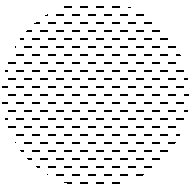
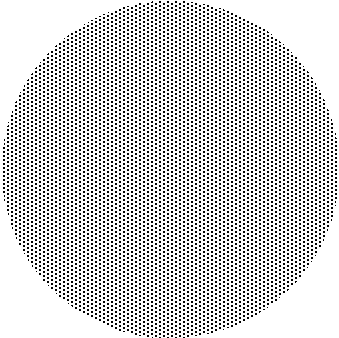
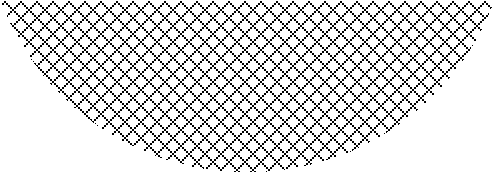
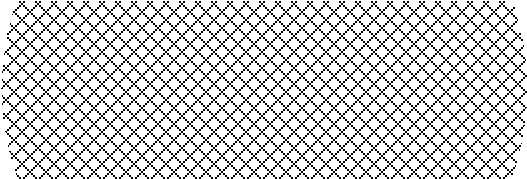
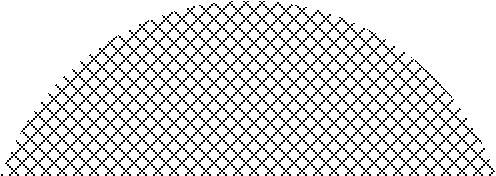
# Лекція № 2

# Багатошарова структура ОС

Обчислювальну систему, що працює під керуванням ОС на основі ядра, можна розглядати як систему, що складається із трьох ієрархічно розташованих шарів:

* нижній шар – утворюється апаратурою;
* проміжний – ядро;
* верхній шар – модулі, що реалізують допоміжні функції.

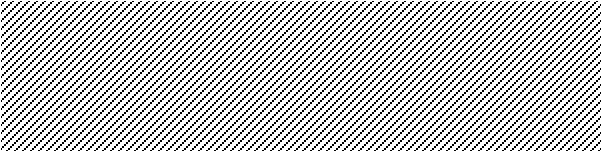
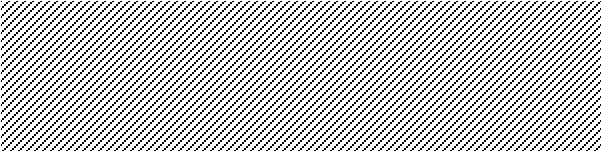


* + допоміжні функції
  + ядро ОС
  + апаратура

Рис. 1.2. Трьохшарова схема обчислювальної системи

Багатошарову структуру, зображену на рис. 1.2, зручно представляти для ілюстрації того факту, що кожен шар може взаємодіяти тільки із сусіднім шаром. При такій організації додатки не можуть безпосередньо взаємодіяти з апаратурою, а тільки через шар ядра.

**Багатошаровий підхід** є універсальним й ефективним способом декомпозиції складних систем. Відповідно до нього, система складається з ієрархії шарів. Кожен шар обслуговує вищележачий шар, виконуючи для нього деякий набір функцій, які утворюють міжшаровий інтерфейс.



до шару

k

+2

шар

k+1

шар

k

міжшаровий

инте

р

фейс

До шару

k

-

1

Рис. 1.3. Концепція багатошарової взаємодії

На основі функцій лежачого нижче шару, наступний шар будує свої функції – більш складні й могутніші, які, у свою чергу, стають примітивами для створення ще більш потужних функцій лежачого вище шару.

Строгі правила взаємодії обумовлюються тільки між шарами. Усередині шару зв'язки між модулями можуть бути довільними. Окремий модуль може виконати свою роботу як самостійно, так і звернутися до іншого модуля свого шару, або до лежачого нижче шару через міжшаровий інтерфейс.

Достоїнства такої системи організації:

* спрощується розробка системи;
* при модернізації системи легко робити зміни усередині кожного шару, не турбуючись про інші шари.

Оскільки ядро являє собою складний багатофункціональний комплекс, цей підхід поширюють і на структуру ядра.

Наприклад:

* **Засоби апаратної підтримки ОС.**Сюди відносять не всі апаратні засоби, а тільки ті, що прямо беруть участь в організації обчислювальних процесів: засоби підтримки привілейованого режиму, систему переривань, засоби перемикання контекстів процесів, засоби захисту областей пам'яті.
* **Машинно-залежні компоненти ОС.**Шар утворюють програмні модулі, які відображують специфіку апаратної платформи комп'ютера. Ідеально, цей шар повністю екранує лежачі вище шари ядра від особливостей апаратури. Це дозволяє розробляти ці шари на основі машинно-незалежних модулів для всіх типів апаратних платформ, підтримуваних даною ОС.
* **Базові механізми ядра***.* Шар виконує найбільш примітивні операції ядра, такі як перемикання контекстів процесів, диспетчеризацію переривань, переміщення сторінок пам'яті на диск і зворотньо і т.п. Модулі даного шару не ухвалюють рішення щодо розподілу ресурсів, вони є виконавчими механізмами для модулів верхніх шарів.
* **Менеджери ресурсів.** Шар складається з потужних функціональних модулів, що реалізують стратегічні завдання по керуванню основними ресурсами обчислювальної системи. Тут працюють менеджери (диспетчери) процесів, введення-виведення, файлової системи й оперативної пам'яті. Розбивка на менеджери може бути різною. Наприклад, менеджер файлової системи іноді поєднують із менеджером введення-виведення, а функції керування доступом користувачів до системи в цілому і її окремим об'єктам доручають окремому менеджеру безпеки.

Кожний з менеджерів веде облік вільних і використовуваних ресурсів певного типу й планує їхній розподіл відповідно до запитів програм. Для виконання ухвалених рішень менеджер звертається до нижчележачого шару. Усередині шару менеджерів існують тісні взаємозв'язки, оскільки для виконання процесу потрібен доступ одночасно до декількох ресурсів – наприклад, процесору, пам'яті, введення-виведення тощо.

* **Інтерфейс системних викликів***.* Це самий верхній шар ядра. Він взаємодіє безпосередньо з додатками й системними утилітами, утворюючи прикладний програмний інтерфейс ОС. Функції, які обслуговують системні виклики, надають доступ до ресурсів системи в зручній і компактній формі, без деталей їхнього фізичного розташування. Для здійснення таких дій системні виклики звичайно звертаються по допомогу до функцій шару менеджерів ресурсів.

Надане розбиття ядра ОС на шари є досить умовною. Таке розбиття й розподіл функцій може бути іншим.

Також може бути іншим і спосіб взаємодії шарів. Для прискорення роботи ядра, в ряді випадків, відбувається звернення з верхнього шару до функцій шарів нижнього рівня, минаючи проміжні шари.

Вибір кількості шарів ядра є відповідальною й складною справою: збільшення числа шарів веде до вповільнення роботи ядра, а зменшення – погіршує розширюваність і логічність системи.

Незалежно від того, які функції виконує ОС, вона повинна задовольняти експлуатаційним вимогам. Вона, зокрема, повинна мати наступні якості:

1. **Надійність.** У випадку помилки в програмному або апаратному устаткуванні, система повинна виявити помилку й / або спробувати виправити ситуацію. Або сповістити про це користувачеві й намагатися звести збитки до мінімуму.
2. **Захист.** Користувач не хоче, щоб інші користувачі (якщо він, наприклад, працює в мережі) йому заважали. Тому ОС повинна захищати користувача від впливу чужих помилок і від спроб злочинного втручання.
3. **Ефективність.** ОС – досить складна програма, що використовує значну частину ресурсів для своїх власних потреб. Ресурси, які використовує ОС, не надходять у розпорядження користувача. Отже ОС повинна бути, якомога найбільш ощадливою. Крім того, вона повинна управляти ресурсами користувачів так, щоб звести до мінімуму час затримки й простоїв.
4. **Передбачуваність.** Вимоги, які користувач може пред'являти до системи, у більшості випадків, є непередбачуваними. Але користувач бажає, щоб обслуговування не дуже сильно змінювалося протягом тривалого часу. Зокрема, при введенні програми в машину користувач повинен мати засноване на попередньому досвіді приблизне уявлення про те, коли йому варто очікувати на

результати.

1. **Зручність.** Все ясно, як і те, що універсальних зручностей не існує.

Тут мова може йти про певний клас завдань.

Дещо про виконувані функції:

**Розподіл процесора***.*У випадку “нерозумної” системи, вся вона розподіляється як єдиний ресурс. Користувач або розпоряджається машиною, або чекає, коли вона буде надана в його розпорядження. Таку стратегію дуже легко організувати, але вона не буде ефективно використовувати устаткування. Для забезпечення паралельної з процесором роботи можна зробити так, щоб одна програма виконувала операції вводу- виводу, поки інша займає головний процесор. Хоча реалізація такого підходу є складнішою, маємо перевагу – кожен пристрій використовується

інтенсивніше.

**Керування пам'яттю***.* Керування пам'яттю тісно пов'язано з розподілом процесора. Програми можуть працювати тільки тоді, коли вони перебувають в оперативній пам'яті, але не обов'язково їх тримати там, якщо шанс отримати процесор є незначним. У цьому випадку, виявиться, що займана ними пам'ять пропадає даремно.

Оперативна пам'ять – це теж розподілюваний ресурс. Тому система витрачає час для раціонального розташування інформації, намагаючись тримати корисні програми в оперативній пам'яті й знищувати “вільні проміжки” між програмами. Для цього система може використовувати переміщення програм. Це робиться для зменшення обсягу даремно використовуваної пам'яті. Переміщення легше здійснити, якщо

використовувати спеціальні стратегії організації пам'яті. Вони дозволяють ОС досить гнучко регулювати обмін інформацією між оперативною й допоміжною пам'яттю.

**Зовнішні пристрої.** Методи розподілу пристроїв введення-виведення й каналів зв'язку різні. Завдання використовує периферійний пристрій стільки часу, скільки йому потрібно, а потім відмовляється від нього. Пристрій зі швидким довільним доступом, такий як, накопичувач на дисках, можна спільно використовувати в декількох завданнях за принципом “операція за операцією”, тобто декільком завданням дозволяється використовувати пристрій почергово. Якщо, скажемо, два завдання спробують виконати операцію введення-виведення на одному пристрої одночасно, тоді виникає черга й затримка.

Підхід до розподілу пристроїв впливає як на коректність, так і на ефективність роботи.

Стратегія розподілу впливає й на ефективність використання пристроїв, наприклад, того ж диску.

Ефективний розподіл периферійних пристроїв важко реалізувати з двох причин:

По-перше. за допомогою існуючих математичних методів не можливо провести необхідні дослідження й відшукати оптимальні способи розподілу декількох різних типів пристроїв для загального випадку рішення завдань.

По-друге, ефективність стратегії розподілу дуже важко виміряти.

Можна оцінити тільки загальний зовнішній прояв неправильного розподілу. Якщо ж при оцінці враховувати й вплив взаємодії з оперативною пам'яттю й центральним процесором, то аналітичні й емпіричні методи виявляються ще менш перспективними. Методи розподілу пристроїв

введення-виведення, контролерів і каналів сильно залежать від пристроїв.

**Програмні ресурси.** Часто в операційних системах є системи прикладних програм і бібліотеки програм користувачів. Будучи ресурсами, що підлягають розподілу, ці бібліотеки мають багато спільного з апаратними ресурсами.

Спільне використання інтерпретаторів, редакторів текстів та ін. можна організувати, якщо ці програми допускають паралельне використання.

Якщо в них робоча частина повністю відділена від даних й операцій запису в пам’ять і застосовується тільки до розділу даних, то такі програми допускають паралельне використання. Якщо ж програма не допускає паралельного використання, то кожен її екземпляр може бути в цей момент надано тільки одному користувачеві, так само, як і будь-який апаратний пристрій.

При паралельному використанні кожному користувачеві виділяється особистий екземпляр розділу даних, а з єдиним екземпляром робочої частини всі користувачі можуть працювати в режимі поділу.

## **Супервізор**

Традиційний підхід при проектуванні ОС полягає в тому, що безліч процесів, що виконують основні функції системи, підпорядковуються головній програмі, яку називають супервізором. Здійснюючи централізоване керування, супервізор зв'язує воєдино інші частини системи. Він організує спільну роботу програм, установлюючи привілеї або призначаючи покарання. Він забезпечує засоби зв'язку й синхронізації між процесами й фізичними пристроями. Звичайно й повідомлення, що передані від процесу до процесу, і запуск і закінчення роботи пристроїв, і сигнали від устаткування обробляються супервізором.

Супервізор, звичайно, управляє поділом всіх ресурсів і послуг системи між користувачами.

**Функції супервізора.** Функції супервізора можна загалом розділити на чотири частини:

* контроль і керування;
* організація зв'язків;
* захист й обмеження;
* обслуговуючі програми.

**Контроль і керування.** В обов'язки супервізора входить установлення послідовності й контроль керування завданнями у системі. Для обробки завдань використовується планування порядку виконання завдань, облік споживання ресурсів й інтерпретація мови керування завданнями. Ці функції можна реалізувати незалежно від супервізора або підпорядкувати йому.

Користувач описує свої вимоги на деякій мові керування завданнями. Супервізор інтерпретує ці вимоги й повідомляє різним програмам, що розподіляють ресурси, на які саме ресурси й завдання є запити. Розподільники ресурсів обслуговують запити, можливо, у порядку їх пріоритетності. Програма обліку запам'ятовує кількість спожитих ресурсів.

**Організація зв'язку.** У супервізорі передбачені засоби зв'язку між різними програмами. Коли дві незалежні програми такі, як файлова система й програми керування пам'яттю, хочуть зв'язатися один з одним, вони повинні просити супервізор установити контакт. Після встановлення первісного контакту, програмам дозволяється передавати один одному повідомлення, прийнятим у даній системі способом, скажімо, за допомогою “листів” і загальної “поштової скриньки”. Первісний зв'язок встановлюється через супервізор. Доступна тільки супервізору інформація дозволяє йому також встановлювати контакти між системними програмами й програмами користувачів.

**Захист й обмеження.** Для забезпечення гарантій виконання роботи супервізор повинен накласти деякі обмеження, як на систему, так і на користувачів. Є ряд програм, що оброблюють сигнали від апаратури, особливо відхилення від нормальних умов функціонування. Наприклад, під час роботи програми копіювання може виникнути кілька десятків різних особливих ситуацій, наприклад, збій при читанні або запису, неготовність дисководів до читання або запису, відсутність місця на дискеті для копійованого файлу тощо.

Коли надходить сигнал про помилку (звичайно це переривання), супервізор повинен прийняти рішення про повторення або навіть відхилення завдання. Для всіх цих ситуацій необхідно передбачити відповідні повідомлення й коригувальні дії.

Обмеження повинні накладати на час роботи програми й кількість видаваних нею результатів.

Супервізор може також організувати систему захисту за допомогою спеціальних “паролів”.

Доступ до захищених ресурсів дозволяється тільки за відповідним паролем.

З метою захисту супервізора, в деяких системах існують два режими роботи:

- режим супервізора; - робочий режим.

Завдяки такій обережності, супервізор має особливу владу над привілейованими частинами системи й програмами користувачів. Супервізор може захищати себе, зберігаючи життєво важливу інформацію в захищеній області пам'яті, доступної тільки в режимі супервізора.

**Обслуговування програми.** Крім розподілу ресурсів, супервізор виконує деякі з функцій системи. Він має набір обслуговуючих програм для аварійних ситуацій, для доступу до бібліотек програм, для обробки повідомлень від програм, що працюють в режимі супервізора. Всередині область пам'яті супервізора часто перебуває область, призначена для організації зв'язків і таблиці системи захисту. У ряді випадків, коли потрібно виконати складні дії, супервізор викликає спеціальні оброблюючі програми.

**Використання супервізора - це приклад побудови ОС на принципі централізованого керування.**

Ідеологія централізації ключових функцій системи під контролем супервізора має свої переваги й недоліки. Одна з переваг – проста реалізація захисту. Друга перевага – простота реалізації супервізора в цілому.

З погляду розробки, легше зосередити найважливіші системні функції у підпорядкуванні супервізора, ніж розподіляти їх по всій системі. Найкраще такий підхід виправдовує себе, коли супервізор роблять відносно невеликим. На жаль, є велика спокуса передати супервізору дуже багато функцій, перетворюючи його тим самим власне в ОС.

У централізованого супервізора є й істотні недоліки.