**1. Поясніть етапи завантаження системи Linux**

Процес завантаження операційної системи Linux складається з декількох етапів, які послідовно виконуються для запуску системи та підготовки її до роботи.

**1. BIOS/UEFI – початкове завантаження**

Перший етап починається з включення комп'ютера, коли процесор шукає програму початкового завантаження в постійній пам’яті (ROM) або флеш-пам'яті.

* **BIOS (Basic Input/Output System)** – використовується у старих системах. BIOS виконує Power-On Self Test (**POST**) – тестування апаратної частини (процесор, пам’ять, жорсткі диски, відеокарта). Якщо перевірка успішна, BIOS шукає завантажувальний сектор (MBR) і передає керування завантажувачу.
* **UEFI (Unified Extensible Firmware Interface)** – заміна BIOS у сучасних комп’ютерах. Має графічний інтерфейс, підтримує GPT-розмітку дисків та працює з Secure Boot. UEFI завантажує завантажувач із спеціального розділу **EFI System Partition (ESP)**.

**2. Завантаження завантажувача (GRUB, LILO, systemd-boot тощо)**

Після проходження POST BIOS/UEFI передає керування завантажувачу, який знаходиться у **Master Boot Record (MBR)** або **EFI System Partition (ESP)**.

Основні завантажувачі Linux:

* **GRUB (GRand Unified Bootloader)** – найпопулярніший завантажувач. Він дозволяє обирати різні ядра та конфігурації.
* **LILO (Linux Loader)** – застарілий завантажувач, який не підтримує вибір ядер.
* **systemd-boot (колишній gummiboot)** – завантажувач, оптимізований для UEFI, що працює у systemd-середовищі.

GRUB шукає ядро Linux (зазвичай vmlinuz-<версія>), завантажує його в пам'ять та передає йому керування.

**3. Завантаження ядра Linux (Kernel)**

Ядро – це основа операційної системи. Воно відповідає за керування процесами, пам'яттю, файловими системами, пристроями вводу/виводу.

На цьому етапі ядро виконує наступні дії:

1. **Ініціалізує апаратне забезпечення**, використовуючи драйвери, що вбудовані в ядро або знаходяться в модульних файлах (.ko).
2. **Монтує root-файлову систему** (зазвичай / або тимчасова initramfs).
3. **Запускає init-процес** – це перший процес користувацького рівня (PID 1).

Ядро може використовувати **initramfs** або **initrd** – тимчасову файлову систему, яка містить модулі драйверів для розпізнавання пристроїв (наприклад, RAID, LVM, специфічні дискові контролери).

**4. Ініціалізація системи (init/systemd)**

Після завантаження ядра запускається перший системний процес – **ініціалізаційна система**.

Різні системи ініціалізації:

* **SysVinit** – стара система ініціалізації, яка використовує рівні запуску (**runlevels**).
* **Upstart** – заміна SysVinit від Ubuntu, яка використовувала події.
* **systemd** – сучасний стандарт в більшості дистрибутивів (Debian, Ubuntu, Fedora, Arch).

Systemd виконує такі основні дії:

1. Монтує необхідні файлові системи (/, /home, /boot, /var, /tmp).
2. Запускає сервіси (мережеві служби, демон журналювання, графічний сервер, SSH тощо).
3. Ініціалізує користувацькі процеси.

**Список сервісів можна переглянути командою:**

systemctl list-units --type=service  
  
**Журнали завантаження можна переглянути:**  
  
journalctl -b  
 **5. Завантаження користувацького середовища**

Після запуску всіх базових служб, система готова приймати користувача.

* У **графічному режимі** завантажується **Display Manager** (наприклад, GDM, LightDM, SDDM), який показує екран входу.
* У **консольному режимі** (без графіки) відкривається **tty** (термінальний емулятор), де можна увійти в систему через текстовий інтерфейс.

Після входу запускається користувацька оболонка (bash, zsh) або графічне середовище (GNOME, KDE, XFCE).

**Підсумок етапів завантаження Linux**

| **Етап** | **Опис** |
| --- | --- |
| **BIOS/UEFI** | Ініціалізація обладнання, передача керування завантажувачу |
| **Bootloader (GRUB, systemd-boot)** | Завантаження ядра Linux |
| **Kernel (ядро Linux)** | Ініціалізація пристроїв, монтування root-файлової системи |
| **Init system (systemd, SysVinit)** | Запуск системних служб та демонів |
| **Login/User session** | Вхід у систему через TTY або графічне середовище |

Цей процес виконується за **кілька секунд** (на сучасних системах) або трохи довше на старому залізі.

**2. Як переглянути системні журнали?**

В сучасних дистрибутивах Linux використовується systemd-journald. Основні команди:

* Перегляд загального журналу:
* journalctl
* Журнал поточного завантаження:
* journalctl -b
* Перегляд логів певної служби (наприклад, SSH):
* journalctl -u ssh
* Перегляд в реальному часі:
* journalctl -f
* Логи ядра:
* journalctl -k

Для систем без systemd, журнали можна переглядати в /var/log/, наприклад:

cat /var/log/syslog

cat /var/log/dmesg

**3. -rw------- : Опишіть цей дозвіл. Як додати прапорець виконуваного файлу?**

Розбір дозволів **-rw-------**:

* - → це звичайний файл.
* rw- → (власник) може читати (r) і писати (w), але не виконувати (-).
* --- → (група) не має жодних прав.
* --- → (інші) також не мають доступу.

Щоб зробити файл виконуваним, потрібно додати прапорець виконання (x):

chmod +x файл.sh

Або змінити права повністю:

chmod 700 файл.sh

**4. Яка різниця між apt і dpkg?**

**dpkg – низькорівневий менеджер пакетів:**

dpkg -i пакет.deb # Встановлення

dpkg -r пакет # Видалення

dpkg -l # Перегляд встановлених пакетів

**apt – високорівневий менеджер, який працює з dpkg, але також керує залежностями:**

apt install пакет # Автоматично встановлює пакет та його залежності

apt remove пакет # Видаляє пакет і непотрібні залежності

apt update # Оновлює список доступних пакетів

apt upgrade # Оновлює всі встановлені пакети

Отже, dpkg використовується для ручної роботи з .deb файлами, тоді як apt автоматизує процес керування пакетами.