МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені Семена Кузнеця

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

НАУКА
ЕКОНОМІКА
ПІДПРИЄМСТВО

1930

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені Семена Кузнеця

Лекція № 2

Тема 1: Архітектура ОС

по курсу "**Операційні системи**"

Тема лекції: Архітектура різноманітних операційних систем

Лектор:
Доцент кафедри Інформаційних систем
кандидат технічних наук, доцент
Голубничий Дмитро Юрійович

18.09.2022

НАВЧАЛЬНІ ПИТАННЯ:

1. Архитектура ОС Windows.
2. Архитектура ОС ReactOS.
3. Архитектура ОС Kolibri OS.
4. Архитектура Linux.

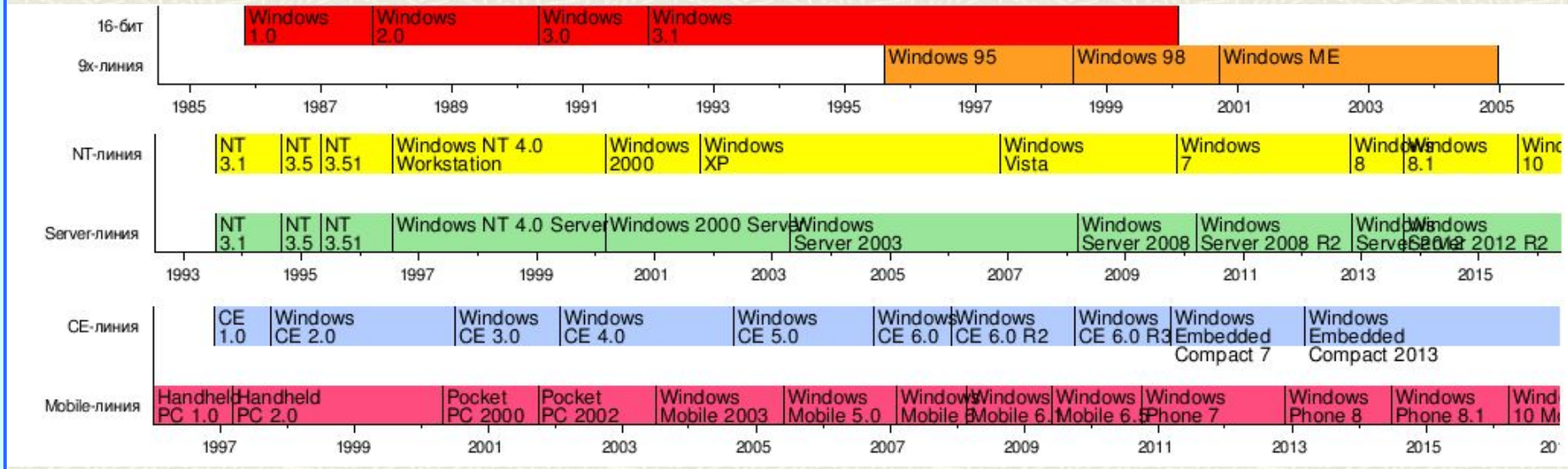
1. Архітектура ОС Windows

Microsoft Windows

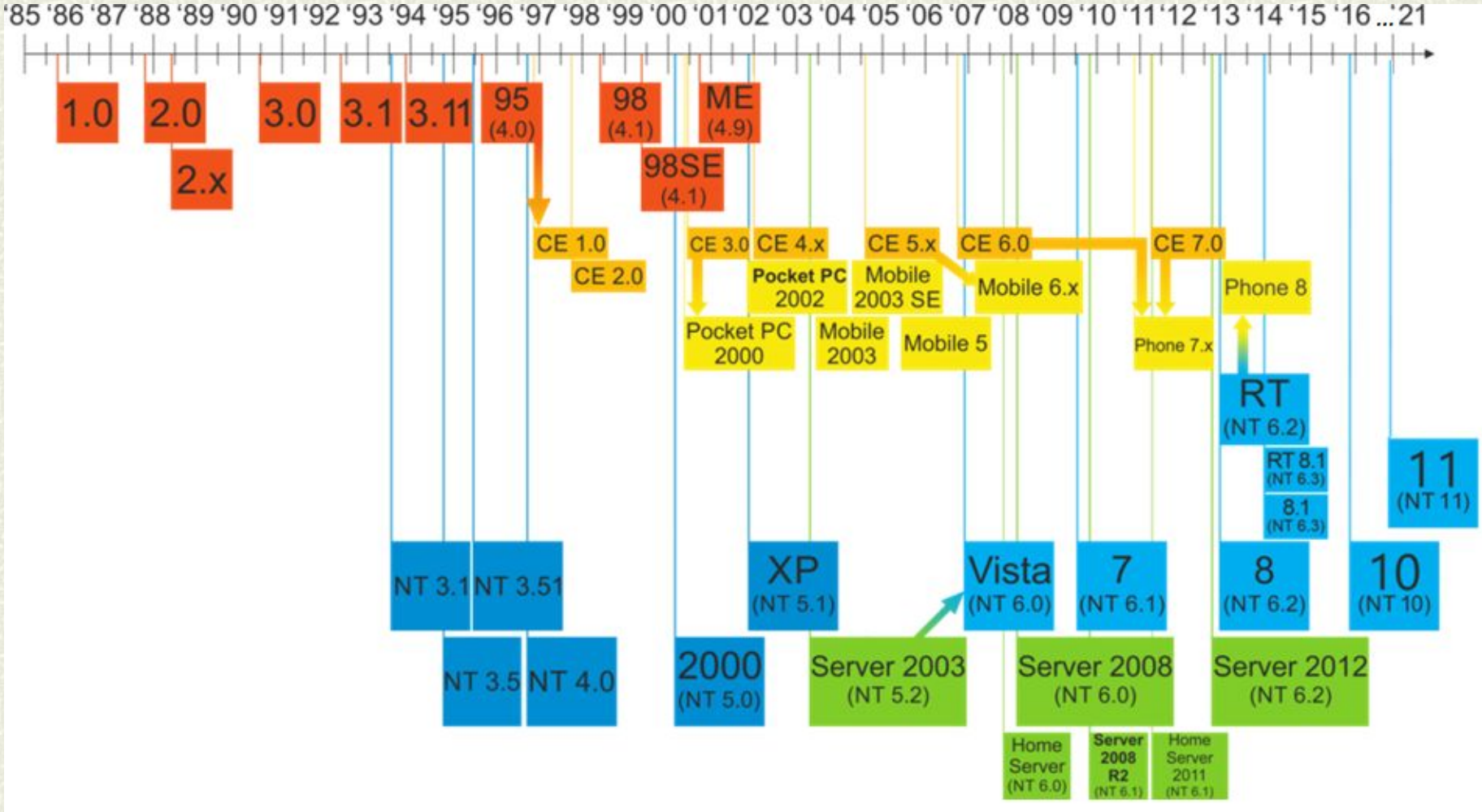
Логотип Microsoft Windows

Група сімейств комерційних операційних систем корпорації Microsoft, орієнтованих на управління за допомогою графічного інтерфейсу.

MS-DOS – є прабатьком Windows. Спочатку, Windows був лише графічним інтерфейсом для MS-DOS. Кожне сімейство обслуговує певний сектор комп'ютерної індустрії.



Історія Windows



Графік виходу і підтримки Windows

Клони Windows

- 1.x (Листопад 1985)
  - версія 1.0
  - версія 1.03
  - 2.X (Листопад 1987)
    - версія 2.0
- 3.x (Травень 1990)
  - версія 3.0
  - Версія 3.1 (листопад
  - Версія 3.11 (for Workgroups)
- NT 3.x (Липень 1993)
  - Версія NT 3.1
  - Версія 3.5 (сентябрь1994)
  - Версія 3.51 (травень 1995)
- 9x (Серпень 1995)
  - версія 95
  - Версія 95 OSR2
  - Версія 98 (червень 1998)
  - Версія Me (вересень 2000)
- NT 4.x (Липень 1996)
  - Версія NT 4.0
- **2000** (NT 5.0) (Лютий 2000)
  - Professional
  - Server
  - Advanced Server
  - DataCenter
- **XP** (NT 5.1) (Жовтень 2001)
  - Home Edition
  - Professional
  - Media Center Edition
  - Starter Edition
  - Professional x64 Edition
- **2003** (NT 5.2) (Квітень 2003)
  - Web Edition
  - Standard Edition
  - Enterprise Edition
  - Datacenter Edition
- **Vista** (NT 6.0) (листопад 2006)
  - Starter
  - Home Basic
  - Home Premium
  - Business
  - Enterprise
  - Ultimate
- **Server 2008** (NT 6.0)
  - Standard
  - Enterprise
  - Datacenter
  - Web Server
  - Storage Server
- **Windows 7** (NT 6.1) (2009)
  - Starter
  - Home Basic
  - Home Premium
  - Professional
  - Enterprise
  - Ultimate
- Server 2008 R2 (NT 6.1)
- **Home Server** (NT 6.1) (2011)
- **Windows 8** (NT 6.2) (2012)
  - Professional
  - Enterprise
- **Server 2012** (NT 6.2) (2012)
  - Foundation
  - Essentials
  - Standard
  - Datacenter
- **Windows 8.1** (NT 6.3) (2013)
- **Server 2012 R2** (NT 6.3) (2013)
- **Windows 10** (NT 10) (2015)
  - Home
  - Professional
  - Enterprise
- **Windows 11** (NT 11) (2021)
  - Home
  - Professional
  - Enterprise
- **Windows Server 2016** (NT 10) (2016)
  - Standard
  - Datacenter
- **Windows Server 2019** (NT 10) (2019)
  - Standard
  - Datacenter
- **Windows Server 2022** (NT 11) (2021)
  - Standard
  - Datacenter

- - 16-ти розрядні ОС
- - 32-ти розрядні ОС з підтримкою 16-ти розрядності
- - 32-ти розрядні ОС
- - 64-ти розрядні ОС
- - 32-ти і 64-ти розрядні ОС

Сумісність - можливість операційної системи виконувати застосунки, розроблені для інших операційних систем.

Види сумісності:

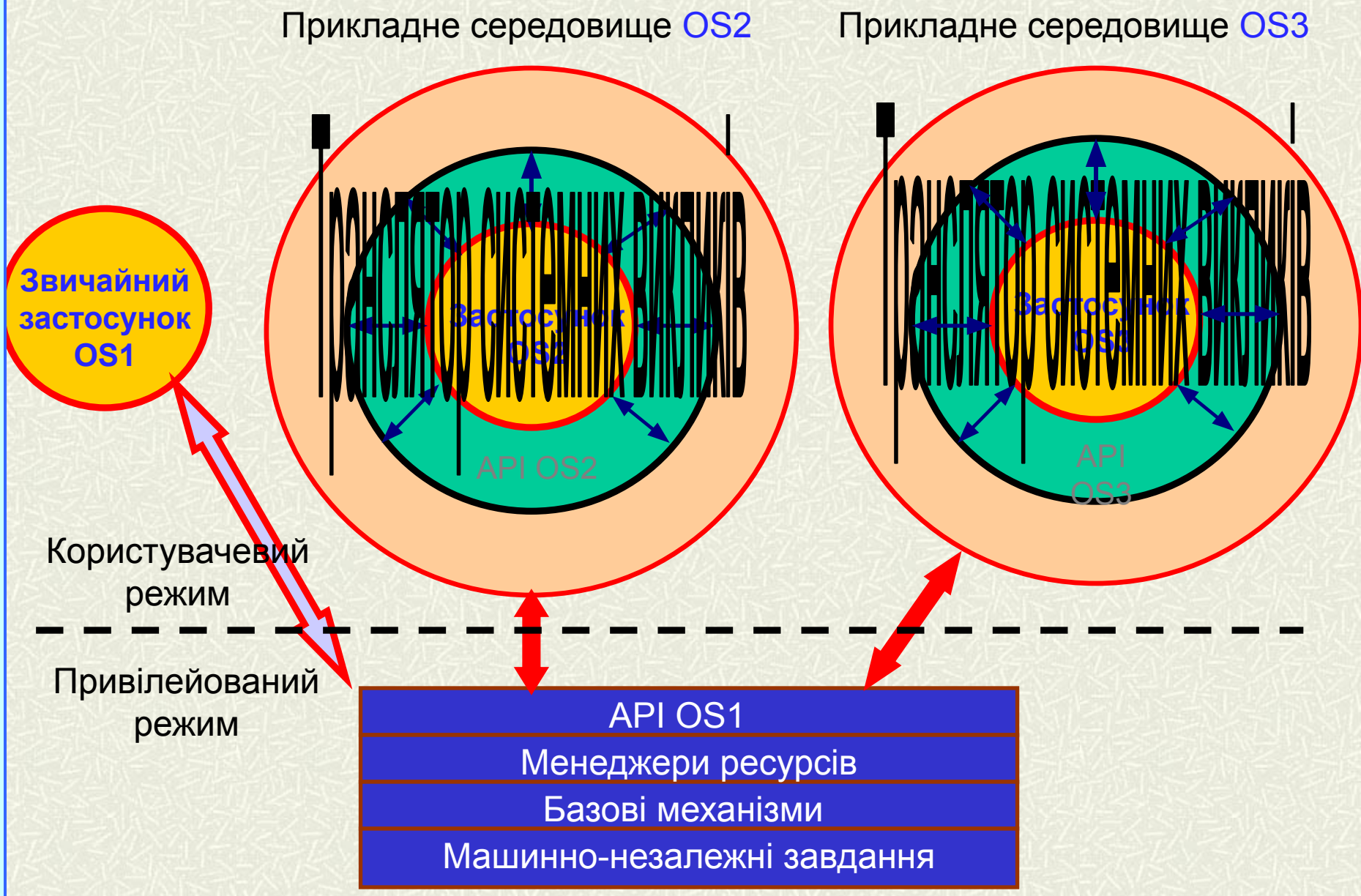
1. На довічним рівні (рівень виконуваної програми).
2. На рівні вихідних текстів (рівень вихідного модуля).

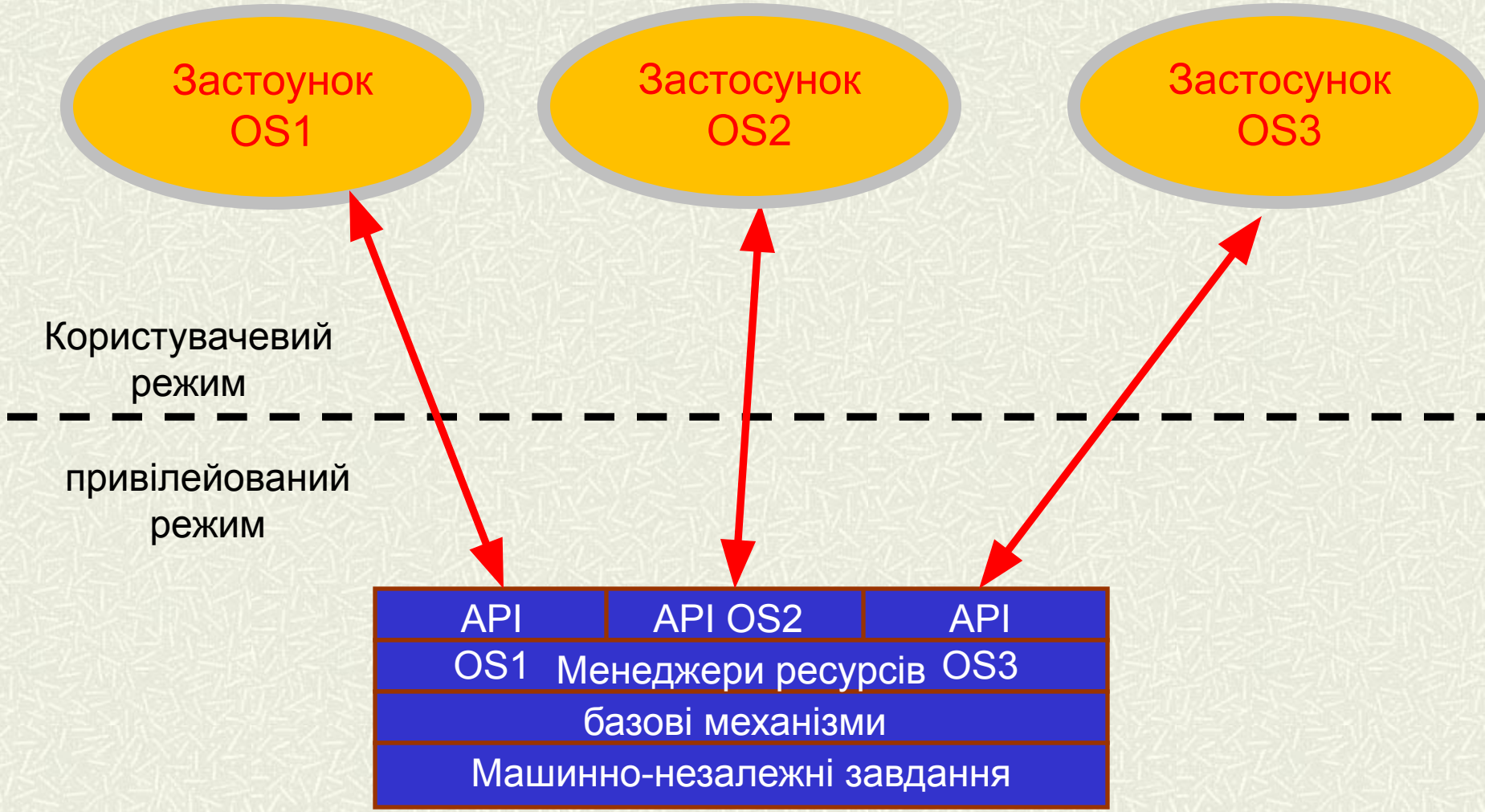
Вид сумісності визначається:

1. Архітектурою центрального процесора.
2. Інтерфейсом прикладного програмування (API).
3. Внутрішньої структурою виконуваного файлу.
4. Наявністю відповідних компіляторів і бібліотек.

Способи досягнення сумісності:

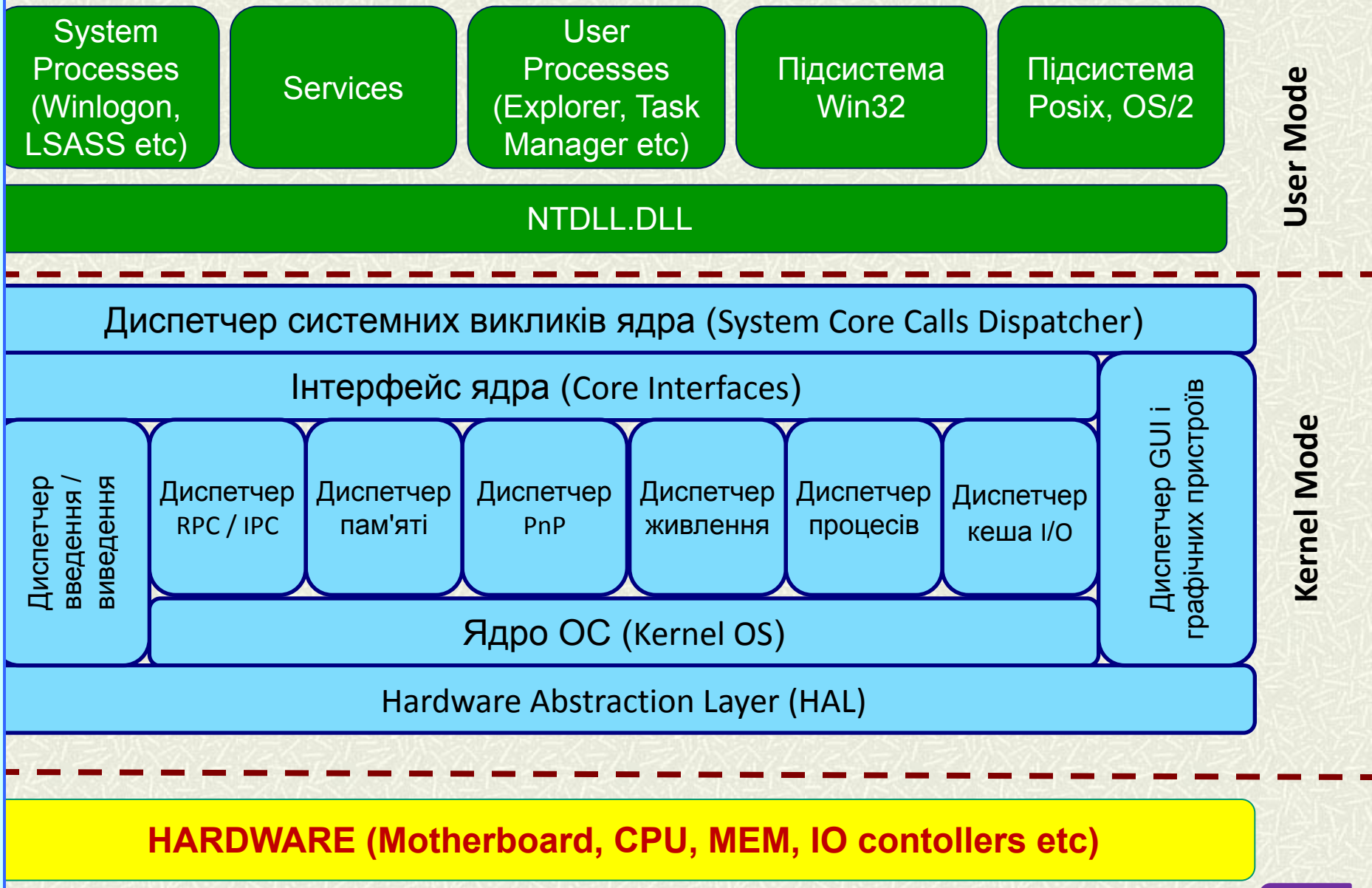
1. Емуляція двійкового коду.
2. Трансляція бібліотек.
3. Створення множинних прикладних середовищ різної архітектури.


Прикладне середовище OS2
Прикладне середовище OS3
Звичайний застосунок OS1
Застосунок OS2
API OS2
Застосунок OS3
API OS3
Користувачевий режим
Привілейований режим
API OS1
Менеджери ресурсів
Базові механізми
Машинно-незалежні завдання

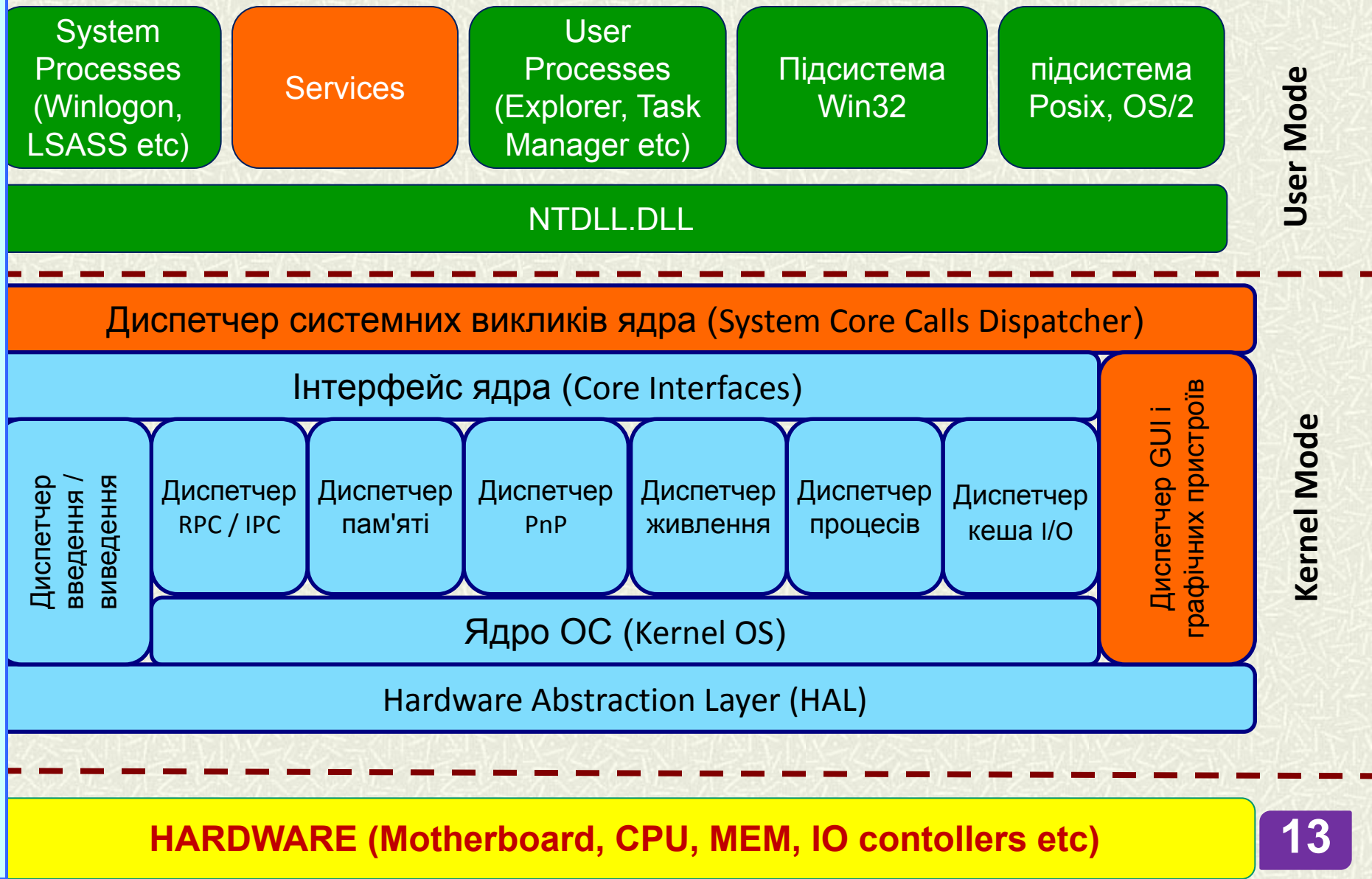

Застоунок
OS1
Застосунок
OS2
Застосунок
OS3
Користувачевий
режим
привілейований
режим
API
OS1
API OS2
API
OS3
Менеджери ресурсів
базові механізми
Машинно-незалежні завдання

Архітектура Windows

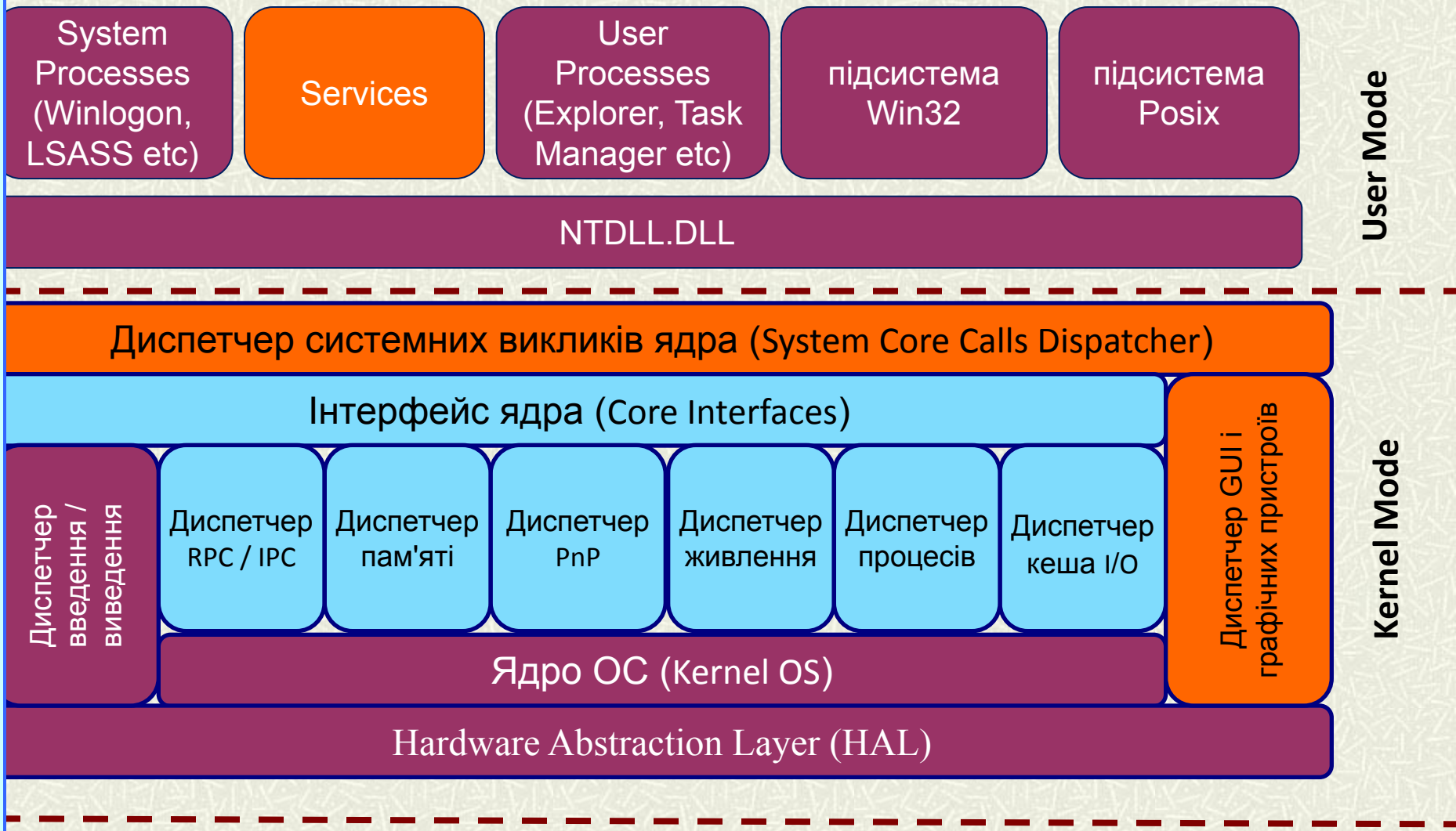
Архітектура Windows 2000



Архітектура Windows XP/ Windows Server 2003

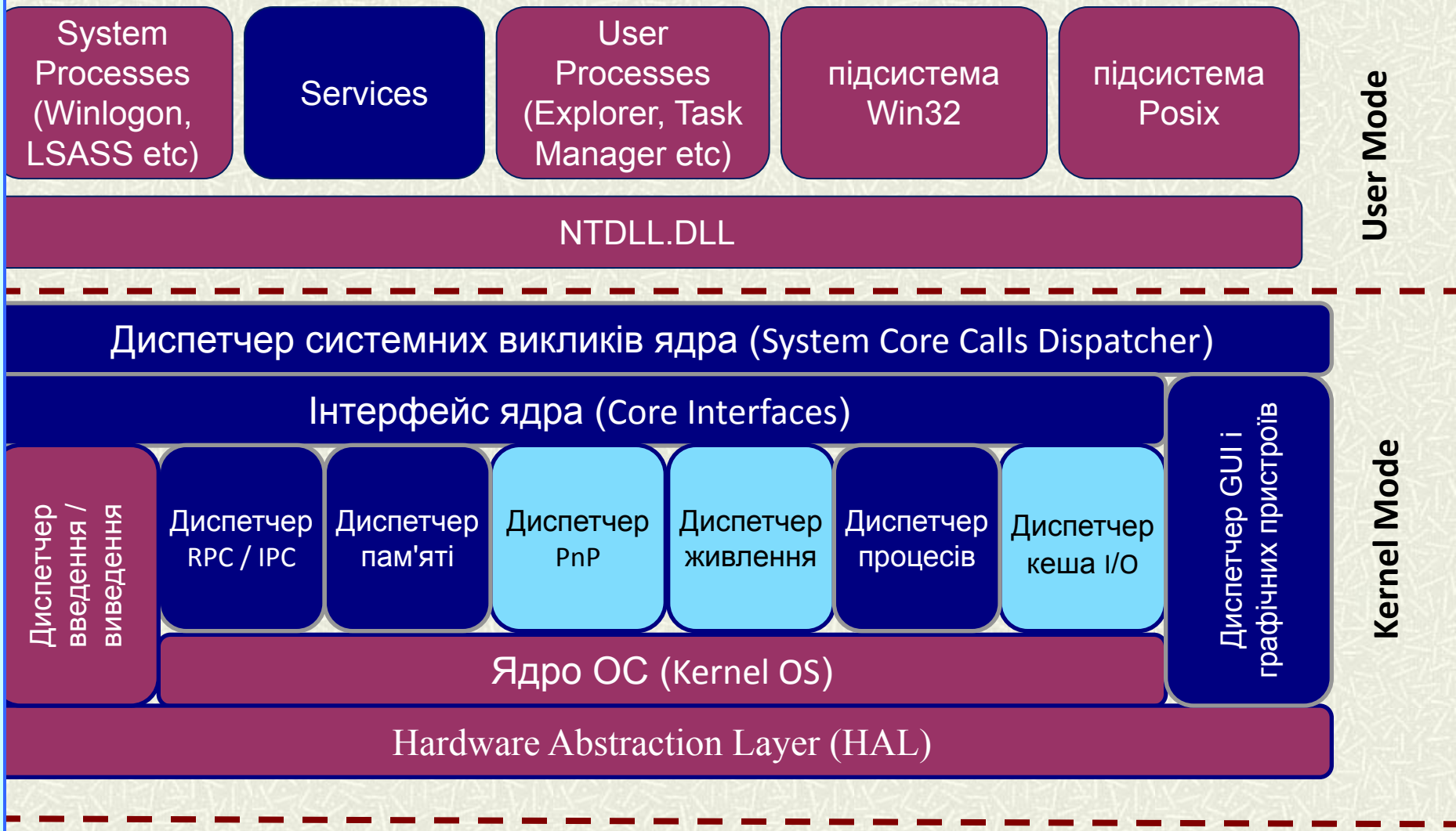


Архітектура Windows Vista / Windows Server 2008



HARDWARE (Motherboard, CPU, MEM, IO contollers etc)

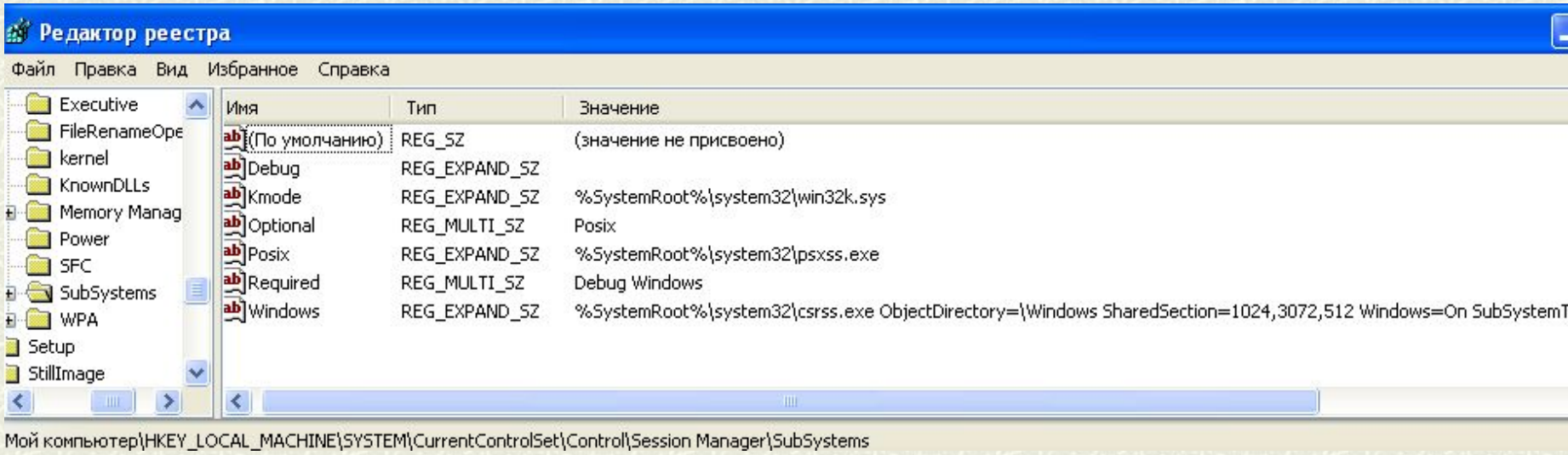
Архітектура Windows 7 (10) / Windows Server 2008 R2



HARDWARE (Motherboard, CPU, MEM, IO contollers etc)

Стартова інформація захищених підсистем зберігається в розділі реєстру:

HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet \ Control \ Session Manager \ SubSystems.



Required – список підсистем, що завантажуються при запуску системи.

Windows – вказується специфікація файлу підсистеми Windows - csrss.exe

Debug – залишається незаповненим. Він використовується для внутрішнього тестування і не виконує ніяких функцій.

Optional – вказує, що підсистеми OS/2 і POSIX запускаються на вимогу

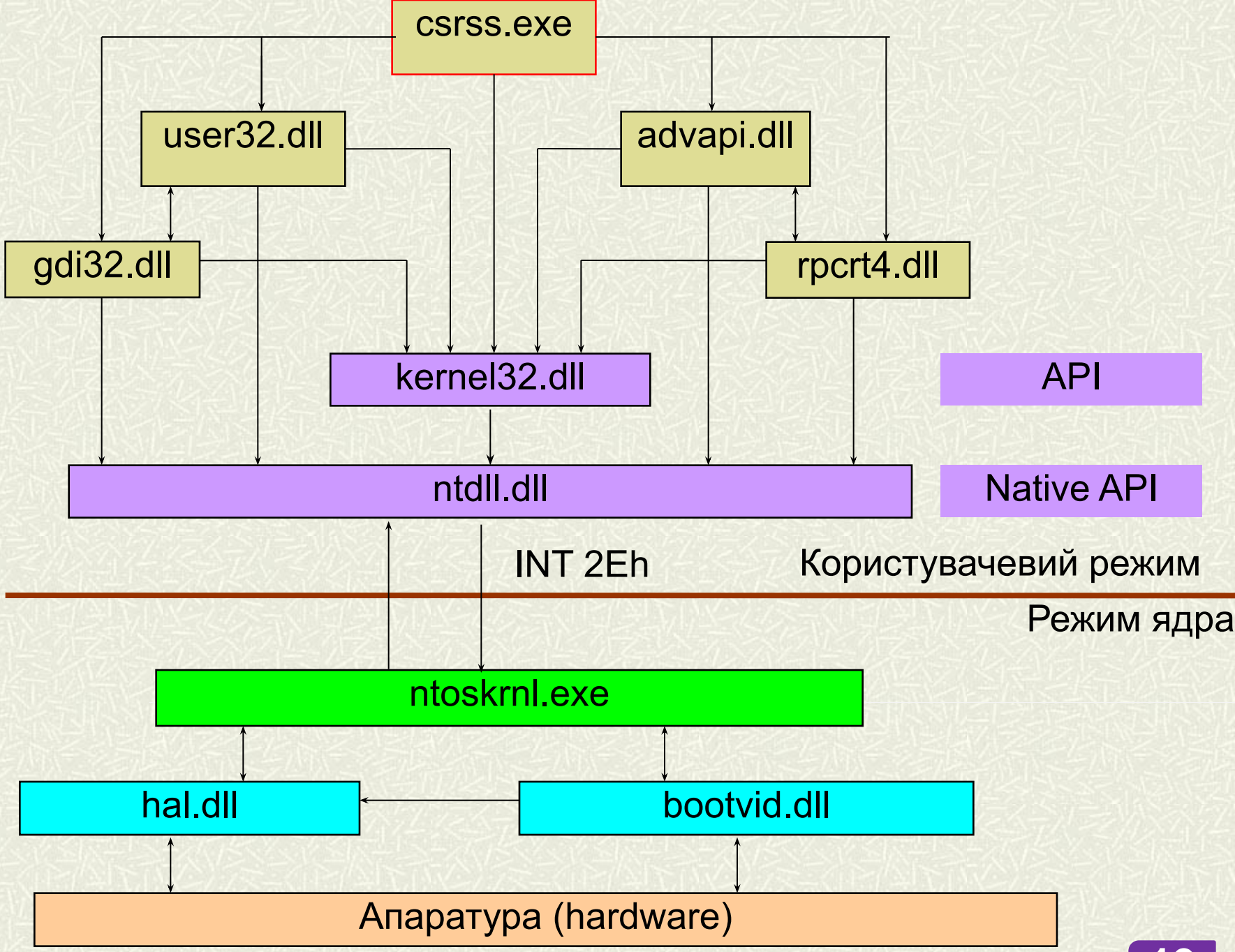
Posix – вказується специфікація файлу підсистеми POSIX, psxss.exe

Kmode – містить ім'я файлу тієї частини підсистеми Windows, яка працює в режимі ядра, - win32k.sys

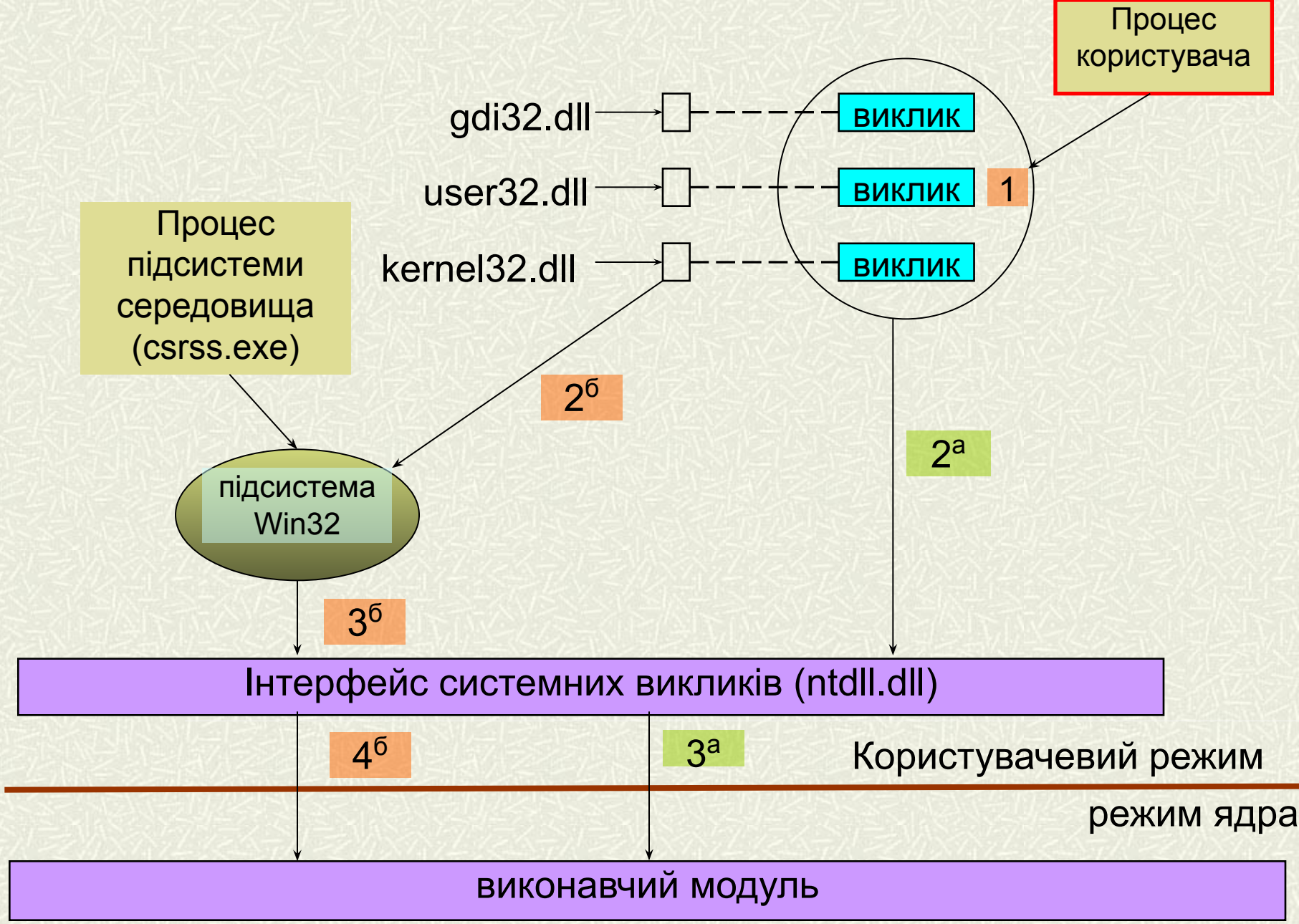
№ п/п	Підсистема виконавчої системи	Призначення
1.	**Диспетчер Об'єктів (Object Manager)**	Керує всіма відомими операційній системі об'єктами (потоки, файли, каталоги, семафори, таймери і т.д.), а також глобальним простором імен.
2.	**Диспетчер вводу / виводу (I/O Manager)**	Служить інтерфейсом між прикладними програмами і драйверами пристроїв. Виступає каркасом для управління пристроями введення / виводу і надає загальні служби введення / виведення.
3.	**Диспетчер Процесів (Process Structure)**	Управляє процесами і потоками, включаючи їх створення і завершення. Займається не стратегіями, які застосовуються по відношенню до процесів, а механізмом для управління ними.
4.	**Диспетчер Віртуальної Пам'яті (Virtual Memory Manager)**	Визначає адресний простір процесу і розподіляє фізичну пам'ять. Реалізує архітектуру віртуальної пам'яті з сторінкової підкачкою на вимогу ОС.
5.	**Диспетчер Кеша (Cache Manager)**	Реалізує глобальний файловий кеш. Зберігає блоки даних, які використовувалися останнім часом, для прискорення доступу до них.
6.	**Диспетчер Безпеки (Security Reference Manager)**	Реалізує модель безпеки на основі **Ідентифікаторів Безпеки (SID)** і **Списків розмежувального Контролю Доступу** (Discretionary Access Control List - DACL). Реалізує механізми безпеки, що задовольняють вимогам класу С2 Помаранчевої книги Міністерства оборони США

Підсистеми Виконавчої Системи Windows і їх призначення

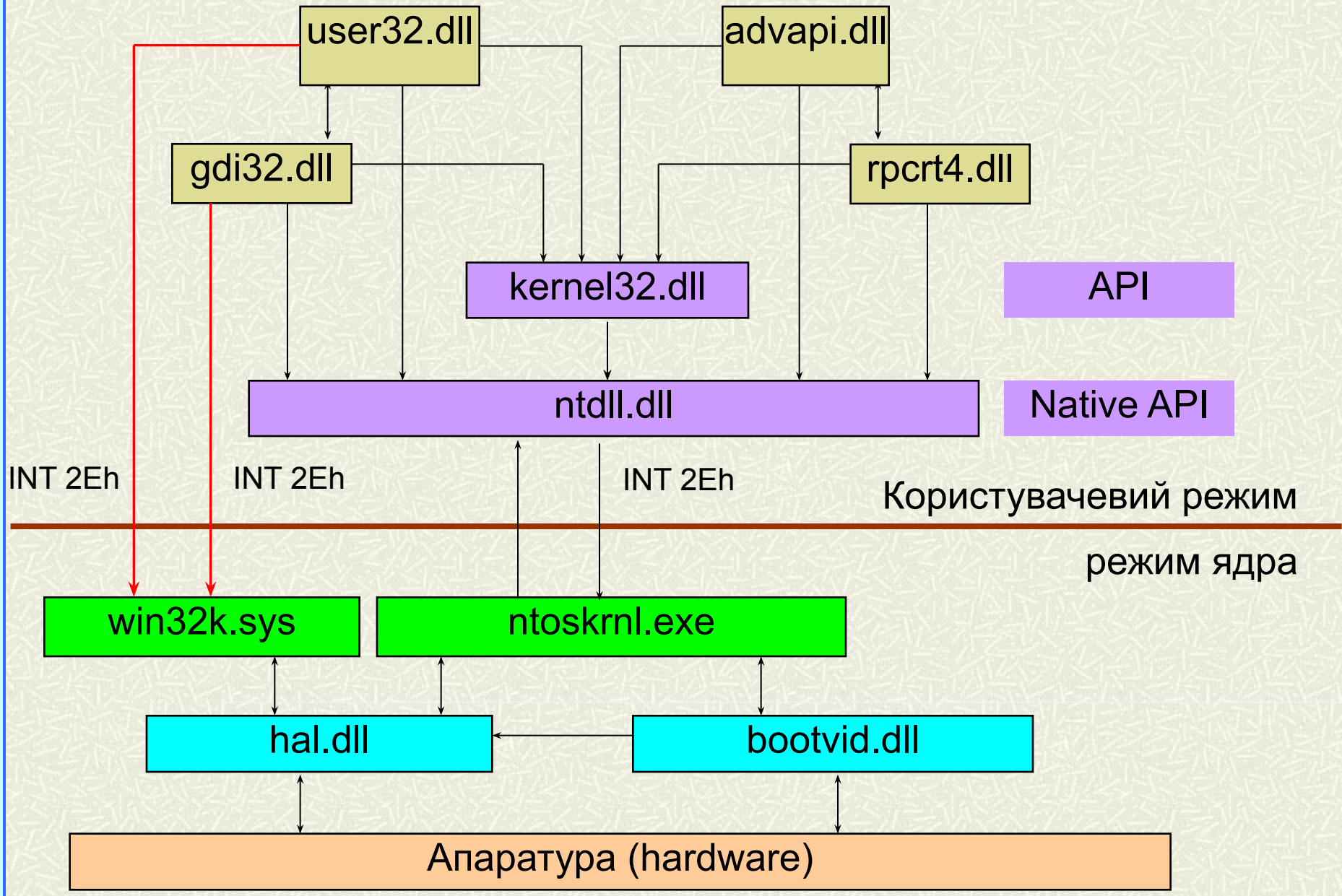
№ п/п	Підсистема виконавчої системи	Призначення
7.	Диспетчер Plug-and-Play (Plug-and-Play Manager)	Повідомляє драйвери пристроїв про включення / відключення нових пристроїв. Для деяких пристроїв перевірка проводиться при завантаженні системи, для інших будь-коли.
8.	Диспетчер Електроживлення (Power Manager)	Контролює стан живлення. Вимикає монітор і диски, при відсутності звернення до них.
9.	Диспетчер Конфігурації (Configuration Manager)	Управляє Реєстром. Додає нові записи і організовує роботу з ключами.
10.	Засіб Виклику Локальних Процедур (Local Procedure Call - LPC) Facility)	Забезпечує високоефективну взаємодію між процесами і їх підсистемами. Застосовується при критичному виконанні системних викликів, коли не використовуються стандартні механізми взаємодії між процесами (IPC)
11.	Підтримка середовища Win32 (Win32 Support)	Реалізує Win32-функції обміну повідомленнями, вікнами та рисування Реалізує інтерфейс графічних пристроїв (GDI).
12.	Виконавчий модуль (Executive Support)	Реалізовує управління чергами, системної областю пам'яті, забезпечує системні робочі потоки, а також надання інтерфейсу до виконуючої системі.



Взаємозв'язок системних модулів



Різні маршрути виконання виклик Win32 API



Залежність системних модулів, включаючи win32k.sys

Програмна заглушка

Застосунок:
Запит на створення / відкриття файлу
CreateFile ()

Kernel32.dll:
Виклик NtCreateFile ()

ntdll.dll:
Виклик системного сервісу

```
NtCreateFile ()
moveax, 20h
lea edx, [esp + 4]
int2Eh
ret2Ch
```

INT 2Eh

Користувачевий режим

режим ядра

Оброблювач заглушки

Таблиця дескрипторів системних сервісів

Диспетчер системних сервісів

0x0020

системний сервіс створення / відкриття файлу

Приклад виклику системного сервісу для створення

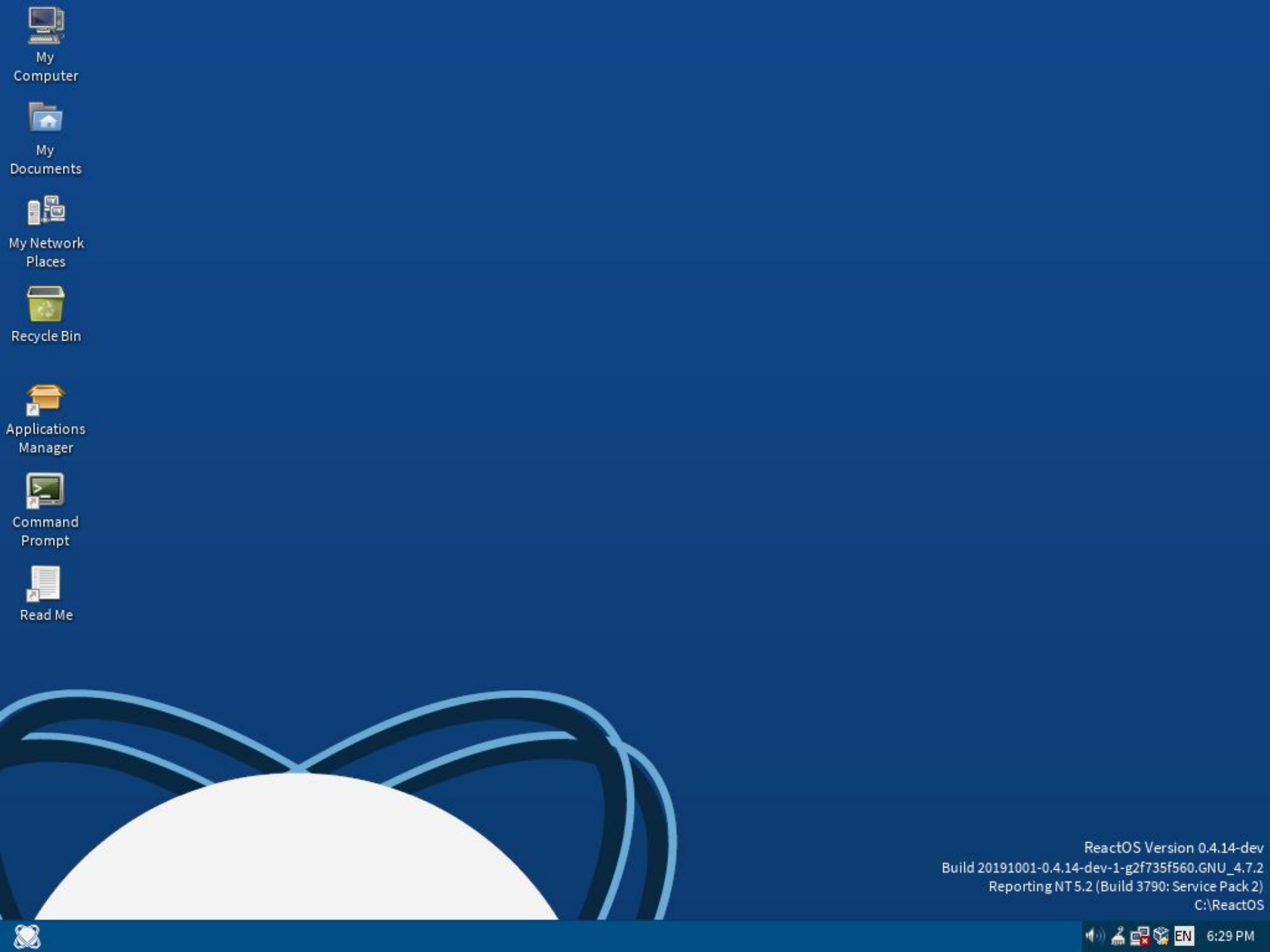
2. Архітектура ОС React OS

ReactOS - **міжнародний проект** вільної і безкоштовної операційної системи з відкритим кодом. ReactOS не є точним клоном Windows, але операційною системою, сумісної з додатками і драйверами Microsoft Windows (зараз це Windows NT версії 5.x і вище - Windows 2000 і далі).

Розробник	ReactOS Foundation
Сімейство ОС	Windows NT
Остання версія	0.4.14 20 липня 2021
Підтримувані платформи	x86, x86-64
Тип ядра	гібридне ядро
Ліцензія	GNU General Public License і ін.
Стан	Бета-версія

Логотип операційної системи ReactOS

Для відстоювання інтересів проекту був створений фонд "ReactOS Deutschland eV", розташований в Німеччині.


My Computer
My Documents
My Network Places
Recycle Bin
Applications Manager
Command Prompt
Read Me
ReactOS Version 0.4.14-dev
Build 20191001-0.4.14-dev-1-g2f735f560.GNU_4.7.2
Reporting NT 5.2 (Build 3790: Service Pack 2)
C:\ReactOS
EN
6:29 PM

ФУНКЦІОНАЛЬНІСТЬ REACTOS

1. Велика частина ядра **повністю сумісна** з Windows 2003 Server SP1:
   - ✔ Виконавча система (Executive)
   - ✔ Модулі ядра (Планувальник, Диспетчер завдань, Переривання тощо)
   - ✔ рівень HAL
   - ✔ Локальний виклик процедур ядра,
   - ✔ Управління процесами і потоками
   - ✔ Підтримка системи введення / виводу (крім PnP).

2. Інші частини ядра повністю **сумісні з NT-архітектурою**: примітний Диспетчер Cache, Менеджер Кофігурації (включаючи реєстр) і Менеджер Пам'яті.

3. Підтримка додатки Win32 в основному, залежить від двох компонентів:
   - ✔ **Win32k** - режим ядра для GUI.
   - ✔ **Win32** бібліотеки (gdi32.dll, user32.dll, kernel32.dll, advapi32.dll)

Характеристика ядра ReactOS

Режим ядра ReactOS реалізований:

- ✔ в одному великому модулі (ntoskrnl),
- ✔ в шарі абстракції апаратного забезпечення (HAL),
- ✔ наборі завантажуваних модулів ядра (драйвера, і інші бібліотеки).

Ядро написано в переносимому мовою C++, компілюється за допомогою MinGW і Microsoft Visual Studio

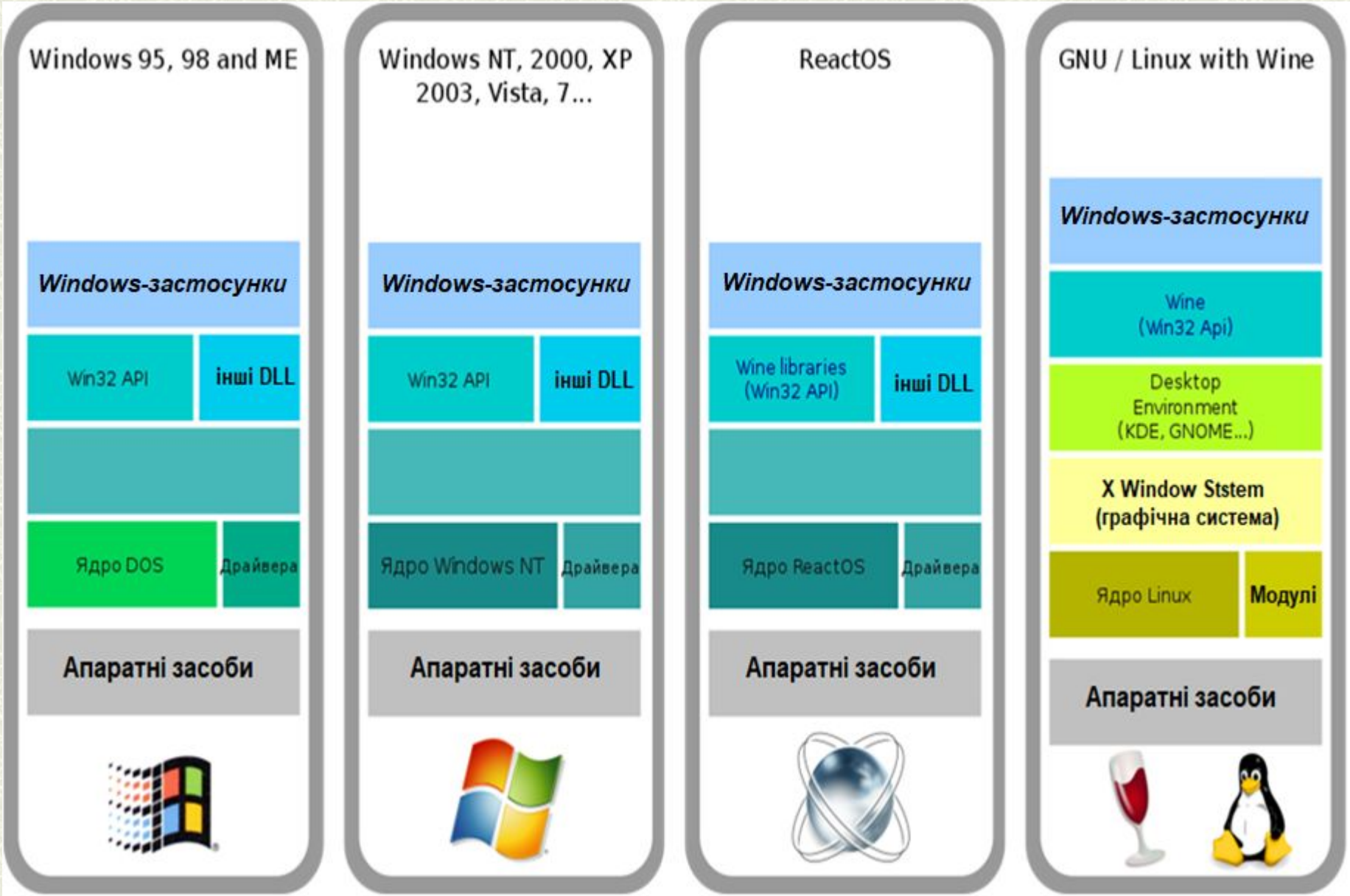
Таким чином для управління специфічними реалізаціями апаратного забезпечення потрібно HAL:

- ✔ переривання,
- ✔ ініціалізація процесора,
- ✔ DMA, доступ до шин PCI,
- ✔ таймери тощо.

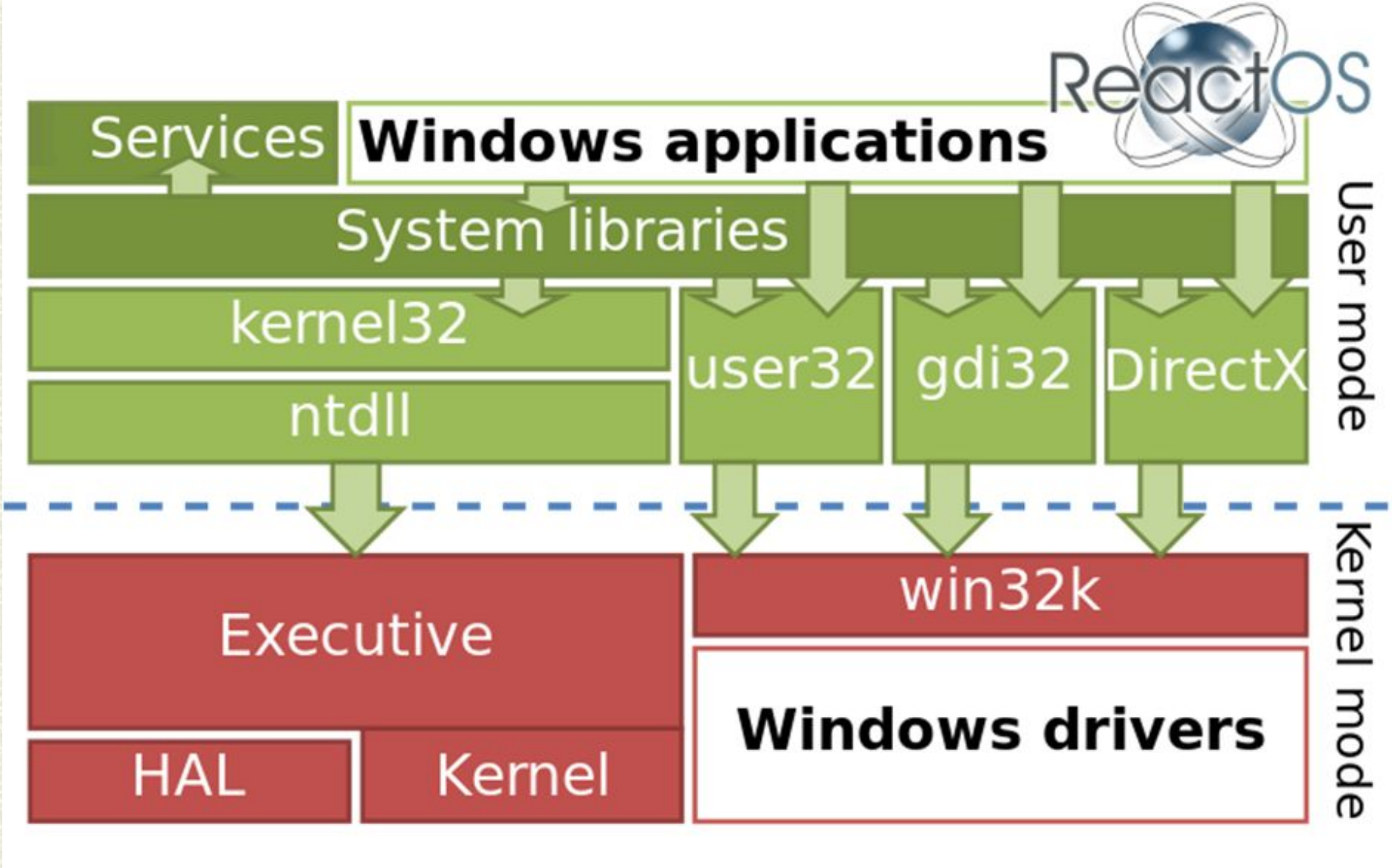
Деякі драйвера спеціалізовані для конкретного типу hardware (Драйвер PCI, драйвер ATAPI, IDE тощо).

У дереві початкових кодів є директорія ***\ntoskrnl***, яка реалізує саме **ядро.** Також є:

- ✔ директорія ***\hal*** для загального коду HAL,
- ✔ **\halx86, \halxbox** для SMP специфічного коду (spinlocks і переривання здійснені в HAL).



Порівняння архітектури ReactOS з іншими ОС

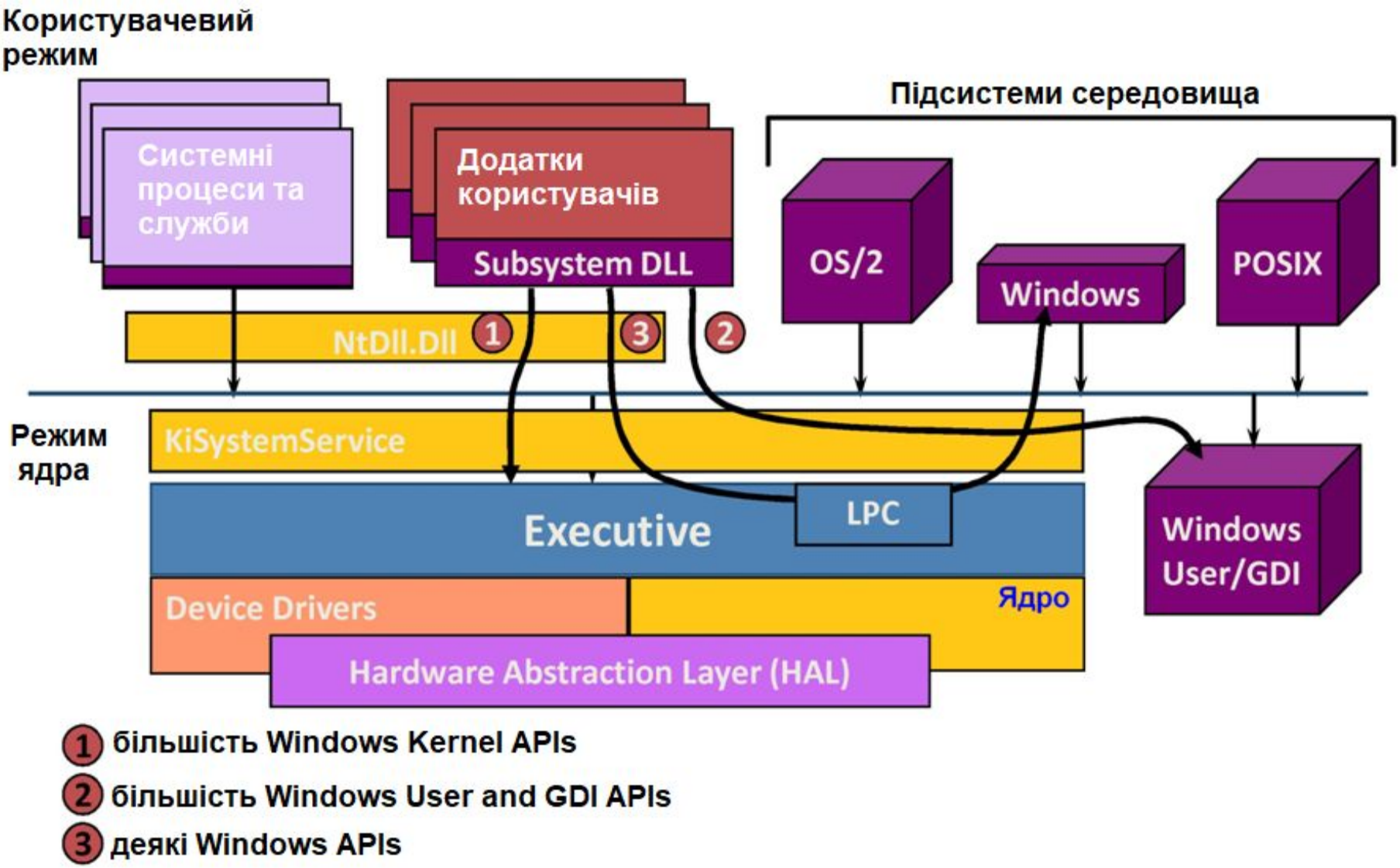


Архітектура ReactOS

білим кольором позначено стороннє програмне забезпечення.

зеленим - компоненти "простору користувача".

червоним - компоненти режиму ядра.



Спрощена архітектура ReactOS

Характеристика підсистем середовища ReactOS

1. API DLLs

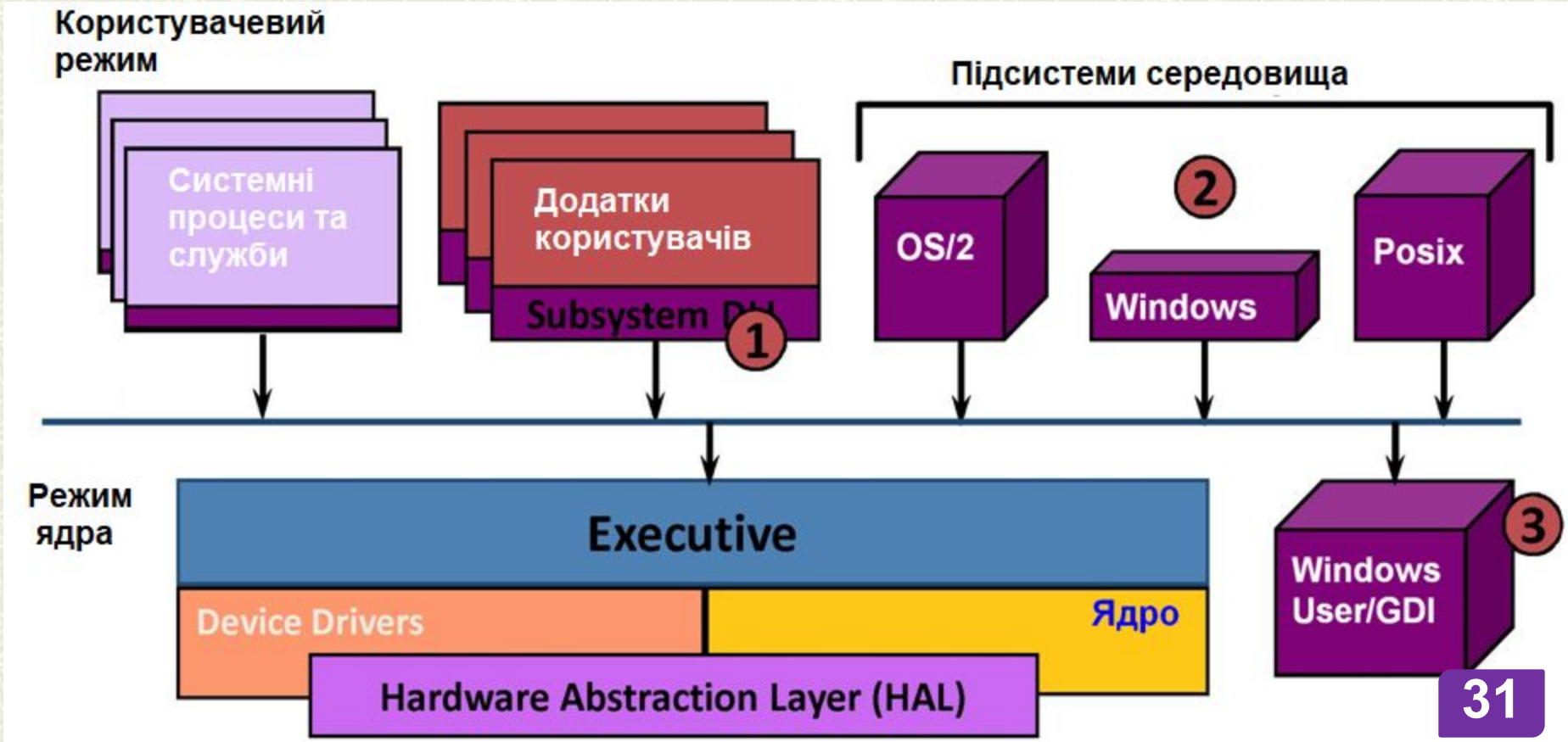
✔для Windows: kernel32.dll, gdi32.dll, user32.dll, і т.п ..

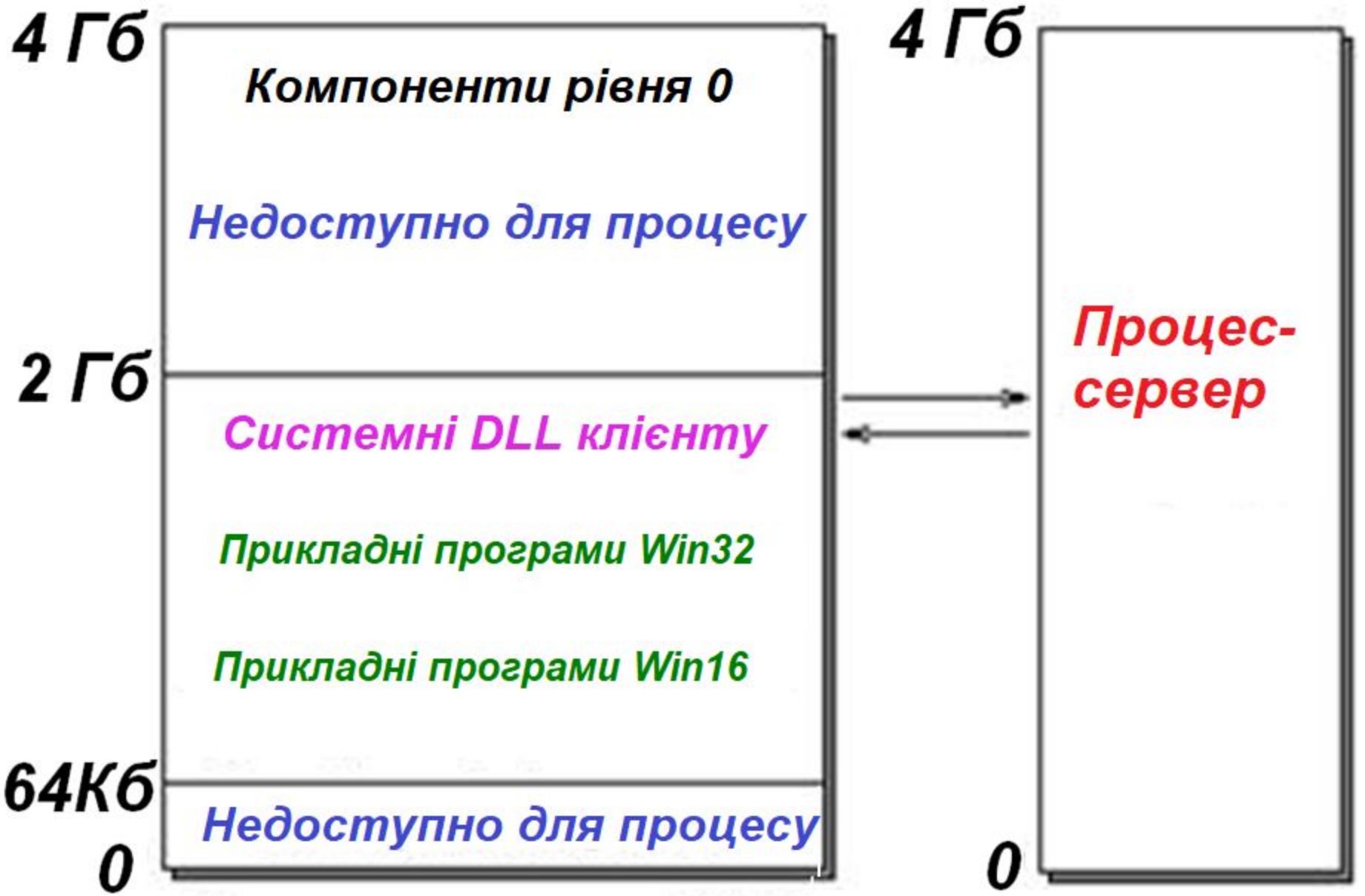
2. Процес Підсистеми середовища

✔для Windows: csrss.exe (Client Server Runtime SubSystem)

3. Тільки для Windows: код режиму ядра GDI

✔win32k.sys - (цей код був колись частиною csrss.exe)





Віртуальний адресний простір користувачевого процесу в ReactOS

Характеристика виконавчої системи ReactOS

Виконавча підсистема (Executive) - верхній рівень ядра, що представляє сервіс ядра підсистем середовища і інших серверів.

Диспетчер об'єктів забезпечує підтримку об'єктно-базованої структури ОС, що представляє ресурси у вигляді об'єктів - абстрактних інкапсульованих типів даних.

Менеджер процесів відстежує об'єкти процесів і потоків.

Менеджер віртуальної пам'яті виконує формування віртуального адресного простору процесу і відображає віртуальні адреси в адресних просторах процесів на фізичні сторінки пам'яті.

Менеджер введення-виведення забезпечує незалежний від пристроїв інтерфейс введення-виведення і відповідає за пересилання запитів на введення-виведення відповідним драйверам.

Менеджер безпеки перевіряє права доступу до об'єктів за запитами інших модулів Виконавчої системи і генерує контрольні повідомлення. Для отримання інформації про права і передачі контрольних повідомлень Менеджер безпеки взаємодіє з**Розпорядником локальної безпеки**.

Проблеми проектування призначеного для користувача режиму в ReactOS

1. **Проблема звернення до ntdll.dll.** Не можна використовувати Wine для kernel32. ReactOS вимагає для своєї реалізації виклик ntdll.dll, які потім виконує ці системні виклики, тоді як Wine передбачає здійснення функцій API безпосередньо в DLL.
2. **Проблема звернення до GDI і бібліотеки USER32.** Технологія NT вимагає безпосереднього звернення до GDI через режим ядра (win32k.sys) . В ReactOS в підсистемі Wine робиться це всередині GDI
3. **Проблема підтримки h-заголовків.** Інші DLL безпосередньо виконуються з підсистеми Wine з мінімальними змінами, щоб підтримувати наші заголовки і формувати систему.
4. **Проблема сумісності з Windows NT.** DLL припускають працювати на рівні сумісності двійкових кодів з Windows.
5. **Проблема підтримки файлових систем.** В ReactOS доступні файлові системи **FAT32, ext2, ISO-9660 (CDFS), NTFS, BtrFs, ReiserFS** і **UFS.**

3. Архітектура ОС Kolibri OS

Kolibri OS – операційна система для PC, повністю написана на **асемблері fasm**, поширювана на умовах GPL. Створена на основі MenuetOS. Є **альтернативної** операційною системою, так як вона використовує власні стандарти і **НЕ** заснована на POSIX.

Розробник	KolibriOS Project Team
Остання версія	0.7.7.0 r9861
Тип ядра	Монолітне ядро
Ліцензія	GNU General Public License і ін.
Стан	Актуальне
Веб сайт	www.kolibrios.org

Логотип операційної системи Kolibri OS

GNU General Public License — одна з найпопулярніших ліцензій на вільне програмне забезпечення, створена *Річардом Столменом* для проєкту GNU.

Мета GNU GPL — надання користувачеві прав на копіювання, зміни й розповсюдження програми та зобов'язань, згідно з якими користувачі всіх похідних від неї програм теж отримають ці права.

ПІДТРИМКА МОВИ В KOLIBRI ОС

Ада

Ci

C++

C--

Forth

Free Pascal

Assembler

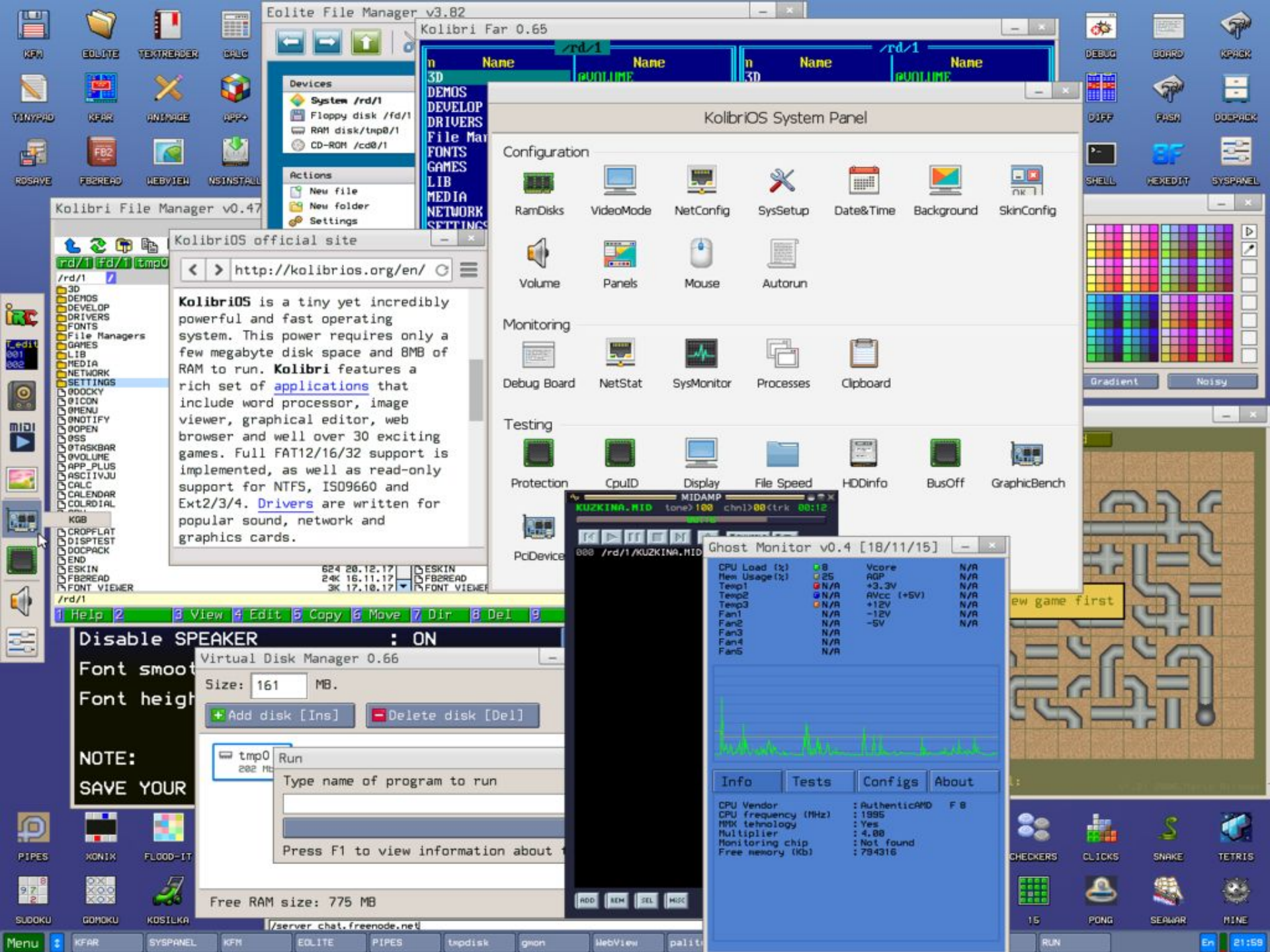
General Public Licence (GPL) - *кожен має необмежене право вивчати, довільно змінювати або змінювати програмне забезпечення, що володіє ліцензією GPL без будь-яких обмежень, але результат такої зміни теж повинен бути викладений під ліцензією GPL*

ІСТОРІЯ KOLIBRI OS

Віллі Туржанмаа	проект АтомОС	Kolibri v.0.1.0.0	Kolibri v.0.2.0.0	Майк Хіббет
2000	*2003*	*2004 травень*	*2004 Серпень*	*2005*

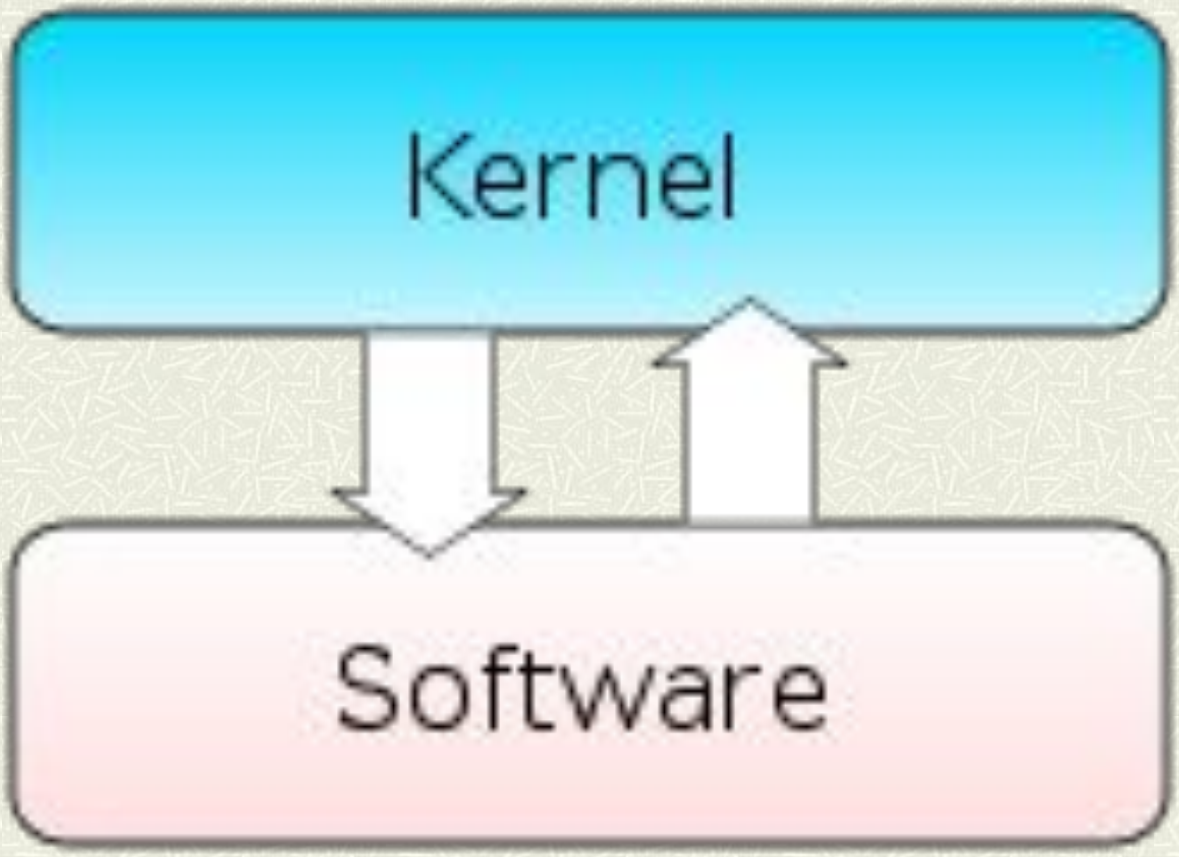
Kolibri v.0.3.1.0	Kolibri v.0.4.0.0	Kolibri v.0.5.1.0	Kolibri v.0.6.1.0	Kolibri v.0.7.0.0	Kolibri v.0.7.7.0+
2005 Березень	*2005 червень*	*2005 жовтень*	*2006 вересень*	*2007 червень*	*2012 жовтень*

t


Eolite File Manager v3.82
Kolibri Far 0.65
KolibriOS System Panel
Configuration
RamDisks
VideoMode
NetConfig
SysSetup
Date&Time
Background
SkinConfig
Volume
Panels
Mouse
Autorun
Monitoring
Debug Board
NetStat
SysMonitor
Processes
Clipboard
Testing
Protection
CpuID
Display
File Speed
HDDinfo
BusOff
GraphicBench
PciDevice
Kolibri File Manager v0.47
KolibriOS official site
http://kolibrios.org/en/
KolibriOS is a tiny yet incredibly powerful and fast operating system. This power requires only a few megabyte disk space and 8MB of RAM to run. Kolibri features a rich set of applications that include word processor, image viewer, graphical editor, web browser and well over 30 exciting games. Full FAT12/16/32 support is implemented, as well as read-only support for NTFS, ISO9660 and Ext2/3/4. Drivers are written for popular sound, network and graphics cards.
Disable SPEAKER : ON
Font smoot
Font heigh
NOTE:
SAVE YOUR
Virtual Disk Manager 0.66
Size: 161 MB.
Add disk [Ins]
Delete disk [Del]
Run
Type name of program to run
Press F1 to view information about
Free RAM size: 775 MB
MIDAMP
KUZKINA.MID
Ghost Monitor v0.4 [18/11/15]
Info
Tests
Configs
About
CPU Vendor : AuthenticAMD F 8
CPU frequency (MHz) : 1995
MMX technology : Yes
Multiplier : 4.00
Monitoring chip : Not found
Free memory (Kb) : 794316
Gradient
Noisy
Menu

ФУНКЦІОНАЛЬНІСТЬ KOLIBRI OS

1. ***БАГАТО ВАРІАНТІВ ЗАВАНТАЖЕННЯ,*** в тому числі з NTFS-розділу.
2. ***ГРАФІЧНИЙ ІНТЕРФЕЙС НА ОСНОВІ VESA.*** Можливість збільшення частоти монітора за рахунок зменшення дозволу.
3. ***СЕРЕДОВИЩЕ РОЗРОБКИ:*** редактор, макро-асемблер (fasm) для збирання ядра і додатків.
4. ***ПІДТРИМКА TCP/IP-СТЕКА*** для деяких мережевих карт.
5. ***ЄМНІСТЬ ОСНОВНОГО дистрибутивів*** – **1,44 Мбайт.**
6. ***МІНІМАЛЬНИЙ ОБСЯГ ОПЕРАТИВНОЇ ПАМ'ЯТІ*** – **8 Мбайт.**
7. ***Багатозадачність,*** можливість створення потоків, паралельне виконання системних викликів.
8. ***сторінковий адресації.***
9. ***Підтримка файлової СИСТЕМ*** FAT12, FAT16, FAT32, NTFS, ISO 9660.
10. ***ПІДТРИМКА звукового кодеку AC'97*** і Intel HD Audio для більшості популярних чіпсетів.
11. ***ПІДТРИМКА ВІДТВОРЕННЯ CD ТА DVD-ДИСКІВ***.
12. ***ПІДТРИМКА USB*** (Open Host Controller Interface (OHCI), Universal Host Controller Interface (UHCI), Enhanced Host Controller Interface (EHCI)).



Монолітна архітектура Kolibri OS

Монолітне ядро - класична і найбільш поширена архітектура ядра операційних систем. Монолітні ядра надають багатий набір абстракцій обладнання.

Всі частини монолітного ядра працюють *в одному адресному* просторі.

модулі Kolibri OS

Модуль управління потоками (Task manager)

Модуль динамічного розподілу пам'яті (Memory manager)

Модуль синхронізаційних примітивів (Synchronization primitives)

Модуль управління переривань (Interrupt controller)

Модуль вводу-виводу (IO manager)

Модуль початкового завантаження

Характеристика ядра Kolibri OS

Модуль управління потоками (Task manager) включає:

1. Планувальник потоків,
2. Набір функцій для управління станом потоку.

планувальник завдань реалізований у вигляді функції, що здійснює вибір активного потоку і перемикання контексту

Модуль динамічного розподілу пам'яті містить функції для динамічного розподілу пам'яті:

1. Alloc() - для виділення пам'яті даного розміру.
2. Free() - для звільнення виділеної пам'яті.

Модуль синхронізаційних примітивів включає:

1. Функції для створення синхронізаційних об'єктів.
2. Управління синхронізаційних примітивами:
   - ✔ семафори,
   - ✔ м'ютекси,
   - ✔ бінарні події.

Характеристика ядра Kolibri OS

Модуль управління переривань представляє:

1. Набір функцій для управління станом системи переривань
2. Функцію для інсталяції обробників переривань

Модуль вводу-виводу надає уніфікований інтерфейс для доступу до пристроїв введення-виведення.

1. Стандартні пристрої: таймер, миша, клавіатура, відеокарта, аудіо, мережеві карти, CD / DVD, HDD
2. Пристрої, з якими система взаємодіє через драйвера.

Модуль початкового завантаження - ініціалізує регістри процесора, а також структури і модулі операційної системи. Він инициализирует головне завдання (функція main) І запускає її.

Заснований на концепції завантаження образу RAM-діску в ОЗП:

- первинний завантажувач знаходить файл kernel.mnt,
- завантажує його в пам'ять і передає йому управління,
- зчитується образ файлу в ОЗУ в реальному режимі
- переміщається образ за межі 1 Мб (обмеження на адресацію в реальному режимі).

4. Архітектура Linux

Linux - сімейство **Unix-подібних операційних систем** на базі **ядра Linux**, що включають той чи інший набір утиліт і програм проекту **GNU**, і, можливо, інші компоненти. Як й ядро Linux, системи на його основі як правило створюються і поширюються відповідно до моделі розробки **вільного** і **відкритого програмного забезпечення**. Linux-системи поширюються в основному безкоштовно у вигляді різних **дистрибутивів** - у формі, готової для установки і зручною для супроводу і оновлень, - і мають свій набір **системних** і **прикладних** компонентів, як вільних, так можливо і власницьких

GPL Free Software

Free as in Freedom

Розробник	Лінус Торвальдс, Проект GNU і The Linux Foundation
Сімейство ОС	Unix-подібна операційна система
Вихідний код	відкритий
Тип ядра	Монолітне
Ліцензія	GNU GPL 2 і GNU GPL
Стан	Ведуться розробки нових версій
Веб сайт	www.kernel.org

GNU General Public License (Універсальна громадська ліцензія GNU або Відкрита ліцензійна угода GNU) - ліцензія на вільне програмне забезпечення, створена в рамках проекту GNU, по якій автор передає програмне забезпечення в суспільну власність. Принцип "успадкування" прав називається "**копілефт**" (**copyleft**)

Дистрибутив Softlanding Linux System +1993

Пітер Мак-Дональд

дистрибутив **Slackware**

Патрік Фолькердінк

версія	дата виходу	
1.00	17 липня	+1993
2.0	2 липня	1994
3.0	30 листопада	1995
4.0	17 травня	1999
7.0	25 жовтня	1999
8.0	1 липня	2001
9.0	19 березня	2003
10.0	23 червень	2004
11.0	2 жовтня	2006
12.0	1 липня	2007
13.0	27 серпня	2009
14.0	28 вересня	2012
14.2	30 червня	2016
15.0	2 лютого	2022

дистрибутив **Debian**

Ян Мердок

версія	кодова ім'я	Дата виходу
0.1		серпень 1993
1.1	Buzz	червень 1996
2.0	Hamm	липень 1998
3.0	Woody	липень 2002
4.0	Etch	квітень 2007
5.0	Lenny	лютий 2009
6.0	Squeeze	лютий 2011
7.0	Wheezy	травень 2013
....	....	...
....	....	...
11.5	Bullseye	серпень 2021

дистрибутив **Red Hat**

Марк Евінг

версія	кодове ім'я	дата виходу
		липень 1994
RHL 0.9	Halloween	жовтень 1994
RHL 2.0		серпень 1995
RHL 3.0.3	Picasso	березень 1996
RHL 4.0		жовтень 1996
RHL 5.1		червень 1998
RHL 6.0	Hedwig	травень 1999
RHL 7.0	Guinness	вересень 2000
RHL 7.0.90		січень 2001
RHL 7.3	Valhalla	травень 2002
RHL 8.0		вересень 2002
RHL 9.0.93	**Severn**	**липень 2003**

Red Hat Enterprise Linux

версія	кодове ім'я	дата виходу
RHEL 2.1	Pensacola	березень 2002
RHEL 3	Taroon	жовтень 2003
RHEL 4	Nahant	лютий 2005
RHEL 5	Tikanga	березень 2007
....	....	...
RHEL 9	Plow	Листопад 2021

Fedora Core

версія	кодове ім'я	дата виходу
FC 1	Yarrow	листопад 2003
FC 2	Tettnang	травень 2004
....	....	...
FC 16	Verne	листопад 2011
....	....	...
FC 20	Heisenbug	грудень 2013
....	....	...
FC 35		травень 2022

Популярність дистрибутивів за останні 6 місяців

станом на 25.08.2022 року

№ з/п	дистрибутив	H.P.D.	№ з/п	дистрибутив	H.P.D.
1	MX Linux	2669	14	KDE neon	591
2	EndeavourOS	2412	15	antiX	473
3	Mint	2270	16	PCLinuxOS	437
4	Manjaro	1509	17	Kali	432
5	Pop!_OS	1274	18	Slackware	424
6	**Ubuntu**	**1262**	19	FreeBSD	392
7	Fedora	1178	20	ArcoLinux	390
8	Debian	1029	21	SparkyLinux	381
9	Garuda	914	22	AlmaLinux	374
10	openSUSE	796	23	Kubuntu	366
11	Zorin	768	24	EasyOS	361
12	Lite	695	25	Solus	358
13	elementary	618	26	Q4OS	341

H.P.D. = Хітів в день

Повний перелік дистрибутивів: https://distrowatch.com/ (всього **275**)

Нумерація версій

A.	B.	C	[.D]
версія ядра	старша версія ревізії ядра	молодша версія ревізії ядра	невеликі виправлення (заплатки ядра)

Парні числа позначають стабільні ревізії - призначені для **промислового використання**

Непарні числа позначають ревізії для розробників

Наприклад,

2. 6. 24. 4

Ubuntu (людяність) – операційна система, заснована на **Debian** GNU/Linux. Основним розробником і спонсором є компанія **Canonical**. В даний час проект активно розвивається і підтримується вільним спільнотою.

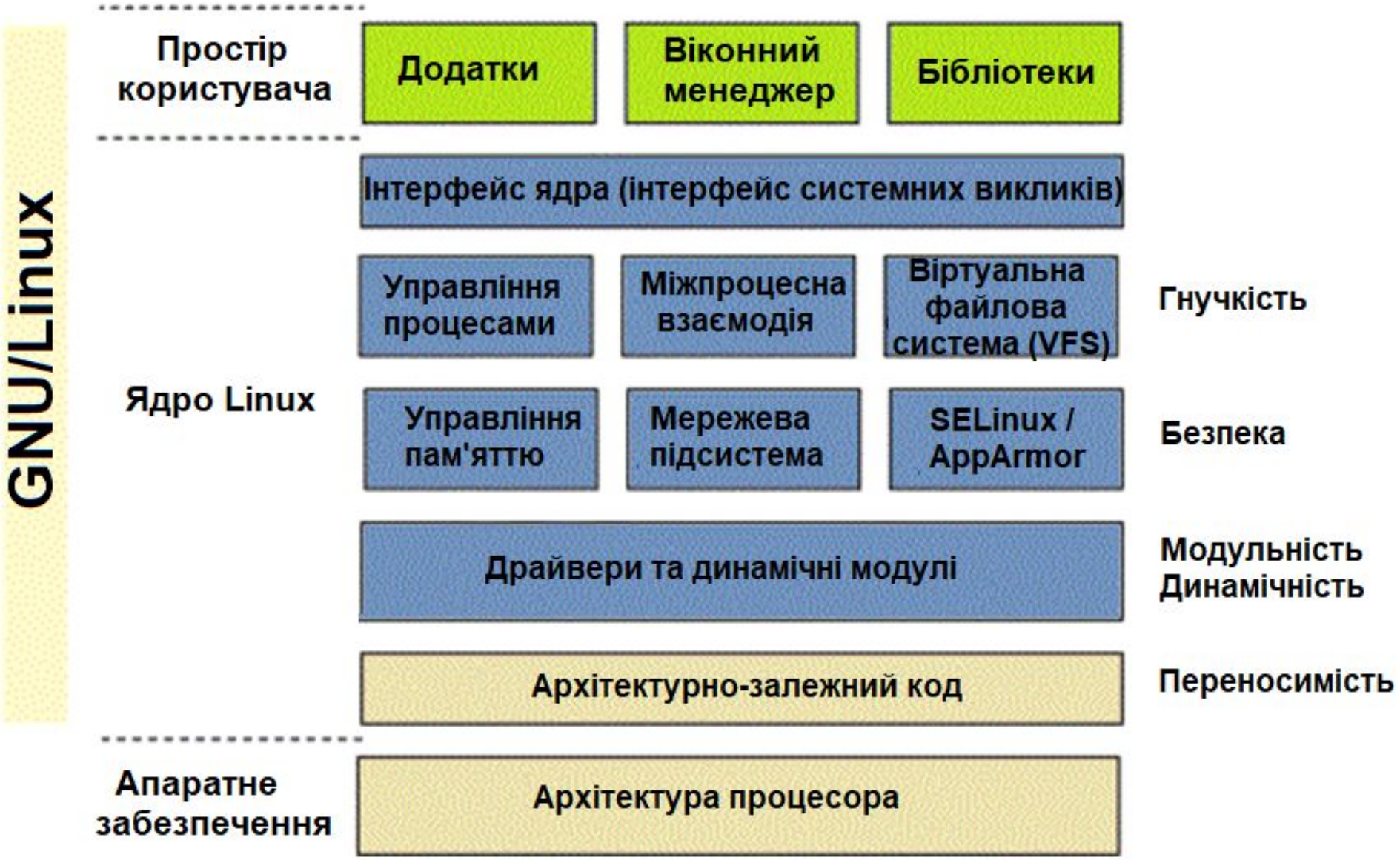
"Коло друзів"

Розробник	Canonical Ltd.
сімейство ОС	Linux
остання версія	22.04 – квітень 2022
Вихідний код	відкритий
Тип ядра	Монолітне
Ліцензія	GNU GPL 3
стан	Ведуться розробки нових версій
Веб сайт	Ubuntu.com

При розробці компонентів Ubuntu активно використовується мова програмування **Python**.

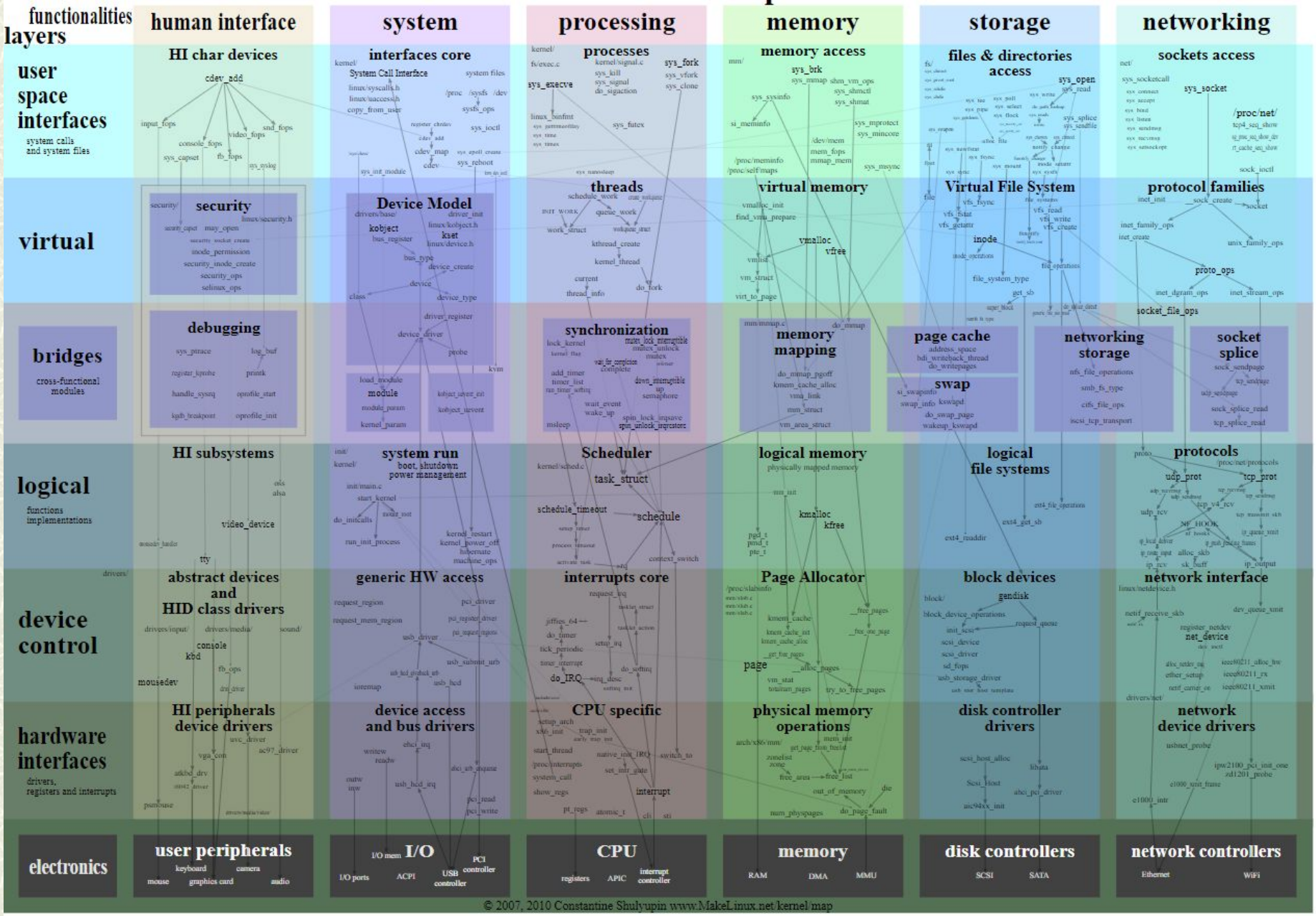
Зовнішній вигляд Робочого столу Ubuntu 22.04 LTS

LTS (Long Term Support - "підтримка на протязі тривалого періоду")



Архітектура Linux

Один з можливих поглядів на архітектуру ядра Linux



Архітектура ядра Linux

(джерело: https://makelinux.github.io/kernel/map/)

Дякую за увагу!