HOMEWORK 3/ YAKOVENKO KRISTINA

**1. Етапи завантаження системи Linux**

Завантаження Linux відбувається в кілька послідовних етапів:

1. **BIOS/UEFI** — це перший рівень, що запускається після ввімкнення комп’ютера. Він перевіряє обладнання (пам’ять, процесор, диски тощо) і визначає, з якого пристрою завантажуватися (жорсткий диск, флешка і т.д.).
2. **Bootloader (GRUB)** — далі передається управління завантажувачу. GRUB (найпоширеніший у Linux) дозволяє вибрати, яку ОС або ядро запускати. Саме він підтягує Linux-ядро в пам’ять.
3. **Kernel (ядро)** — після вибору системи завантажується ядро Linux. Воно починає керувати ресурсами системи, ініціалізує драйвери для пристроїв, монтує кореневу файлову систему.
4. **Init-система (наприклад, systemd)** — ядро передає керування програмі ініціалізації. У сучасних дистрибутивах це найчастіше systemd. Вона запускає всі фонові сервіси, налаштовує мережу, монтує інші файлові системи, запускає демонів тощо.
5. **Login** — коли всі системні процеси запущені, з’являється екран входу (консоль або GUI), де користувач вводить логін і пароль.
6. **Shell або Desktop** — після успішного входу запускається інтерпретатор команд (bash) або графічне середовище (GNOME, KDE тощо), і користувач отримує доступ до системи.

**2. Як переглянути системні журнали в Linux**

У Linux журнали (логи) містять важливу інформацію про роботу системи, служб, помилки та події.

**Основні способи перегляду:**

* **journalctl** — сучасна утиліта для перегляду логів системи (для systemd-дистрибутивів):
* Щоб побачити останні помилки: journalctl -xe
* Логи конкретного сервісу: journalctl -u ssh

**Також можна переглядати логи вручну:**

* Усі системні журнали зберігаються у директорії:

cd /var/log

ls

Наприклад:

cat syslog

cat auth.log

cat dmesg

Деякі з цих файлів можуть бути доступні лише суперкористувачу (sudo cat файл).

**3. Пояснення -rw-------**

-rw------- - Це права доступу до файлу.

|  |  |
| --- | --- |
| - | Файл |
| rw- | Власник може читати (r) і записувати (w) |
| ------- | Інші користувачі не мають доступу |

**4. Різниця між apt і dpkg:**

Linux-дистрибутивах на основі Debian (наприклад, Ubuntu) встановлення програм зазвичай відбувається за допомогою менеджерів пакетів. Два найпоширеніші інструменти — це apt і dpkg, але вони працюють по-різному.

**apt** (Advanced Package Tool) — це менеджер пакетів високого рівня. Його головна перевага — робота з репозиторіями. Коли ми використовуємо команду apt install, система автоматично підключається до віддалених серверів (репозиторіїв), знаходить потрібний пакет, завантажує його разом із усіма необхідними залежностями і встановлює. Наприклад, якщо якась програма потребує ще кілька додаткових бібліотек для роботи, apt сам усе це врахує і встановить автоматично. Крім того, за допомогою apt можна оновлювати систему (apt update, apt upgrade) або видаляти пакети (apt remove).

**dpkg** — це інструмент нижчого рівня, який працює лише з локальними .deb файлами. Тобто, перед тим як скористатися dpkg, потрібно самостійно завантажити файл встановлення (наприклад, з офіційного сайту програми). Потім команда dpkg -i назва\_файлу.deb встановить цей пакет. Але якщо йому не вистачає якихось залежностей — dpkg не встановить їх автоматично, і доведеться додатково вручну шукати і встановлювати ці пакети.

| **Ознака** | **apt** | **dpkg** |
| --- | --- | --- |
| Тип інструменту | Менеджер пакетів високого рівня | Низькорівневий інструмент |
| Працює з репозиторіями | Так (підключає онлайн-джерела) | Ні (тільки локальні .deb-файли) |
| Автоматично вирішує залежності | Так | Ні |
| Типова команда | apt install firefox | dpkg -i firefox.deb |
| Може оновлювати всі пакети | Так (apt upgrade) | Ні |

Отже, apt зручніший для щоденного використання і роботи з інтернетом, тоді як dpkg застосовується, коли потрібно встановити щось вручну, наприклад, офіційний пакет, якого немає в репозиторіях.