

Лекція 2. Операційна система Linux

01. Створення та розвиток ОС Linux
02. Принципи побудови ОС Linux
03. Організація роботи ОС Linux
04. Ядро ОС Linux
05. Системні служби ОС Linux (Linux system utilities)
06. Система управління пакетами
07. Оболонка (Shell)
08. Графічний інтерфейс користувача
09. Віконна підсистема X Window System (X11)
10. Графічне середовище робочого стола
11. Дистрибутиви ОС Linux
12. Рекомендована література

01.1. Створення та розвиток ОС Linux

UNIX(1971):

- 1) UNIX System V Release 4 (SVR4) (1989);
- 2) Berkeley Software Distribution (BSD) (1977-1995)

Річард Столмен (Richard Stallman) (1953, вік: 67) →
GNU Project (1983, MIT) → Free Software Foundation
(www.fsf.org)

Ендрю Таненбаум (Andrew Tanenbaum) (1944, вік: 76),
«Operating Systems: Design and Implementation» →
ОС MINIX

01.2. Створення та розвиток ОС Linux



Лінус Торвальдс (Linus Benedict Torvalds) (1969, вік: 50)

01.3. Створення та розвиток ОС Linux

1991 рік: Лінус купує клон IBM PC з процесором Intel 80386, отримує копію ОС MINIX, відвідує виступ Річарда Столмена

квітень 1991 року: Лінус (21 рік, 2-ий курс, University of Helsinki) починає писати перший код майбутньої системи Linux (context switch, x86 assembly language)

вересень 1991 року: версія 0.11 (трохи більше 10 тис. рядків коду)

1994 рік: версія ядра 1.0 (176.250 рядків коду)

01.4. Створення та розвиток ОС Linux

1996 рік: Лінус Торвальдс отримує ступінь магістра в області комп'ютерних наук (University of Helsinki), назва магістерської роботи: «Linux: A Portable Operating System»

1997-2003: робота на приватну компанію (Каліфорнія)

з 2003 року: робота у Linux Foundation (колишня Open Source Development Labs)

Непростий характер Лінуса: суперчка з Ендрю Таненбаумом (Tanenbaum–Torvalds debate), жорсткі висловлювання

01.5. Створення та розвиток ОС Linux

Назва: Linux vs. GNU/Linux

Організації:

Free Software Foundation, Open Source Initiative,
Linux Foundation, Apache Software Foundation та ін.

Стандарти UNIX/Linux:

- UNIX System V Release 4 (SVR4)
- POSIX (Portable Operating System Interface)
- Filesystem Hierarchy Standard (FHS)

01.6. Створення та розвиток ОС Linux

Версії ядра ОС Linux:

1994 рік: версія ядра 1.0 (176.250 рядків коду)

1999 рік: версія ядра 2.2 (1.800.847 рядків коду)

2003 рік: версія ядра 2.6 (5.929.913 рядків коду)

.....

2013 рік: версія ядра 3.10 (15.803.499 рядків коду)

2015 рік: версія ядра 4.1 (19.500.000 рядків коду)

.....

2020 рік: версія ядра 5.8 (>27.800.000 рядків коду)

02.1. Принципи побудови ОС Linux

Linux успадкувала у UNIX «ідеологію» побудови:

UNIX: ОС для групи досвідчених програмістів, які спільно працюють над складним проектом

на противагу «ідеології»:

Windows: «персональна» ОС для користувача-початківця або офісного працівника

02.2. Принципи побудови ОС Linux

Принципи, які характеризують «ідеологію» UNIX/Linux [Таненбаум]:

- 1) максимально можлива уніфікованість інтерфейсів користувача та програмних інтерфейсів → принцип найменшої несподіваності (principle of least surprise);
- 2) модульність та гнучкість (можливість конфігурувати різні набори модулів за потребою [досвідченого] користувача) → принцип: одна програма = одна функція;
- 3) мінімалізм у назвах команд та способах взаємодії користувача з ОС.

03.1. Організація роботи ОС Linux

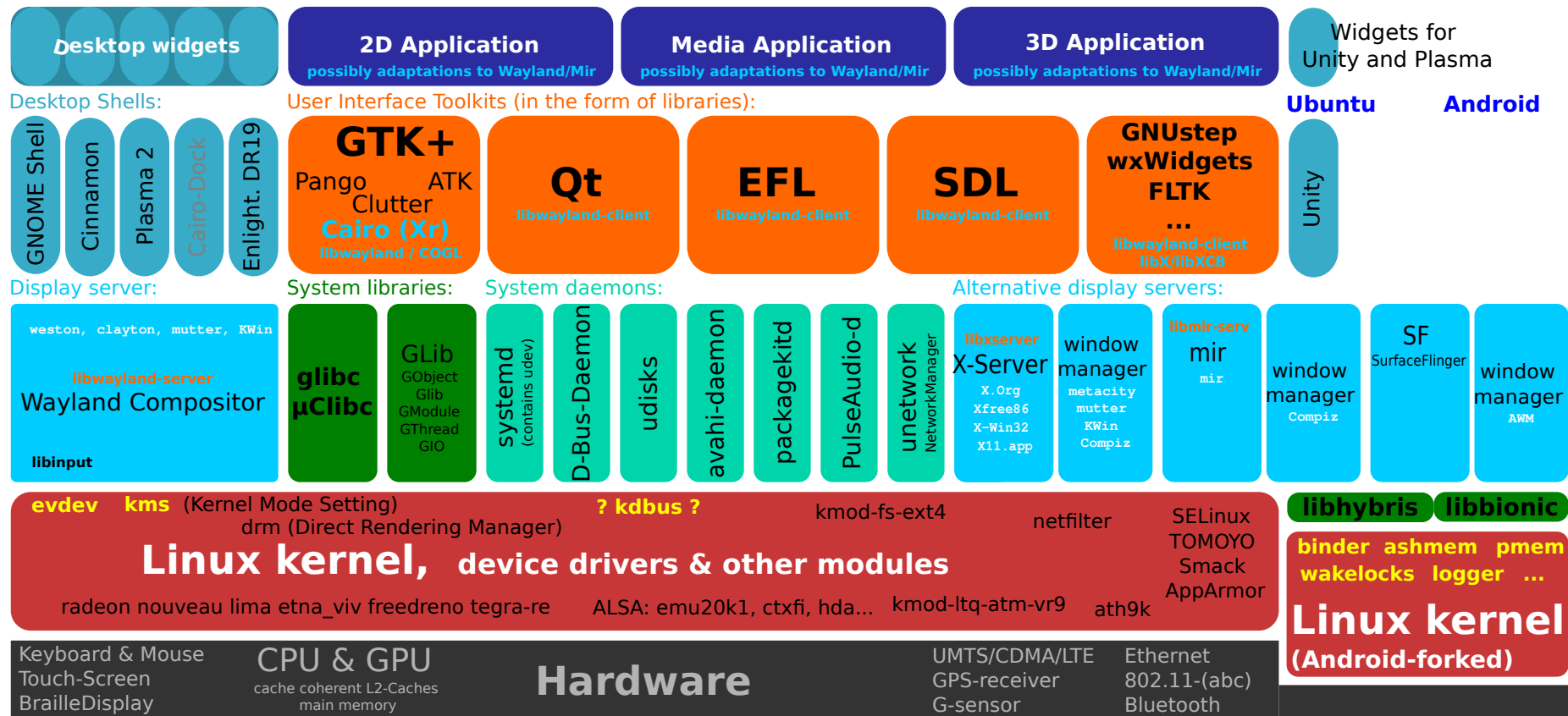
1. Ядро (Kernel): модулі ядра, які реалізують основні функції ОС
2. Системні бібліотеки (System libraries): libc, бібліотека функцій — системних викликів, та ін. Системні бібліотеки реалізують функціональність яка: 1) може не потребувати участі ядра; 2) потребує участі ядра (виклик функції транслюється у системний виклик)
3. Системні служби (System utilities): реалізують окремі спеціалізовані функції по управлінню системою (для яких не потрібний режим ядра). Можуть бути призначені для:
1) запуску один раз (старт/зупинка системи, конфігурація, налаштування і т.п.); 2) постійної роботи у фоновому режимі (daemons)

03.2. Організація роботи ОС Linux

system- management programs	user processes	user utility programs	compilers
system shared libraries			
Linux kernel			
loadable kernel modules			

Програмні компоненти ОС Linux (проста схема)

03.3. Організація роботи ОС Linux



Програмні компоненти ОС Linux (складна схема)

03.4. Організація роботи ОС Linux

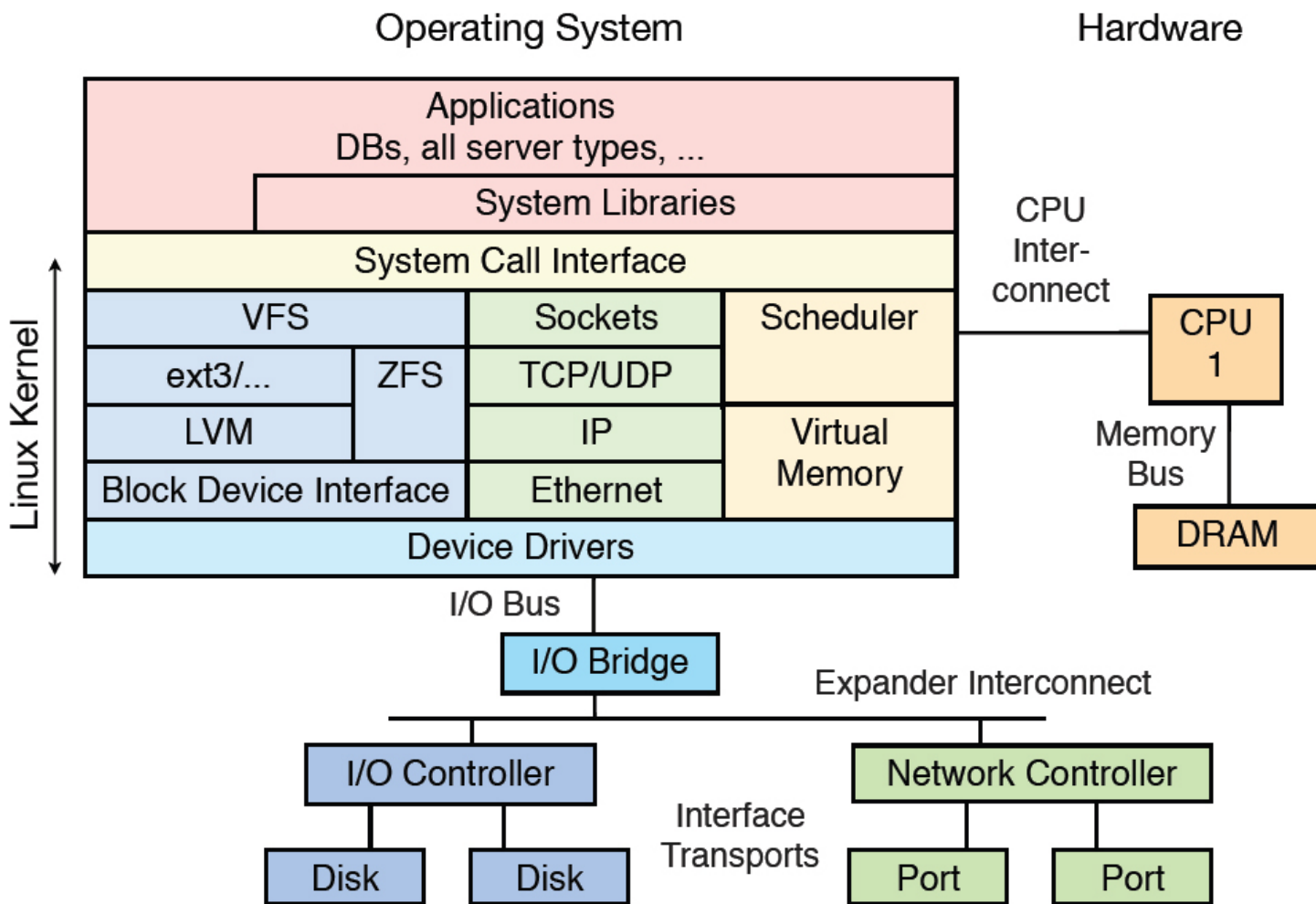
Ієрархія інтерфейсів:

1. Інтерфейс користувача: користувач - програма
2. Прикладні програмні інтерфейси: програма - прикладна функція
3. Системні бібліотечні функції: програма - системна функція
4. Інтерфейс системних викликів: програма - ядро ОС
System Call Interface (SCI)
5. Апаратний інтерфейс: апаратно-залежний код ядра - апаратура

04.1. Ядро ОС Linux

- Ядро ОС Linux має монолітну архітектуру та реалізує базову функціональність системного рівня,
- доступ до якої програмам користувацького рівня надається через інтерфейс системних викликів (System Call Interface, Linux kernel API), що містить близько 380 системних викликів.
- Програмний код ядра ОС Linux та відповідна документація є відкритими.
- Вони доступні на веб-сайті Linux Kernel Organization (www.kernel.org), яка представляє інтереси об'єднання розробників ядра.

04.2. Ядро ОС Linux: Структура ядра



04.3. Ядро ОС Linux: Основні функціональні компоненти (1-3)

1. Управління обчислювальними процесами: `fork/exit`, `scheduler`, `cgroups` та ін.
2. Управління пам'яттю (Memory management): `Virtual memory`, підвантаження та кеш сторінок, взаємодія з `Memory Manager`.
3. Управління пристроями (Device control): `block/character devices`, драйвера пристроїв вводу/виводу, `evdev` (обробник вхідних даних від драйверів пристроїв вводу), `I/O scheduler`.

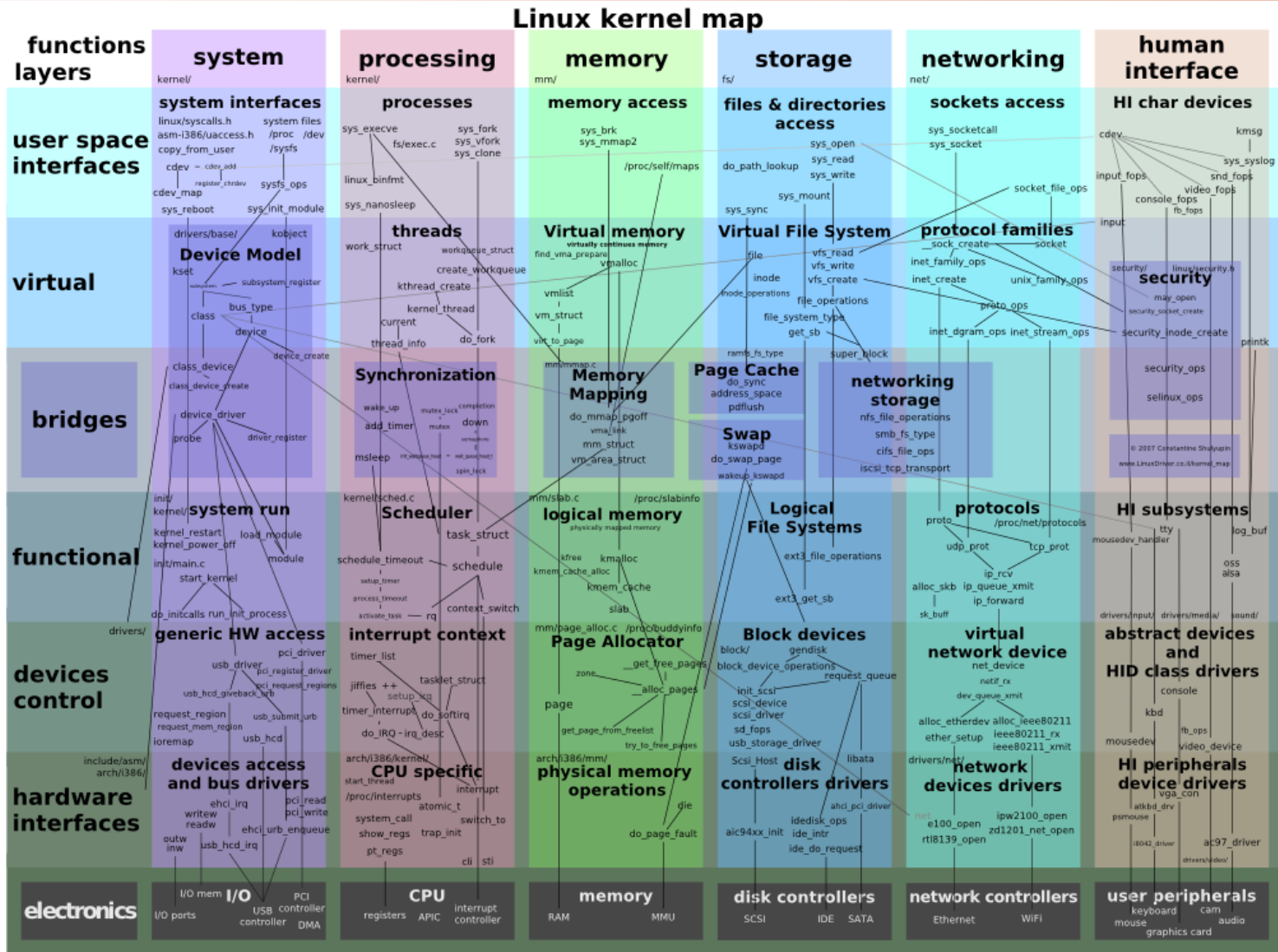
04.4. Ядро ОС Linux: Основні функціональні компоненти (4-6)

4. Управління ФС: virtual file system (VFS), драйвера файлових систем, LVM (Logical Volume Manager)
5. Мережна підсистема: реалізація стеку протоколів TCP/IP, sockets, netlink (аналог Unix domain sockets), network scheduler, netfilter
6. Графічна підсистема: Direct Rendering Manager (DRM) (взаємодія з GPU та графічною пам'яттю), Kernel mode setting (KMS) (взаємодія з контролером дисплею)

04.5. Ядро ОС Linux: Основні функціональні компоненти (7-9)

7. Аудіо-підсистема: Advanced Linux Sound Architecture (ALSA) (програмна «обгортка» драйверів аудіо-пристроїв).
8. Апаратно-залежна частина ядра (підтримка різних процесорних архітектур).
9. Підтримка роботи з Loadable kernel modules (LKM) (динамічне завантаження/вивантаження бінарних модулів ядра).

04.6. Ядро ОС Linux: схема функціональних залежностей в коді ядра



04.7. Ядро ОС Linux: параметри ядра

- Ядро ОС Linux — це складний програмний комплекс з великою кількістю параметрів, за допомогою яких можна змінювати режими його роботи та налагоджувати його функціональність в різних аспектах.
- Значення параметрів ядра та контроль за їх змінами грають важливу роль у забезпеченні ефективної роботи системи, її надійності та безпеки.
- Існує три способи конфігурації параметрів ядра:
 - 1) під час компіляції ядра з вихідних кодів,
 - 2) під час завантаження ядра в момент старту системи за допомогою параметрів командного рядка ядра,
 - 3) під час роботи системи за допомогою системної утиліти **sysctl**.
- Конфігурація (зміна) параметрів ядра вимагає прав адміністратора системи та має виконуватись з великою обережністю.

04.8. Ядро ОС Linux: параметри ядра

- Під час роботи системи параметри ядра ОС Linux відображаються у директорії **proc/sys/** віртуальної файлової системи **proc/** у вигляді окремих системних записів (файлів), які згруповані у декілька груп (директорій).
- Групи параметрів ядра у **proc/sys/** частково відповідають основним функціональним компонентам ядра:
 - управління обчислювальними процесами: більшість параметрів з групи **/proc/sys/kernel/**, в тому числі параметри диспетчеризації процесів **/proc/sys/kernel/sched_*** та **/proc/sys/kernel/sched_domain/**,

04.9. Ядро ОС Linux: параметри ядра

- управління пам'яттю: група параметрів [/proc/sys/vm/](#),
- управління вводом/виводом та пристроями: група параметрів [/proc/sys/dev/](#),
- файлова підсистема: група параметрів [/proc/sys/fs/](#),
- мережна підсистема: група параметрів [/proc/sys/net/](#).

Пояснення щодо самих параметрів ядра та їх значень можна знайти в Документації ядра ОС Linux:
www.kernel.org/doc/html/latest/admin-guide/sysctl/index.html.

04.10. Ядро ОС Linux: параметри ядра

Приклади визначення значень параметрів ядра:

1) kernel.threads-max - максимальна кількість програмних потоків в системі:

```
$cat /proc/sys/kernel/threads-max  
29147
```

2) kernel.yama.ptrace_scope — параметр модуля безпеки Yama, що визначає схему дозволів на використання системного виклику ptrace:

```
$cat /proc/sys/kernel/yama/ptrace_scope  
0
```

04.11. Ядро ОС Linux: Loadable Kernel Modules (LKM)

- В архітектурі ядра ОС Linux реалізовано механізм завантажуваних модулів ядра (loadable kernel modules, LKM), які можуть бути під'єднані чи від'єднані від ядра за потребою під час роботи системи.
- Основне призначення модулів ядра — забезпечення підтримки нових апаратних засобів у складі системи (драйвери пристроїв, які підключаються до комп'ютера).
- Завантажуваний модуль — це об'єктний файл з розширенням `.ko` (kernel object).
- Для роботи з модулями використовуються команди `insmod/rmmod` чи `modprobe` (додати та видалити модуль з ядра).

04.12. Ядро ОС Linux: Loadable Kernel Modules (LKM)

- Файли завантажуваних модулів зберігаються у директорії **/lib/modules/**.
- Список доступних для завантаження модулів можна подивитись в наступний спосіб:

```
$ls -R /lib/modules/4.9.0-11-amd64/kernel/ | grep .ko
```

- Загальну кількість завантажуваних модулів можна визначити так:

```
$ls -R /lib/modules/4.9.0-11-amd64/kernel/ | grep -c .ko
```

04.13. Ядро ОС Linux: Loadable Kernel Modules (LKM)

- Модулі, які завантажені і виконуються у складі ядра, відображаються у системні записи (файли) директорії **/proc/modules** віртуальної файлової системи **proc/**.
- Їх список можна подивитись, або виконавши:
`$less /proc/modules`
або у більш зручному вигляді за допомогою команди **lsmod**.
- Інформацію про параметри окремого завантажуваного модуля можна отримати або командою **modinfo**, наприклад:
`$modinfo usbcore`
(потрібні права адміністратора), або виконавши:
`$/sbin/modinfo usbcore`

04.14. Ядро ОС Linux: Loadable Kernel Modules (LKM)

- Використання завантажуваних модулів (LKM) може призвести до «забруднення» ядра.
- Ядро позначає себе «забрудненим» (tainted), коли з ним відбувається щось, що впливає на можливість ефективного усунення помилки або зависання ядра шляхом аналізу його програмного коду.
- Наприклад, у разі завантаження до ядра пропрієтарного модуля (стан «забруднення» ядра #0 (G/P, 1, proprietary module was loaded)) і виникнення помилки у роботі ядра, буде важко або взагалі неможливо встановити причину помилки, оскільки програмний код такого модуля є закритим.

04.15. Ядро ОС Linux: Loadable Kernel Modules (LKM)

- З точки зору безпеки стан «забруднення» ядра #13 (E, 8192, unsigned module was loaded) може бути ознакою завантаження в ядро руткіту (rootkit) або виконання експлойту ядра (kernel exploit).
- Встановити чи є ядро «забрудненим» можна за допомогою параметру ядра `kernel.tainted`. Якщо `kernel.tainted=0`, то ядро «чисте», інакше (`kernel.tainted>0`) ядро є «забрудненим» і значення `kernel.tainted` визначає стан «забруднення».
- Перелік станів «забруднення» (Table for decoding tainted state) можна знайти тут:
<https://www.kernel.org/doc/html/latest/admin-guide/tainted-kernels.html>

04.16. Ядро ОС Linux: Loadable Kernel Modules (LKM)

Приклад:

1) визначення стану «забруднення» ядра:

```
$cat /proc/sys/kernel/tainted  
4096
```

2) декодування стану «забруднення» ядра за допомогою скрипту kernel-chktaint:

```
$ sh kernel-chktaint
```

```
Kernel is "tainted" for the following reasons:
```

```
 * externally-built ('out-of-tree') module was loaded  (#12)
```

```
For a more detailed explanation of the various taint flags see  
Documentation/admin-guide/tainted-kernels.rst in the the Linux  
kernel sources
```

```
or https://kernel.org/doc/html/latest/admin-guide/tainted-kernels.html
```

```
Raw taint value as int/string: 4096/'G
```

O

'

04.17. Ядро ОС Linux: підсумки

1. Архітектура: монолітне ядро з динамічно завантажуваними модулями (LKM).
2. Висока продуктивність: основна ОС для суперкомп'ютерів з top500.
3. Переносимість: ядро Linux портоване на більшість існуючих апаратних платформ.
4. Відкритість для нових рішень: використовується для дослідження та розробки нових програмних технологій системного рівня.

05.1. Системні служби ОС Linux (Linux system utilities)

1. **systemd** → служба ініціалізації та запуску інших системних служб (прийшов на заміну процесу init)
2. **journald** → служба логування подій
3. **logind** → служба, яка обслуговує логування користувачів у систему
4. **networkd** → служба, яка забезпечує конфігурування мережних інтерфейсів
5. **timedated** → служба встановлення та контролю часу/дати
6. **udev** → служба контролю пристроїв

05.2. Системні служби ОС Linux (Linux system utilities)

- 7. D-Bus daemon → служба, що забезпечує міжпроцесорну взаємодію (inter-process communication (IPC)) та віддалений виклик процедур (remote procedure call (RPC))
- 8. PackageKitd → служба управління роботою менеджерів пакетів (установка, оновлення та видалення програмних компонент)
- 9. PulseAudio daemon → служба управління аудіо-підсистемою («надбудова» над ALSA)
- 10. UDisks daemon → служба контролю за підключенням та управління пристроями зовнішньої пам'яті
- . . .

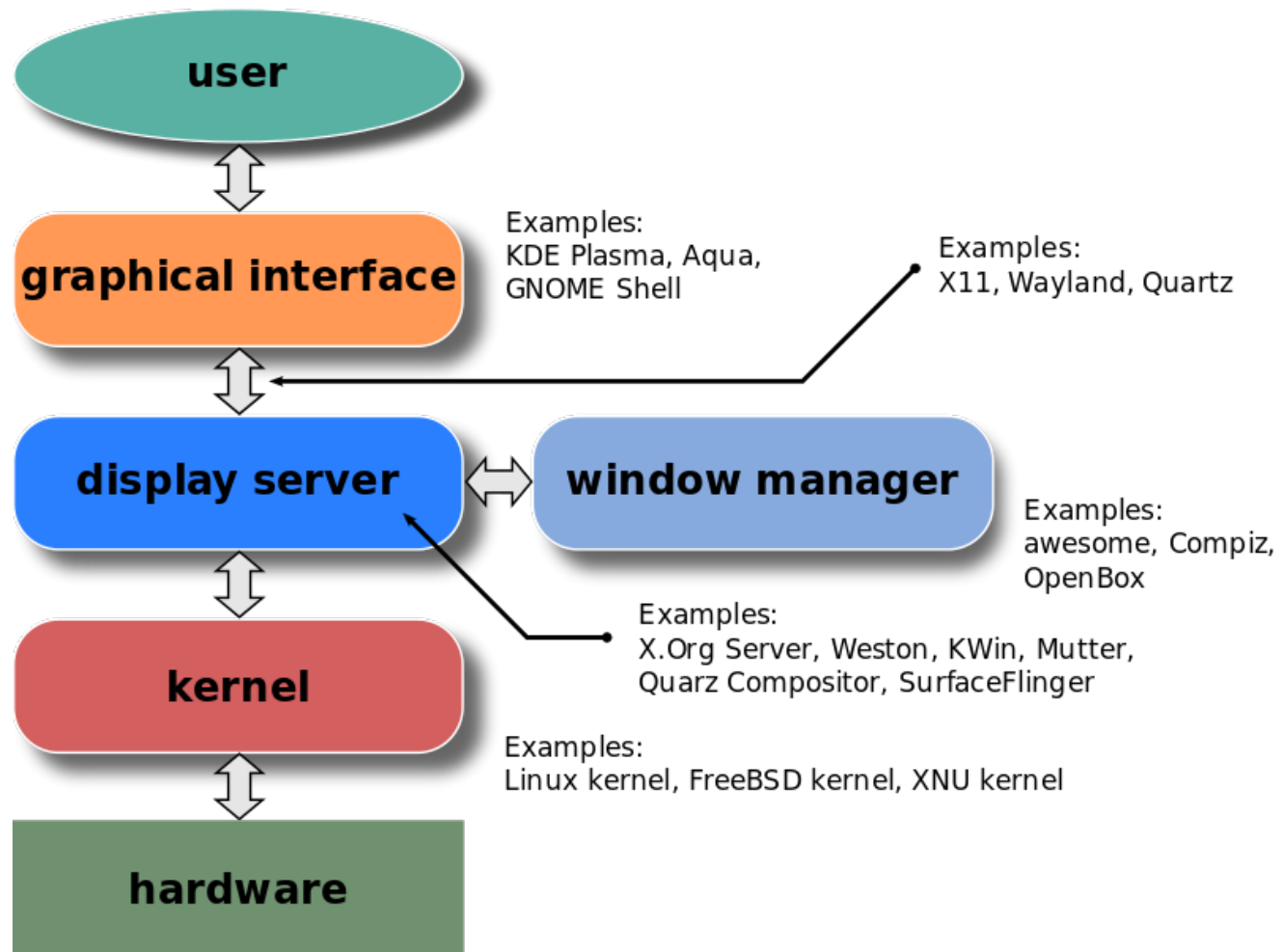
06. Система управління пакетами

1. Призначення: автоматизація робіт по установці, видаленню, налаштуванню та оновленню програм, програмних компонент та програмних бібліотек.
2. Пакет: це ПЗ + метадані (ім'я, дата, версія, залежності та ін.)
3. Принцип роботи: підтримується каталог (база даних) встановленого ПЗ, на основі аналізу якого + вказівок користувача виконується оновлення, установка чи видалення ПЗ
4. Основна проблема: підтримка складних взаємозалежностей між різними пакетами (за умов величезної кількості пакетів і зв'язків між ними)
5. Приклади: dpkg (Debian), RPM (RedHat, Fedora, openSUSE), Pacman (Arch Linux)

07. Оболонка (Shell)

1. Shell: 1) command-line interface; 2) Graphical shells
2. Інтерфейс командного рядка: термінал або емулятор терміналу
3. Unix shell: стандарт → **sh** (Bourne shell), інші відомі приклади: **bash** (Linux, Mac OS X), **tcsh** (FreeBSD)
4. Shell script: скриптова мова програмування, яка дозволяє автоматизувати функції управління ОС + прикладними програмами і даними

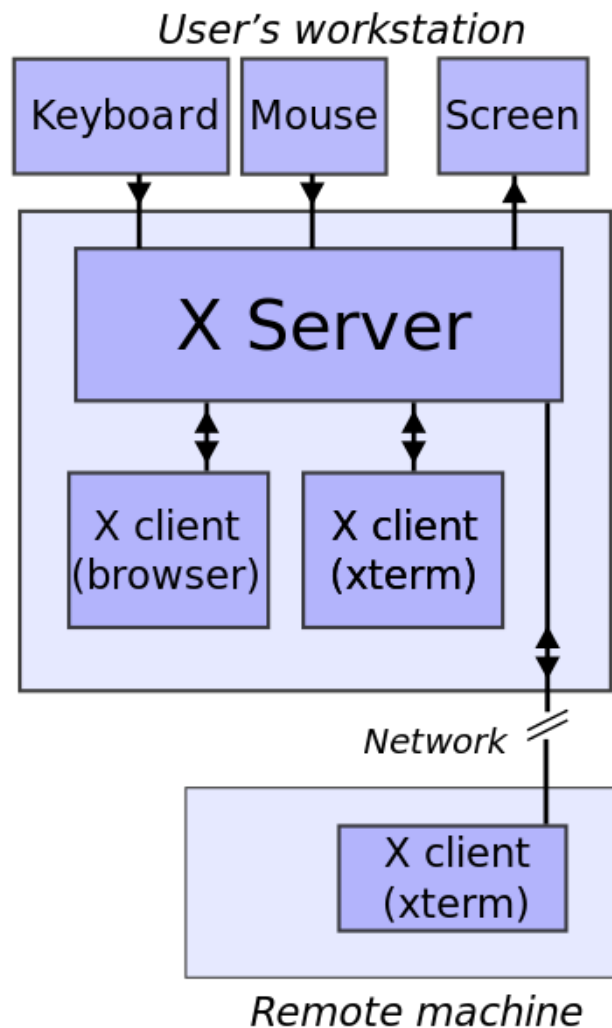
08. ОС Linux: Графічний інтерфейс користувача



09.1. Віконна система X Window System

1. Призначення: надання програмних інструментів та протоколів для забезпечення графічного інтерфейса користувача
2. Принцип роботи: максимальна функціональна декомпозиція
3. Складові: 1) X-server забезпечує відображення графічної інформації на дисплей та взаємодію з пристроями вводу (клавіатура, мишка); 2) «клієнт» - «поставляє матеріал» для відображення на дисплей та сприймає команди від пристроїв вводу; 3) менеджер вікон (window manager) визначає розміщення, зовнішній вигляд (форму, кольори і т.д.) та деталі відображення вікон

09.2. Віконна система X Window System



4. X-server взаємодіє з клієнтами по стандартному протоколу (X11)

5. «Клієнт» може бути програмою, яка виконується на іншому комп'ютері

6. Еталонна реалізація: X.Org Server (www.x.org)

10. Графічне середовище робочого стола (Desktop environment)

1. Надбудова над віконною системою (X Window System), яка визначає правила відображення вікон та інших елементів інтерфейсу, а також допустимі операції по взаємодії з ними.
2. В таких графічних середовищах, як правило, використовується метафора «робочого стола» (desktop environment): desktop, windows, icons, toolbars, folders, wallpapers, widgets + drag and drop
3. Найбільш розповсюджені desktop environments:
 - 1) K Desktop Environment (**KDE**) (openSUSE, Kubuntu)
 - 2) **GNOME 3** (Debian)
 - 3) **Unity** (Ubuntu)

11.1. Дистрибутиви ОС Linux

1. Що таке дистрибутив ОС Linux (Linux distribution)?
2. Чим дистрибутиви відрізняються один від одного?
3. Найбільш популярні дистрибутиви (distrowatch.com):
 - 1) Debian (→ Ubuntu → Linux Mint)
 - 2) Fedora (RedHat)
 - 3) openSUSE (SUSE)
 - 4) Arch Linux
4. Дистрибутиви Linux з платною підпискою на технічне обслуговування: RedHat, SUSE

11.2. Дистрибутиви ОС Linux

5. Легкі дестрибутиви Linux (Lightweight Linux distributions): займають мало місця на носії, потребують для роботи мінімальних параметрів системи (CPU, Main memory), the entire system can be run from RAM → популярні для створення LiveCD/LiveUSB → **Puppy Linux** (210 MB), **Lightweight Portable Security** (LPS) (500 MB), **Tiny Core Linux** (15 MB), **Bodhi Linux** (585 MB)

6. Спеціалізовані дистрибутиви: OpenWrt (primarily used on embedded devices to route network traffic)

7. “Дистрибутиви” для мобільних обчислювальних пристроїв: Android, Tizen, Firefox OS

12. Рекомендована література

1. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, 10th Edition, Wiley, 2018. – 942 p.
([Chapter 20. The Linux System](#))
2. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 4th Edition, Pearson, 2014. - 1136 p.
([Chapter 10. Case Study 1: UNIX, Linux, and Android](#))
3. Maurice J. Bach, The Design of the UNIX Operating System, Prentice Hall, 1986. - 486 p.
4. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd edition, O'Reilly Media, 2005. - 944 p.
5. Robert Love, Linux Kernel Development, 3rd edition, Addison-Wesley Professional, 2010. - 440 p.