

# МЕТОДИЧНІ ОСНОВИ ВИВЧЕННЯ ТЕМИ “ОПЕРАЦІЙНІ СИСТЕМИ” У ПІДГОТОВЦІ МАЙБУТНЬОГО ВЧИТЕЛЯ

Семеріков С.О.

Криворізький державний педагогічний університет

Стрімкий розвиток інформатики як науки, вдосконалення засобів обчислювальної техніки та програмного забезпечення призводить до швидкого застарівання навичок роботи з конкретними програмними продуктами. Відомий спеціаліст у галузі програмування Дональд Кнут, відмічаючи цю тенденцію, підкреслює необхідність навчання концептуальних основ інформатики, її математичної та алгоритмічної складових, що формують загальні прийоми та методи роботи майбутнього спеціаліста.

На жаль, значна частина навчальної літератури з інформатики не відповідає цим вимогам, пропонуючи варіант вивчення курсу інформатики без ознайомлення з її теоретичними основами та переносячи центр уваги на заучування інтерфейсу окремих прикладних програм. Це призводить до такого негативного явища, як заміна вивчення інваріантної частини курсу інформатики варіативною, а, отже, штучного “розмиття” вимог до знань та вмінь з предмету.

*Операційна система* (ОС) є першим, з чим стикається будь-який учень на початку роботи з комп’ютером, проте означення операційної системи в курсі інформатики вводиться не одразу. Так, М.І. Жалдак та Н.В. Морзе у підручнику “Інформатика–7” спочатку вводять поняття інформаційної (обчислювальної) системи як системи пристроїв і описів правил управління ними, а сукупність описів правил управління пристроями інформаційної системи називають її програмною частиною або програмним забезпеченням [1, с. 24].

В процесі вивчення курсу введені поняття збагачуються, наповнюючись досвідом практичної роботи, тому виникає необхідність формалізації поняття операційної системи, як це було зроблено, наприклад, у навчальному посібнику

М.І. Жалдака та Ю.С. Рамського [2].

На жаль, у багатьох студентів поняття операційної системи залишається досить розмитим, і цьому є об'єктивні причини. По-перше, операційні системи розроблені для великої кількості різноманітних комп'ютерних систем: однокористувацьких робочих станцій та персональних комп'ютерів, багатокористувацьких систем середнього масштабу, універсальних обчислювальних машин та суперкомп'ютерів, систем реального часу тощо. По-друге, неперервно продовжується інтенсивний розвиток усіх напрямків досліджень, що відносяться до комп'ютерів: в будові операційних систем з'явилися нові важливі компоненти, технології та розробки [3].

Незважаючи на таку різноманітність систем та постійні зміни, деякі фундаментальні положення залишаються незмінними, і саме на них варто зосередити увагу студентів. Для цього ми пропонуємо вивчення теми "Операційні системи" за таким планом:

## **1. Поняття операційної системи. Багатошарова архітектура комп'ютерної системи.**

В цьому питанні доцільно розглянути *операційну систему* як розподільвач ресурсів, керуючу програму, засіб підвищення ефективності обчислювальної системи та полегшення праці програміста.

При розгляді багатошарової архітектури комп'ютерної системи використовуємо запропоновану в [3] схему розташування шарів:

прикладне програмне забезпечення
компілятори, редактори, командні інтерпретатори (системне програмне забезпечення)
операційна система
машинні мови
мікропрограми
фізичні пристрої

## **2. Етапи розвитку операційних систем.**

На кожному етапі виділятимемо його характерні ознаки:

*Ранні системи (1945-1955)* – відсутність операційних систем та захисту програм.

*Друге покоління систем (1956-1965)* – системи пакетної обробки, канали введення-виведення, буферизація, переривання, бібліотеки, мови керування завданнями.

*Третє покоління систем (1966-1980)* – багатозадачність, розподіл часу, віртуальна пам'ять, багатопроцесорність.

*Четверте покоління* та подальший розвиток операційних систем (з 1981) – масивно-паралельні системи, комп'ютерні мережі, розподілені обчислення.

### **3. Концепції операційної системи.**

Кожна з концепцій операційної системи (а це, насамперед, програми, процеси, потоки, файли, системні виклики, оболонки, ядро, пам'ять, управління пам'яттю) може бути предметом детального розгляду, проте нашою задачею є означення їх суттєвих властивостей.

Під *програмою* розумітимемо:

- набір інструкцій та даних, що зберігаються у фалі на диску;
- файл, що має атрибут “виконуваний”;
- файл, що містить впорядковану послідовність інструкцій ядра;
- файл з текстом програми мовою програмування;
- об'єктний файл (текст програми, відтрансльований в машинні коди);
- виконуваний файл, що містить об'єктний код з приєднаними бібліотеками.

*Процес* розглядатимемо як програму під час виконання, що створюється ядром як середовище її виконання, може бути призупинений ОС та складається з трьох сегментів: коду, даних користувача та системних даних (зокрема, тих, що відсутні у програмі).

*Потік* означимо як послідовність виконуваних інструкцій всередині процесу.

Концепцію *файлу*, якою студенти вже володіють, доцільно розширити введенням поняття дескриптора файлу як унікального цілого числа, що однозначно ідентифікує файл в операціях введення-виведення та запропонувати серію прикладів застосування дескрипторів файлів для операцій із регулярними

файлами, каталогами, каналами, символними та блочними пристроями тощо.

*Системні виклики* розглядатимемо як сукупність розширених інструкцій операційної системи, що забезпечують інтерфейс між програмами та ядром системи.

При розгляді *оболонки (інтерпретаторів команд)* слід звернути увагу на те, що вони є звичайними програмами, які можуть бути замінені іншими, тому до оболонок формують такі вимоги:

- підтримка запрошення до діалогу;
- перенаправлення введення-виведення;
- можливість виконання фонових задач;
- підтримка щонайменше трьох типів команд: виконуваних файлів, сценаріїв та вбудованих команд оболонки.

Найбільш містким з розглядуваних є концепція *ядра* – коду, який:

- постійно знаходиться в основній пам'яті;
- контролює виконання процесів шляхом їх створення, завершення, призупинення та взаємодії;
- планує виконання процесів на центральному процесорі;
- розподіляє пам'ять для виконуваних процесів;
- обслуговує файлову систему;
- дає процесам керований доступ до периферійних пристроїв;
- прозоро надає ряд системних послуг, що узагальнюють концепцію файлу.

*Пам'ять* пропонується характеризувати розміром, швидкістю та вартістю. Це дозволяє впорядкувати різні типи пам'яті у порядку зменшення вартості:

Тип	Розмір	Швидкість доступу
Регістрова пам'ять	Байти	Майже зі швидкістю процесора
Кеш-пам'ять	Кілобайти	Наносекунди
Оперативна пам'ять	Мегабайти	Мікросекунди
Магнітні диски	Гігабайти	Мілісекунди

Тип	Розмір	Швидкість доступу
Магнітні стрічки /Оптичні диски	Без обмежень	–

Окремо розглядаються віртуальна та дискова кеш-пам'ять.

*Управління пам'яттю* є однією з головних послуг операційної системи, що забезпечує:

- ізоляцію процесів;
- автоматичне виділення та контроль пам'яті;
- підтримку модульного програмування;
- керування захистом та доступом;
- довготермінове зберігання.

#### **4. Структура операційної системи.**

Для кожного типу систем виділятимемо їх характерні ознаки.

*Мінімальні операційні системи* – найбільш примітивна форма операційних систем, що являє собою безструктурний набір незахищених підпрограм, які важко адаптувати до мультипрограмного та багатокористувацького середовища.

*Багатошарові системи* являють собою ієрархію шарів, що розташовані один над одним, або серію концентричних кіл, в яких рівень привілеїв зростає до центра.

*Віртуальні машини* утворюють для кожного процесу віртуальний центральний процесор, надаючи мінімальний набір операцій, що утворюють трирівневу ієрархію:

- найнижчий рівень – віртуалізація центрального процесора для всіх процесів;
- середній рівень – віртуалізація пам'яті для всіх процесів;
- вищий рівень – віртуальні пристрої введення-виведення.

*Системи клієнт-сервер* мають мінімальне ядро, що забезпечує взаємодію між клієнтом (користувацьким процесом) та сервером (обробником запитів).

#### **5. Засоби введення-виведення.**

Розгляд цього пункту доцільно почати з найпростішого *програмованого*

*введення-виведення*, при якому центральний процесор безпосередньо керує контролерами пристроїв. Далі переходимо до *введення-виведення, керованого перериваннями*, коли центральний процесор керує процесом введення-виведення, посилаючи команди відповідним модулям. Завершується пункт розглядом *каналів прямого доступу до пам'яті*, коли центральний процесор передає повноваження модулю DMA по взаємодії із закріпленим за ним блоком пам'яті.

## **6. Мультипрограмування. Мультипрограмні ОС та їх задачі. Операційні системи як віртуальні машини.**

В останньому пункті коротко характеризуємо *апаратні засоби підтримки мультипрограмування* (механізми пріоритетних переривань, захисту коду та даних, динамічний перерозподіл адрес) та *вимоги до мультипрограмних ОС*, зумовлені ними.

Наведена структура вивчення теми дозволяє наповнити поняття операційної системи змістом на основі вивчення загальних для усіх систем властивостей, що найбільш повно відповідає вимогам до фундаментальної підготовки майбутнього вчителя інформатики.

### Література:

1. Жалдак М.І., Морзе Н.В. Інформатика–7. Експериментальний навч. посібник для учнів 7 класу загальноосвітньої школи. – К.: Діасофт, 2000. – 208 с.
2. Жалдак М.І., Рамський Ю.С. Інформатика: Навч. посібник / За ред. М.І. Шкіля. – К.: Вища шк., 1991. – 319 с.
3. Столлингс В. Операционные системы, 4-е изд. – М.: Вильямс, 2002. – 848 с.