

# ЛЕКЦІЯ 1. ОСНОВНІ КОНЦЕПЦІЇ, ЕВОЛЮЦІЯ, РІЗНОВИДИ ОПЕРАЦІЙНИХ СИСТЕМ

## 1.1. Поняття операційної системи, її призначення та функції

### 1.1.1. Поняття операційної системи

Причиною появи операційних систем була необхідність створення зручних у використанні комп'ютерних систем (під *комп'ютерною системою* будемо розуміти сукупність апаратного і програмного забезпечення комп'ютера). Комп'ютерні системи від самого початку розроблялися для розв'язання практичних задач користувачів. Оскільки робити це за допомогою лише апаратного забезпечення виявилось складно, були створені прикладні програми. Для таких програм знадобилися загальні операції керування апаратним забезпеченням, розподілу апаратних ресурсів тощо. Ці операції згрупували в рамках окремого рівня програмного забезпечення, який і стали називати операційною системою.

Далі можливості операційних систем вийшли далеко за межі базового набору операцій, необхідних прикладним програмам, але проміжне становище таких систем між прикладними програмами й апаратним забезпеченням залишилося незмінним.

Можна дати таке означення операційної системи. *Операційна система (ОС)* – це програмне забезпечення, щоб реалізувати зв'язок між прикладними програмами й апаратними засобами комп'ютера.

### 1.1.2. Призначення операційної системи

Операційні системи забезпечують, по-перше, зручність використання комп'ютерної системи, по-друге, ефективність і надійність її роботи.

Перша функція властива ОС як розширеній машині, друга – ОС як розподільвача апаратних ресурсів.

### 1.1.3. Операційна система як розширена машина

За допомогою операційної системи у прикладного програміста (а через його програми і в користувача) має створюватися враження, що він працює з розширеною машиною.

Апаратне забезпечення комп'ютера недостатньо пристосоване до безпосереднього використання у програмах. Наприклад, якщо розглянути роботу із пристроями введення-виведення на рівні команд відповідних контролерів, то можна побачити, що набір таких команд обмежений, а для багатьох пристроїв - примітивний (є навіть вислів: «апаратне забезпечення потворне»). Операційна система приховує такий *інтерфейс апаратного забезпечення*, замість нього програмістові пропонують *інтерфейс прикладного програмування* (рис. 1.1), що використовує поняття вищого рівня (їх називають абстракціями).



Рис. 1.1. Взаємодія ОС із апаратним забезпеченням і застосуваннями

Наприклад, при роботі з диском типовою абстракцією є файл. Працювати з файлами простіше, ніж безпосередньо з контролером диска (не потрібно враховувати переміщення головок дисководу, запускати й зупиняти мотор тощо), внаслідок цього програміст може зосередитися на суті свого прикладного завдання. За взаємодію з контролером диска відповідає операційна система.

Виділення абстракцій дає змогу досягти того, що код ОС і прикладних програм не потребуватиме зміни при переході на нове апаратне забезпечення. Наприклад, якщо встановити на комп'ютері дисковий пристрій нового типу (за умови, що він підтримується ОС), всі його особливості будуть враховані на рівні ОС, а прикладні програми продовжуватимуть використовувати файли, як і раніше. Така характеристика системи називається *апаратною незалежністю*. Можна сказати, що ОС надають апаратно-незалежне середовище для виконання прикладних програм.

#### 1.1.4. Операційна система як розподільувач ресурсів

Операційна система має ефективно розподіляти ресурси. Під ресурсами розуміють процесорний час, дисковий простір, пам'ять, засоби доступу до зовнішніх пристроїв. Операційна система виступає в ролі менеджера цих ресурсів і надає їх прикладним програмам на вимогу.

Розрізняють два основні види розподілу ресурсів. У разі *просторового розподілу* ресурс доступний декільком споживачам одночасно, при цьому кожен із них може користуватися частиною ресурсу (так розподіляється пам'ять). У разі *часового розподілу* система ставить споживачів у чергу і згідно з нею надає їм змогу користуватися всім ресурсом обмежений час (так розподіляється процесор в однопроцесорних системах).

При розподілі ресурсів ОС розв'язує можливі конфлікти, запобігає несанкціонованому доступу програм до тих ресурсів, на які вони не мають прав, забезпечує ефективну роботу комп'ютерної системи.

### 1.2. Історія розвитку операційних систем

Перші операційні системи з'явилися в 50-ті роки і були *системами пакетної обробки*. Такі системи забезпечували послідовне виконання програм у пакетному режимі (без можливості взаємодії з користувачем). У певний момент часу в пам'яті могла перебувати тільки одна програма (системи були однозадачними), усі програми виконувалися на процесорі від початку до кінця. За такої ситуації ОС розглядали просто як набір стандартних служб, необхідних прикладним програмам і користувачам.

Наступним етапом стала підтримка багатозадачності. У багатозадачних системах у пам'ять комп'ютера стали завантажувати декілька програм, які виконувалися на процесорі навперемінно. При цьому розвивалися два напрями: *багатозадачна пакетна обробка* і *розподіл часу*. У багатозадачній пакетній обробці завантажені програми, як і раніше, виконувалися в пакетному режимі. У режимі розподілу часу із системою могли працювати одночасно кілька користувачів, кожному з яких надавався діалоговий термінал (пристрій, що складається із клавіатури і дисплея).

Підтримка багатозадачності потребувала реалізації в ОС засобів координації задач. Можна виділити три складові частини такої координації.

1. Захист критичних даних задачі від випадкового або навмисного доступу інших задач.
2. Забезпечення обміну даними між задачами.
3. Надання задачам справедливої частки ресурсів (пам'яті, процесора, дискового простору тощо).

Ще одним етапом стала поява *ОС персональних комп'ютерів*. Спочатку ці системи, як і ОС першого етапу, були однозадачними й надавали базовий набір стандартних служб (на цьому етапі важливим було впровадження графічного інтерфейсу користувача). Подальший розвиток апаратного забезпечення дав змогу використати в таких системах засоби, розроблені для більших систем, насамперед багатозадачність і, як наслідок, координацію задач.

Є правило розвитку ОС для конкретної апаратної платформи: для більшості нових апаратних платформ ОС спочатку створюють як базовий набір стандартних служб, координацію задач реалізують у ній пізніше. Зазначимо, що це правило вірне, якщо апаратна платформа дозволяє реалізувати багатозадачний режим.

Багато сучасних ОС спочатку розроблялися для персональних комп'ютерів або були перенесені на них з інших апаратних платформ. Основну увагу у цьому курсі буде приділено двом групам операційних систем: UNIX-сумісним системам, насамперед Linux, та серії Windows NT/2000/XP фірми Microsoft (далі – лінія Windows XP).

### 1.3. Класифікація сучасних операційних систем

Розглянемо класифікацію сучасних операційних систем залежно від області їхнього застосування.

Насамперед відзначимо *ОС великих ЕОМ* (мейнфреймів). Основною характеристикою апаратного забезпечення, для якого їх розробляють, є продуктивність введення-виведення: великі ЕОМ оснащують значною кількістю периферійних пристроїв (дисків, терміналів, принтерів тощо). Такі комп'ютерні системи використовують для надійної обробки значних обсягів даних, при цьому ОС має ефективно підтримувати цю обробку (в пакетному режимі або в режимі розподілу часу). Прикладом ОС такого класу може бути OS/390 фірми IBM.

До наступної категорії можна віднести *серверні ОС*. Головна характеристика таких ОС – здатність обслуговувати велику кількість запитів користувачів до спільно використовуваних ресурсів. Важливу роль для них відіграє мережна підтримка. Є спеціалізовані серверні ОС, з яких виключені елементи, не пов'язані з виконанням їхніх основних функцій (наприклад, підтримка застосувань користувача). Нині для реалізації серверів частіше застосовують універсальні ОС (UNIX або системи лінії Windows XP).

Наймасовіша категорія – *персональні ОС*. Деякі ОС цієї категорії розробляли з розрахунком на непрофесійного користувача (лінія Windows 95/98/Me фірми Microsoft, яку далі називатимемо Consumer Windows), інші є спрощеними версіями універсальних ОС. Особлива увага в персональних ОС приділяється підтримці графічного інтерфейсу користувача і мультимедіа-технологій.

Виділяють також *ОС реального часу*. У такій системі кожна операція має бути гарантовано виконана в заданому часовому діапазоні. ОС реального часу можуть керувати польотом космічного корабля, технологічним процесом або демонстрацією відеороликів. Існують спеціалізовані ОС реального часу, такі як QNX і VxWorks.

Ще однією категорією є *вбудовані ОС*. До них належать керуючі програми для різноманітних мікропроцесорних систем, які використовують у військовій техніці, системах побутової електроніки, смарт-картах та інших пристроях. До таких систем ставлять особливі вимоги: розміщення в малому обсязі пам'яті, підтримка спеціалізованих засобів введення-виведення, можливість прошивання в постійному запам'ятовувальному пристрої. Часто вбудовані ОС розробляються під конкретний пристрій; до універсальних систем належать Embedded Linux і Windows CE.

## 1.4. Функціональні компоненти операційних систем

Операційну систему можна розглядати як сукупність функціональних компонентів, кожен з яких відповідає за реалізацію певної функції системи. У цій лекції описані найважливіші функції сучасних ОС і компоненти, що їх реалізують.

Спосіб побудови системи зі складових частин та їхній взаємозв'язок визначає архітектура операційної системи. Підходи до реалізації архітектури ОС будуть розглянуті в лекції 2.

### 1.4.1. Керування процесами й потоками

Як ми вже згадували, однією з найважливіших функцій ОС є виконання прикладних програм. Код і дані прикладних програм зберігаються в комп'ютерній системі на диску в спеціальних виконуваних файлах. Після того як користувач або ОС вирішать запустити на виконання такий файл, у системі буде створено базову одиницю обчислювальної роботи, що називається *процесом* (process).

Можна дати таке означення: процес – це програма під час її виконання.

Операційна система розподіляє ресурси між процесами. До таких ресурсів належать процесорний час, пам'ять, пристрої введення-виведення, дисковий простір у вигляді файлів. При розподілі пам'яті з кожним процесом пов'язується його *адресний простір* – набір адрес пам'яті, до яких йому дозволено доступ. В адресному просторі зберігаються код і дані процесу. При розподілі дискового простору для кожного процесу формується список відкритих файлів, аналогічним чином розподіляють пристрої введення-виведення.

Процеси забезпечують захист ресурсів, якими вони володіють. Наприклад, до адресного простору процесу неможливо безпосередньо звернутися з інших процесів (він є захищеним), а при роботі з файлами може бути задано режим, що забороняє доступ до файлу всім процесам, крім поточного.

Розподіл процесорного часу між процесами необхідний через те, що процесор виконує інструкції одну за одною (тобто в конкретний момент часу на ньому може фізично виконуватися тільки один процес), а для користувача процеси мають виглядати як послідовності інструкцій, виконувани паралельно. Щоб домогтися такого ефекту, ОС надає процесор кожному процесу на деякий короткий час, після чого перемикає процесор на інший процес; при цьому виконання процесів відновлюється з того місця, де їх було перервано. У *багатопроцесорній системі* процеси можуть виконуватися паралельно на різних процесорах.

Сучасні ОС крім багатозадачності можуть підтримувати *багатопотоковість* (multithreading), яка передбачає в рамках процесу наявність кількох послідовностей інструкцій (*потоків*, threads), які для користувача виконуються паралельно, подібно до самих процесів в ОС. На відміну від процесів потоки не забезпечують захисту ресурсів (наприклад, вони спільно використовують адресний простір свого процесу).

### 1.4.2. Керування пам'яттю

Під час виконання програмного коду процесор бере інструкції й дані з оперативної (основної) пам'яті комп'ютера. При цьому така пам'ять відображається у вигляді масиву байтів, кожен з яких має адресу.

Як уже згадувалося, основна пам'ять є одним із видів ресурсів, що розподіляються між процесами. ОС відповідає за виділення пам'яті під захищений адресний простір процесу і за вивільнення пам'яті після того, як виконання процесу буде завершено. Обсяг пам'яті, доступний процесу, може змінюватися в ході виконання, у цьому разі говорять про динамічний розподіл пам'яті.

ОС повинна забезпечувати можливість виконання програм, які окремо або в сукупності перевищують за обсягом доступну основну пам'ять. Для цього в ній має бути реалізована технологія *віртуальної пам'яті*. Така технологія дає можливість розміщувати в основній пам'яті тільки ті інструкції й дані процесу, які потрібні в поточний момент часу, при цьому вміст іншої частини адресного простору зберігається на диску.

### 1.4.3. Керування введенням-виведенням

Операційна система відповідає за керування пристроями введення-виведення, підключеними до комп'ютера. Підтримка таких пристроїв в ОС звичайно здійснюється на двох рівнях. До першого, нижчого, рівня належать *драйвери пристроїв* – програмні модулі, які керують пристроями конкретного типу з урахуванням усіх їхніх особливостей. До другого рівня належить *універсальний інтерфейс введення-виведення*, зручний для використання у прикладних програмах.

ОС має реалізовувати загальний інтерфейс драйверів введення-виведення, через який вони взаємодіють з іншими компонентами системи. Такий інтерфейс дає змогу спростити додавання в систему драйверів для нових пристроїв.

Сучасні ОС надають великий вибір готових драйверів для конкретних периферійних пристроїв. Що більше пристроїв підтримує ОС, то більше в неї шансів на практичне використання.

### 1.4.4. Керування файлами та файлові системи

Для користувачів ОС і прикладних програмістів дисковий простір надається у вигляді сукупності *файлів*, організованих у *файлову систему*.

Файл – це набір даних у файловій системі, доступ до якого здійснюється за іменем. Термін «файлова система» може вживатися для двох понять: принципу організації даних у вигляді файлів і конкретного набору даних (зазвичай відповідної частини диска), організованих відповідно до такого принципу. У рамках ОС може бути реалізована одночасна підтримка декількох файлових систем.

Файлові системи розглядають на логічному і фізичному рівнях. Логічний рівень визначає зовнішнє подання системи як сукупності файлів (які звичайно перебувають у каталогах), а також виконання операцій над файлами і каталогами (створення, вилучення тощо). Фізичний рівень визначає принципи розміщення структур даних файлової системи на диску або іншому пристрої.

### 1.4.5. Мережна підтримка

#### Мережні системи

Сучасні операційні системи пристосовані до роботи в мережі, їх називають *мережними операційними системами*. Засоби мережної підтримки дають ОС можливість:

- надавати локальні ресурси (дисковий простір, принтери тощо) у загальне користування через мережу, тобто функціонувати як сервер;
- звертатися до ресурсів інших комп'ютерів через мережу, тобто функціонувати як клієнт.

Реалізація функціональності сервера і клієнта базується на *транспортних засобах*, які відповідають за передачу даних між комп'ютерами відповідно до правил, обумовлених мережними протоколами.

## Розподілені системи

Мережні ОС не приховують від користувача наявність мережі, мережна підтримка в них не визначає структуру системи, а збагачує її додатковими можливостями. Є також *розподілені ОС*, які дають змогу об'єднати ресурси декількох комп'ютерів у *розподілену систему*. Вона виглядає для користувача як один комп'ютер з декількома процесорами, що працюють паралельно. Розподілені та багатопроцесорні системи є двома основними категоріями ОС, які використовують декілька процесорів. Вони мають багато спільного й будуть розглянуті в подальших лекціях.

### 1.4.6. Безпека даних

Під безпекою даних в ОС розуміють забезпечення надійності системи (захисту даних від втрати у разі збоїв) і захист даних від несанкціонованого доступу (випадкового чи навмисного).

Для захисту від несанкціонованого доступу ОС має забезпечувати наявність засобів *аутентифікації* користувачів (такі засоби дають змогу з'ясувати, чи є користувач тим, за кого себе видає; зазвичай для цього використовують систему паролів) та їхньої *авторизації* (дозволяють перевірити права користувача, що пройшов аутентифікацію, на виконання певної операції).

### 1.4.7. Інтерфейс користувача

Розрізняють два типи засобів взаємодії користувача з ОС: *командний інтерпретатор* (shell) і *графічний інтерфейс користувача* (GUI).

Командний інтерпретатор дає змогу користувачам взаємодіяти з ОС, використовуючи спеціальну командну мову (інтерактивно або через запуск на виконання командних файлів). Команди такої мови змушують ОС виконувати певні дії (наприклад, запускати програми, працювати із файлами).

Графічний інтерфейс користувача надає йому можливість взаємодіяти з ОС, відкриваючи вікна і виконуючи команди за допомогою меню або кнопок. Підходи до реалізації графічного інтерфейсу доволі різноманітні: Наприклад, у Windows-системах засоби його підтримки вбудовані в систему, а в UNIX вони є зовнішніми для системи і спираються на стандартні засоби керування введення-виведенням.

## Висновки

**Операційна система – це рівень програмного забезпечення, що перебуває між рівнями прикладних програм й апаратного забезпечення комп'ютера. Головне її призначення – зробити використання комп'ютерної системи простішим і підвищити ефективність її роботи.**

**До основних функціональних компонентів ОС належать: керування процесами, керування пам'яттю, керування введенням-виведенням, керування файлами і підтримка файлових систем, мережна підтримка, забезпечення захисту даних, реалізація інтерфейсу користувача.**