

Тема 2. Взаємодія програм з операційною системою

План

1. Архітектура Windows
2. Виконання програм різних типів
3. Прикладний програмний інтерфейс Windows

1. Архітектура Windows

В операційних системах Windows NT (2000, XP, Server 2003 та ін.) код користувацьких додатків відокремлений від коду ядра операційної системи. Код прикладних програм виконується в непривілейованому режимі користувача (user mode) і обмежений в доступі до ресурсів, а код ядра ОС виконується в привілейованому режимі ядра (kernel mode) і має доступ до всіх ресурсів системи. Проте, коли програма режиму користувача здійснює системні виклики, то відповідний потік переводиться в режим ядра, а по завершенні системного виклику повертається в режим користувача.

Операційна система складається із двох основних частин: ядра (NT Executive) і підсистем оточення, які працюють в режимі користувача. Ядро керує процесами, пам'яттю, файловою системою, введенням-виведенням та ін. Підсистеми оточення є окремими процесами, які дозволяють користувачеві виконувати певні системні функції.

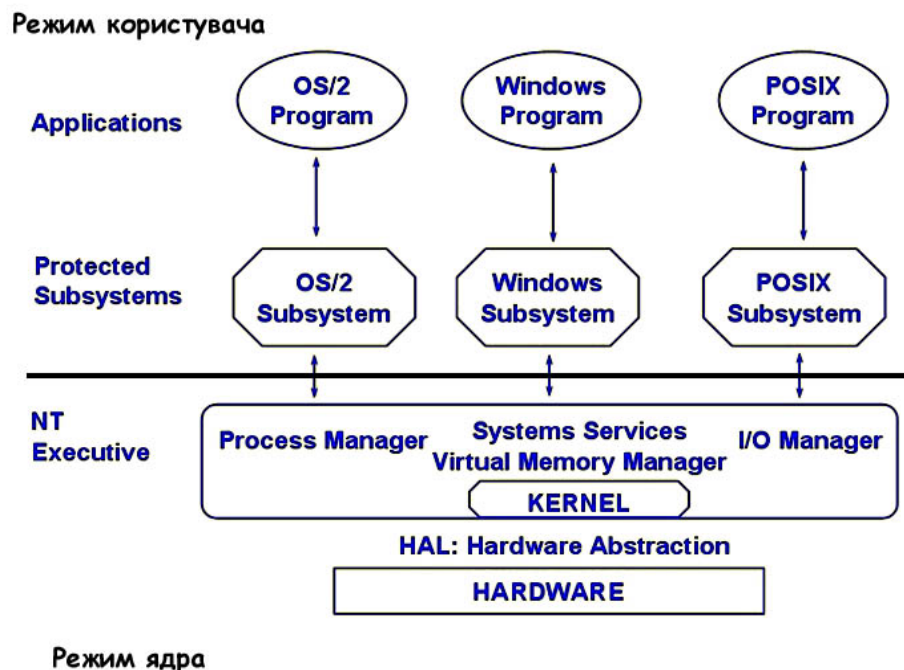


Рис. 2.1. Архітектура Windows.

Виділяють окремий рівень ОС – рівень апаратних абстракцій HAL (Hardware Abstraction Layer). Функція цього рівня полягає в тому, щоб надати абстрактні (машинно-незалежні) апаратні пристрої (служби), які використовуються іншою частиною операційної системи. Наявність HAL забезпечує легку міграцію ОС на інші платформи (змін зазнає лише HAL, який адаптується під кожен апаратну платформу).

2. Виконання програм різних типів

Взаємодія програми з операційною системою залежить від типу програми та операційної системи. Під керуванням Windows сімейства NT можуть виконуватися програми кількох типів: 16-розрядні MS-DOS, 16-розрядні Windows 3.1, 32-розрядні Windows, 64-розрядні Windows, 32-розрядні POSIX, 32-розрядні OS/2.

16-розрядні програми реального режиму під керуванням MS-DOS. Операційна система MS DOS – це однозадачна операційна система. Після запуску програми на виконання керування повністю передається прикладній програмі, яка отримує доступ до всіх ресурсів комп'ютера. Програма може взаємодіяти з апаратними засобами (здійснювати введення-виведення) через функції операційної системи (т. зв. програмні переривання), функції базової системи введення-виведення (BIOS) або IOS), а також напряму (через порти введення-виведення).

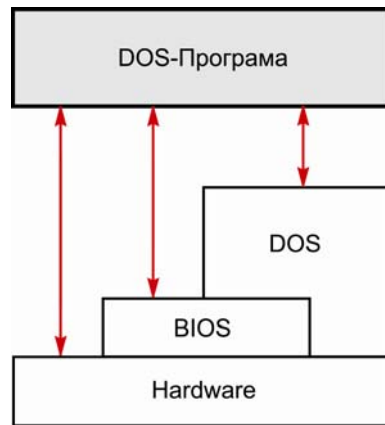


Рис. 2.2. Виконання програми під керуванням DOS.

16-розрядні програми реального режиму під керуванням Windows. Операційна система Windows NT (2000 і наступні) запускає DOS-програми у захищеному режимі. Спочатку запускається Win32-процес ntvdm – віртуальна машина DOS (NT Virtual DOS machine), який сканує програму MS-DOS і виконує її команди. Якщо програма виконує звичайні команди, вони виконуються процесором. Якщо програма використовує системні виклики (переривання DOS або BIOS), то вони перенаправляються віртуальній машині, яка перехоплює ці переривання і формує відповідні їм системні виклики Win API (метод трампліна чи батута). Нарешті, якщо програма хоче оминути ОС і напряму звернутися до портів введення-виведення або відеопам'яті, що заборонено в захищеному режимі, то такі дії блокуються і виконання програми припиняється з відповідним повідомленням.

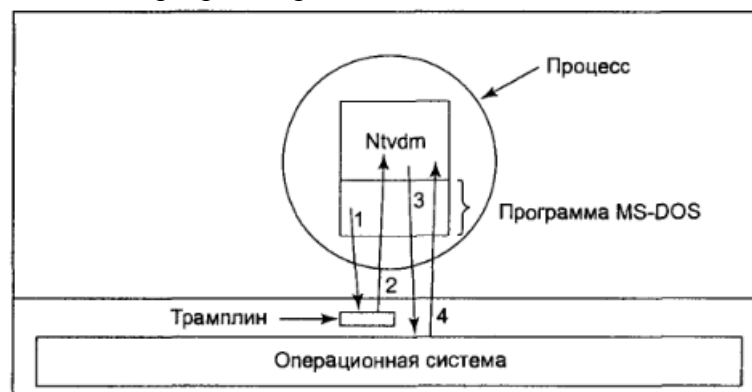


Рис. 2.3. Виконання програми на віртуальній машині NTVDM.

32-розрядні програми під керуванням Windows. Операційна система Windows використовує концепцію віртуальної пам'яті. Для кожного процесу створюється окремий лінійний адресний простір, який не перетинається з адресними просторами інших програм та самої ОС. Відображенням адрес віртуальної пам'яті на адреси наявної в комп'ютері фізичної пам'яті займається диспетчер пам'яті. Якщо фізична пам'ять (мікросхеми пам'яті) вичерпується, вступає в дію механізм підкачки – окремі блоки пам'яті вивільнюються і зберігаються в спеціальному файлі на диску (swapping). Таким чином, всім процесам пам'яті вистачає.

Кожна сторінка пам'яті позначається тегом, який визначає режим роботи процесора з цією сторінкою (користувача або ядра). Якщо програма використовує системні виклики, код і дані якої знаходяться в ядрі ОС, відбувається перемикання програми в режим ядра. Після виконання коду системної функції здійснюється зворотний процес – перемикання в режим користувача. Таким чином прикладна програма використовує обидва режими, але команди програми користувача завжди виконуються в режимі користувача, а команди ядра – в режимі ядра.

3. Прикладний програмний інтерфейс Windows

Операційна система Windows NT не дозволяє прикладним програмам звертатись до апаратних ресурсів системи. Звичайно ж, Windows має власний набір внутрішніх функцій (як у DOS чи BIOS), але вони не публікуються, і можуть змінюватися від версії до версії. Натомість Microsoft визначила набір функціональних викликів, через які програми користувача одержують доступ до системних ресурсів. Ці виклики одержали назву прикладного програмного інтерфейсу (Application Programming Interface, API або Win32 API). Ці виклики детально задокументовані і опубліковані. Вони являють собою бібліотечні процедури, які виконують системні функції в режимі ядра або в просторі користувача.

Опубліковані функції Win32 API не змінюються, лише доповнюються новими функціями. Це дає можливість виконувати програми написані для різних операційних системах, наприклад, Windows 9X (95, 98, Me), Windows NT (2000, XP, 2003 та ін.), Windows CE (palmtops, вбудовані системи та ін.). Win32-програми можуть також виконуватись в середовищі Win64.

Windows API містить буквально тисячі функцій, які умовно можна поділити на кілька основних категорій:

- базові сервіси (Base Services);
- сервіси компонентів (Component Services);
- інтерфейс користувача (User Interface);
- інтерфейс графічних пристроїв (Graphics Device Interface);
- мережеві сервіси (Networking);
- повідомлення та співпраця (Messaging and Collaboration);
- мультимедійні сервіси (Multimedia Services);
- Web-сервіси (Web Services).

Основні сервіси забезпечують доступ до фундаментальних ресурсів Windows. Вони включають файлові системи, пристрої, процеси, нитки, реєстр та обробку помилок.

Сервіси компонентів надають засоби для створення додатків, що можуть взаємодіяти між собою на основі COM-технології (Microsoft Component Object Model). COM – це основна технологія для OLE та ActiveX (Internet-enabled components). COM-додатки мають задокументований і опублікований в системі інтерфейс, через який можна працювати з об'єктами цих додатків. Прикладом можуть бути компоненти Delphi – сервери Word, Excel, Power Point і т.д. На жаль, вони практично не задокументовані.

Інтерфейс користувача забезпечує функціональність для створення вікон і елементів керування.

Інтерфейс графічних пристроїв GDI (Graphics Device Interface) – інтерфейс (бібліотека функцій) для роботи з графікою. Містить функції для взаємодії з графічними пристроями (рисування ліній тексту, фігур), наприклад монітор, принтер або графічний файл. Виконує роль посередника між програмою і драйверами дисплею і принтера, забезпечує незалежність від апаратних графічних пристроїв.

Мережеві сервіси – це програмний інтерфейс мережевого програмного забезпечення. Забезпечує незалежну від протоколів взаємодію програм через мережу.