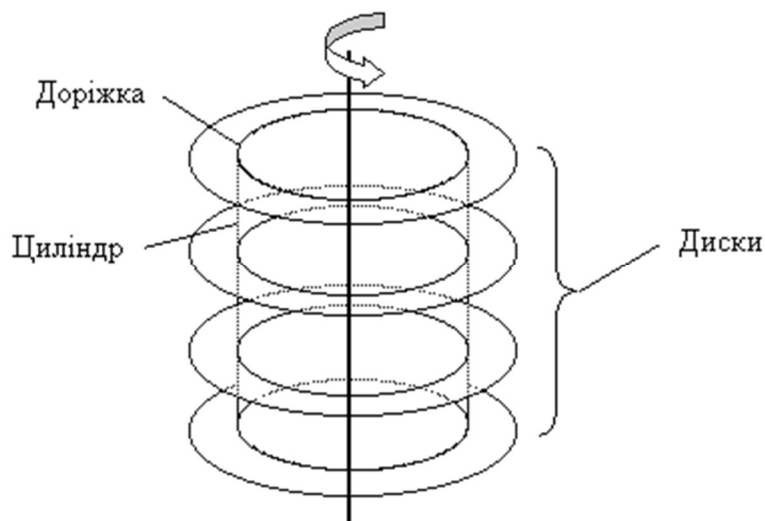


Рисунок 8.1 – Структура циліндрів і доріжок вінчестера

Операційна система при роботі з диском використовує власну одиницю дискового простору, яка називається **кластером**. Доріжки і сектори створюються

у результаті виконання процедури фізичного або низькорівневого форматування диску. Низькорівневий формат диску не залежить від типу операційної системи.

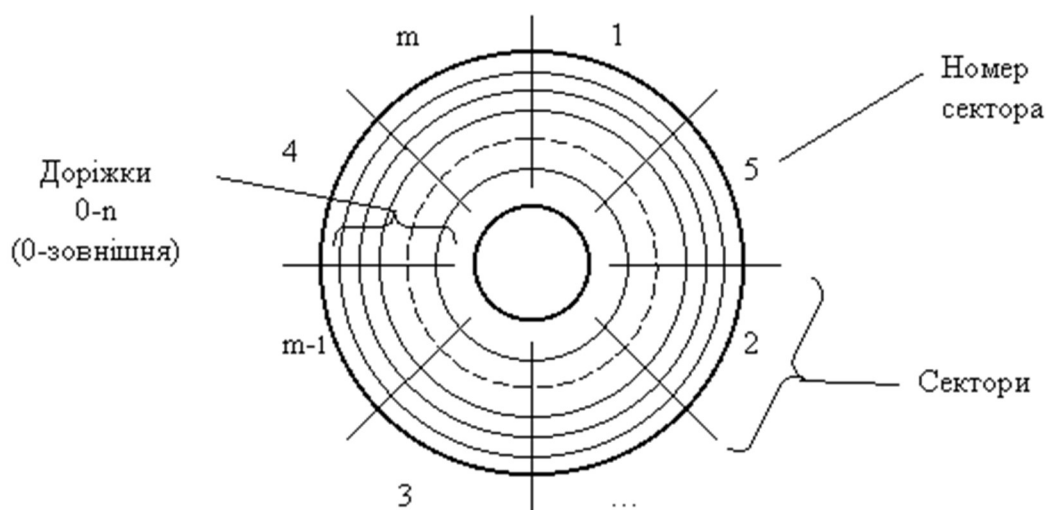


Рисунок 8.2 – Структура доріжок і секторів на диску

Розмітку диска під конкретний тип файлової системи виконує процедура високорівневого або логічного форматування.

Логічне форматування – це процес перетворення вже розміченого дискового простору у відповідності зі стандартами операційної системи. При високорівневому форматуванні визначається розмір кластера і на диск записується інформація для роботи з файловою системою, окрім того на диск записується завантажувач операційної системи.

Перед форматуванням диску для певної файлової системи, його можна розбити на розділи.

Розділ – це безперервна частина фізичного диску, яку операційна система представляє користувачеві як логічний пристрій. Цей логічний пристрій функціонує так, якщо б це був логічний диск.

Операційні системи різного типу використовують єдине для всіх уявлення про розділи, але створюють на його основі логічні пристрої специфічні для кожної системи.

Для організації **логічного диска** кожна операційна система, розділяє його на **дві частини: системну область та область даних (data)** (рис. 8.3).

У свою чергу **системна область** має такі розділи:

- завантажувальний запис – початкова область логічного диска, що містить невелику програму, яка ініціює запуск операційної системи, містить блок-параметри диска та системний завантажувач;
- область керування розміщенням файлів;
- кореневий каталог – зміст інформації, що міститься в області даних.

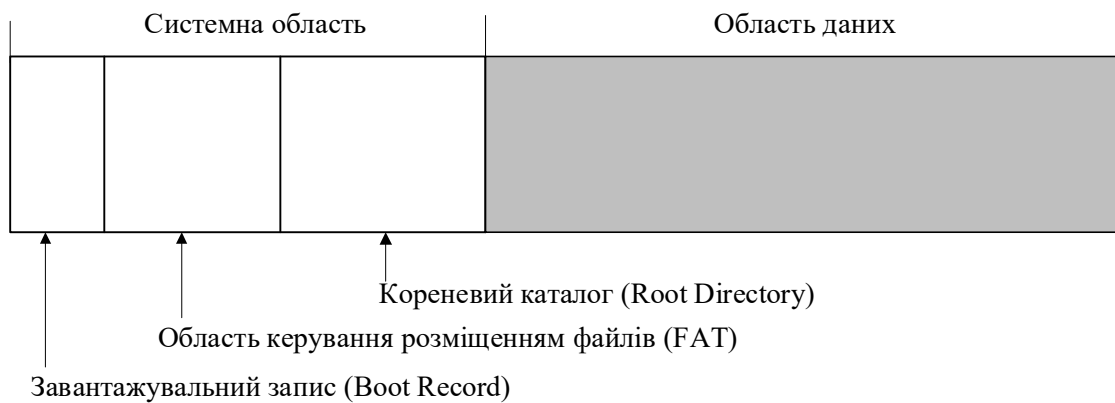


Рисунок 8.3 – Структура логічного диска

Операційна система може підтримувати різні статуси розділів, особливим чином помічаючи розділи, що можуть бути використані для завантаження операційної системи або ті, в яких можливо лише зберігати данні.

8.3 Поняття файлу. Типи файлів. Атрибути. Правила запису імен файлів в операційних системах. Поняття каталогу. Повне ім'я файлу. Маршрут у файлової системі. Шаблони. Деревоподібна структура збереження інформації

Інформація у пам'яті комп'ютера зберігається у файлах.

Файл – це поіменована область на диску або іншому носії інформації. У файлах можуть зберігатися документи, готові до виконання програми і довільні інші дані.

Файли бувають різних типів:

- звичайні,
- спеціальні,
- файли-каталоги.

Звичайні файли у свою чергу поділяються на **текстові та двійкові**.

Текстові файли складаються з рядків символів, які представлено в ASCII-кодуванні. Це можуть бути документи, вихідні тексти програм та ін. Текстовий файл можна прочитати на екрані та роздрукувати на принтері.

Атрибути файлу – це параметри, за якими файл відрізняється від множини інших файлів.

Двійкові файли не використовують ASCII-кодування, вони мають складну внутрішню структуру, наприклад, об'єктний код програми або архівний файл.

Всі операційні системи повинні вміти розпізнавати хоча б один тип файлів – їх власні виконувані файли.

У різних файлових системах можуть використовуватись такі характеристики файлів (рис. 8.4):

- інформація про дозволи;
- пароль для доступу;
- власник файлу;

- ознака: «тільки для читання», «прихований», «системний», «архівний», «двійковий», «тимчасовий»;
- довжина запису;
- ознака блокування;
- час створення, останнього доступу, останньої зміни;
- поточний розмір.

Позначення файлу для операційних систем називається **іменем** файлу. Імена файлів у Windows складаються з двох частин: імені і розширення. В імені може бути від одного до восьми символів. Розширення починається з точки, за якою записано від одного до трьох символів. **Допустимі символи в імені і розширенні:**

- великі і малі букви;
- цифри;
- символи: -, \$, #, &, @, !, %, (,), {, }, ' , " , ~, ^, +, ,, ;, =, [,] .

Недопустимі символи: \, /, :, *, ?, <, >, |.

Розширення імені файлу не є обов'язковим. Воно описує вміст файлу. Багато програм дозволяють за розширенням файлу викликати відповідну програму і відразу завантажити в неї даний файл.

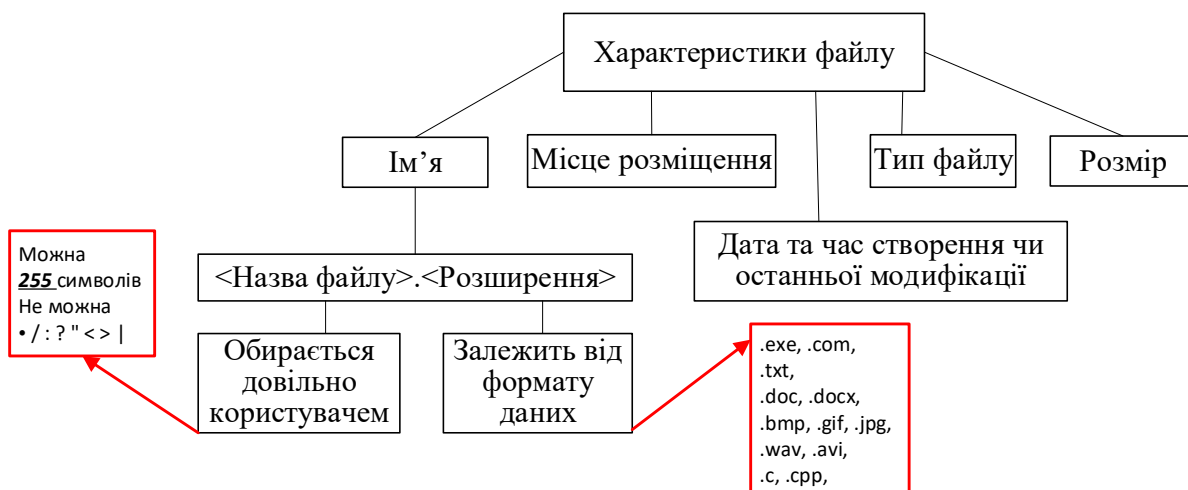


Рисунок 8.4 – Характеристики файлу

Деякі поєднання символів не можна використовувати як імена файлів, оскільки операційна система використовує їх для позначення пристроїв. Імена пристроїв дозволяють при заданні команд ОС здійснити введення і виведення інформації не тільки з файлами на дисках, але й з різними пристроями комп'ютера, що іноді буває досить зручно.

Зарезервовані імена файлів:

- PRN – принтер;
- LPT1-LPT4 – пристрої, що підключаються до паралельних портів 1-4 (переважно це принтери);
- COM1-COM4 – пристрої, що підключаються до послідовних портів 1-4;
- CON при введенні – клавіатура, при виведенні – екран;

NUL – «порожній» пристрій; всі операції введення/виведення для цього пристрою ігноруються.

Імена файлів реєструються на дисках у **каталогах**. В операційній системі Windows каталоги називаються папками.

Каталог – це спеціальне місце на диску, в якому зберігаються імена файлів, відомості про розмір, дату і час створення або останньої модифікації, атрибути файлів.

Якщо у каталозі зберігається ім'я файлу, то кажуть, що цей файл міститься у даному каталозі. На кожному диску може бути декілька каталогів. У кожному каталозі може бути кілька файлів, але кожен файл реєструється тільки в одному каталозі.

Всі каталоги, крім кореневого, є файлами спеціального вигляду. Кожен каталог має ім'я і він може бути зареєстрованим в іншому каталозі.

Поточний каталог – це каталог, з яким користувач працює у даний момент часу.

Шлях – це послідовність імен каталогів або «..», які розділено символами «\».

Шлях задає **маршрут** від поточного каталогу, або від кореневого каталогу, до того каталогу, в якому знаходиться потрібний файл. Якщо шлях починається з символу «\», то маршрут обчислюється від кореневого каталогу диска, інакше – від поточного каталогу. Кожне ім'я каталогу у шляху відповідає входженню у підкаталог з таким самим іменем. Символ «..» відповідає входженню у надкаталог.

Повне ім'я файлу має такий вигляд: ім'я_диска: шлях\ім'я файлу

Шаблон імені файлу – це деякий набір символів, якому мають відповідати один або декілька файлів. Застосовується при роботі утиліт або команд, які оперують файлами.

У багатьох командах можливе використання символів * і ? для позначення групи файлів.

* означає довільне число довільних символів.

? означає довільний один символ або відсутність символу.

Приклад 8.1 Позначення

*.txt вказує на всі файли у поточному каталозі, які мають розширення .txt;

ab*.* – всі файли, ім'я яких починається з ab;

. – всі файли;

ab???.bat – всі файли з розширенням .bat, ім'я яких починається на ab і складається з п'яти символів.

Для кожного файлу у каталозі зберігаються його атрибути. Використовуються такі **4 атрибути файлів**:

- read-only – тільки для читання, забороняє вносити зміни у файл;
- hidden – прихований, для системних файлів;
- system – системний, для системних файлів;
- archive – архівувати, використовується програмами резервного копіювання

Ієрархія каталогів може бути деревом або мережею. Каталоги утворюють **дерево**, якщо файлові дозволено входити тільки в один каталог, і **мережу** – якщо файл може входити одразу в декілька каталогів. В операційній системі **Windows** каталоги утворюють деревоподібну структуру, а в **UNIX** – мережеву. Як і будь-який інший файл, каталог має символічне ім'я та однозначно ідентифікується складеним ім'ям, яке містить ланцюжок символічних імен усіх каталогів, крізь які проходить шлях від кореня до даного каталогу.

8.4 Файлова система. Логічна організація файлу

Файлова система – це набір домовленостей, що визначають організацію даних на носіях інформації, правила збереження файлів та каталогів на диску, які відомості зберігаються про файли та каталоги, яким має бути формат каталогів та службової інформації, .

Для використання дисків, які розмічено за допомогою деякої файлової системи, операційна система або спеціальна програма повинна підтримувати цю файлову систему.

Призначення файлової системи: забезпечити користувачеві зручний інтерфейс при роботі з даними, що містяться на диску, а також забезпечити сумісне використання файлів декількома користувачами або процесами.

Основними задачами файлової системи є:

- зберігання інформації;
- структурування файлів;
- забезпечення прав доступу до даних;
- захист даних;
- створення однорідного середовища для різних носіїв інформації.

Логічна організація – це спосіб представлення файлу у вигляді певних логічних записів.

Логічний запис – це найменший елемент даних, якими може оперувати програміст при обміні із зовнішніми пристроями. Навіть якщо фізичний обмін з пристроєм здійснюється великими одиницями, операційна система забезпечує доступ до окремої логічного запису.

Існують декілька схем логічної організації файлів. Записи можуть бути фіксованою або змінною довжини, можуть бути розташовані послідовно або з використанням індексних таблиць.

Файлова система FAT.

Систему було розроблено ще на початку 80-х, ця файлова система досить примітивна, оскільки вона була створена для зберігання даних на дискетах. Ця файлова система називається FAT, оскільки найважливішою в ній є таблиця розміщення файлів на диску, по-англійськи – file allocation table, скорочено FAT. Ця таблиця містить інформацію про те, які ділянки (кластери) диску вільні, і про ланцюги кластерів, що утворюють файли і каталоги.

У файловій системі FAT імена файлів і каталогів повинні складатися не більше як із 8 символів, плюс три символи в розширенні імені.

Недоліки: підтримує лише короткі імена файлів та каталогів, малопродуктивна для великих дисків та непристосована для багатозадачної роботи.

Файлова система NTFS.

Цю систему почали використовувати у 1993 році. NTFS розглядає файли як об'єкти зі своїми властивостями. NTFS дозволяє стискати окремі файли та каталоги для економії простору.

NTFS має вбудовані можливості розмежовувати доступ до даних для різних користувачів і груп користувачів, а також призначати квоти (обмеження на максимальний обсяг дискового простору, що займають ті чи інші користувачі). NTFS використовує систему журналювання для підвищення надійності файлової системи. У файловій системі NTFS відсутнє розділення на атрибути.

Переваги: робота з великими дисками, маленький розмір кластеру, теоретично може працювати з дисками розміром до 17 ТБ, стабільність.

Файлові системи UNIX.

UNIX System V реалізує механізм віртуальної файлової системи VFS, що дозволяє ядру системи підтримувати декілька файлових систем (рис. 8.5).

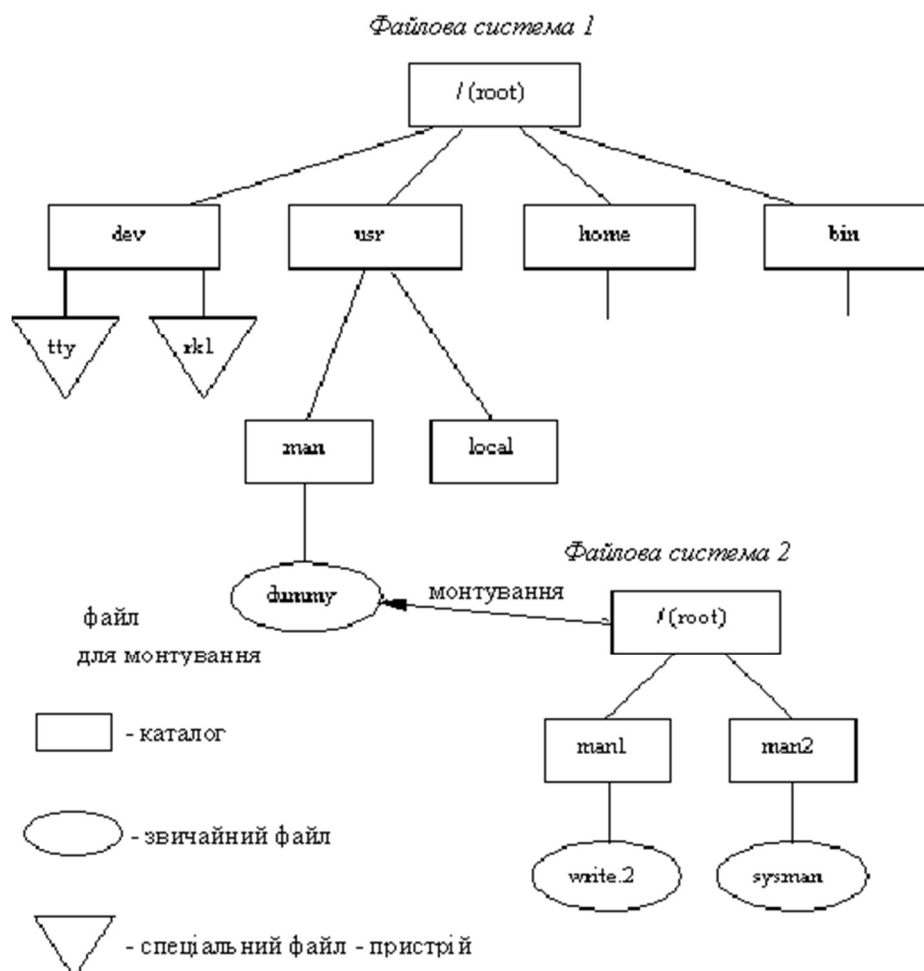


Рисунок 8.5 – Традиційна файлова система s5

Файлова система *s5* має ієрархічну структуру, у якій рівні створюються за рахунок каталогів, що містять інформацію про файли більш низького рівня. Каталог самого верхнього рівня називається кореневим і має ім'я *root*. Ієрархічна структура зручна для багатокористувацької роботи: кожен користувач локалізується у своєму каталозі і разом з тим усі файли в системі логічно зв'язані. Кореневий каталог файлової системи завжди міститься на системному пристрої (диск, що має таку ознаку). Інші файли можуть міститися на цьому чи іншому пристрої. Для зв'язку ієрархій файлів, що розташовані на різних носіях, застосовується монтування файлової системи, яке виконується системним викликом *mount*.

В **UNIX** для файлу існує **три типи імені** – коротке, повне і відносне.

- **Коротке ім'я** ідентифікує файл у межах одного каталогу. Воно може складатися не більш ніж з 14 символів і містити так називаний суфікс, відокремлюваний крапкою.

- **Повне ім'я** однозначне визначає файл. Воно складається з ланцюжка імен каталогів, через які проходить маршрут від кореневого каталогу до даного файлу. Імена каталогів розділяються символами «/», при цьому ім'я кореневого каталогу не вказується, наприклад, **/mnt/rk2/test.c**, де **mnt** і **rk2** – імена каталогів, а **test.c** – ім'я файлу. Кожному повному імені в ОС відповідає лише один файл, однак файл може мати кілька різних імен, тому що посилання на той самий файл можуть міститися в різних каталогах (тверді зв'язки).

- **Відносне ім'я** файлу зв'язане з поняттям «поточний каталог», тобто каталог, ім'я якого задавати не потрібно, тому що воно мається на увазі за замовчуванням. Ім'я файлу щодо поточного каталогу називається відносним. Воно являє собою ланцюжок імен каталогів, через які проходить маршрут від поточного каталогу до даного файлу. Відносне ім'я на відміну від повного не починається із символу «/». Так, якщо в попередньому прикладі прийняти за поточний каталог **/mnt**, те відносне ім'я файлу **test.c** буде **rk2/test.c**.

Файлова система AdvFS – файлова система, для Unix-подібної операційної системи OSF/1 (пізніше перейменована в Digital UNIX/Tru64 UNIX).

Особливості:

- журналювання для швидкого відновлення у випадку відмови;
- можливість відновлення видалених файлів;
- висока продуктивність;
- динамічна структура, що дозволяє адміністратору керувати системою «на льоту»
- можливість створення «знімків» файлової системи «на льоту»
- здійснення дефрагментації під час активності користувачів.

8.5 Фізична організація файлу

Фізична організація файлу – це принципи розміщення файлів, каталогів і системної інформації на реальному пристрої (диску).

Основні критерії ефективності фізичної організації файлів:

- швидкість доступу до даних;
- обсяг адресної інформації файлу;
- ступінь фрагментування дискового простору;
- максимально можливий розмір файлу.

Фрагментація – це наявність великого числа несуміжних ділянок вільної пам'яті дуже маленького розміру. Ці фрагменти настільки маленькі, що жодна з нових програм, не може поміститися у жодній з ділянок, хоч і сумарний обсяг фрагментів може перевищувати необхідний обсяг пам'яті.

Існує декілька **способів фізичної організації файлів**:

1. **Безперервне розміщення**, при якому файлу надається послідовність кластерів диску, що утворюють суцільну ділянку дискової пам'яті.

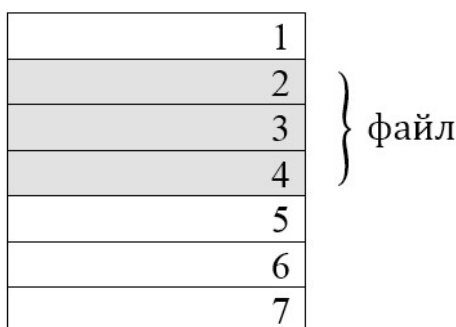


Рисунок 8.6 – Безперервне розміщення файлу

Переваги безперервного розміщення:

- висока швидкість доступу;
- відсутність фрагментації на рівні файлу;
- мінімальний обсяг адресної інформації, достатньо зберігати тільки номер першого кластера та розмір файлу.

Недоліки:

- неможливо визначити якого розміру повинна бути суцільна область, що виділяється файлу, тому що різні файли мають різні розміри.

2. **Розміщення файлу у вигляді пов'язаного списку кластерів дискової пам'яті**, при такому способі на початку кожного кластера міститься покажчик на наступний кластер.

Переваги:

- адресна інформація мінімальна, тому розташування задається тільки номером першого кластера;
- фрагментація на рівні кластера відсутня.

Недоліки:

- складність організації доступу до довільної ділянки файлу;
- кількість даних файлу у кластері не дорівнює степеню двійки, а багато програм читають дані саме кластерами.

3. **Використання пов'язаного списку індексів**. Цей спосіб є модифікацією попереднього, файлу теж виділяється пам'ять у вигляді пов'язаного списку кластерів, номер першого кластера запам'ятовується у записі

каталогу, де зберігаються характеристики файлу, інша адресна інформація відділена від кластерів файлів, з кожним кластером диска пов'язаний індекс, ці індекси розташовуються в окремому образі диска, коли пам'ять вільна, то всі індекси мають нульові значення.

Якщо деякий кластер N призначено деякому файлу, то індекс цього кластера стає рівним або номеру наступного кластера даного файлу, або приймає спеціальне значення, якщо кластер є останнім у файлі.



Рисунок 8.7 – Використання пов'язаного списку індексів

Переваги:

- мінімальна адресна інформація;
- відсутність фрагментації на рівні кластера;
- відсутність проблем при зміні розміру файлу;
- вільний доступ до довільного кластеру файлу.

Недоліки: фрагментація на рівні файлів (файл може розбиватись на несуміжні фрагменти).

4. Перерахування номерів кластерів.

Переваги:

- висока швидкість доступу до довільного кластеру файлу;
- відсутність фрагментації на рівні кластера.

Недоліки:

- довжина адреси залежить від розміру файлів.

Отже, на фізичному рівні файлова система визначає номер фізичного блока, який містить потрібний логічний запис, і зсув логічного запису у фізичному блоці. Для розв'язання цієї задачі використовуються результати роботи логічного рівня – зсув логічного запису у файлі, адреса файлу на зовнішньому пристрої, а також відомості про фізичну організацію файлу.

Після визначення номера фізичного блока, файлова система звертається до системи введення/виведення для виконання операції обміну із зовнішнім пристроєм. У відповідь на цей запит до буфера файлової системи буде передано потрібний блок, у якому обирається потрібний логічний запис.

8.6 Загальна модель файлової системи

Будь-яку файлову систему можна представити багаторівневою моделлю (рис. 8.8), в якій кожен рівень надає деякий інтерфейс більш вищому рівню, а сам у свою чергу для виконання своєї роботи використовує інтерфейс більш нижчого рівня.

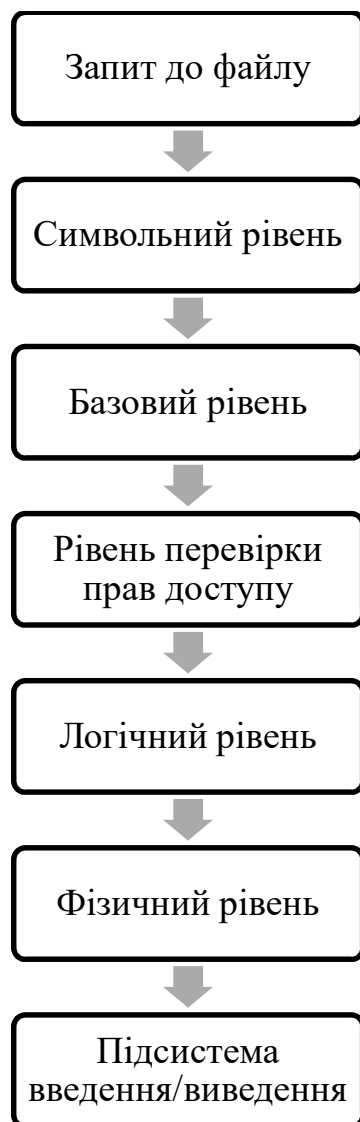


Рисунок 8.8 – Загальна модель файлової системи

Завдання *символьного рівня* є визначення за символьним іменем файлу його унікального імені.

На *базовому рівні* за унікальним іменем файлу визначаються його характеристики: права доступу, розмір, адресу та ін. При відкритті файлу його характеристики переміщуються з диска в оперативну пам'ять.

Наступний рівень – *перевірка прав доступу* до файлу. Для цього порівнюються повноваження користувача або процесу, що запитує файл, зі списком дозволених видів доступу до файлу.

На *логічному рівні* визначаються координати логічного запису, що запитується, у файлі, при цьому фізичне розташування файлу представляється у

вигляді безперервної послідовності байт. Алгоритм роботи даного рівня залежить від логічної організації файлів.

На *фізичному рівні* файлова система визначає номер фізичного блока, який містить необхідний логічний запис, і зсув логічного запису у фізичному блоці, і після визначення номера фізичного блока, файлова система звертається до системи введення/виведення для виконання операції обміну з пристроєм; у відповідь на цей запит у буфер файлової системи передається потрібний блок, в якому на підставі отриманого при роботі фізичного рівня зсуву, обирається необхідний логічний запис.

Корисні посилання

1. Організація файлової системи у Linux.
URL : https://linuxguide.rozh2sch.org.ua/#_організація_файлової_системи_у_linux
2. Файлові системи персональних комп'ютерів.
URL : <http://intranet.tdmu.edu.ua/data/kafedra/theacher/medkat/marcenuk/Українська/Лекції/BasicCourse/Filesystems.htm>

Тестові завдання

1. Жорсткий диск розбивають на логічні диски для:
 - а) зручності;
 - б) дефрагментації;
 - в) фрагментації;
 - г) для швидкої роботи програм.
2. Сукупність доріжок, що розміщено на однаковій відстані від осі диска, називають:
 - а) кластером;
 - б) головною доріжкою;
 - в) циліндром;
 - г) кільцевим шляхом.
3. Розмітку диска під конкретний тип файлової системи виконує процедура:
 - а) низькорівневого форматування;
 - б) дефрагментація;
 - в) логічного форматування;
 - г) немає такої процедури.
4. Відсутній у системній області такий розділ:
 - а) завантажувальний запис;
 - б) область керування розміщенням файлів;
 - в) кореневий каталог;
 - г) розділ форматування.
5. Не існують файли такого типу:
 - а) звичайні;
 - б) спеціальні;

- в) файли-каталоги;
 - г) нормованого.
6. Іменем файлу називається:
 - а) позначення файлу для операційної системи;
 - б) логічний диск, на якому файл розміщено;
 - в) каталог, в якому міститься файл;
 - г) немає такого поняття.
 7. Скільки символів в сучасній ОС може містити ім'я файлу:
 - а) 255;
 - б) 8;
 - в) 16;
 - г) 32.
 8. Оберіть недозволене ім'я файлу:
 - а) авп_346;
 - б) _345;
 - в) \$%.txt;
 - г) -№;%;*(.
 9. Ієрархія каталогів в ОС Windows має:
 - а) деревоподібну структуру;
 - б) мережеву структуру;
 - в) змішану структуру;
 - г) ланцюжкову структуру.
 10. Оберіть файлову систему, яку не підтримує ОС Windows:
 - а) FAT;
 - б) NTFS;
 - в) FAT32;
 - г) EXT4.

Практичні завдання

1. Знайдіть приклади до кожного виду операційних систем.
2. Знайдіть інформацію про SSD диски. Складіть таблицю порівняння HDD дисків з SSD.
3. Знайдіть інформацію про файлову систему ext4. Для якої операційної системи її використовують? Запишіть її переваги та недоліки.

Контрольні запитання

1. Що таке фізичний диск? Що таке логічний диск?
2. Що таке логічне форматування?
3. Що таке файл? Які бувають типи файлів?
4. Які характеристики файлів використовують файлові системи?
5. Чому символ «\» є недопустимим у назві файлу?
6. Що таке розширення файлу?
7. Що таке файлова система? Які файлові системи для ОС Linux ви знаєте?
8. Які способи фізичної організації файлів вам відомі?

Тема 9. Файлові менеджери

Мета: ознайомитись з поняттям, призначенням, основними можливостями файлових менеджерів

План

1. Загальний огляд файлових менеджерів: призначення і можливості.
2. Структура панелі файлового менеджера.
3. Загальні прийоми роботи з файловими менеджерами.
4. Запуск програм і виконання команд операційної системи.
5. Зміна конфігурації (налагодження) файлового менеджера.

9.1 Загальний огляд файлових менеджерів: призначення і можливості

Для роботи з файлами, папками та дисками широке розповсюдження отримали програми-оболонки. Вони дозволяють у зручному режимі виконувати популярні операції: перегляд вмісту каталогу на дисках, перехід з одного каталогу в інший, копіювати, переміщувати та видаляти файли, запускати програми та ін.; і надають широкий набір додаткових можливостей.

Вбудовані засоби операційної системи (наприклад *Провідник* у Windows рис. 9.1) полегшують процес управління файлами та папками, але є недостатньо зручними для використання. Тому, на сьогоднішній день створено багато різних програм, що значно спрощують виконання основних файлових операцій. Такі програми називають *файловими менеджерами*.

Файлові менеджери – програми, що забезпечують зручний доступ до файлів і папок.

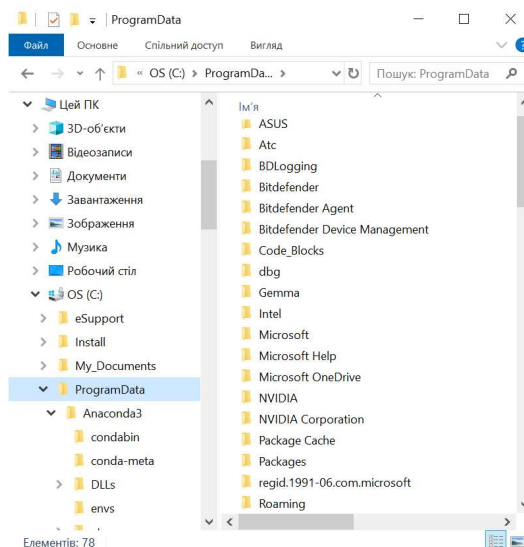


Рисунок 9.1 – Провідник у Windows

Призначення файлових менеджерів: прискорення та полегшення виконання файлових операцій (копіювання, переміщення, перейменування, перегляду, видалення)

Файлових менеджерів дуже багато, але найчастіше користуються такими програмами: Far Manager, Norton Commander, Total Commander, Midnight Commander, Volkov Commander, Explorer++ та ін. (рис. 9.2).

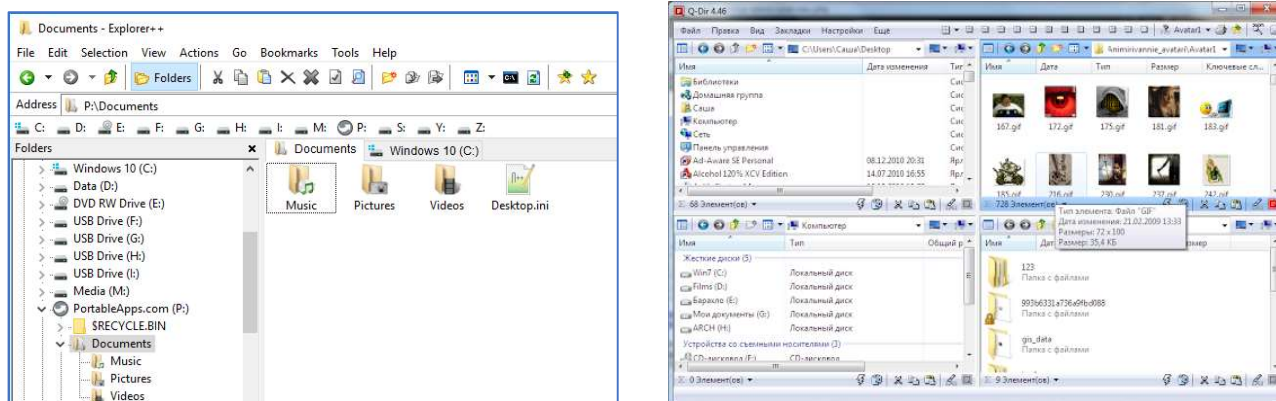


Рисунок 9.2 – Зовнішній вигляд файлових менеджерів Explorer++ та Q-Dir відповідно

Ці програми відрізняються графічним інтерфейсом, функціональними можливостями, обсягом місця, що займають на диску, операційною системою, у середовищі якої працюють. Прийоми роботи у цих програмах майже однакові та відрізняються незначно. Більшість клавіатурних комбінацій може використовуватись у всіх програмах. Функціональні можливості кожного з них істотно розширюються за допомогою зовнішніх модулів – **плагінів**.

Основні можливості файлових менеджерів:

- виконання операцій з файлами: створення, відкриття, редагування, переміщення, перейменування, копіювання, видалення, зміна атрибутів та властивостей, пошук файлів та призначення прав без набору відповідних команд з клавіатури;
- переглядати і редагувати текстові файли;
- робота з мережею;
- робота з дисками;
- робота з архівами;
- резервне копіювання;
- керування принтерами;
- виконання команд операційної системи;
- наочне зображення на екрані монітора вмісту файлів та каталогів на диску.

9.2 Структура панелі файлового менеджера

Після завантаження програми на екрані з'являються дві панелі, однакові у зовнішньому вигляді – оточене рамкою вікно. Панель, на якій знаходиться маркер – поточна. Зміна поточної панелі здійснюється за допомогою клавіші *Tab*.

У заголовку панелі (рис. 9.3) відображається *шлях до поточного каталогу*, зміст якого відображується на панелі. При зміні каталогу змінюється і заголовок

панелі. На панелях відображається вміст поточного каталогу. Внизу панелей розміщено *рядок міні-статусу*, в якому надається інформація про позначений файл чи каталог. Для файлу відображається ім'я, розмір у байтах, дата та час утворення.

Під панелями знаходиться *командний рядок* для введення команд з клавіатури. Цей рядок завжди починається із запрошення, в якому зазвичай міститься шлях до поточного каталогу і яке зберігається увесь час роботи.

Ще нижче розміщується *рядок статусу*, у якому міститься інформація про поточний статус *функціональних клавіш*. Їх призначено для прискорення виконання команд, які найчастіше використовують. Призначення цих клавіш змінюється відповідно режиму роботи програми. Кнопка **Alt** змінює призначення відповідно до потреб користувача. Взагалі у всіх файл-менеджерах призначення функціональних клавіш однакове, хоча можуть бути певні відмінності.

Одним із найважливіших елементів вікна є командне меню, яке викликається за допомогою клавіші **F9**. Це меню є списком деяких об'єктів, між якими потрібно зробити вибір, встановивши маркер на відповідний пункт і натиснути клавішу *Enter*. Після чого з'являється меню, у якому відображено команди.

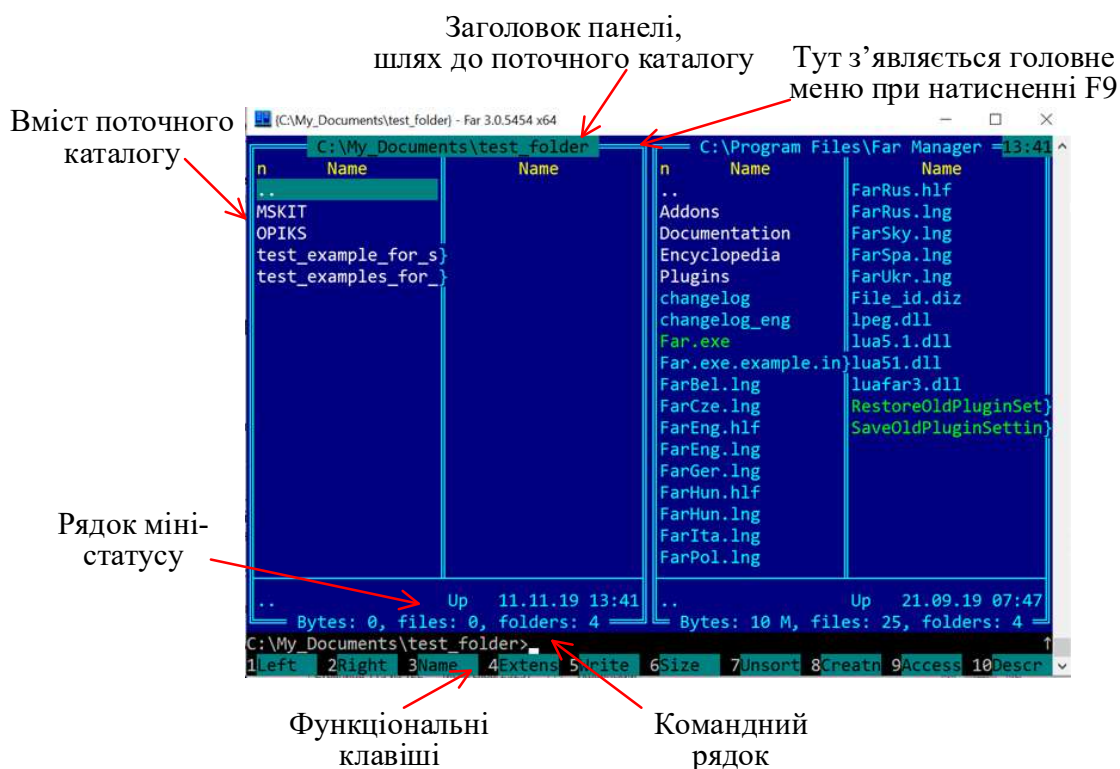


Рисунок 9.3 – Структура вікна файлового менеджера

Зміст командного меню може змінюватись для різних файлових менеджерів, але у всіх є: *Files, Right, Left, Options, Commands*.

Меню *Right, Left* містять команди керування зовнішнім виглядом панелей. Наприклад, сортування об'єктів панелі, дерево папок, вмикання та вимикання панелей та ін.

Меню *Files* містить команди для роботи з файлами та папками (майже дублюють команди функціональних клавіш): архівація файлів; встановлення атрибутів; позначення групи файлів та ін.

Меню *Commands* – спеціальні команди для: виклику дерева каталогів, пошуку файлів, порівняння файлів поточного каталогу та ін.

Меню *Options* (*Параметри, Конфігурація*) – команди налагодження програми: встановлення опцій екрану; встановлення вигляду панелей; встановлення запрошення; опції текстового редактора та ін.

9.3 Загальні прийоми роботи файлових менеджерів

Керування панелями:

Tab – перехід з однієї панелі на сусідню

Alt+F2 – виведення на правій панелі вмісту іншого диска

Alt+F1 – виведення на лівій панелі вмісту іншого диска

Таблиця 9.1 – Призначення функціональних клавіш

F1	help	довідка по роботі з програмою
F3	View	перегляд файлів без редагування
F4	Edit	редагування файлів
F5	Copy	копіювання файлів
F6	RenMove	перейменування, переміщення каталогу
F7	MkDir	створення каталогу
F8	Delete	вилучення файлу чи каталогу
F9	PullDn	вихід у командне меню
F10	Quit	вихід з програми

Призначення функціональних клавіш змінюється при натисненні клавіш *Alt* або *Ctrl*.

9.4 Запуск програм і виконання команд операційної системи

Файлові менеджери мають у своєму арсеналі потужні інструменти для написання та компіляції програм. Як приклад розглянемо програму FAR Manager. Для того, щоб **створити програму**, наприклад, на мові програмування C++ потрібно спочатку надати відповідні **асоціації** для файлів *.cpp та *.exe.

Отже, викликаємо меню F9. Оберіть *Команди/ Асоціації файлів*. Натисніть *Enter* (рис. 9.4).

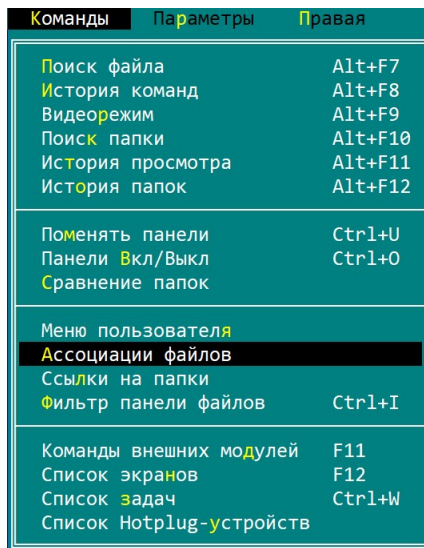


Рисунок 9.4 – Меню налаштування асоціацій файлів

Для переходу до меню, де прописується шлях до компілятора та налагоджувача, потрібно натиснути клавішу *Insert*. Далі для файлів *.cpp та *.exe потрібно вказати повний шлях до програми компілятора та налагоджувача відповідно (рис. 9.5).

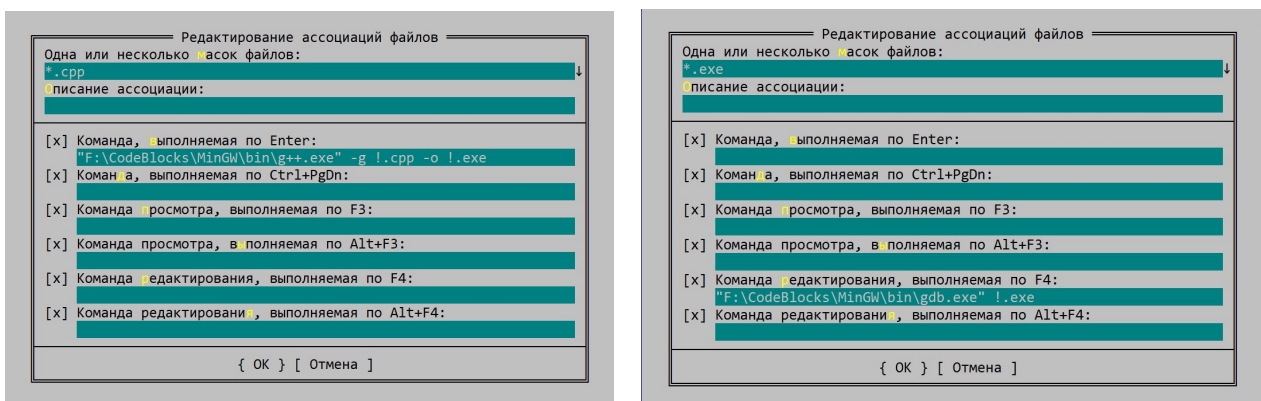


Рисунок 9.5 – Шлях до програми компілятора та налагоджувача відповідно

У вікно до *.cpp потрібно написати:

«шлях до файлу g++.exe» -g !.cpp -o !.exe

У вікно до *.exe потрібно написати:

«шлях до файлу dbg.exe» !.exe

Записи потрібно робити акуратно, дотримуючись синтаксису, особливо варто звернути увагу на наявність пробілів.

Важливо! **Обов'язково** потрібно зберегти зміни, інакше їх буде обнулено після закриття Far (Параметри/Зберегти параметри або Shift+F9).

Після того, як попередні налаштування зроблено, потрібно створити текстовий файл (Shift+F4) з відповідним розширенням (наприклад, testfar.cpp). У файлі, що відкрився, створити програму, зберігаючи синтаксис мови програмування (рис. 9.6). Для налаштування параметрів редактора можна скористатись меню *Параметри*.


```
F:\...ocks\testfar.cpp 1251 Стр 7/7 Кол 40 С 40 22:00
#include <iostream>

int main ()
{
    std::cout << "Hello world"<<std::endl;
    return 0;
}
```

Рисунок 9.6 – Код програми

Далі закрити редактор (F4), у переліку файлів знайти потрібний і натиснути *Enter*. Консольне вікно швидко з’явиться і зникне, для його перегляду потрібно натиснути Ctrl+O (рис. 9.7). На екрані з’явилось таке повідомлення:

```
F:\CodeBlocks>"F:\CodeBlocks\MinGW\bin\g++.exe" -g testfar.cpp -o testfar.exe
```

Рисунок 9.7 – Успішна компіляція програми

Якщо під час компіляції виявлено помилки, то необхідно файл з кодом відкрити для редагування, зробити відповідні правки, зберегти зміни та повторити процес.

Якщо виникає помилка подібна до зображеної на рисунку 9.8, то шлях до відповідної бібліотеки потрібно записати у каталозі *Path* змінних оточення діалогового вікна *Цей комп’ютер*, після чого потрібно закрити файловий менеджер, а потім запустити його знову.

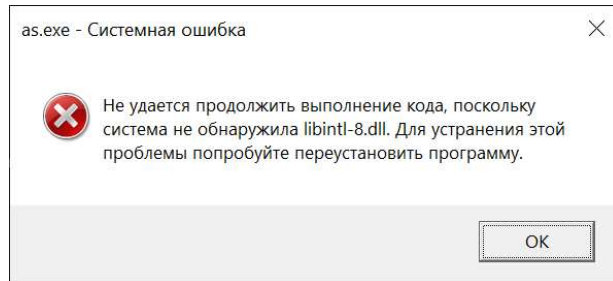


Рисунок 9.8 – Системна помилка

Якщо помилок немає, то поруч з файлом *.cpp з’являється файл *.exe (рис. 9.9).

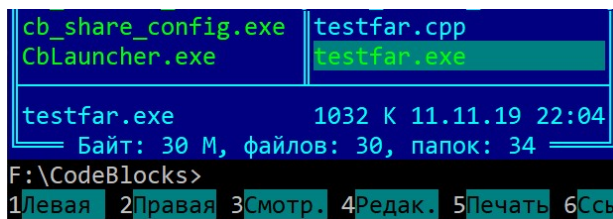


Рисунок 9.9 – Виконуваний файл testfar.exe

Нарешті можна побачити результат роботи програми на екрані. Для цього потрібно обрати відповідний файл та натиснути *Enter* (рис. 9.10).

```
F:\CodeBlocks>testfar.exe
Hello world
```

Рисунок 9.10 – Запуск файлу testfar.exe

Так само можна запускати будь-яке програмне забезпечення. На рисунку 9.4 продемонстровано наявність командного рядка, отже є можливість скористатись усім командним функціоналом операційної системи.

9.5 Зміна конфігурації файлового менеджера

Команди налагодження програми можна розділити на:

- **команди налагодження панелей** – зазвичай містяться у меню *Ліва (Права)* та передбачають встановлення:
 - способу відображення об'єктів панелі (короткий, повний, дерево та ін.);
 - способу сортування (послідовності відображення);
 - зміни диску.

Відображення прихованих файлів можна забезпечити за допомогою команди *Показати приховані файли (Параметри/Налаштування панелі/Показати приховані і системні файли)*.

- **команди налагодження інтерфейсу** – містяться у меню *Параметри/Налаштування інтерфейсу*: визначити параметри екрана; вказати, які файли повинно бути відображено; задати параметри миші та принтера.

Більш «глибоке» налаштування відбувається за допомогою виклику меню *far:config* безпосередньо з командного рядка менеджера.

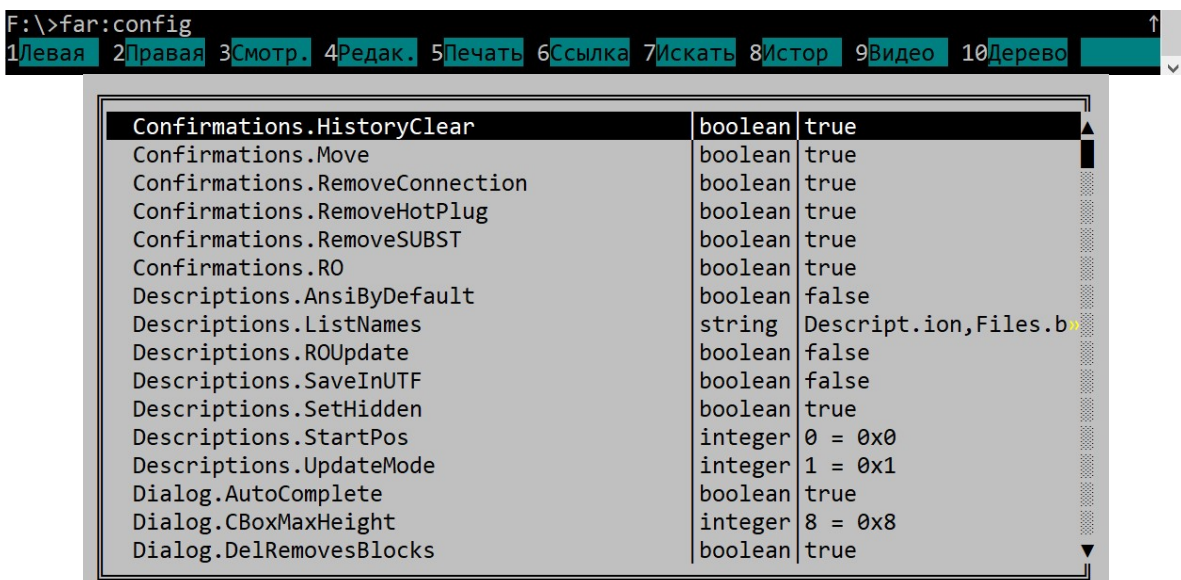


Рисунок 9.11 – Параметри конфігурації Far

Корисні посилання

1. Комбінації клавіш Far. URL : <http://www.ptfmva.ru/upload/iblock/f6b/combinations-far.pdf>
2. 10 корисних функцій Far Manager. URL : <https://lifehacker.ru/15-poleznyx-funkcij-far-manager/>
3. Керівництво зі встановлення та налаштування Far Manager для програмування на C++ та Python. URL : http://acm.petrsvu.ru/site/study/first_year/vclearn/far_and_mingw/
4. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Методи та засоби комп'ютерних інформаційних технологій». URL : https://moodle.znu.edu.ua/pluginfile.php?file=/337580/mod_resource/content/2/Metod_rekomend_MSKiT.pdf
5. Методичні вказівки до виконання лабораторних робіт з дисципліни «Прикладне програмне забезпечення» URL : http://eir.zntu.edu.ua/bitstream/123456789/436/1/%D0%9F%D0%9F%D0%97_FAR_2015.pdf

Тестові завдання

1. Провідник у Windows – це:
 - а) засіб для перегляду конфігурації системи;
 - б) файловий менеджер;
 - в) текстовий редактор;
 - г) табличний процесор Linux Mint.
2. Файловий менеджер забезпечує:
 - а) зручні засоби для налаштування програмного забезпечення;
 - б) зручний доступ до файлів і папок;
 - в) засіб фізичної організації файлу;
 - г) засіб логічної організації файлу.
3. Не є можливістю файлового менеджера:
 - а) робота з архівами;
 - б) виконання команд операційної системи;
 - в) робота з мережею;
 - г) перегляд та редагування текстових файлів.
4. Головне меню викликається натисненням клавіші:
 - а) F9;
 - б) Alt+PrintScreen;
 - в) Shift+Ins;
 - г) Shift+ F9.
5. Переміститися на іншу панель файлового менеджера можна натисненням клавіші:
 - а) Shift;
 - б) Caps Lock;
 - в) Tab;
 - г) Ctrl+Shift.

6. Змінити диск на лівій панелі можна комбінацією клавіш:
 - а) Shift;
 - б) Alt+F1;
 - в) F9;
 - г) Shift+Ins.
7. Відобразити дерево папок деякого каталогу можна натисненням клавіш:
 - а) Ctrl+T;
 - б) Ctrl+O;
 - в) Ctrl+R;
 - г) Ctrl+Q.
8. Створити текстовий файл можна комбінацією:
 - а) Shift+F4;
 - б) Shift+ F9;
 - в) Alt+ Ctrl;
 - г) Shift.
9. Запустити деяку програму у Far можна таким чином:
 - а) натиснути Enter на файлі з розширенням .exe;
 - б) натиснути F4 на файлі з розширенням .exe;
 - в) натиснути F5 на файлі з розширенням .exe;;
 - г) натиснути F6 на файлі з розширенням .exe;.
10. Для «глибокої» конфігурації Far потрібно викликати меню:
 - а) tree;
 - б) regedit;
 - в) far:config;
 - г) group policy.

Практичні завдання

1. Створіть текстовий файл у Far Manager та скопіюйте до нього текст програми з одномірним масивом згідно свого варіанта. Скомпілюйте та запустіть програму.
2. У меню far:config знайдіть параметр, який дозволяє показувати дерево папок.
3. За допомогою меню асоціацій для файлів *.txt налаштуйте відкриття тільки у Блокноті.

Контрольні запитання

1. Що таке файловий менеджер?
2. Які основні можливості файлових менеджерів вам відомі?
3. Чи можливо змінити атрибути файлів у Far?
4. Яким чином заборонити відображення системних файлів?
5. У чому полягає призначення функціональних клавіш у програмі?
6. Чим відрізняються режими перегляду F3 та редагування файлу F4?

Тема 10. Мова команд операційної системи

Мета: вивчити поняття про командний рядок, основні команди операційної системи, засвоїти основні прийоми роботи з командними інтерпретаторами Windows та Linux.

План

1. Командний рядок. Загальна структура команд операційної системи.
2. Внутрішні та зовнішні команди. Основні правила використання команд операційної системи.
3. Довідкова система командного рядка. Загальносистемні команди операційної системи.
4. Зарезервовані імена пристроїв операційної системи. Введення кількох команд в одному рядку.
5. Порівняння систем команд операційних систем Windows та GNU/Linux.

10.1 Командний рядок. Загальна структура команд операційної системи

«Основним виконавцем» операційної системи є **командний процесор** — інтерпретатор командних рядків.

Командний рядок – це одна з можливостей операційної системи, яка забезпечує введення команд ОС та інших комп'ютерних команд.

Основна перевага командного рядка полягає у тому, що він дозволяє вводити всі команди без участі графічного інтерфейсу, що забезпечує більш швидку роботу та має велику кількість додаткових можливостей, які не можуть бути здійснені у графічному інтерфейсі.

Функції командного рядка:

- приймання та виконання команд, що надходять з клавіатури чи з командного файлу;
- виконання внутрішніх команд операційної системи, які він обслуговує;
- завантаження й виконання зовнішніх команд операційної системи та прикладних програм, що розміщуються у виконуваних файлах.

Основний файл командного рядка – консоль **cmd.exe** переводить команди користувача у зрозумілий системі вигляд, знаходиться у папці зі встановленою ОС: {буква_системного_диска}:\WINDOWS\system32\cmd.exe (рис. 10.1)

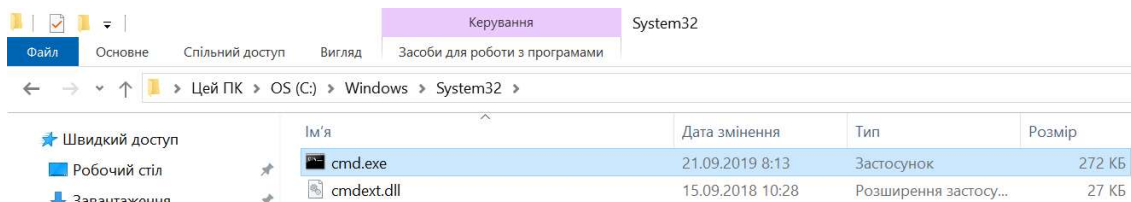


Рисунок 10.1 – Шлях до файлу командного рядка

Запустити командний рядок можна, безпосередньо у папці system32 або за допомогою комбінації клавіш Win+R/cmd. Результатом виконання останньої послідовності дій буде поява вікна, в якому буде запрошення до введення команди.

Для запуску командного рядка в ОС Linux потрібно викликати термінал комбінацією клавіш Alt+Ctrl+T.

Для того, щоб **запустити командний рядок від імені Адміністратора**, потрібно викликати контекстне меню для файлу cmd.exe та обрати пункт *Запустити з правами адміністратора*.

Командний рядок призначений для діалогу користувача з комп'ютером мовою команд операційної системи. **Складається** він із запрошення операційної системи і рядка редагування.

Запрошення операційної системи – це певна текстова інформація (найчастіше позначення поточного диска, каталогу й символу «>»), що передусє рядку редагування і вказує на те, що операційна система запрошує до введення команд. Для введення команди операційної системи необхідно набрати її в командному рядку й натиснути клавішу *Enter*.

Команда операційної системи – основний засіб роботи у середовищі операційної системи, який викликає відповідну їй програму для виконання. Команда набирається з клавіатури у командному рядку, поруч із запрошенням. Наявність в консолі значка > (Windows) або \$ (Linux) свідчить про те, що інтерфейс командного рядка може приймати команди. Іншу важливу інформацію, наприклад, поточний каталог (або розташування), де буде виконуватися команда, також може бути вказано у запрошенні (рис. 10.2).

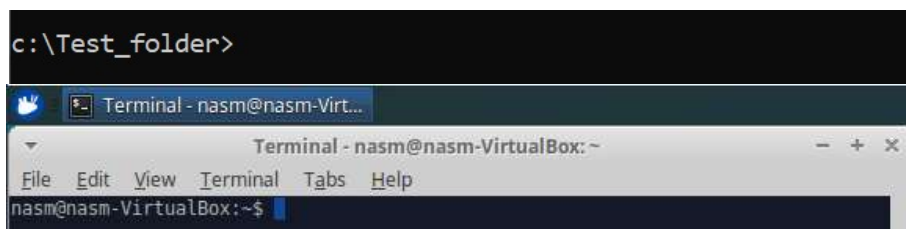


Рисунок 10.2 – Інтерфейс командного рядка у Windows та Linux відповідно

Наприклад, дізнатися версію ОС можливо за допомогою команди **ver** (рис. 10.3)

```
C:\Windows\system32\cmd.exe
c:\Test_folder>ver
Microsoft Windows [Version 10.0.17763.805]
```

Рисунок 10.3 – Інформація про версію ОС

Для того, щоб повторити команду, яку було викликано раніше, можна скористатись стрілками ↑ або ↓.

Для того, щоб **скопіювати деякі записи командного рядка**, потрібно натиснути **Ctrl+M**, після чого курсором миші виділити потрібний текст та скопіювати його у буфер обміну.

З командного рядка можна запустити більшість прикладних програм, для цього достатньо ввести назву відповідного exe-файлу. Наприклад, написавши у консолі `notepad.exe`, запуститься програма *Блокнот*. Звісно, може знадобитись вказати повний шлях до виконуваного файлу.

10.2 Внутрішні та зовнішні команди. Основні правила використання команд операційної системи

Команди операційної системи поділяють на:

- **внутрішні**, які розпізнає і виконує командний процесор (`cmd.exe`). Прикладами таких команд є: **cls** – очищення екрану, **time** – зміна та перегляд поточного часу, **dir** – перегляд вмісту папки.
- **зовнішні**, які оформлено як окремі програми, за замовчуванням знаходяться у тій же папці, що і `cmd.exe`. Наприклад, **xcopy** – копіювання файлів, **at** – запуск програм та команд у певний час.

Команда має ім'я, а також може містити перемикачі й параметри, що відокремлюються від її імені й один від одного пробілами. Параметри визначають об'єкти, на які діє команда, а перемикачі змінюють режим її роботи

У загальному вигляді команду операційної системи можна записати так:

Command [Par1] [Par2]...[/Sw1] [/Sw2]...,

де **Command** – ім'я команди; **Par1**, **Par2** – параметри (аргументи) команди; **Sw1**, **Sw2** – перемикачі або ключі (перед перемикачем повинна стояти риска «/»).

Правила використання конкретної команди називаються її **синтаксисом**. Для опису синтаксису команди застосовують певні означення.

Кутові дужки <...> у синтаксисі команди є ознакою **обов'язкового** використання записаного у них виразу. **Необов'язкові параметри** записують у **квадратних дужках** [...]. Для вибору одного з перелічених параметрів використовують символ альтернативи «|», для продовження за аналогією – три крапки.

При введенні команди ні кутові, ані квадратні дужки не пишуться.

Аргументи задають об'єкти, над якими потрібно виконати дії задані командою (наприклад, специфікація файлу, каталогу). Аргументи у командному

рядку є позиційними і повинні записуватись у послідовності вказаній синтаксисом. Вони відділяються від інших елементів хоча б одним пробілом.

Ключі уточнюють або модифікують дію команди. Кожен ключ починається із слеша (/), а тому пропуск перед ним необов'язковий. Відносний порядок запису ключів у команді у деяких випадках значення не має.

Команду можна коригувати тільки до натиснення клавіші *Enter*. При натисненні *Enter* після набору команди вона вводиться у дію. Можливості операційної системи дозволяють текст команди посилати у буфер одночасно з її набором, а отже – надається можливість її повторення шляхом натиснення призначеної для цього клавіші.

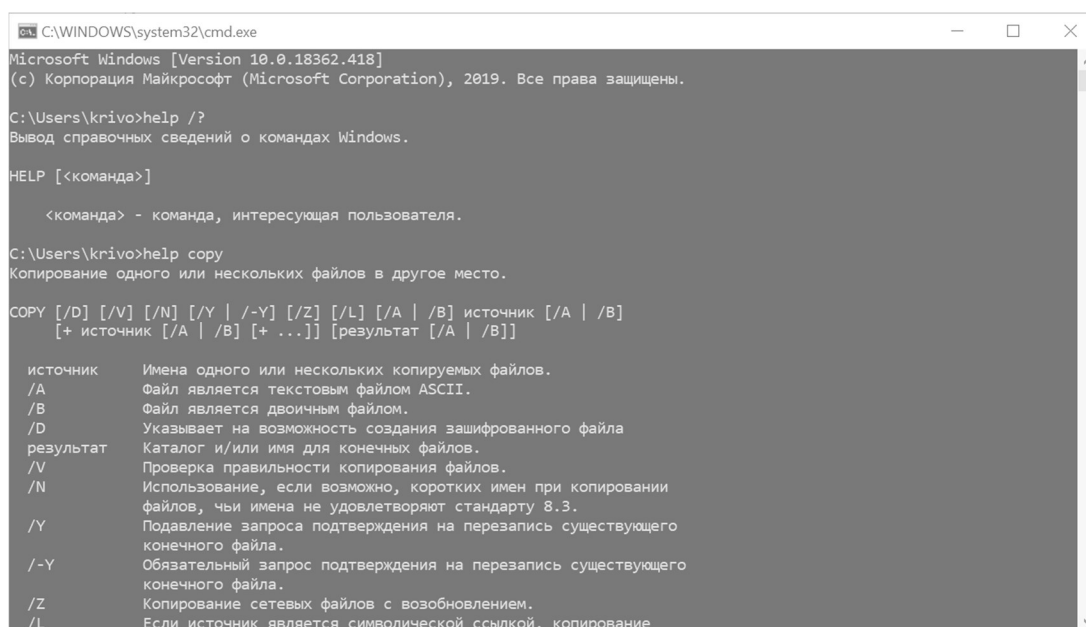
Введення команди.

1. Команди друкуються на клавіатурі латинськими буквами.
2. Ключі або перемикачі відокремлюються від команди пробілами.
3. Команди починають виконуватись після натиснення клавіші *Enter*.
4. Переривання виконання команди або програми здійснюється, зазвичай, натисненням Ctrl+C або Ctrl+Break.

Ці правила актуальні як для роботи в ОС Windows, так і для Linux.

10.3 Довідкова система командного рядка. Загальносистемні команди операційної системи

Довідку про команди операційної системи можна отримати як у командному рядку, так і за допомогою загальної довідкової системи ОС. Команда **help** виводить список усіх команд операційної системи зі стислим описом їх призначення. Щоб отримати інформацію про синтаксис команди, потрібно у командному рядку ввести **help_command** або **command_/?** (рис. 10.4)



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.18362.418]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

C:\Users\krivo>help /?
Вывод справочных сведений о командах Windows.

HELP [<команда>]

<команда> - команда, интересующая пользователя.

C:\Users\krivo>help copy
Копирование одного или нескольких файлов в другое место.

COPY [/D] [/V] [/N] [/Y | /-Y] [/Z] [/L] [/A | /B] источник [/A | /B]
[+ источник [/A | /B] [+ ...]] [результат [/A | /B]]

источник      Имена одного или нескольких копируемых файлов.
/A            Файл является текстовым файлом ASCII.
/B            Файл является двоичным файлом.
/D            Указывает на возможность создания зашифрованного файла
результат     Каталог и/или имя для конечных файлов.
/V            Проверка правильности копирования файлов.
/N            Использование, если возможно, коротких имен при копировании
файлов, чьи имена не удовлетворяют стандарту 8.3.
/Y            Подавление запроса подтверждения на перезапись существующего
конечного файла.
/-Y          Обязательный запрос подтверждения на перезапись существующего
конечного файла.
/Z            Копирование сетевых файлов с возобновлением.
/L            Если источник является символической ссылкой, копирование
```

Рисунок 10.4 – Введення довідки про командів help та copy

За об'єктом дії команди операційної системи поділяють на загальносистемні та команди обслуговування файлів, каталогів, дисків.

Загальносистемні команди ОС.

1. Виведення відомостей про версію операційної системи – **ver**.
2. Виведення і встановлення поточної дати – **date_ [/t|дата]**.

Команда без параметрів виводить поточну дату й запитує, чи потрібно ввести нову дату. Щоб залишити поточну дату без змін, натискають *Enter*. З перемикачем **/T** дата виводиться без запиту про нову дату.

3. Виведення і встановлення системного часу – **time_ [/t|час]**.

Команда без параметрів виводить на екран поточний системний час і запитує, чи потрібно ввести новий час. Щоб залишити поточний час, потрібно натиснути *Enter*. З перемикачем **/t** час виводиться без запиту про новий час.

4. Очищення екрана – **cls**. Видаляє інформацію з екрана монітора.

5. Встановлення режиму перевірки правильності запису файлів на диск – **verify_ [on|off]**. Вмикає (**verify_on**) і вимикає (**verify_off**) режим перевірки або виводить на екран (**verify**) поточний стан режиму.

6. Перенаправлення введення та виведення.

Якщо після команди є знак «>», після якого вказано ім'я файлу чи пристрою, то буде створено файл із зазначеним ім'ям з інформацією, що мала бути виведена командою на екран, або всю цю інформацію буде передано у зазначений пристрій.

Приклад 10.1 Команда **dir c:\test_>_test.dir** створює файл **test.dir**, у який записується список файлів і підкаталогів каталогу **c:\test**.

Якщо файл з таким ім'ям вже існує, тоді вміст файлу буде знищено і туди буде записано нову інформацію. Щоб додати нову інформацію до наявного файлу, потрібно використати знак «>>».

Таблиця 10.1 – Оператори перенаправлення введення/виведення команд

Оператор перенаправлення	Опис
>	Записує дані на виході команди замість вікна консолі у файл або на пристрій, наприклад, на принтер.
<	Зчитує потік вхідних даних команди з файлу, а не з клавіатури.
>>	Додає вихідні дані команди у кінець файлу, не видаляючи при цьому існуючої інформації з файлу.
>&	Зчитує дані на виході одного дескриптора як вхідні дані іншого.
<&	Зчитує вхідні дані одного дескриптора як вихідні дані іншого.
	Зчитує вихідні дані однієї команди та записує їх на вхід іншої команди. Ця процедура відома під назвою «канал»

Приклад 10.2

1. Виведення у файл **тест.txt** списку всіх папок із поточної директорії

```
dir /A:D /B >тест.txt
```

2. Виведення довідки з команди **tasklist** у файл **t_hlp.txt**

```
tasklist /? > t_hlp.txt
```

3. Зміна дати на 12.11.2019

```
echo 12.11.2010 | date
```

В останньому прикладі використано оператор конвеєризації. За його допомогою повідомлення, що виводяться першою командою, можна передавати як вхідні дані для другої команди. У даному випадку як вхідні дані для команди **date**, яка очікує на вході нову дату у форматі **дд.мм.гггг**, використано виведення першої команди **echo**, яка виводить на екран заданий рядок-параметр. Якщо заздалегідь записати нову дату у файл **date.txt**, то зміна дати можлива і такою командою: **date < date.txt**

Оператор конвеєра є найбільш ефективним у комбінації з фільтруючими командами. Команди-фільтри дають можливість сортувати, переглядати та забирати частину вихідної інформації інших команд.

Таблиця 10.2 – Команди-фільтри

Команда	Опис
more	Відображає вміст файлу або команди в одному вікні командного рядка за один раз
find	Пошук вказаних символів у файлах або виведенні команди
findstr	Пошук зразків тексту з використанням регулярних виразів
sort	Сортування файлів або виведення команди за алфавітом

Приклад 10.3 Команда **dir | find «.2019»** відфільтрує у виведенні команди **dir** всі файли та папки, у рядках опису яких зустрічається рядок **.2019** (створено у 2019 році).

10.4 Зарезервовані імена пристроїв операційної системи. Введення кількох команд в одному рядку

В операторах перенаправлення замість імені файлу можна вказувати стандартні позначення пристроїв комп'ютера. У Windows підтримуються такі імена пристроїв: **PRN** – принтер, **LPT1-LPT3** – відповідні паралельні порти, **AUX** – пристрій, який під'єднано до паралельного порту, **COM1-COM3** – відповідні послідовні порти, **CON** – термінал: при введенні це клавіатура, при виведенні – монітор, **NUL** – порожній пристрій, всі операції введення/виведення для нього ігноруються.

Дуже часто розв'язання деякої практичної задачі вимагає виконання декількох команд. У цьому випадку їх об'єднують у спеціальні командні файли, але мова про них буде нижче. Windows надає можливість об'єднати декілька

команд в одному рядку, припускаючи їхнє послідовне виконання, а також, дозволяючи обумовлювати виконання одних команд результатами роботи інших.

Таблиця 10.3 – Оператори умовного виконання команд

Символ	Синтаксис	Визначення
&	<i>команда1 & команда2</i>	Використовується для розділення декількох команд в одному командному рядку. В cmd.exe спочатку виконується перша, а потім – друга. WINDOWS
;	<i>команда1; команда2</i>	Незалежно від того, як виконалася попередня команда, друга команда повинна буде запуститися (послідовне виконання команд). LINUX
&&	<i>команда1 && команда2</i>	Запускає команду, яка йде після символу && , тільки якщо попередню команду було виконано успішно.
 	<i>команда1 команда2</i>	Запускає команду, яка йде після символу , тільки якщо попередню команду було виконано з помилкою.
()	<i>(команда1 & команда2)</i>	Використовується для групування або вкладення команд.

Приклад 10.4 Якщо у корні диска **d** є файли з розширенням **txt**, то скопіювати їх у папку **tmp**, а самі файли видалити

```
dir d:\*.txt && (xcopy d:\*.txt d:\tmp & del d:\*.txt)
```

У цьому прикладі використано групування команд і у випадку успішного виконання команди **dir** (тобто файли із заданою маскою знайдено), буде виконано одразу дві команди – копіювання файлів **xcopy** і видалення файлів **del**.

10.5 Порівняння систем команд операційних систем Windows та GNU/Linux

Як було написано вище робота у командному рядку починається із запрошення. Робота у командному рядку терміналу Linux може виконуватися за допомогою великої кількості команд, більшість з яких мають відповідний аналог в ОС Windows. Робота з опціями та параметрами відбувається аналогічно роботі у Windows (у довідковій системі є інформація про всі команди з відповідними параметрами та ключами).

Команди терміналу Linux чутливі до регістру: **sudo** (команда запуску від імені адміністратора), **SUDO**, **sUdO** – не одне й те ж саме, спрацює тільки перший варіант команди. У більшості випадків у записі команд використовуються малі

літери, але іноді регістр все ж переключється, у разі використання разом із командою опцій і параметрів.

При використанні імен файлів і каталогів також важливо правильно вводити назви **file1** і **File1** – різні файли, навіть, якщо вони знаходяться в одному каталозі.

При роботі у терміналі підтримується автодоповнення, яке автоматично дописує імена файлів і каталогів, назви команд. Якщо натиснути *Tab*, то ім'я файлу, команди або опції допишеться самостійно. У разі коли є, наприклад, кілька файлів, що починаються однаково, автозаповнення продовжить слово лише до останнього однакового у всіх іменах символу. Якщо натиснути два рази клавішу *Tab*, можна переглянути усі можливі варіанти назв.

Наприклад, якщо у домашньому каталозі користувача (/home/teacher) ввести *Зав* і натиснути клавішу *Tab*, автозаповнення спрацює лише для одного слова *Завантаження* (звісно, якщо такий каталог існує), назва якого повністю з'явиться в рядку.

Далі наводиться таблиця відповідності команд у різних операційних системах.

Таблиця 10.4 – Основні команди ОС Windows та Linux

Дія, що виконується	Команда ОС Windows	Команда ОС Linux
Загальні команди		
Виклик довідки	/?, HELP	man, info, --help
Виклик командного інтерпретатора	CMD	bash, csh, sh
Налаштування терміналу параметрів	MODE	stty
Вихід з командного інтерпретатора	EXIT	exit
Пауза у роботі	PAUSE	sleep
Зміна системної дати	DATE	date
Зміна системного часу	TIME	date
Очищення екрана	CLS	clear
Виведення повідомлення на екран	ECHO	echo
Перегляд конфігурації комп'ютера	SYSTEMINFO	lshw, sysinfo
Диспетчер завдань	TASKMGR	top
Робота з файлами і каталогами		
Інформація про поточну директорію	CD	cd
Перегляд змісту каталогу	DIR	dir, ls
Зміна поточної директорії	CD disk:	cd
Зміна поточного диску	CD	cd
Створення каталогу	MKDIR	mkdir
Виклик текстового редактора	NOTEPAD	nano

Дія, що виконується	Команда ОС Windows	Команда ОС Linux
Графічне зображення структури каталогів або шляху	TREE	
Повернутись до батьківської директорії	CD	cd
Видалити директорію, файл	RD, RMDIR	rmdir
Виведення на екран вмісту файлу	TYPE, MORE	cat, less, more
Перейменування одного або кількох файлів	REN, RENAME	mv
Копіювання одного або кількох файлів	COPY, XCOPY, ROBOCOPY	cp, cat
Зміна атрибутів файлу	ATTRIB	chmod
Створення текстового файлу	COPY CON	touch
Видалення файлу	DEL	rm

Звісно у таблиці наводяться тільки назви команд, а всі параметри та додаткові опції (ключі) можна уточнити, скориставшись довідковою системою.

Приклад 10.5 Текстовий файл у консолі Windows можна створити командою **copy con file.txt** – копіювання з пристрою **con** (консоль) у файл **file.txt** відбувається введенням символів безпосередньо з клавіатури у файл. Для завершення введення використовується **Ctrl+Z** або натиснення F6.

Приклад 10.6 Розглянемо два способи створити текстовий файл безпосередньо у терміналі. Напевно, найкоротшою командою для створення файлу в Linux є: **>name**

Створення текстового файлу та перевірка його наявності (команда **ls**) проілюстровано на рисунку 10.5 (а-в).

```

Terminal - nas@nas-VirtualBox: ~
File Edit View Terminal Tabs Help
nas@nas-VirtualBox:~$ >name
nas@nas-VirtualBox:~$ ls
Desktop Downloads Music Pictures Templates
Documents lab1.asm name Public Videos

```

а)

```

Terminal - nas@nas-Virt...
File Edit View Terminal Tabs Help
GNU nano 2.5.3 File: name
Це текстовий файл
Його створено спеціально
для теми "Мова команд операційної системи"

```

в)

```

Terminal - nas@nas-Virt...
File Edit View Terminal Tabs Help
nas@nas-VirtualBox:~$ >name
nas@nas-VirtualBox:~$ ls
Desktop Downloads Music Pictures Templates
Documents lab1.asm name Public Videos
nas@nas-VirtualBox:~$ file name
name: empty
nas@nas-VirtualBox:~$ nano name
nas@nas-VirtualBox:~$ nano name
nas@nas-VirtualBox:~$ nano name
nas@nas-VirtualBox:~$ file name
name: UTF-8 Unicode text
nas@nas-VirtualBox:~$

```

б)

Рисунок 10.5 – Робота з текстовим файлом у консолі Linux Xubuntu

Також для створення текстового файлу можна скористатись командою **touch**.

Приклад 10.7 Графічне зображення структури каталогу F:\CODEBLOCKS\FUNCTION_MASSIV можна отримати за допомогою

команди **tree f:\CodeBlocks\function_massiv**, а додавши опцію **F** додатково отримаємо ще й список файлів кожного каталогу (рисунок 10.6).



```
F:\>tree f:\CodeBlocks\function_massiv
Структура папок тома Storage
Серийный номер тома: DE05-64FA
F:\CODEBLOCKS\FUNCTION_MASSIV
├── bin
│   └── Debug
├── obj
│   └── Debug
```

```
F:\>tree f:\CodeBlocks\function_massiv /F
Структура папок тома Storage
Серийный номер тома: DE05-64FA
F:\CODEBLOCKS\FUNCTION_MASSIV
├── function_massiv.cbp
├── function_massiv.layout
├── main.c
├── bin
│   └── Debug
│       └── function_massiv.exe
├── obj
│   └── Debug
│       └── main.o
```

Рисунок 10.6 – Дерево папок і файлів каталогу

Корисні посилання

1. Методичні рекомендації до виконання лабораторних робіт.
URL : https://moodle.znu.edu.ua/pluginfile.php?file=/337580/mod_resource/content/2/Metod_rekomend_MSKiT.pdf
2. Команди ОС Windows. URL : <https://ab57.ru/cmdlist.html>
3. Команди ОС Linux. URL : <http://blog.sedicomm.com/2017/07/04/komandy-linux-ot-a-do-z-obzor-s-primerami/>
4. Команди для навігації та переміщення по каталогам у Linux (cd, ls, pwd, file). URL : <https://zametkinapolyah.ru/izuchaem-sistemu-linux/komandy-dlya-navigacii.html>

Тестові завдання

1. Назвіть командний інтерпретатор для Windows:
 - а) cmd.exe;
 - б) bash;
 - в) console application;
 - г) notepad.exe.
2. Назвіть командний інтерпретатор для Linux:
 - а) cmd.exe;
 - б) bash;
 - в) console application;
 - г) notepad.exe.
3. Запустити командний рядок у Windows можна комбінацією клавіш:
 - а) Alt+T;
 - б) Ctrl+Alt;
 - в) Win+R/cmd;
 - г) Ctrl+Alt+T.
4. Запустити командний рядок у Linux можна комбінацією клавіш:
 - а) Alt+T;
 - б) Ctrl+Alt;
 - в) Win+R/cmd;

- г) Ctrl+Alt+T.
5. Яким елементом позначаються обов'язкові параметри у синтаксисі команди ОС:
- а) { };
 - б) [];
 - в) < >;
 - г) всі параметри є необов'язковими.
6. Кожен ключ у команді Windows позначається:
- а) /;
 - б) %;
 - в) *;
 - г) №.
7. Переривання виконання команди або програми здійснюється натисненням:
- а) Alt+Shift;
 - б) Ctrl+T;
 - в) Ctrl+Break;
 - г) Ctrl+Z.
8. Довідку для команди **format** можна викликати таким чином:
- а) sos format;
 - б) print format;
 - в) help format;
 - г) command format.
9. У Linux переглянути зміст поточного каталогу можна за допомогою команди:
- а) ls;
 - б) man;
 - в) help;
 - г) file.
10. Якою командою можна створити каталог в ОС Windows та Linux:
- а) mkdir;
 - б) cd;
 - в) ls;
 - г) dir.

Практичні завдання

1. Знайдіть інформацію про 5 внутрішніх команд ОС, перевірте їх роботу.
2. За допомогою команди tree виведіть графічне зображення дерева папок диска D до текстового файлу.
3. Команду із прикладу 10.3 модернізуйте таким чином, щоб результат було виведено до файлу.

Контрольні запитання

1. Що таке командний процесор? Які його функції?
2. Що таке командний рядок?
3. Що означає запрошення операційної системи?
4. Як отримати список команд операційної системи?
5. Як спрямувати виведення повідомлень команд операційної системи у файл?
6. Які елементи (крім команди) можна прописувати у командному рядку?
7. Як отримати довідку про синтаксис певної команди?

Тема 11. Призначення та основні принципи написання командних файлів

Мета: вивчити поняття про командний файл, набути навичок розв'язання типових задач адміністрування операційної системи з використанням командних файлів.

План

1. Поняття про командний файл.
2. Windows: основні команди, створення і запуск командних файлів.

Приклади командних файлів.

3. Linux: створення і запуск командних файлів. Приклади.

11.1 Поняття про командний файл

Командний (пакетний) файл являє собою звичайний текстовий файл з розширенням **bat** (або **cmd**).

Пакетний файл може містити:

- внутрішні команди (команди роботи з файлами, каталогами, команди системного значення);
- зовнішні команди;
- спеціальні команди, які виконуються тільки коли прописані у самому пакетному файлі;
- команди розгалуження (**if**);
- мітки переходу (**goto**);
- звертання до виконуваних файлів і виклик інших пакетних файлів.

На відміну від файлів ***.exe** та ***.com** – командні файли містять не машинні коди, а текст який інтерпретується командним процесором.

Таким чином, пакетний файл має схожість з програмами на алгоритмічних мовах програмування, але для його виконання не потрібно встановлювати середовище програмування або додатковий інтерпретатор.

У найпростішому випадку у командний файл можна записати декілька команд, які будуть послідовно виконуватись.

В ОС **Linux** виконуваним може бути навіть текстовий файл. З точки зору Linux виконуваний файл – це будь-який файл, який позначений, як виконуваний і який Linux зможе запустити на виконання. Це означає, що у кожного файлу є спеціальна властивість-перемикач, яка ніяк не залежить ні від імені, ні від вмісту, ні від розширення, і яка відповідає за виконуваність.

Якщо на файлі встановлено властивість виконуваності, то це програма, яку можна запустити звичайним чином, а якщо не встановлено – то це всього лише текстовий файл. Тобто, зробити файл виконуваним або ж навпаки, прибрати можливість його виконання, можна зміною всього однієї його властивості.

У кожного файлу є **три групи прав доступу** – для користувача-власника, для групи-власника і для всіх інших, кожна група складається у свою чергу з

трьох прав: на читання (r), зміну (w) та виконання (x). Ці атрибути визначають, що система може робити з цим файлом. Право на виконання і є тим самим перемикачем виконуваності.

11.2 Windows: основні команди, створення і запуск командних файлів. Приклади командних файлів

Процес створення та запуск командного файлу містить такі дії:

1. Створити текстовий документ у будь-якому текстовому редакторі. Наприклад, Блокнот.
 2. Ввести бажаний набір команд операційної системи, при потребі додати коментарі, тестові виведення на друк та ін.
 3. Зберегти файл із розширенням **.bat**.
 4. Запустити файл у командному рядку з директорії, де створено файл.
- Для цього у консолі потрібно ввести назву файлу та натиснути *Enter*.

Приклад 11.1 Нехай, наприклад, файл **arc.bat** містить такий набір інструкцій:

```
d:
cd d:\tmp
d:\winrar\rar a arc.rar *.txt
del *.txt
```

Якщо запустити на виконання файл **arc.bat**, то він вкляде в архів **arc.rar** всі текстові файли з папки **tmp**, а потім видалить ці файли. Тепер можна налаштувати (наприклад, за допомогою команди **at**) щоденний або щотижневий запуск цього файлу у визначений час, внаслідок чого папка **d:\tmp** не буде переповнюватись текстовими файлами.

Таким чином, командні файли є зручним засобом автоматизації виконання рутинних задач адміністрування. Але можливості командних файлів ширше, ніж простий перерахунок послідовних команд.

Також слід додати про можливість додання коментарів у тексти скриптів командних файлів. Рядки коментарів повинні починатись з ключового слова **REM** і дозволяють природньою мовою додати пояснення у текст скрипту.

Приклад 11.2 Додамо коментар до програми з прикладу 11.1.

```
REM Додамо в архів arc.bat всі текстові файли
d:\winrar\rar a arc.rar *.txt
```

Для редагування командного файлу потрібно викликати для нього контекстне меню та обрати пункт *Редагувати*.

Приклад 11.3 Командний файл, який шукає у поточному каталозі файл з назвою **laba** і з будь-яким розширенням. Якщо такий файл існує, то у *Блокноті* створюється файл **laba1**, у кінець якого дописується поточна дата та час. Якщо такий файл не знайдено, то на екран виводиться відповідне повідомлення.

```
IF EXIST laba.* GOTO MI
ECHO laba not found.
GOTO END
:MI
```

```

ECHO laba found
if exist laba.txt (
    start notepad.exe labal.txt
    date /T >> laba.txt
    time /T >> laba.txt
)
:END

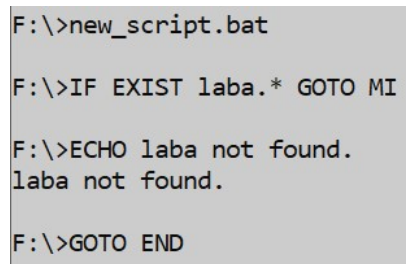
```

Послідовне виконання у пакетному файлі можна перервати декількома способами. По-перше, можна використати команду переходу **goto мітка**. Ця команда в умовах слабо структурованих файлів використовується досить часто. Мітка, на яку здійснюється перехід по цій команді оформлюється таким чином:

:мітка

Очевидно, що перехід повинно бути обумовлено командою **if**, інакше не виконаються команди, які містяться між командою переходу та міткою у випадку переходу вперед або відбудеться зациклювання при переході назад. Як мітку можна використати рядок **:MI**, яка передає керування у кінець поточного пакетного файлу.

Результат роботи програми продемонстровано на рисунку 11.1



```

F:\>new_script.bat

F:\>IF EXIST laba.* GOTO MI

F:\>ECHO laba not found.
laba not found.

F:\>GOTO END

```

Рисунок 11.1 – Демонстрація роботи скрипту

Як видно з екрану, командний процесор дублює всі команди на екран. Це досить незручно. У цьому випадку за допомогою команди **echo** можна вимкнути дублювання на екрані команд. Якщо почати скрипт з команди **echo off**, то у вікно не будуть виводитись команди, що виконуються (крім самої команди **echo off**). Вимкнути дублювання самої команди **echo off** можна, розмістивши знак **@** на початку рядка. Цей знак подавляє дублювання команди тільки зі свого рядка. Таким чином, якщо почати командний файл з команди **@echo off**, то дублювання команд буде повністю подавлено.

Команда **echo.** (з крапкою) виводить на екран порожній рядок.

Виведення повідомлень командою **echo** можна перенаправити у файл або інший пристрій виведення. Для цього використовується механізм, який було описано вище, перенаправлення введення/виведення (символи **>** та **>>**).

Приклад 11.4 Файл містить набір команд, які у файл **report.txt** записують вміст папки **work**, поточну дату та час.

```

@ECHO OFF
ECHO Folder «work» contents as of> report.txt
date /t>> report.txt t

```



```
time /t>> report.txt
dir d:\work >>report.txt
```

Виконання скрипту можна перервати, натиснувши **Ctrl+C** або **Ctrl+Break**. На екрані з'явиться запит на переривання виконання командного файлу. Якщо відповісти «Так», то виконання буде закінчено, якщо «Ні», то виконання буде продовжено з наступної команди.

У командному файлі можна викликати інший командний файл, вказавши його ім'я. Після закінчення роботи файлу, що було викликано, виконання вихідного файлу продовжено не буде. Якщо вимагається продовжити виконання вихідного файлу, то це можна зробити командою **call**.

Приклад 11.5 Файл виводить список файлів та папок, які було створено або модифіковано у 2019 році, у файл **task.txt**.

```
@echo off
echo Files created in 2019 >task.txt
dir | find ".2019">>task.txt
```

11.3 Linux: створення і запуск командних файлів. Приклади

Процес створення командного файлу у Linux майже нічим не відрізняється від процесу у Windows. Різниця у тому, що розширення не є обов'язковим, а також у синтаксисі команд.

Повідомлення на екран виводяться командою **echo**. Команди зі скрипту на екрані консолі не дублюються. Коментар починається із символу **#**.

Приклад 11.6 Сценарій **script.sh** містить такий набір інструкцій

```
#!/bin/bash
echo It works!
```

Файл сценарію починається з послідовності **#!**, яка у світі UNIX називається **sha-bang**, що вказує системі який інтерпретатор слід використовувати для виконання сценарію. Сигнатури **#!** може не бути, якщо користувач не використовує специфічних команд.

Далі виводиться на екран тестове повідомлення. Звісно, файл має бути запущено з того каталогу, де він зберігається. Роботу програми проілюстровано на рисунку 11.2

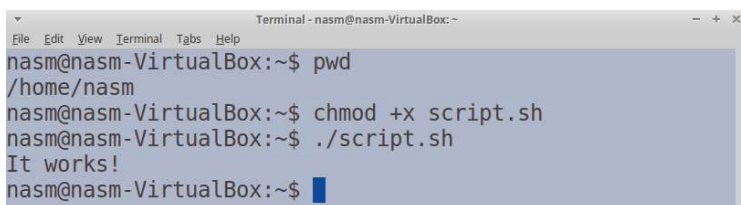


Рисунок 11.2 – Демонстрація роботи сценарію **script.sh**

Декілька пояснень до зображеного на рисунку 11.2. Спочатку командою **pwd** перевірено, чи співпадає поточний каталог із каталогом, в якому збережено

файл. Щоб файл став виконуваним, для нього необхідно встановити права на виконання файлів як програм, що було зроблено за допомогою команди

```
chmod +x script.sh,
```

якщо файл знаходиться не у поточному каталозі, то потрібно вказати повний шлях. Далі для файлу **script.sh** надано атрибут «виконання» (параметр «+x»), і нарешті передано інтерпретатору на виконання.

Приклад 11.7 Командний файл у вмісті домашньої папки шукає файли, в іменах яких є комбінація **.txt**. Нагадаємо, що у файлів в Linux немає розширень, а текстові комбінації в іменах після крапки використовуються суто для зручності користувача для швидкої ідентифікації вмісту файлу.

```
#!/bin/bash  
ls -l | find -name "*.sh" >> task.txt
```

Команда канал передає список файлів на вхід команди, яка шукає в їхніх іменах потрібну комбінацію, а інформація, яку отримано, виводиться у текстовий файл.

Приклад 11.8 У командному файлі подано такий набір інструкцій:

```
#!/bin/bash  
touch task2.txt  
if [ -f task2.txt ]; then  
    echo "файл task2.txt існує"  
fi  
if ! [ -f task2.txt ]; then  
    echo "файл task2.txt не існує"  
fi
```

Декілька коментарів до коду: спочатку створюється файл **task2.txt**, далі за допомогою команди **[]** з опцією **-f** перевірено існування саме файлу, а коли відповідь отримано – виводиться на екран відповідне повідомлення. Зазначимо, що потрібно уважно слідкувати за наявністю пробілів у синтаксисі команди та додаткових опцій до неї.

Для командних файлів Windows та Linux використовуються ті ж самі алгоритмічні конструкції (розгалуження, цикли та ін.), що і у мовах програмування.

Корисні посилання

1. Синтаксис команд Linux. URL : https://linuxguide.rozh2sch.org.ua/#_синтаксис_команд
2. Робота користувача в ОС Windows. URL : <http://arkov.narod.ru/OS-Lab-123.pdf>
3. Вступ до інтерфейсу командного рядка. URL : https://tutorial.djangogirls.org/uk/intro_to_command_line/

Тестові завдання

1. Командний файл Windows має розширення:
 - а) cmd;
 - б) bash;
 - в) bat;
 - г) exe.
2. Коментар у командному файлі Linux починається символом:
 - а) !;
 - б) ?;
 - в) *;
 - г) #.
3. Усі файли, що мають розширення **.png** можна позначити маскою:
 - а) *.png;
 - б) ?.png;
 - в) ???.png;
 - г) %.png.
4. Запустити командний рядок у Linux можна комбінацією клавіш:
 - а) Alt+T;
 - б) Ctrl+Alt;
 - в) Win+R/cmd;
 - г) Ctrl+Alt+T.
5. Коментар у пакетному файлі Windows:
 - а) REM;
 - б) STEM;
 - в) TEN;
 - г) SEM.
6. Вимкнути дублювання всіх команд з bat-файлу можна, розмістивши у першому рядку:
 - а) ECHO OFF;
 - б) ECHO ON;
 - в) @ECHO OFF;
 - г) №.
7. У якому варіанті друга команда виконається тільки, якщо перша завершилась успішно?
 - а) com1 || com2;
 - б) com1 & com2;
 - в) com1:com2;
 - г) com1&&com2.
8. Змінити права доступу можна командою:
 - а) echo;
 - б) chmod;
 - в) exit;
 - г) ls.

9. Переглянути назву поточного каталогу термінала можна за допомогою команди:

- а) pwd;
- б) man;
- в) help;
- г) file.

10. Якою командою можна створити файл у Linux?

- а) touch;
- б) dir;
- в) ls;
- г) clear.

Практичні завдання

1. Поясніть на прикладах, чим відрізняються символи > та >>.
2. Створіть командний файл Windows, який виводить у консоль привітання українською мовою. Налаштуйте відповідне кодування.
3. Приклад 11.5 модифікуйте таким чином, щоб пошук відбувався для *текстових* файлів, які створено у 2019 році.

Контрольні запитання

1. Чим відрізняються зовнішні і внутрішні команди командного процесора?
2. Що таке командний файл?
3. Як створити та запустити командний файл у Windows?
4. Як створити та запустити командний файл у Linux?
5. Як написати коментар?
6. Як вимкнути дублювання команд?
7. Що таке послідовне виконання команд?
8. Яким чином команди можна розмістити в одному рядку?
9. Що таке команда канал?

Предметний покажчик

А

Адаптер, 23
Алгоритм, 37
Архітектура фон Неймана, 19
Архітектура потоків даних, 20
Асемблери, 35
Атрибути файлу, 99

Б

Базовий рівень програмного забезпечення, 27
Байт, 12
Біт, 12

В

Вікно, 17
Вікно повідомлення, 72

Г

Гарвардська архітектура, 20
Графічний інтерфейс користувача, 67

Д

Діалогове вікно, 72
Драйвер, 28

Е

Евристичний аналізатор, 86
Електронна таблиця, 51

І

Інтерпретатор, 36
Інтерфейс, 67
Інформатика, 4
Інформаційна система, 5
Інформаційна технологія, 5
Інформація, 6

К

Каталог, 100
Команда операційної системи, 120
Командний процесор, 95
Командний рядок, 119
Командний файл, 131

Компілятор, 36
Компоновщик, 35
Комп'ютерна технологія, 6
Комп'ютерний вірус, 83
Контекстне меню, 75

Л

Логічна організація файлу, 102
Логічне форматування, 98
Логічний запис, 102

М

Макрос, 63
Материнська плата, 22
Меню, 75
Міжпрограмний інтерфейс, 27
Мікропроцесор, 22

Н

Накопичувач, 23
Налагоджувач, 35

О

Оперативна пам'ять, 23
Операційна система, 29

П

Паралельна архітектура, 21
Плагін, 111
Поведінковий аналізатор, 86
Постійний запам'ятовувальний пристрій, 23
Поточний каталог, 100
Прикладний рівень програмного забезпечення, 30
Принципи фон Неймана, 18
Програмне вікно, 74
Програмне забезпечення, 27

Р

Репутаційні технології, 87
Розрядність процесора, 22

С

Сигнатурний аналіз, 86
Система програмування, 35
Система управління базами даних, 36

Системний рівень програмного забезпечення, 28
Службове програмне забезпечення, 29
Стиснення даних, 87
Структура комп'ютера, 21

Т

Таблиця ASCII, 14
Табличний процесор, 51
Тактова частота процесора, 22
Текстовий документ, 41
Текстовий процесор, 41
Текстовий редактор, 41
Терафлопс, 11
Транслятор, 35

У

Утиліта, 29

Ф

Файл, 99
Файлова система, 95
Файловий менеджер, 110
Фізична організація файлу, 104
Фізичний диск, 96
Форматування даних, 59
Фрагментація, 104

Х

Хмара, 33
Хмарні технології, 87

Ш

Шаблон, 101
Шлях, 100

Я

Ярлик, 71

Рекомендована література

Основна:

1. Баженов В. А., Венгерський П. С., Горлач В. М. Інформатика: комп'ютерна техніка, комп'ютерні технології : підручник для студентів вищого навчального закладу. Київ : Каравелла, 2006. 250 с.
2. Білик В. М., Костирко В. С. Інформаційні технології та системи. Київ : Центр учбової літератури, 2008. 188 с.
3. Шинкаренко Г. А., Шишов О. В. Інформатика. Комп'ютерна техніка. Комп'ютерні технології. Київ : Каравела, 2011. 592 с.
4. Дибкова Л. М. Інформатика і комп'ютерна техніка. Київ : Академвидав, 2005. 416 с.
5. Войтюшенко Н. М. Інформатика і комп'ютерна техніка. Київ : Центр учбової літератури, 2009. 564 с.

Додаткова:

1. Бусигін Б. С., Коротенко Г. М., Коротенко Л. М. Прикладна інформатика. Дніпропетровськ : Національний гірничий університет, 2004. 559 с.
2. Коноплянко З. Д. Інформатика та комп'ютерна техніка. Українсько-Англійсько-Російський термінологічний тлумачний словник. URL : https://www.studmed.ru/view/konoplyanko-zd-nformatika-ta-kompyuterna-tehnka-ukrayinsko-anglysko-rosyskiy-termnologchniy-tlumachiny-slovník_25eba0b4674.html (дата звернення : 19.04.2019)
3. Карпенко С. М., Іванов Є. О. Основи інформаційних систем і технологій. Київ : МАУП, 2006. 264 с.
4. Лясин Д. Н., Саньков С. Г. Командные файлы OS MS Windows. Волгоград : Волгоградский государственный технический университет, 2012. 29 с.

Навчальне видання
(українською мовою)

Козлова Ольга Станіславівна
Кривохата Анастасія Григорівна
Кудін Олексій Володимирович

МЕТОДИ ТА ЗАСОБИ КОМП'ЮТЕРНИХ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Навчальний посібник
для здобувачів ступеня вищої освіти бакалавра
спеціальності «Інженерія програмного забезпечення»
освітньо-професійної програми «Програмна інженерія»

Рецензент *О.С. Пшенична*
Відповідальний за випуск *А.О. Лісняк*
Коректор *А.Г. Кривохата*