

# ОПЕРАЦІЙНА СИСТЕМА

## Файли і файлові структури

Всі програми і дані в зовнішній пам'яті комп'ютера зберігаються у вигляді *файлів*. *Файл* – сукупність даних, яка має своє унікальне ім'я і зберігається на зовні запам'ятовуючих пристроях.

*Ім'я файла* складається з двох частин назви файла та розширення. Назва і розширення відокремлюються крапкою. Правила утворення назви файла:

- максимальна кількість символів в імені 256;
- використовуються літери кирилиці або латиниці, арабські цифри та деякі інші символи;
- назва файла не може містити символи: \ \* / : ? “ < >.

Розширення файла вказує на тип даних які зберігаються в файлі.

Файли можуть бути різних розмірів.

На фізичному рівні блоки одного файла можуть розташовуватись неперервною областю або зберігатися окремими фрагментами. Процес «збирання» окремих фрагментів файлів разом називається *дефрагментацією*.

Всі дані про розташування файлів на зовнішньому запам'ятовуючому пристрої зберігаються в каталозі (папці) або директорії. *Каталог* – список елементів, кожний з яких описує характеристики конкретного файла, що використовуються для організації доступу до нього: ім'я файла, його тип, місцеположення на зовнішньому запам'ятовуючому пристрої, розмір. Каталогів може бути велика кількість, і вони зв'язані у структури, наприклад, ієрархічну систему каталогів. Кожний каталог розглядається як файл і має власне ім'я. Повне ім'я каталогу або файла в такій структурі задається шляхом переходів між каталогами і файлами в логічній структурі каталогів.

Кожний файл також характеризується *атрибутами*: *тільки для читання* (заборона на зміну вмісту файла та його вилучення), *прихований* (заборона на виведення на екран при перегляді вмісту каталогу), *архівний* (позначення файла або папки для архівація, наприклад, деякі програми використовують цей атрибут для позначення групи файлів або папок, що підлягають архівуванню).

## Імена зовнішніх запам'ятовуючих пристроїв

Імена зовнішніх запам'ятовуючих пристроїв позначаються літерами латинського алфавіту з двокрапками: А:, В:, С:, D:, Е:, F: і т.д.

А:, В: - дисководи для гнучких дисків.

С:, D: і т.д. – жорсткі диски або логічні диски вінчестера.

Наступна літера після імені останнього розділу жорсткого диску – дисковод для читання оптичних дисків.

Наступна літера після імені оптичного дисководу – ім'я тимчасово під'єднаних пристроїв через USB-порт.

## ***Поняття операційної системи***

*Операційна система* – комплекс програмних засобів і даних, які забезпечують керування роботою апаратної та інформаційної складових інформаційної системи, координують їх взаємодію, забезпечують виконання функції передавання команд і даних між користувачем і комп'ютером.

*Функції операційної системи:*

- розподіл ресурсів, управління складовими апаратного забезпечення інформаційної системи;
- завантаження програм на виконання;
- управління виконанням прикладних програм;
- забезпечення організації збереження даних для подальшого їх використання;
- здійснення зв'язку між користувачем і комп'ютером.

## ***Типи операційних систем***

Операційні системи можна класифікувати за різними ознаками:

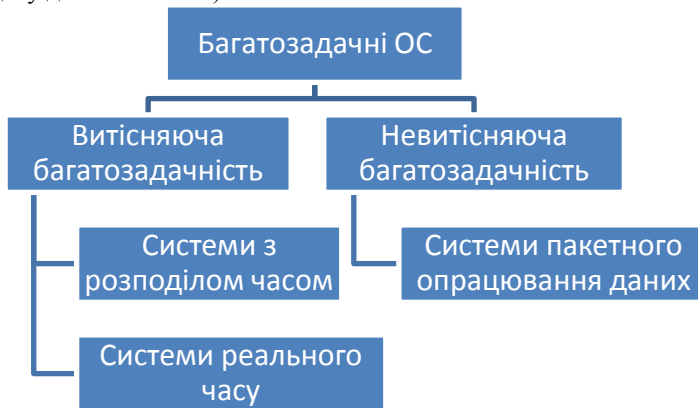
1) *За особливостями реалізації алгоритмів керування основними ресурсами комп'ютера* (процесора, пам'яттю, зовнішніми пристроями).

2) *За особливостями апаратних платформ* (персональних комп'ютерів, міні-комп'ютерів, майнфреймів, кластерів (сукупність декількох інформаційних систем, які працюють сумісно для виконання спільних додатків, і для користувачів представляються єдиною системою) і мереж комп'ютерів). Окремо виділяють ОС, розроблені таким чином, щоб вони могли легко переноситись з комп'ютера одного типу на комп'ютер іншого типу, так звані *мобільні* ОС, наприклад, UNIX.

3) *За кількістю одночасно виконуваних задач* (однозадачні (MS-DOS, MSX-DOS, CP/M) – наступна програма запускається після завершення виконання попередньої; багатозадачні (OS/2, UNIX) – запускається на виконання кілька програм одночасно і забезпечується

при цьому розподіл між ними системних ресурсів і створюється ілюзія одночасного виконання кількох програм).

4) *За способом реалізації багатозадачності: витісняюча* (перемикання процесора з одного процесу на інший здійснюється за допомогою ОС, а не процесу, що в даний момент виконується) і *невитісняюча* (процес, що вданий момент виконується процесором, буде виконуватись до тих пір, поки за допомогою цього процесу не буде здійснено передачу управління ОС для вибору наступного процесу для виконання).



*Системи з розподілом часом* (UNIX, VMS) – системи в яких кожній задачі виділяється квант процесорного часу і жодна задача не займає процесор на довго.

*Системи реального часу* (QNX, RT/11) – системи у яких є фіксований набір наперед розроблених програм, а вибір програми на виконання здійснюється на основі аналізу поточного стану контролюваного об'єкта. Ці системи, в основному, використовуються для управління різними технічними об'єктами: наукова експериментальна установка, доменний процес.

*Системи пакетного опрацювання даних* (OS EC) – системи в яких спочатку формується пакет задач з вимогами до системних ресурсів, а потім для одночасного виконання вибираються задачі, в яких різні вимоги до системних ресурсів, наприклад, обчислювальні задачі і задачі з інтенсивним введенням-виведенням даних. Вибір наступних задач залежить від вільних системних ресурсів, тобто вибираються «вигідні» задачі. Критерієм ефективності таких систем є максимальна кількість задач виконаних за одиницю часу.

5) *За кількістю працюючих користувачів* – однокористувацькі і багатокористувацькі. В багатористувацьких ОС передбачено захист даних кожного користувача від несанкціонованого доступу інших користувачів.

## Складові операційної системи

*Ядро* – основний компонент операційної системи, контролює всі події, які відбуваються в інформаційній системі і забезпечує спільне використання ресурсів виконуваними програмами.

*Драйвери пристроїв* – програми, які забезпечують керування апаратними пристроями комп'ютера і обмін даними між процесором і периферійними пристроями.

*Оболонка* (командний інтерпретатор) – модуль ОС, за допомогою якого здійснюється введення команд, які інтерпретуються оболонкою і передаються на виконання ядру. Може бути реалізований у вигляді окремого модуля або вбудований до ядра ОС.

*Системні утиліти* – програми призначені для виконання певних службових операцій, що непередбачені оболонкою користувача.



## Процеси і потоки

Всі комп'ютери можуть одночасно виконувати кілька операцій, наприклад, читання з диску і виведення тексту на екран монітору або принтеру. В багатозадачних системах процесор перемикається між програмами.

ОС здійснює модель послідовних процесів. *Процес* – програмний модуль, що може виконуватись центральним процесором, з даними, необхідними для виконання: вмістом стеку, вмістом адресного й інших регістрів процесора тощо.

Запуск програми на виконання: після вказування імені файлу, що містить виконуваний код, здійснюються наступні дії:



такі дані:

- › про стан процесу;
- › лічильник команд;
- › розподіл пам'яті;
- › стан відкритих файлів;
- › вказівник стеку;
- › про використання і розподіл ресурсів.

*Стани процесу:*

*Новий* – процес щойно створений.

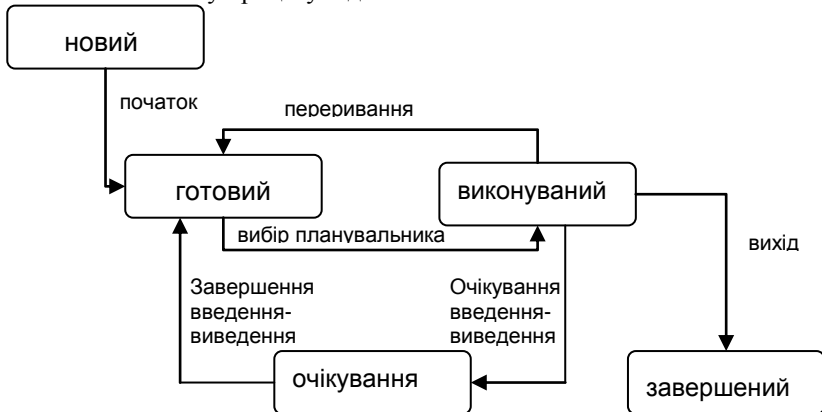
*Виконуваний* – активний стан процесу, під час якого процес керує всіма необхідними ресурсами і безпосередньо виконується.

*Очікуваний* – пасивний стан процесу, процес заблокований, він не може виконуватись внаслідок очікування певної події: завершення операції введення-виведення, отримання повідомлення від іншого процесу, звільнення необхідного йому ресурсу.

*Готовий* – пасивний стан процесу, процес заблокований у зв'язку із зовнішніми стосовно нього обставинами: процес має всі необхідні для нього ресурси, він готовий виконуватись, однак процесор зайнятий виконанням іншого процесу.

*Завершений* – процес завершив свою роботу.

Зміни стану процесу під час його виконання



Підсистеми керування процесами планує виконання процесів, тобто розподіляє процесорний час між кількома одночасно існуючими в системі процесами, а також займається створенням і знищенням процесів, забезпечує процеси необхідними системними ресурсами, підтримує взаємодію між процесами. Розподіл процесів між наявними ресурсами називають *плануванням процесів*. Планування процесів містить у собі вирішення наступних задач:

- › визначення моменту часу для зміни виконуваного процесу;
- › вибір процесу на виконання з черги готових процесів;
- › переключення контекстів «старого» і «нового» процесів.

В різних операційних системах по різному вирішується питання вибору з черги процесу на виконання.

В ОС забезпечується автономне виконання процесів, їх ізолюваність один від одного: кожний процес виконується у своєму віртуальному просторі, для кожного процесу виділяються свої ресурси. Однак задачі, розв'язувана в рамках одного процесу, може мати внутрішній паралелізм, що дозволяє прискорити її вирішення. Для цього використовується *багатопоточний механізм*. Потік – об'єкт, які отримують час процесора на тих же умовах, що і процеси.

## ***Управління пам'яттю***

Пам'ять комп'ютера має ієрархічну структуру.



Частина ОС, яка відповідає за управління пам'яттю, називається *менеджером пам'яті*.

Функції ОС щодо керування пам'яттю:

- відстеження вільної і зайнятої пам'яті;
- виділення пам'яті процесам і звільнення пам'яті при завершенні процесів;
- витіснення процесів з оперативної пам'яті на диск, якщо розміри основної пам'яті недоступні для розміщення в ній усіх процесів, і повернення їх у оперативну пам'ять, коли в ній звільняється місце;
- налагодження адрес програми на конкретну область фізичної пам'яті.

Для управління пам'яттю використовують стратегії *без використання зовнішньої пам'яті* і *з використанням зовнішньої пам'яті*.

Стратегії без використання зовнішньої пам'яті:

- *розподіл пам'яті фіксованими розділами* (поділ пам'яті на розділи фіксованої величини, кожний розділ призначено для виконання однієї задачі);
- *розподіл пам'яті розділами змінного розміру* (при завантаженні задачі виділяється необхідна їй пам'ять при умові, що вистачає місця в пам'яті, інакше задача не приймається на виконання і очікує у черзі вивільнення пам'яті).

Стратегії з використанням зовнішньої пам'яті:

- *свопінг* або *підкачка*: кожний процес повністю переноситься до оперативної пам'яті, виконується деякий час, потім повністю повертається на зовнішній запам'ятовуючий пристрій;
- *використання віртуальної пам'яті*: збереження частин програм, які використовуються в даний момент, в оперативній пам'яті, а інші – на зовнішніх запам'ятовуючих пристроях.

## ***Організація введення/виведення***

Як було зазначено раніше, однією з основних функцій ОС забезпечення процесів введення/виведення.

Пристрої введення/виведення поділяють на дві категорії: *блочні пристрої* і *символьні пристрої*.

В *блочних пристроях* дані зберігаються у вигляді блоків фіксованого обсягу, причому у кожного блоку є своя адреса. Дані кожного блоку може бути прочитана незалежно від інших. До блочних пристроїв відносяться, наприклад, магнітний диск.

В *символьних пристроях* приймання даних відбувається у вигляді потоку символів без блоків. Ці пристрої **не мають адреси** і не здійснює пошук даних.

Одним з основних механізмів забезпечення роботи зовні пристроїв є використання переривань. *Переривання* – спеціальний сигнал, або команда, процесору для негайного перемикавання процесора на опрацювання події, що сформувала переривання (опрацювання переривання) за певним алгоритмом.

Пристрої введення\виведення складаються з *механічної* і *електронної* частини.

*Механічна* частина – сам пристрій, а *електронна* – контролер або адаптер. Контролери або встановлюються на материнській платі або розташовані на самому пристрої. За допомогою контролера здійснюється перетворення даних у форму зручну для опрацювання пристроєм. Наприклад, за допомогою контролера монітора здійснюється зчитування даних з пам'яті у вигляді байт, що містять символи, які потрібно відобразити на екрані монітора. В контролері відбувається перетворення зчитаних даних у сигнали, які використовуються для модуляції променя електронної трубки.

*Принципи* програмного забезпечення введення\виведення:

- › *незалежність від пристроїв* – використання будь-яких пристроїв введення\виведення в програмі здійснюється без вказування конкретного пристрою, всі проблеми пов'язані із визначенням відмінностей цих пристроїв здійснюється за допомогою ОС;



- ▶ *єдинообразне іменування* – імена пристроїв або файлів повинні бути просто текстовим рядком або цілим числом і не залежати від фізичного пристрою;
- ▶ *опрацювання помилок в роботі пристроїв здійснюється від нижчого до вищого рівня програмного забезпечення* – помилки повинні опрацьовуватись як можна ближче до апаратури, тільки якщо за допомогою нижнього рівня програмного забезпечення не можна усунути помилку здійснюється передавання даних про помилку до вищого рівня програмного забезпечення;
- ▶ *буферізації* – копіювання даних великого обсягу до **буферу**, а потім аналіз їх вмісту.
- ▶ *використання блокуючих (синхронних) і неблокуючих (асинхронних) способів передавання даних* – при неблокуючому передаванні даних центральний процесор починає операцію передавання даних і перемикається на інший процес, поки не настане переривання завершення операції передавання даних, при блокую чому – виконання програми автоматично припиняється поки дані не потраплять у буфер програми.
- ▶ *використання виділених пристроїв і пристроїв колективного користування* – при використанні пристроїв колективного користування з ними може працювати одночасно кілька користувачів, а з виділеними пристроями – тільки окремі користувачі, до тих пір , поки не завершить свою роботу один користувач, пристрій не може бути надано у користування іншому.

*Способи введення\виведення:*

- ▶ *програмне введення\виведення* – здійснюється за допомогою центрального процесора, який очікує готовності пристрою під час проведення операції;
- ▶ *керувий перериванням введення\виведення* – центральний процесор починає передавання введення\виведення і перемикається на інший процес, поки переривання від пристрою не передасть сигнал про закінчення операції ведення\виведення;
- ▶ *прямий доступ до пам'яті* – окрема мікросхема керує перенесенням цілого блоку даних і ініціює переривання після закінчення перенесення блоку.

## ***Драйвери пристроїв***

Програми управління пристроями введення\виведення називають *драйверами*. Кожний драйвер керує пристроями одного

типу або одного класу. Драйвери пристроїв є нижчим шаром, над яким розміщено програми, а під яким – контролери пристроїв.

*Функції драйверів:*

- опрацювання запитів читання і запису незалежного від пристрою програмного забезпечення;
- ініціалізація пристрою;
- управління енерговживанням пристрою і реєстра цією подій;
- перевірка вхідних параметрів при зверненні до пристрою, якщо параметри помилкові відбувається повернення помилки до ОС, якщо параметри правильні, здійснюється управління роботою пристрою;
- перевірка використання пристрою на даний момент, тобто контроль за роботою пристрою.

*Управління пристроєм* – це надання йому серії команд. В драйвері визначається послідовність команд в залежності від того, що повинно бути виконано. Після визначення команд вони записуються до контролера і далі може відбуватися наступне:

- драйвер блокується до тих пір, поки контролер буде виконувати визначені команди і переривання від пристрою його не розблокує;
- якщо робота контролера достатня швидка, то робота драйвера не блокується, а здійснюється за допомогою драйвера опрацювання наступного запиту.

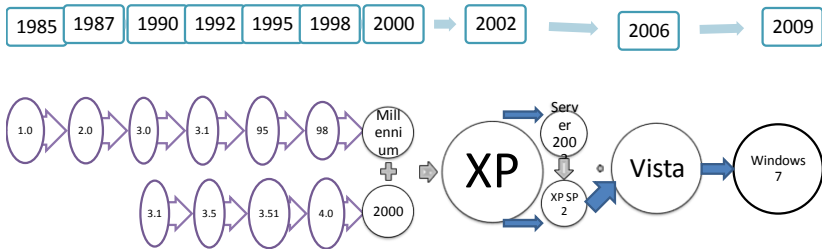
## ***Файлові системи***

Частина ОС, за допомогою якої забезпечується робота з файлами і збереження даних на зовнішніх запам'ятовувочих пристроях, називається *файловою системою*.

Принцип організації файлової системи залежить від ОС.

## ***Операційні системи сімейства Windows***

# Історія розвитку операційних систем сімейства Windows



## Основні технологічні принципи Windows

- Plug and Play (вмикай і працюй) – автоматичне розпізнання і налаштування будь-якого пристрою комп'ютера.
- Object Link and Embedding (OLE) (зв'язок або вбудовування об'єктів) – забезпечення розташування, вбудовування або зв'язування використовуваного документа з документом, що створений за допомогою будь-якої іншої програми Windows, при цьому зберігається можливість змінювати і редагувати вбудований документ або об'єкт засобами програми за допомогою якої він був створений.
- Drag and Drop (перетягнути і відпустити) – метод фізичного і логічного переміщення об'єктів, що використовується для їх копіювання або переміщення в інше положення.
- WYSIWYG (What You See Is What You Get) (що бачиш, те і отримаєш) – зображення документа на екрані при його перегляді і редагуванні, та зображення документа на папері при його друкуванні практично не відрізняються.

### Файлова система FAT

FAT (file allocation table) - таблиця розміщення файлів. Цей термін відноситься до лінійної табличної структури із відомостями про файли – імена файлів, їх атрибути та ніші дані, за допомогою яких визначається місцезнаходження файлів або їх фрагментів в FAT. Елемент FAT визначає фактичну область диска, в якій зберігається початок фізичного файлу.

В файловій системі FAT логічний дисковий простір будь-якого логічного диска поділяється на дві області:

- системну область;
- область даних.

*Системна область* створюється при форматуванні і поновлюється при маніпулюванні файловою структурою. *Область даних* містить файли і каталоги, підпорядковані кореневому, і доступна через користувацький інтерфейс.

Системна область складається із наступних компонентів:

- завантажувальний запис;
- зарезервовані сектори;
- таблиці розташування файлів (FAT);
- кореневого каталогу.

Таблиця розташування файлів являє собою карту (образ) області даних, в якій описано стан кожної ділянки області даних. Область даних розбита на кластери. В таблиці FAT кластери, які належать одному файлу (некореневому каталогу), зв'язані у ланцюги. Для вказування номера кластера в системі управління файлами FAT16 використовується 16-бітове слово, а отже, можна мати до 65536 кластерів.

Так як FAT використовується при доступі до диску дуже інтенсивно, вона завантажується до ОЗП і знаходиться там максимально довго.

Кореневий каталог відрізняється від іншого каталогу тим, що він розташований у фіксованому місці логічного диска і має фіксовану кількість елементів. Для кожного файлу і каталогу в файловій системі зберігаються відомості з наступною структурою:

- ім'я файлу або каталогу – 11 байт;
- атрибути файлу – 1 байт;
- резервне поле – 1 байт;
- час створення – 3 байта;
- дата створення – 2 байта;
- дата останнього доступу – 2 байта;
- зарезервовано – 2 байта;
- час останньої модифікації – 2 байта;
- номер початкового кластера в FAT – 2 байта;
- розмір файлу – 4 байта.

Структура системи каталогів є ієрархічною.

*Файлова система FAT32*

FAT32 є повністю незалежною 32-разрядною файловою системою и містить велику кількість удосконалень в порівнянні із FAT16. Принципова відмінність FAT32 в тому, що більш ефективно використовується дисковий простір: у FAT32 використовуються

кластери меншого розміру, що приводить до економії дискового простору.

У FAT32 можна переміщувати кореневий каталог і використовувати резервну копію FAT замість стандартної. Використання розширеного завантажувального запису FAT32 дозволяє створювати копії критичних структур даних, що підвищує стійкість дисків до порушень структури FAT у порівнянні із попередніми версіями. Кореневий каталог являє собою звичайний ланцюг кластерів, тому може знаходитися в будь-якому місці диска, що знімає обмеження на розмір кореневого каталогу.

#### *Файлова система NTFS*

*Файлова система NTFS (New Technology File System)* містить цілу низку значних удосконалень, які суттєво відрізняють її від інших файлових систем. Файли зберігаються в каталогах, але робота на дисках великого об'єму в NTFS відбувається набагато ефективніше:

- є засоби для обмеження доступу до файлів і каталогів;
- введено механізми, які суттєво підвищують надійність файлової системи;
- знято багато обмежень на максимальну кількість дискових секторів і/або кластерів.

#### *Основні характеристики файлової системи NTFS:*

- *надійність* - високошвидкісні комп'ютери і системи спільного використання повинні бути достатньо надійними, для цієї мети введено механізм транзакцій (*транзакція* - набір команд або операцій), при якому ведеться журналювання файлових операцій (всі команди кожної транзакції протоколюються і здійснюються тільки у випадку їх 100% коректності. Якщо команда транзакції не правильна, то за допомогою журналу вже виконаних команд ОС повертається в початковий стан (до початку транзакції. Повернення назад називається відкатом.);

- *розширена функціональність* - в NTFS введено: вдосконалену відмовостійкість, емуляція інших файлових систем, потужна модель захисту, паралельне опрацювання потоків даних, створення файлових атрибутів, які визначено користувачем;

- *підтримка стандарту POSIX* - до числа базових засобів відносяться необов'язкове використання імен файлів із врахуванням регістра, збереження часу останнього звернення до файлу і механізм альтернативних імен, який дозволяє посилатися на один и той же файл з використанням різних імен;

- *гнучкість* - розподіл дискового простору відрізняється великою гнучкістю: розмір кластера може змінюватися від 512 байт до 64 Кбайт.

NTFS добре працює з великими масивами даних і великими томами. Максимальний розмір тому (і файлу) – 16 Эбайт. (1 Эбайт

дорівнює 2 в 64 або 16000 млрд. гігабайт.). Кількість файлів в кореновому і некореновому каталогах не обмежено. Тому, що в основу структури каталогів NTFS закладено ефективну структуру даних, яка називаються «бінарне дерево», час пошуку файлів в NTFS не пов'язано лінійною залежністю з їх кількістю.

В системі NTFS є деякі засоби для самовідновлення і підтримки різних механізмів перевірки цілісності системи.

Файлова система NTFS підтримує об'єктну модель захисту і всі тома, каталоги і файли використовуються як самостійні об'єкти NTFS. Права доступу до томів, каталогів і файлів залежать від залікового запису користувача і тієї групи, до якої він належить.

У файлової системі NTFS є вбудовані засоби стиснення, які можна застосовувати до томів, каталогів і файлів.

#### *Етапи завантаження ОС Windows*

- 1. Вмикання живлення.* За допомогою блоку живлення здійснюється перевірка електричних параметрів системи. Якщо вони в нормі, то передається сигнал Power Good процесору. Час між вмиканням живлення і передаванням сигналу, як правило, 0.1-0.5 секунд.
- 2. До таймеру мікропроцесору поступає сигнал Power Good.* З отриманням цього сигналу таймер перестає посилати сигнал Reset процесору, відбувається вмикання процесору.
- 3. Виконання коду ROM BIOS процесором.* Процесор завантажує ROM BIOS починаючи з адреси FFFF:0000. По цій адресі прописано тільки перехід на адресу коду BIOS ROM.
- 4. Виконання початкового тестування пристроїв.* Кожна помилка, яка зустрічається на даному етапі повідомляється певними звуковими кодами, так як відео система ще не ініціалізована.
- 5. За допомогою BIOS здійснюється пошук адаптерів для яких може бути потрібне завантаження власного BIOS.* Самим типовим випадком є відео карта. Процедура завантаження починається із сканування пам'яті із адреси C000:0000 до C780:0000 для пошуку відео ROM. Таким чином завантажуються системи всіх адаптерів.
- 6. ROM BIOS перевіряє вмикання це або перезавантаження.* Ця процедура - два байта за адресою 0000:0472. Будь-яке значення відмінне від 1234h вказує на те, що це є перезавантаженням комп'ютера.
- 7. Якщо це включення то за допомогою ROM BIOS відбувається запуск повного POST (Power On Self Test).* Якщо це перезавантаження, то з POST процедури

виключається перевірка пам'яті . Процедуру POST можна поділити на три компоненти:

- За допомогою відео тесту ініціалізується відео адаптер, тестується карта і відео пам'ять, відображається конфігурація або помилки, які виникають.

- При ідентифікації BIOS-а визначається і відображається версія прошивки, виробника і дату виготовлення BIOS .

- За допомогою тесту пам'яті відбувається перевірка пам'яті і визначається розмір встановленої пам'яті.

Якщо в процесі POST перевірки виникають не суттєві помилки, то вони відображаються на екрані, але продовжується процес завантаження. Якщо помилки є суттєвими, то процес завантаження зупиняється, що, як правило, супроводжується серією біп-кодів.

8. *Зчитування відомостей про конфігурацію із CMOS* (невелика область пам'яті (64 байт) живиться від батарейки на материнській платі). Найголовніше для завантаження зберігається - порядок, в якому повинні перевірятися приводи, який з них повинен бути першим - дисковод, CD-ROM або вінчестер.
9. *Якщо першим є жорсткий диск, BIOS перевіряє самий перший сектор диска на наявність Master Boot Record (MBR). Для дисковода перевіряється Boot Record в першому секторі. Master Boot Record – перший сектор на циліндрі 0, 0 головка, 512 байт розміром. Якщо він є, то завантажується до пам'яті за адресою 0000:7C00, потім перевіряється на правильну сигнатуру - два останніх байта повинні бути 55AAh. Відсутність MBR або байтів перевірки є сигналом зупинки процесу завантаження, видається відповідне попередження. MBR складається з двох частин - системного завантажувача (partition loader або Boot loader), програми, яка отримує управління при завантаженні з цього жорсткого диска; таблиці розділів (партицій), яка містить відомості про логічні диски, які є на жорсткому диску.*
10. *При правильному MBR знайдений запис завантажується до пам'яті комп'ютера. Процес установки кількох операційних систем на один комп'ютер, як правило, замінює завантажувач на свою програму, за допомогою якої можна вибрати з якого диску продовжується процес завантаження.*
11. *За допомогою Boot Loader перевіряється таблиця партицій в пошуках потрібної. Завантажувач далі шукає завантажувальний запис (Boot Record) на самому першому*

*секторі розділу.* В даному випадку Boot Record це 512 байт - таблиця з описом розділу (кількість байт в секторі, кількість секторів в кластері і т.п.) і перехід на перший файл операційної системи (IO.SYS в DOS).

12. *Операційна система.* Управління передається операційній системі.
13. *Boot Record перевіряється на правильність і якщо код є правильним, то код завантажувального сектору використовується як програма.* Завантаження Windows XP контролюється файлом NTLDR, який знаходиться в кореневому каталозі системного розділу. NTLDR працює в чотири етапи:
  - *початкова фаза завантаження:* за допомогою NTLDR відбувається перемикання процесору в захищений режим. Потім завантажується відповідний драйвер файлової системи для роботи з файлами будь-якої файлової системи, яка підтримується XP. ОС Windows XP може працювати з FAT-16, FAT-32 і NTFS
  - *вибір системи:* якщо в кореневому каталозі є BOOT.INI, то його зміст завантажується до пам'яті. Якщо в ньому є запис більш ніж про одну операційну систему, за допомогою NTLDR зупиняється процес завантаження: виводиться меню з вибором і відбувається, очікування введення від користувача, певний проміжок часу. Якщо такого файлу немає, то NTLDR продовжує завантаження з першої партиції першого диску, як правило це C:\. Якщо в процесі вибору користувач вибрав Windows NT, 2000 або XP, то перевіряється натиснення клавіші F8 і виведення відповідного меню з опціями завантаження.
    - *Normal* - продовження завантаження в звичайному режимі.
    - *Logged* (включити протоколювання загрузки/Bootlog.txt) - нормальне завантаження із створенням в кореневій директорії системного диска файлу bootlog.txt з докладним протоколом ходу завантаження ОС.
    - *Safe Mode* (безпечний режим) - режим захисту від збоїв, при якому в пам'ять завантажується мінімальний набір наднадійних драйверів: драйвер стиснених



дисків Db1Space.bin або DrvSpace.bin, розширеній пам'яті himem.sys, файлової системи ifshlp.sys, драйвери клавіатури, миші, адаптера. Директиви з конфігураційних файлів config.sys, autoexec.bat, system.ini (деякі настройки і параметри Windows) ігноруються.

- *Safe mode with Network Support* (із завантаженням мережевих драйверів) - режим захисту від збоїв з мережевою підтримкою.
- *Safe mode command prompt* (з підтримкою командного рядка) - режим захисту від збоїв з підтримкою командного рядка.
- *VGA Mode* (Включити режим VGA) – 800X600, 16bit
- *Завантаження останньої вдалої конфігурації*
- *Відновлення служби каталогів* (на контроллері домена Windows)
- *Режим відладки*

- Після кожного вдалого завантаження XP створює копію даної комбінації драйверів і системних налаштувань відому як Last Known Good Configuration
- *визначення пристроїв*: за допомогою NTLDR знаходиться і завантажується програма NTDETECT.COM для визначення пристроїв комп'ютера. За допомогою NTDETECT.COM створюється список компонентів, які потім використовуються в ключі HARDWARE гілки HKEY\_LOCAL\_MACHINE реєстру
- *вибір конфігурації*: якщо комп'ютер має більш ніж один профіль обладнання, виконання програми зупиняється на меню вибору конфігурації.

14. Після вибору конфігурації NTLDR починає завантаження ядра XP (NTOSKRNL.EXE). В процесі завантаження ядра (але перед ініціалізацією) NTLDR залишається головним в управлінні комп'ютером. Крім ядра завантажується і Hardware Abstraction Layer (HAL.DLL). Два файли знаходяться в директорії System32.

15. NTLDR завантажує драйвери пристроїв, як завантажувальні. Завантаживши їх NTLDR передає управління комп'ютером далі. Кожний драйвер має ключ в

HKEY\_LOCAL\_MACHINE\SYSTEM\Services. Якщо значення Start дорівнює SERVICE\_BOOT\_START, то пристрій вважається завантажувальним. Для кожного такого пристрою на екрані друкується точка. NTOSKRNL в процесі завантаження проходить через дві фази - так звану фазу 0 і фазу 1. В першій фазі ініціалізується лише та частина мікроядра і виконавчі підсистеми, для роботи яких потрібні основні служби і продовження завантаження. Викликається HAL для підготовки контролера переривань. Ініціалізується Memory Manager, Object Manager, Security Reference Monitor і Process Manager. Фаза 1 починається коли відбувається підготовка системи до опрацювання переривань пристроїв. Якщо на комп'ютері встановлено більш одного процесору вони ініціалізуються. Всі виконавчі підсистеми реініціалізуються в наступному порядку:

- Object Manager
- Executive
- Microkerne
- Security Reference Monitor
- Memory Manager
- Cache Manager
- LPCS
- I/O Manager
- Process Manager

16. *Ініціалізація Менеджера введення/виведення починає процес завантаження всіх системних драйверів. С того моменту де зупинився NTLDR завантажуються драйвера за пріоритетами.* Збій в завантаженні драйвера призводить до перезавантаження і поновлення Last Known Good Configuration.

17. *Ініціалізації ядра - запуск Session Manager Subsystem (SMSS).* Підсистема відповідає за створення користувацького оточення, яке забезпечує інтерфейс NT. SMSS працює в користувацькому режимі, але на відміну від інших додатків SMSS вважається частиною операційної системи, що дозволяє їй запустити графічну підсистему і login. SMSS завантажує win32k.sys - графічну підсистему. Драйвер перемикає комп'ютер в графічний режим, SMSS запускаються всі сервіси, які повинні автоматично завантажуватися при старті. Якщо всі пристрої і сервіси

запустилися вдало процес завантаження вважається вдалим і створюється Last Known Good Configuration.

18. Процес завантаження не вважається завершеним до тих пір, поки користувач не вів своє ім'я до системи. Процеси ініціалізується файлом WINLOGON.EXE, який запускається як сервіс і підтримується Local Security Authority (LSASS.EXE), який і показує допоміжне вікно для здійснення входу до системи. Це допоміжне вікно виводиться, коли Services Subsystem завантажує мережеву службу.

*Призначення деяких системних папок*

<i>Ім'я</i>	<i>Призначення</i>
<i>Config</i>	Містить файли конфігурації пристроїв ПК. Настроюються конфігурації в панелі управління в настройках Система
<i>Cookies</i>	Тут знаходяться спеціальні cookie-файли, які приймає ІЕ, під час з'єднання з Інтернет для зручності роботи з сайтами. Цю теку можна іноді безболісно видаляти.
<i>Cursors</i>	Тут зберігаються курсори для миші.
<i>Fonts</i>	Тут зберігаються встановлені в системі шрифти.
<i>Help</i>	Тут зберігаються файли довідки ОС і іноді інших програм.
<i>History</i>	Папка журналу раніше проглянутих посилань в Інтернет.
<i>Inf</i>	В цій папці знаходяться inf-файли, які містять параметри установки пристроїв в Windows. Не треба плутати inf-файли з драйверами, які зазвичай мають розширення *.drv.
<i>Media</i>	Тут знаходяться звукові файли Windows
<i>ShellNew</i>	доповнення до меню <i>Створити</i> , за допомогою якого можна створювати документи, папки, ярлики та ін.
<i>System</i>	Тут зберігаються драйвера, програмне ядро Windows, різні частини додатків (dll бібліотеки), частини Панелі управління (cpl.* файли) і ін.
<i>System32</i>	доповнення до папки System, містить в основному драйвера
<i>Temp</i>	місце для зберігання tmp-файлів (тимчасові файли, які створюються додатками під час роботи). Цю теку можна видаляти, якщо жодна з працюючих програм її не використовує в даний момент.
<i>Головне меню</i>	вміст <i>Головного меню</i>
<i>Обране</i>	папка для зберігання деяких ярликів

<i>DeskTop</i>	реальна фізична папка на жорсткому диску, відповідна вмісту нереального <i>Робочого столу</i>
<i>All Users</i>	Тут зберігаються папки <i>Робочий стіл</i> і <i>Головне меню</i> , які використовуються всіма користувачами ПК.
<i>Application Data</i>	Тут зберігаються різні додаткові програмні надбудови, які створюються самими ж програмами
<i>Catroot</i>	Тут зберігаються каталоги безпеки Windows, стислі в архів формату Cabinet.
<i>Downloaded Program Files</i>	Тут тимчасово зберігаються елементи ACTIVEX і додатки Java, автоматично завантажені з Інтернету при перегляді деяких сторінок.
<i>Offline Web Pages</i>	Тут зберігаються сторінки з автономним доступом.
<i>Tasks</i>	папка “Призначені завдання”
<i>Web</i>	папка Windows, яка містить Web-сторінки, шпалери для <i>Робочого столу</i> , Web-налаштування ОС
<i>Msagent</i>	папка для програми Microsoft Agent Server.
<i>Options</i>	Тут зберігаються автоматично створюваний дистрибутив Windows.
<i>Twain_32</i>	тут знаходяться деякі елементи TWAIN, що використовують ds-файли
<i>Addins</i>	Тут зберігаються деякі додатки до Windows.
<i>AppPatch</i>	Тут зберігаються деякі додаткові бібліотеки до програм
<i>Connection Wizard</i>	Тут зберігається частина програми-майстра для підключення Connection Wizard.
<i>Driver Cache</i>	архів драйверів для Windows.
<i>Msapps</i>	Тут містяться додатки Microsoft, що входять до складу інших програм
<i>Prefetch</i>	Папка містить файли-прогнози запуску додатків.