

2.2 ПРИЗНАЧЕННЯ І ФУНКЦІЇ ОПЕРАЦІЙНОЇ СИСТЕМИ

Сьогодні існує велика кількість різних типів операційних систем, що відрізняються сферами застосування, апаратними платформами і методами реалізації. Природно, це обумовлює і значні функціональні відмінності цих ОС. Навіть у конкретної операційної системи набір виконуваних функцій частенько визначити не так просто. Та функція, яка сьогодні виконується зовнішнім по відношенню до ОС компонентом, завтра може стати її невід'ємною частиною і навпаки.

Призначення ОС можна розкласти на такі складові.

1. **Зручність.** ОС робить використання комп'ютера простими і зручними.
2. **Ефективність.** ОС дозволяє ефективно використати ресурси комп'ютера.
3. **Можливість розвитку.** ОС має бути організована так, щоб вона дозволяла ефективну розробку, тестування і впровадження нових програм-додатків і системних функцій, причому це не повинно заважати нормальному функціонуванню ОС.

Кінцевий користувач може бути не знайомий з деталями апаратного забезпечення комп'ютера. Комп'ютер для нього є набором апаратних засобів і системного програмного забезпечення. Програму-додаток можна створити на будь-якій системі програмування.

Створення програми-додатка в машинних кодах, яка повністю відповідає за управління апаратним забезпеченням комп'ютера, є досить складним завданням. Для спрощення процедури створення додатків існує набір системних програм, у тому числі утиліти. Останні реалізують функції ОС, які часто використовуються для роботи з файлами і пристроями введення-виведення. Програміст використовує ці засоби для розробки власних програм, а вони під час свого виконання звертаються до утиліт для виконання певних функцій.

2.2.1 Основні функції операційних систем

Оскільки ОС виступає в ролі посередника, що полегшує програмістам і програмам-додаткам доступ до різних можливостей комп'ютерної системи, то при вивченні операційних систем дуже важливо з усього різноманіття виділити ті функції, які властиві всім операційним системам як класу продуктів. Список таких можливостей, які надаються ОС, можна представити таким чином.

1. Розробка програм. Сприяє програмістові при розробці програм, ОС надає йому різноманітні інструменти і сервіси. Ці сервіси реалізовані у вигляді спеціальних системних програм (системи компіляції, відладки тощо) і утиліт (обслуговування файлів і тому подібне), які підтримуються операційною системою, але не входять в її ядро. Ці програми є інструментальними засобами для розробки додатків.

2. Виконання програм. Для запуску програм необхідно виконати ряд дій. Потрібно завантажити в основну пам'ять коди команд і дані, ініціювати зовнішні пристрої і файли, а також підготувати інші ресурси. ОС виконує цю рутинну роботу замість користувача.

3. Доступ до пристроїв введення-виведення. Для управління роботою кожного пристрою введення-виведення потрібний свій особливий набір команд або сигналів управління. ОС надає користувачеві стандартний інтерфейс, який приховує усі ці деталі і забезпечує програмістові доступ до пристроїв введення-виведення за допомогою простих команд читання і запису.

4. Контрольований доступ до файлів. При роботі з файлами з боку ОС передбачаються не лише глибокі знання і розуміння роботи пристроїв введення-виведення (диски, CD і тому подібне), але і знання структур даних, які записані у файлах. Багатозадачні ОС, крім того, можуть забезпечувати роботу механізмів захисту при зверненні до файлів.

5. Системний доступ. ОС керує доступом до спільно доступної обчислювальної системи в цілому, а також до окремих її ресурсів. Вона може забезпечити захист ресурсів і даних від несанкціонованого звернення до них, а також розв'язувати конфліктні ситуації (у багатозадачних ОС це можуть бути драйвери віртуальних пристроїв).

6. Знаходження помилок і їх обробка. При роботі комп'ютерної системи можуть виникати різноманітні збої, до яких можна віднести як внутрішні, так і зовнішні, що виникають в апаратурі. Можливі також програмні помилки (арифметичне переповнювання, спроба звернення до елементу пам'яті, доступ до якої заборонений). Реакція ОС на помилку може бути різною (заміщення сегменту або сторінки пам'яті, просте повідомлення про помилку, аварійне завершення програми).

7. Облік використання ресурсів. Сучасна ОС повинна мати засоби обліку різних ресурсів і відображення параметрів продуктивності обчислювальної системи. Така інформація є у край важливою, особливо у зв'язку з підвищенням її пропускної спроможності.

Таким чином, можна зробити висновок, що операційна система комп'ютера є комплексом взаємозв'язаних програм, який діє як інтерфейс між додатками і користувачами з одного боку, і апаратурою комп'ютера з іншого боку. Відповідно до цього визначення ОС виконує в основному дві групи функцій:

- надання користувачеві або програмістові замість реальної апаратури комп'ютера розширеної віртуальної машини, з якою зручніше працювати і яку легше програмувати;

- підвищення ефективності використання комп'ютера шляхом раціонального управління його ресурсами відповідно до деякого критерію. Розглянемо деякі з цих функцій детальніше.

2.2.2 ОС як віртуальна машина

Для того щоб успішно розв'язувати свої задачі, сучасний користувач або навіть прикладний програміст може обійтися без досконального знання апаратних пристроїв комп'ютера. Йому не обов'язково знати, як функціонують різні електронні блоки і електромеханічні вузли комп'ютера. Більше того, дуже часто користувач може не знати навіть системи команд процесора. Користувач- програміст звик мати справу з потужними високорівневими функціями, які йому надає операційна система (рис. 2.1).

Так, наприклад, при роботі з диском програмістові, що пише додаток для роботи під управлінням ОС, або кінцевому користувачеві ОС досить представляти його у вигляді деякого набору файлів, кожен з яких має ім'я. Послідовність дій при роботі з файлом полягає в його відкритті, виконанні однієї або декількох операцій читання або запису, а потім в закритті файлу.

Такі деталі, як використовується при записі частотна модуляція або поточний стан двигуна механізму переміщення магнітних головок читання/запису, не повинні хвилювати програміста. Саме операційна система приховує від програміста велику

частину особливостей апаратури і надає можливість простої і зручної роботи з необхідними файлами.



Рисунок 2.1 – ОС як віртуальна машина

Якби програміст працював безпосередньо з апаратурою комп'ютера, без участі ОС, то для організації читання блоку даних з диска програмістові довелося б використати більше десятка команд з вказівкою певної кількості параметрів: номера блоку на диску, номера сектора на доріжці і т. п. А після завершення операції обміну з диском він повинен був би передбачити у своїй програмі аналіз результату виконаної операції.

Враховуючи, що контролер диска здатний розпізнавати більше двадцяти різних варіантів завершення операції, можна вважати програмування обміну з диском на рівні апаратури не найтривіальнішим завданням. Не менш обтяжливою виглядає і робота користувача, якби йому для читання файлу з терміналу потрібно було задавати числові адреси доріжок і секторів.

Операційна система позбавляє програмістів не лише від необхідності безпосередньо працювати з апаратурою дискового накопичувача, надаючи їм простий файловий інтерфейс, але і бере на себе усі інші рутинні операції, пов'язані з управлінням іншим апаратним обладнанням комп'ютера: фізичною пам'яттю, таймерами, принтерами тощо.

У результаті реальна машина, що здатна виконувати тільки невеликий набір елементарних дій, які визначаються її системою команд, перетворюється на

віртуальну машину, що виконує широкий набір набагато потужніших функцій. Віртуальна машина теж управляється командами, але це вже команди іншого, вищого рівня: видалити файл з певним ім'ям, запустити на виконання деяку прикладну програму, підвищити пріоритет завдання, вивести текст з файлу на друк. Таким чином, призначення ОС полягає в наданні користувачеві/програмістові деякої розширеної віртуальної машини, яку легше програмувати і з якою легше працювати, чим безпосередньо з апаратурою, що становить реальний комп'ютер.

2.2.3 ОС як диспетчер ресурсів

Операційна система не лише надає користувачам і програмістам зручний інтерфейс до апаратних засобів комп'ютера, але і є механізмом, що розподіляє ресурси комп'ютера.

До числа основних ресурсів сучасних обчислювальних систем можуть бути віднесені такі ресурси: процесори, основна пам'ять, таймери, набори даних, диски, принтери, мережеві пристрої і деякі інші. Ресурси розподіляються між процесами. Процес (завдання) є базовим поняттям більшості сучасних ОС і часто коротко визначається як програма в стадії виконання. Програма – це статичний об'єкт, що є файлом з кодами і даними. Процес – це динамічний об'єкт, який виникає в операційній системі після того, як користувач або сама операційна система вирішує «запустити програму на виконання», тобто створити нову одиницю обчислювальної роботи.

У багатьох сучасних ОС для позначення мінімальної одиниці роботи ОС використовують термін «потік», або «нитка», при цьому змінюється суть терміну «процес». Детальніше це ми розглянемо пізніше. В інших розділах ми будемо дотримуватися спрощеного тлумачення, відповідно до якого для позначення програми, що виконується, тільки термін «процес».

Управління ресурсами обчислювальної системи з метою найефективнішого їх використання є основним призначенням операційної системи. Наприклад, мультипрограмна операційна система організовує одночасне виконання відразу декількох процесів на одному комп'ютері, по черзі перемикаючи процесор з одного процесу на інший, виключаючи простої процесора, що викликаються зверненнями процесів до введення-виведення. ОС також відстежує і розв'язує конфлікти, що

виникають при зверненні декількох процесів до одного і того ж пристрою введення-виведення або до одних і тих же даних. Критерій ефективності, відповідно до якого ОС організовує управління ресурсами комп'ютера, може бути різним. Наприклад, в одних системах важливий такий критерій, як пропускна спроможність обчислювальної системи, в інших – час її реакції. Відповідно до вибраного критерію ефективності операційні системи по-різному організовують обчислювальний процес.

Управління ресурсами включає розв'язання наступних загальних, не залежних від типу ресурсу, задач:

- планування ресурсу – тобто визначення, якому процесу, коли і в якій кількості (якщо ресурс може виділятися частинами) слід виділити цей ресурс;
- задоволення запитів на ресурси;
- відстежування стану і облік використання ресурсу, тобто підтримка оперативної інформації про те, зайнятий чи вільний ресурс, і яка доля ресурсу вже розподілена;
- розв'язання конфліктів між процесами.

Для розв'язання цих загальних задач управління ресурсами різні ОС використовують різні алгоритми, особливості яких і визначають вигляд ОС в цілому, включаючи характеристики продуктивності, сферу застосування і навіть призначений для користувача інтерфейс. Наприклад, вживаний алгоритм управління процесором значною мірою визначає чи може ОС використовуватися як система розподілу часу, система пакетної обробки або система реального часу. Задача організації ефективного спільного використання ресурсів декількома процесами є дуже складне, і ця складність породжується в основному випадковим характером виникнення запитів на споживання ресурсів. У мультипрограмній системі утворюються черги заявок від одночасно виконуваних програм до розподілених ресурсів комп'ютера: процесора, сторінки пам'яті, до принтера, до диска. Операційна система організовує обслуговування цих черг за різними алгоритмами: в порядку вступу, на основі пріоритетів, кругового обслуговування тощо.

Таким чином, управління ресурсами складає важливу частину функцій будь-якої операційної системи, особливо мультипрограмної. Більшість функцій управління

ресурсами виконуються операційною системою автоматично і прикладному програмістові недоступні.

2.2.4 Управління процесами

Найважливішою частиною операційної системи, що безпосередньо впливає на функціонування обчислювальної машини, є підсистема управління процесами.

Для кожного новостворюваного процесу ОС генерує системні інформаційні структури, які містять дані про потреби процесу в ресурсах обчислювальної системи, а також про фактично виділені йому ресурси.

Щоб виконати процес, операційна система повинна призначити йому область оперативної пам'яті, в якій будуть розміщені коди і дані процесу, а також надати йому необхідну кількість процесорного часу. Крім того, процесу може знадобитися доступ до таких ресурсів, як файли і пристрої введення-виведення. В інформаційні структури процесу часто включаються допоміжні дані, що характеризують історію перебування процесу в системі. Наприклад, яку частку часу процес витратив на операції введення-виведення, а яку на обчислення, його поточний стан (активний або заблокований), міра привілейованості процесу (значення пріоритету). Дані такого роду можуть враховуватися операційною системою при ухваленні рішення про надання ресурсів процесу.

У мультипрограмній операційній системі одночасно може існувати декілька процесів. Частина процесів породжується за ініціативою користувачів та їх додатків. Такі процеси називають призначеними для користувача. Інші процеси, що називаються системними, ініціалізувалися самою операційною системою для виконання своїх функцій.

Оскільки процеси часто одночасно претендують на одні і ті ж ресурси, то в обов'язки ОС входить підтримка черг заявок процесів на ресурси, наприклад, черги до процесора, до принтера, до послідовного порту.

Важливим завданням операційної системи є захист ресурсів, що виділені цьому процесу, від інших процесів. Одним з ресурсів процесу, що найретельніше захищаються, є області оперативної пам'яті, в якій зберігаються коди і дані процесу. Сукупність усіх областей оперативної пам'яті, що виділені операційною системою

процесу, називається його *адресним простором*. Говорять, що кожен процес працює у своєму адресному просторі, маючи на увазі захист адресних просторів, який здійснюється ОС. Захищаються і інші типи ресурсів, такі як файли, зовнішні пристрої тощо. ОС може не лише захищати ресурси, що виділені одному процесу, але і організовувати їх спільне використання, наприклад, дозволяти доступ до деякої області пам'яті декільком процесам.

Упродовж періоду існування процесу його виконання може бути багаторазово перерване і продовжене. Для того щоб відновити виконання процесу, необхідно відновити стан його операційного середовища. Стан операційного середовища ідентифікується станом реєстрів і програмного лічильника, режимом роботи процесора, покажчиками на відкриті файли, інформацією про незавершені операції введення-виведення, кодами помилок виконуваних цим процесом системних викликів тощо. Ця інформація називається *контекстом процесу*. Говорять, що при зміні процесу відбувається перемикання контекстів.

Операційна система бере на себе також функції синхронізації процесів, що дозволяють процесу призупиняти своє виконання до настання якої-небудь події в системі, наприклад, завершення операції введення-виведення, здійснюється за її запитом операційної системою.

Для реалізації складних програмних комплексів корисно буває організувати їх роботу у вигляді декількох паралельних процесів, які періодично взаємодіють один з одним і обмінюються деякими даними. Оскільки ОС захищає ресурси процесів і не дозволяє одному процесу писати в пам'ять або читати з пам'яті іншого процесу, то для оперативної взаємодії процесів ОС повинна надавати особливі засоби, які називають засобами міжпроцесної взаємодії.

Таким чином, підсистема управління процесами планує виконання процесів, тобто розподіляє процесорний час між декількома одночасно існуючими в системі процесами, займається створенням і знищенням процесів, забезпечує процеси необхідними системними ресурсами, підтримує синхронізацію процесів, а також забезпечує взаємодію між процесами.

2.2.5 Управління пам'яттю

Пам'ять для процесу є таким же важливим ресурсом, як і процесор, оскільки процес може виконуватися процесором тільки в тому випадку, якщо його коди і дані (не обов'язково всі) знаходяться в оперативній пам'яті.

Управління пам'яттю включає розподіл наявної фізичної пам'яті між усіма існуючими в системі в даний момент процесами, завантаження кодів і цих процесів у відведені їм області пам'яті, налаштування адресно-залежних частин кодів процесу на фізичні адреси виділеної області, а також захист областей пам'яті кожного процесу.

Існує велика різноманітність алгоритмів розподілу пам'яті. Вони можуть відрізнятися, наприклад, кількістю областей пам'яті, що виділяються процесу. В одних випадках пам'ять виділяється процесу у вигляді однієї безперервної області, а в інших – у вигляді декількох несуміжних областей. У деяких системах розподіл пам'яті здійснюється сторінками фіксованого розміру, а в інших – сегментами змінної довжини.

Одним з найпопулярніших способів управління пам'яттю в сучасних операційних системах є так звана віртуальна пам'ять. Наявність в ОС механізму віртуальної пам'яті дозволяє програмістові писати програму так, ніби в його розпорядженні є однорідна оперативна пам'ять великого об'єму, що часто істотно перевищує об'єм наявної фізичної пам'яті. Насправді усі дані, що використовуються програмою, зберігаються на диску і при необхідності частинами (сегментами або сторінками) відображаються у фізичну пам'ять.

При переміщенні кодів і даних між оперативною пам'яттю і диском підсистема віртуальної пам'яті виконує трансляцію віртуальних адрес, отриманих в результаті компіляції і компонування програми, у фізичні адреси елементів оперативної пам'яті. Захист пам'яті – це вибіркова здатність оберігати виконуване завдання від запису або читання пам'яті, призначеної іншому завданню. Правильно написані програми не намагаються звертатися до пам'яті, призначеної іншим процесам. Проте реальні програми часто містять помилки, в результаті яких такі спроби іноді робляться. Засоби захисту пам'яті, які реалізовані в операційній системі, повинні забороняти несанкціонований доступ процесів до чужих областей пам'яті.

Таким чином, функціями ОС з управління пам'яттю є: відстежування вільної і зайнятої пам'яті; виділення пам'яті процесам і звільнення пам'яті при завершенні процесів; захист пам'яті; витіснення процесів з оперативної пам'яті на диск, а також налаштування адрес програми на конкретну область фізичної пам'яті.

2.2.6 Управління файлами і зовнішніми пристроями

Здатність ОС до «екранування» складнощів реальної апаратури дуже яскраво проявляється в одній з основних підсистем ОС – файловій системі. Операційна система віртуалізує окремий набір даних, що зберігаються на зовнішньому накопичувачі, у вигляді файлу – простої неструктурованої послідовності байтів, що має символічне ім'я. Для зручності роботи з даними файли групуються в каталоги, які, у свою чергу, утворюють групи – каталоги вищого рівня. Користувач може за допомогою ОС виконувати над файлами і каталогами такі дії, як пошук за іменем, видалення, виведення вмісту на зовнішній пристрій (наприклад, на дисплей), зміну і збереження вмісту.

Файлова система ОС виконує перетворення символічних імен файлів, з якими працює користувач або прикладний програміст, у фізичні адреси даних на диску, організовує спільний доступ до файлів, захищає їх від несанкціонованого доступу.

При виконанні своїх функцій файлова система тісно взаємодіє з підсистемою управління зовнішніми пристроями, яка за запитами файлової системи здійснює передачу даних між дисками і оперативною пам'яттю.

Підсистема управління зовнішніми пристроями, що називається також підсистемою введення-виведення, виконує роль інтерфейсу до усіх пристроїв, підключених до комп'ютера. Спектр цих пристроїв дуже великий. Номенклатура накопичувачів, що випускаються, на жорстких, гнучких і оптичних дисках, принтерів, сканерів, моніторів, плотерів, модемів, мережевих адаптерів і більше спеціальних облаштувань введення-виведення, таких як, наприклад, аналого-цифрові перетворювачі, може налічувати сотні моделей. Ці моделі можуть істотно відрізнитися набором і послідовністю команд, за допомогою яких здійснюється обмін інформацією з процесором і пам'яттю комп'ютера, швидкістю роботи, кодуванням

передаваних даних, можливістю спільного використання і певною кількістю інших деталей.

Програма, що управляє конкретною моделлю зовнішнього пристрою і враховує усі його особливості, називається *драйвером* цього пристрою (від англійського drive – управляти, вести). Драйвер може управляти єдиною моделлю пристрою, наприклад модемом U-1496E компаній ZyXEL, або ж групою пристроїв певного типу. Для користувача дуже важливо, щоб операційна система включала якомога більше різноманітних драйверів, оскільки це гарантує можливість підключення до комп'ютера великого числа зовнішніх пристроїв різних виробників. Від наявності відповідних драйверів багато в чому залежить успіх операційної системи на ринку. Наприклад, відсутність багатьох необхідних драйверів зовнішніх пристроїв була однією з причин низької популярності OS/2. Створенням драйверів пристроїв займаються як розробники конкретної ОС, так і фахівці компаній, що випускають зовнішні пристрої. Операційна система повинна добре підтримувати певний інтерфейс між драйверами і іншою частиною ОС, щоб розробники з компаній-виробників пристроїв введення- виведення могли поставляти разом зі своїми пристроями драйвери для цієї операційної системи.

Прикладні програмісти можуть користуватися інтерфейсом драйверів при розробці своїх програм. Але це не дуже зручно – такий інтерфейс є низькорівневими операціями, обтяженими великою кількістю деталей.

Підтримка високорівневого уніфікованого інтерфейсу прикладного програмування до різноманітних пристроїв введення-виведення є одним з найважливіших завдань ОС. З часу появи ОС UNIX такий уніфікований інтерфейс у більшості ОС будується на основі концепції файлового доступу. Ця концепція полягає в тому, що обмін з будь-яким зовнішнім пристроєм виглядає як обмін з файлом, що має ім'я і є неструктурованою послідовністю байтів. Файлом може виступати як реальний файл на диску, так і алфавітно-цифровий термінал, друкуючий пристрій або мережевий адаптер.

2.2.7 Захист даних і адміністрування

Безпека даних обчислювальної системи забезпечується засобами відмовостійкості ОС, спрямованими на захист від збоїв і відмов апаратури і помилок програмного забезпечення, а також засобами захисту від несанкціонованого доступу.

Першою при захисті даних від несанкціонованого доступу є процедура логічного входу. ОС повинна переконатися, що до системи намагається увійти користувач, вхід якого дозволений адміністратором. Функції захисту ОС взагалі дуже тісно пов'язані з функціями адміністрування, оскільки саме адміністратор визначає права користувачів при їх зверненні до різних ресурсів системи – файлів, каталогів, принтерів, сканерам тощо. Крім того, адміністратор обмежує можливості користувачів у виконанні тих або інших системних дій.

Наприклад, користувачеві може бути заборонено виконувати процедуру завершення роботи ОС, встановлювати системний час, завершувати чужі процеси, створювати облікові записи користувачів, змінювати права доступу до деяких каталогів і файлів. Адміністратор може також урізувати можливості інтерфейсу користувача, прибравши, наприклад, деякі пункти з меню операційної системи, що виводиться на дисплей користувача.

Важливим засобом захисту даних є *функції аудиту ОС*, що полягають у фіксації усіх подій, від яких залежить безпека системи. Наприклад, спроби вдалого і невдалого логічного входу в систему, операції доступу до деяких каталогів і файлів, використання принтерів тощо. Список подій, які необхідно відстежувати, визначає адміністратор ОС.

Підтримка відмовостійкості реалізується ОС, як правило, на основі резервування. Найчастіше у функції ОС входить підтримка декількох копій даних на різних дисках або різних дискових накопичувачах. Резервуються також принтери і інші пристрої введення-виведення. При відмові одного з надлишкових пристроїв ОС повинна швидко і прозорим для користувача способом зробити реконфігурацію системи і продовжити роботу з резервним пристроєм. Особливим випадком забезпечення відмовостійкості є використання декількох процесорів, тобто мультипроцесування, коли система продовжує роботу при відмові одного з процесорів. Підтримка відмовостійкості входить в

обов'язки системного адміністратора. До складу ОС входять утиліти, що дозволяють адміністраторові виконувати регулярні операції резервного копіювання для забезпечення швидкого відновлення важливих даних.

2.2.8 Інтерфейс прикладного програмування

Прикладні програмісти використовують у своїх додатках звернення до ОС, коли для виконання тих або інших дій їм потрібен особливий статус, який має тільки операційна система. Наприклад, у більшості сучасних ОС усі дії, які пов'язані з управлінням апаратними засобами комп'ютера, може виконувати тільки ОС. Окрім цих функцій прикладний програміст може скористатися набором сервісних функцій ОС, які спрощують написання додатків. Функції такого типу реалізують універсальні дії, що часто вимагаються в різних додатках, такі, наприклад, як обробка текстових рядків. Ці функції могли б бути виконані і самим додатком, проте набагато простіше використати вже готові, відлагоджені процедури, включені до складу операційної системи. В той же час навіть за наявності в ОС відповідної функції програміст може реалізувати її самостійно в рамках додатка, якщо запропонований операційною системою варіант його не цілком влаштовує.

Можливості операційної системи, що доступні прикладному програмістові у вигляді набору функцій, називаються *інтерфейсом прикладного програмування* (Application Programming Interface, **API**). Від кінцевого користувача ці функції приховані за оболонкою алфавітно-цифрового або графічного інтерфейсу користувача.

Для розробників додатків усі особливості конкретної операційної системи представлені особливостями її API. Тому операційні системи з різною внутрішньою організацією, але з однаковим набором функцій API, здаються нам однією і тією ж ОС, що спрощує стандартизацію операційних систем і забезпечує переносимість додатків між різними ОС, такими, що відповідають певному стандарту на API.

Наприклад, наслідування загальних стандартів API UNIX, одним з яких є стандарт **Posix** (Portable Operating System Interface – Інтерфейс переносимих операційних систем), дозволяє говорити про деяку узагальнену операційну систему

UNIX, хоча численні версії цієї ОС від різних виробників іноді істотно відрізняються внутрішньою організацією.

Додатки виконують звернення до функцій API за допомогою системних викликів. Спосіб, яким додаток отримує послуги операційної системи, дуже схожий на виклик підпрограм. Інформація, яка потрібна ОС і яка складається з ідентифікатора команди і даних, поміщається в певне місце пам'яті, в реєстри і/або стек. Потім управління передається операційній системі, яка виконує необхідну функцію і повертає результати через пам'ять, реєстри або стеки. Якщо операція проведена не успішно, то результат включає індикацію помилки.

2.2.9 Інтерфейс користувача

Операційна система повинна забезпечувати зручний інтерфейс не лише для застосовних програм, але і для людини, працюючої за терміналом. Ця людина може бути кінцевим користувачем, адміністратором ОС або програмістом.

У ранніх операційних системах пакетного режиму функції інтерфейсу користувача були зведені до мінімуму і не вимагали наявності терміналу. Команди мови управління завданнями набивалися на перфокарти, а результати виводилися на друкуючий пристрій.

Сучасні ОС підтримують розвинені функції інтерфейсу користувача для інтерактивної роботи за терміналами двох типів: алфавітно-цифровими і графічними.

При роботі за алфавітно-цифровим терміналом користувач має у своєму розпорядженні систему команд, потужність яких відображає функціональні можливості цієї ОС. Командна мова ОС дозволяє запускати і зупиняти додатки, виконувати різні операції з файлами і каталогами, отримувати інформацію про стан ОС (кількість працюючих процесів, об'єм вільного простору на дисках тощо), адмініструвати систему. Команди можуть вводитися не лише в інтерактивному режимі з терміналу, але і прочитуватися з так званого командного файлу, що містить деяку послідовність команд.

Програмний модуль ОС, відповідальний за читання окремих команд або послідовності команд з командного файлу, іноді називають командним інтерпретатором.

Введення команди може бути спрощене, якщо операційна система підтримує графічний інтерфейс користувача. В цьому випадку користувач для виконання потрібної дії за допомогою миші вибирає на екрані потрібний пункт меню або графічний символ.