Temat: Instalacja i deinstalacja dodatkowych składników systemu w Linux

ĆWICZENIE NR 23 – Instalacja i deinstalacja dodatkowych składników systemu Ubuntu 24.04 LTS (wersja zaktualizowana 2026)

Cel ogólny lekcji: Opanowanie metod instalacji, aktualizacji i usuwania oprogramowania w systemie Ubuntu 24.04 LTS z wykorzystaniem APT, Snap, Flatpak, dpkg oraz pracy z repozytoriami i plikami .deb, w bezpiecznym środowisku wirtualnym z wykorzystaniem migawek VirtualBox.

Cele szczegółowe. Uczeń potrafi:
- aktualizować listę pakietów i instalować oprogramowanie APT,
- wyszukiwać informacje o pakietach i ich wersjach,
- instalować i zarządzać aplikacjami Snap,
- korzystać z Flatpak i repozytorium Flathub,
- instalować aplikacje ręcznie z plików .deb,
- używać dpkg i devscripts/dget,
- korzystać z Synaptic,
- rozumie różnicę między update, upgrade, full-upgrade, do-release-upgrade,
- zna polecenia aktualizujące system, ale ich nie wykonuje,
- bezpiecznie pracuje na migawkach VirtualBox.

W zeszycie wykonaj notatkę z poniższych czynności. W zeszycie ucznia można zanotować postępy, napotkane trudności, a także ewentualne pytania lub refleksje dotyczące instalacji i zarządzania oprogramowaniem w systemie Ubuntu 24.04.

SEKCJA 0 – Praca z migawkami VirtualBox

Środowisko działa w VirtualBox. Prowadzący przygotował migawkę początkową.
Uczniowie wykonują wszystkie polecenia, a migawkę przywracają dopiero na końcu ćwiczenia.

Przywróć pierwszą migawkę maszyny

SEKCJA 0B – Polecenia, których NIE wykonujemy (tylko zapisujemy! Proszę zachować sens nie musi być słowo w słowo zanotowane)

Zapisz, ale nie wykonuj:

sudo apt upgrade

nie zmienia wersji systemu, tylko aktualizuje zainstalowane pakiety w ramach tej samej wersji Ubuntu do najnowszych wersji dostępnych w repozytoriach.
Jeśli chcesz zachować identyczną wersję pakietów, z którymi zaczynałeś ćwiczenia - możesz pominąć upgrade.
Czy to trwa długo? Zależy od kilku czynników: Liczby pakietów do aktualizacji. Prędkości internetu. Wydajności dysku i CPU.
Typowo: kilka minut przy małych aktualizacjach, kilkanaście minut przy dużych (np. po dłuższym czasie bez aktualizacji).
Co się dzieje, gdy NIE wykonasz apt upgrade?
Brak zmian:

- system pozostaje dokładnie w stanie, w jakim jest teraz,
- pakiety nie zostają zaktualizowane,
- ćwiczenia będą zachowywały się przewidywalnie (np. różne wersje narzędzi nie zaczną działać inaczej).

Potencjalne minusy:

- możesz mieć nowszą listę pakietów (po apt update), ale starsze zainstalowane wersje,
- zdarzają się sytuacje, że po instalacji nowych programów pojawia się komunikat o brakujących zależnościach - wtedy i tak system może poproszyć o aktualizację części bibliotek,
- brak aktualizacji bezpieczeństwa (dla szkolnych środowisk nie jest to zwykle krytyczne).

sudo apt full-upgrade

aktualizuje pakiety, ale w przeciwieństwie do zwykłego apt upgrade:
może usuwać pakiety i instalować nowe zależności, aby dokończyć aktualizację.
Czyli: zmienia wersje pakietów, dodaje brakujące zależności, usuwa pakiety, które przeszkadzają w aktualizacji.
Dlatego jest to polecenie bardziej agresywne i ingerujące w system.
Kiedy używać apt full-upgrade?
✓ przy dużych aktualizacjach pakietów (np. nowe jądro, nowe sterowniki)
✓ gdy system mówi, że pakiety są „trzymane” (held back)
✓ gdy trzeba zaktualizować dużo zależności naraz
Kiedy NIE używać?

❌ podczas ćwiczeń szkolnych, kiedy chcesz zachować przewidywalność

❌ na serwerach produkcyjnych bez testów

❌ jeśli boisz się, że jakaś aplikacja przestanie działać

```
sudo do-release-upgrade
```

Polecenie do aktualizacji Ubuntu do nowszej wersji. Jak to działa?

apt upgrade > aktualizuje pakiety w tej samej wersji Ubuntu

apt full-upgrade > może usuwać pakiety, ale wciąż zostajesz na tej samej wersji Ubuntu

do-release-upgrade > przechodzi do NOWEGO Ubuntu (nowy numer LTS / nowa wersja)

release-upgrade wymaga czystego systemu bez zalegających aktualizacji.

To jedyne oficjalne i bezpieczne narzędzie do zmiany wersji Ubuntu.

Jeśli nowa wersja LTS jeszcze nie jest widoczna, czasem trzeba dopisać:

```
sudo do-release-upgrade -d
```

(-d używa najnowszej dostępnej wersji stabilnej - NIE wersji beta)

SEKCJA 1 – APT

apt (Advanced Packaging Tool) System zarządzania pakietami APT jest menedżerem do obsługi pakietów. Ma możliwość pobrania, instalacji pakietów oraz automatycznej instalacji brakujących zależności. Wykonaj polecenia służące do aktualizacji i dodawania oprogramowania:

1.1 Zaktualizuj liste pakietów

```
sudo apt update
```

Polecenie pobiera aktualne informacje o pakietach z repozytoriów systemu.

1.2 Wyświetl liste zainstalowanych pakietów

```
apt list --installed
```

Polecenie wypisze nazwy pakietów, ich wersje oraz informacje o tym, czy pochodzą z repozytoriów systemowych.

1.3 Zainstaluj edytor tekstu Geany

```
sudo apt install geany
```

Uruchom program wpisując w terminalu:

```
geany
```

zamknij program

1.4 Wyświetl listę pakietów, które mają dostępne aktualizacje:

```
apt list --upgradable
```

1.5.1 Wyszukaj pakiety, których nazwa lub opis pasuje do wpisanej frazy (przestarzałe):

```
apt-cache search zfs
```

```
apt-cache search finch
```

1.5.1 Wyszukaj pakiety, których nazwa lub opis pasuje do wpisanej frazy (rekomendowane):

```
apt search zfs
```

```
apt show zfs
```

W Ubuntu nie istnieje realny pakiet o nazwie zfs.

Istnieją jedynie konkretne pakiety implementujące ZFS, takie jak: zfsutils-linux, zsys, zfs-zed,

To dlatego: apt show zfs zwraca komunikat:

State: nie jest pakietem rzeczywistym (wirtualnym)

N: Nie udało się wybrać wersji kandydującej pakietu "zfs"

To normalne - pakiet wirtualny jest tylko etykietą, wskazującą na grupę funkcjonalności, ale sam nie zawiera plików.

```
apt search finch
```

```
apt show finch
```

1.6 Sprawdź polityki wersji pakietu

```
apt-cache policy imagemagick
```

1.7 Spróbuj usunąć pakiet:

```
sudo apt remove --purge firefox
```

Jeśli widzisz komunkat w stylu:

Oczekiwanie na blokadę bufora: Nie udało się uzyskać blokady /var/lib/dpkg/lock-frontend

Ubuntu domyślnie uruchamia automatyczne aktualizacje (unattended-upgrades).

Gdy są uruchomione, APT **zamykany jest na czas instalacji**, aby uniknąć uszkodzenia systemu.

1.8 Proces unattended-upgr sam się zakończy. Sprawdź status:

```
ps aux | grep unattended
```

1.9 Nie chcesz czekać zakończ proces (bezpieczna opcja)

```
sudo systemctl stop unattended-upgrades
```

1.10 Potwierdz usunięcie lub usuń uruchamiajac ponownie polecenie:

```
sudo apt remove --purge firefox
```

SEKCJA 2 – Repozytoria i deb-src

Repozytoria to serwery, z których Ubuntu pobiera program, deb-src to ich wersje źródłowe - pozwalają pobierać kody źródłowe pakietów, a nie tylko gotowe programy.

2.1 Włącz repozytora źródłowe deb-src

```
sudo sed -Ei 's/^# deb-src /deb-src /' /etc/apt/sources.list
```

```
sudo apt update
```

2.2 Spróbuj pobrać kodów źródłowych pakietów

```
apt source audacious
```

```
apt source audacious-plugins
```

Jeśli widzisz komunkat w stylu:

E: W pliku sources.list należy umieścić jakieś URI dla "deb-src"

a. Otwórz plik głównego repozytorium:

```
sudo nano /etc/apt/sources.list.d/ubuntu.sources
```

b. W tym pliku znajdziesz sekcję podobną do:

Types: deb

URIs: http://archive.ubuntu.com/ubuntu/

Suites: noble noble-updates noble-security

Components: main restricted universe multiverse

Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

Aby włączyć źródła, dopisz drugą sekcję **deb-src**:

Types: deb-src

URIs: http://archive.ubuntu.com/ubuntu/

Suites: noble noble-updates noble-security

Components: main restricted universe multiverse

Signed-By: /usr/share/keyrings/ubuntu-archive-keyring.gpg

Uwaga: format .sources nie używa hashy (#).

Każda sekcja to osobny blok.

c. Zapisz i wyjdź:

CTRL + O, ENTER, potem CTRL + X

d. Zaktualizuj listę repozytoriów:

```
sudo apt update
```

e. Pobierz źródła:

```
apt source audacious
```

```
apt source audacious-plugins
```

2.3 Dodaj repozytorium Brave

a. Zainstaluj program curl bez zadawania pytań, z uprawnieniami administratora.

```
sudo apt install curl -y
```

b. Utwórz plik repozytorium Brave w formacie .list i wpisz do niego linię repozytorium APT:

```
sudo curl -fsSL https://brave-browser-apt-release.s3.brave.com/brave-browser-archive-keyring.gpg \
 -o /usr/share/keyrings/brave-browser-archive-keyring.gpg
```

Co robi to polecenie `curl -fsSL` ?
-f — kończy działanie, jeśli wystąpi błąd HTTP (np. 404),
-s — tryb „silent” (bez zbędnych komunikatów),
-S — pokazuje błąd, jeśli wystąpi,
-L — podąża za ewentualnymi przekierowaniami.
Polecenie pobiera klucz kryptograficzny GPG, którego Ubuntu używa do weryfikowania autentyczności pakietów Brave.
Opcja -o /usr/share/keyrings/brave-browser-archive-keyring.gpg
nakazuje zapisać pobrany klucz w katalogu systemowym keyrings, zgodnie z aktualną polityką bezpieczeństwa APT (oddzielne keyringi per repozytorium).

```
echo "deb [signed-by=/usr/share/keyrings/brave-browser-archive-keyring.gpg] \
 https://brave-browser-apt-release.s3.brave.com/ stable main" \
 | sudo tee /etc/apt/sources.list.d/brave-browser-release.list
```

Jak działa to polecenie echo "deb [...]" ?
Tworzy poprawny wiersz definicji repozytorium dla APT.
[signed-by=...] Określa, że APT ma używać właśnie pobranego klucza GPG do weryfikacji pakietów z tego repozytorium.
https://brave-browser-apt-release.s3.brave.com/ stable main To właściwy adres repozytorium Brave oraz sekcja stable main, czyli standardowe źródło pakietów.
| sudo tee /etc/apt/sources.list.d/brave-browser-release.list Zapisuje wynik do osobnego pliku listy repozytoriów (zalecana praktyka – każdy vendor w osobnym pliku).

2.4 Prawidłowo zainstaluj przeglądarkę Brave w Ubuntu:

```
sudo apt update
```

```
sudo apt install brave-browser -y
```

SEKCJA 3 – Ubuntu Software

Ubuntu Software (App Center) to sklep z aplikacjami dla Ubuntu, który pozwala łatwo instalować i usuwać programy jednym kliknięciem, bez użycia terminala.

3.1 Uruchom "Ubuntu Software"

Wyszukaj aplikację Brave (jest > instalowane ręcznie w sekcji 2).

3.2 Zainstaluj dowolną aplikację z listy.

3.3 Odinstaluj z listy dowolną aplikację za pomocą Ubuntu Software.

SEKCJA 4 – Snap

Snap to system instalacji aplikacji w Ubuntu, który dostarcza programy w postaci samodzielnych pakietów działających na każdej wersji systemu. Aplikacje Snap uruchamiane są w bezpiecznym *sandboxie* i aktualizują się automatycznie.

Sandbox to bezpieczne, odizolowane środowisko, w którym aplikacja działa oddzielnie od reszty systemu. Dzięki temu nawet gdy aplikacja zrobi coś niepożądanego, nie może uszkodzić Ubuntu ani dostać się do danych użytkownika.

4.1 Sprawdź wersję Snap

```
snap version
```

4.2 Usuń Firefox i zainstaluj ponownie w wersji Snap (wersja domyślna)

```
sudo snap remove firefox
```

Zainstaluje najnowszą dostępną wersję Firefox z oficjalnego kanału Snap.

```
sudo snap install firefox
```

4.3 Wyszukaj VLC i zainstaluj

```
snap find vlc
```

```
sudo snap install vlc
```

4.4 Zaktualizuj wszystkie zainstalowane aplikacje w systemie obsługiwane przez Snap, uruchom terminal i wykonaj polecenie:

```
sudo snap refresh
```

SEKCJA 5 – Flatpak

Flatpak to uniwersalny sposób instalowania aplikacji w Linuxie – działa na każdej dystrybucji i uruchamia programy w bezpiecznym, odizolowanym środowisku (*sandbox*). Dzięki temu pozwala instalować **najnowsze i niezależne od systemu wersje aplikacji**.

5.1 Zainstaluj Flatpak

```
sudo apt install flatpak -y
```

Flathub to **centralne repozytorium aplikacji Flatpak**, w którym znajdują się najnowsze, wieloplatformowe programy dla Linuxa. Umożliwia łatwą instalację, aktualizację i uruchamianie aplikacji w jednakowych wersjach na różnych dystrybucjach.

5.2 Dodaj repozytorium Flathub do Flatpak (niezbędne do instalacji większości aplikacji Flatpak), wykonaj w terminalu:

```
flatpak remote-add --if-not-exists flathub https://flathub.org/repo/flathub.flatpakrepo
```

5.3 Zainstaluj edytor tekstu Gedit z Flathub (Flatpak) (czasochłonna)

```
flatpak install -y flathub org.gnome.gedit
```

Po zakończeniu instalacji uruchom program:

```
flatpak run org.gnome.gedit
```

5.4 Zainstaluj program GIMP z Flathub za pomocą Flatpak

```
flatpak install -y flathub org.gimp.GIMP
```

Po zakończeniu instalacji uruchom GIMP:

```
flatpak run org.gimp.GIMP
```

5.5 Odinstaluj Gedit i GIMP

```
flatpak uninstall org.gnome.gedit
```

```
flatpak uninstall org.gimp.GIMP
```

SEKCJA 6 – Instalacja .deb

Instalacja pliku **.deb** to ręczne Dodaj programu do systemu - pobierasz gotowy pakiet i instalujesz go poleceniem dpkg, niezależnie od repozytoriów Ubuntu.

6.1 Instalacja Google Chrome

a. Pobierz pakiet instalacyjny Google Chrome (DEB) przy użyciu wget

```
wget https://dl.google.com/linux/direct/google-chrome-stable_current_amd64.deb
```

b. Ręczna instalacja pakietu DEB przy użyciu dpkg bezpośrednio w systemie (z pominięciem repozytoriów)

```
sudo dpkg -i google-chrome-stable_current_amd64.deb
```

c. Wymuszone naprawianie niespójności pakietów i instalacja brakujących zależności.

```
sudo apt --fix-broken install -y
```

SEKCJA 6A – Instalacja z lokalnego pliku .deb przy użyciu APT (offline)

APT potrafi instalować pakiety z **lokalnych plików .deb**, **automatycznie** dobierając zależności (w przeciwieństwie do dpkg). To bezpieczniejsza metoda instalacji z pliku.

Cel: Zainstaluj **Midnight Commander (mc)** z lokalnych plików .deb (bez repozytoriów).

Kroki:

6A.1 Pobierz wymagane pakiety do katalogu roboczego (np. /home/ubuntu)

```
sudo apt-get update
```

```
cd /home/ubuntu
```

```
apt-get download mc mc-data mailcap
```

6A.2 Zainstaluj wszystkie pliki jednocześnie przy użyciu APT (z lokalnych źródeł)

```
sudo apt install ./mc_*.deb ./mc-data_*.deb ./mailcap_*.deb
```

6A.3 Weryfikacja

```
mc --version
```

Uwagi:

- **Użycie prefiksu ./ jest obowiązkowe** - wskazuje APT, że instalujesz **z plików lokalnych**, a nie z repozytoriów sieciowych.

- Jeśli instalujesz **jako root**, może pojawić się ostrzeżenie o piaskownicy _apt: N: Pobieranie ma miejsce poza piaskownicą, jako root … - pkgAcquire::Run (13: Brak dostępu) To **nie jest błąd**. Aby go zredukować, możesz nadać plikom czytelne uprawnienia:

```
sudo chown root:root /home/ubuntu/*.deb
```

```
sudo chmod 644 /home/ubuntu/*.deb
```

Pytania kontrolne:

1. Dlaczego apt install ./plik.deb jest bezpieczniejsze niż dpkg -i?
2. Co robi prefiks ./ w poleceniu apt install?
3. Jak sprawdzisz, czy mc został poprawnie zainstalowany?

SEKCJA 6B – Instalacja z lokalnego repozytorium APT (offline, własny katalog)

Możesz stworzyć **własne repozytorium lokalne** (folder z pakietami .deb), wygenerować indeks i dodać je do źródeł APT. Dzięki temu APT będzie **rozwiązywał zależności offline** i traktował katalog jak normalne repozytorium.

Cel: Utwórz repozytorium lokalne i zainstaluj mc z tego repozytorium.

Kroki:

6B.0 Sprzątanie poprzednich prób (opcjonalnie, ale zalecane)

```
sudo rm -f /home/ubuntu/localrepo/.deb
```

```
sudo rm -f /home/ubuntu/localrepo/Packages
```

```
sudo apt-get clean
```

6B.1 Utwórz katalog repozytorium i skopiuj wymagane pakiety (spójne wersje)

```
mkdir -p /home/ubuntu/localrepo
```

```
cd /home/ubuntu/localrepo
```

Wariant A: użyj sprawdzonych adresów (mc 4.8.33-1 działa na 24.04)

```
wget 'http://archive.ubuntu.com/ubuntu/pool/universe/m/mc/mc_4.8.33-1_amd64.deb'
```

```
wget 'http://archive.ubuntu.com/ubuntu/pool/universe/m/mc/mc-data_4.8.33-1_all.deb'
```

```
wget
'http://archive.ubuntu.com/ubuntu/pool/main/m/mailcap/mailcap_3.70+nmu1ubuntu1_
all.deb'
```

(Alternatywa: jeśli któryś adres 404 - pobierz kandydatów wersji automatycznie)

```
sudo apt-get update

apt-get download mc mc-data mailcap
```

6B.2 Ustaw właściciela i prawa (żeby _apt czytał pliki bez ostrzeżeń)

```
sudo chown root:root /home/ubuntu/localrepo/*

sudo chmod 644 /home/ubuntu/localrepo/*.deb
```

6B.3 Wygeneruj indeks repozytorium (APT czyta 'Packages' oraz 'Packages.gz')

```
cd /home/ubuntu/localrepo

sudo dpkg-scanpackages . /dev/null | sudo tee Packages > /dev/null

sudo gzip -c Packages | sudo tee Packages.gz > /dev/null
```

6B.4 Dodaj repo lokalne do APT (w osobnym pliku; unikniesz duplikatów)

```
echo "deb [trusted=yes] file:/home/ubuntu/localrepo ./" | sudo tee
/etc/apt/sources.list.d/localrepo.list
```

6B.5 Odśwież listy pakietów i zainstaluj mc z repo lokalnego

```
sudo apt-get update

sudo apt-get install mc
```

6B.6 Weryfikacja

```
mc --version

apt-cache policy mc mc-data
```

Uwagi (ważne przy Noble 24.04):

- Upewnij się, że **mc i mc-data** mają **zgodne wersje**. Dla 24.04 poprawnie działa para 4.8.33-1. *Jeśli system żąda innej wersji (np. 3:4.8.30-1ubuntu0.1), pobierz dokładnie taką parę i wymuś ją:* sudo apt-get install mc=3:4.8.30-1ubuntu0.1 mc-data=3:4.8.30-1ubuntu0.1

- Jeśli pojawi się komunikat **„Nie odnaleziono pliku …/Packages”**, sprawdź, czy w katalogu repo masz **oba** pliki: Packages **i** Packages.gz (krok 3), i wykonaj ponownie sudo apt-get update.
- Ostrzeżenia o piaskownicy _apt przy repo file:/ są **niegroźne** (rozwiązanie: krok 2 - chown/chmod).

Pytania kontrolne:

1. Jak wygenerować plik Packages i Packages.gz dla repozytorium lokalnego?
2. Jaki wpis dodajesz do źródeł APT, aby wskazać katalog repozytorium lokalnego?
3. Co zrobisz, gdy APT zgłasza konflikt wersji mc vs mc-data?

SEKCJA 7 – dpkg

dpkg to podstawowy, niskopoziomowy instalator pakietów w Ubuntu, który służy do ręcznej instalacji i usuwania plików .deb, bez automatycznego rozwiązywania zależności.

7.1 Wyświetlenie listy pakietów:

```
dpkg --list
```

7.2 Pobierz i instalacja pliku .deb i wymuszone naprawianie niespójności pakietów i instalacja brakujących zależności:

```
wget https://repo.skype.com/latest/skypeforlinux-64.deb
```

```
sudo dpkg -i skypeforlinux-64.deb
```

```
sudo apt --fix-broken install -y
```

7.3 Sprawdź stanu pakietu za pomocą dpkg:

```
dpkg --status skypeforlinux
```

7.4 Usunięcie pakietu za pomocą dpkg:

```
sudo dpkg --remove skypeforlinux
```

SEKCJA 8 – devscripts i dget

devscripts to zestaw narzędzi dla twórców pakietów w Ubuntu, a dget służy do pobierania pakietów źródłowych .dsc bezpośrednio z repozytoriów Debiana lub Ubuntu.

8.1 Wykonaj instalacje pakietu narzędzi deweloperskich devscripts:

```
sudo apt install devscripts -y
```

8.2 Pobierz źródła pakietu Debian:

a. Pobierz źródła pakietu Debian z podanego adresu URL za pomocą dget:

```
dget https://archive.debian.org/debian/pool/main/h/haskell-tldr/haskell-tldr_0.4.0-2.dsc
```

Pobrałeś pakiet poprawnie, ale dget zakończył się błędem: Nie można sprawdzić podpisu: Brak klucza publicznego Validation FAILED!!

To jest normalne przy starych pakietach z Debian Archive.

Archiwum zawiera stare, historyczne pakiety z podpisami, których klucze są już nieważne lub nie są w systemie. Sam Debian ostrzega, że takie źródła mogą nie przechodzić walidacji.

b. Pobierz źródła pakietu Debian z pominięciem weryfikacji podpisu i automatycznym wypakowaniem:

```
dget -u https://archive.debian.org/debian/pool/main/h/haskell-tldr/haskell-tldr_0.4.0-2.dsc
```

-u (czyli unpack, ale bez weryfikacji podpisu)

SEKCJA 9 – Synaptic

Synaptic to graficzny menedżer pakietów APT - pozwala instalować, usuwać i aktualizować programy w Ubuntu za pomocą wygodnego interfejsu, bez użycia terminala.

9.1 Zainstaluj Menedżer pakietów Synaptic:

```
sudo apt install synaptic
```

9.2 Zainstaluj aplikację VLC przez Synaptic:

a. Otwórz Synaptic
   - Menu > Administrator lub System > Menedżer pakietów Synaptic
   - Wprowadź swoje hasło (sudo).

b. Wyszukaj VLC
   - W górnym pasku kliknij **Szukaj**.
   - Wpisz: **vlc**
   - Kliknij **Szukaj** lub Enter.

c. Zaznacz VLC do instalacji
   - Na liście znajdź pakiet **vlc** (opis: „odtwarzacz multimedialny…”).
   - Kliknij na nim **prawym przyciskiem myszy** > wybierz **Zaznacz do instalacji**.

- Synaptic może zapytać o dodatkowe pakiety. Kliknij **Zaznacz**.

d. Rozpocznij instalację

- Kliknij **Zastosuj** (u góry).
- W oknie podsumowania kliknij ponownie **Zastosuj**.
- Poczekaj, aż pobieranie i instalacja się zakończą.

e. Uruchom VLC aby potwierdzić, że instalacja przebiegła pomyślnie

Po instalacji możesz go znaleźć tutaj: Menu > **VLC media player**

lub uruchomić z terminala:

vlc

SEK CJA 10 – Instalacja VirtualBox

VirtualBox to program do uruchamiania maszyn wirtualnych - pozwala instalować i testować inne systemy operacyjne bez zmiany głównego systemu.

10.1 WERSJA STANDARDOWA (klasyczna .list)

a. Dodaj klucza GPG (wymagane w Ubuntu 24.04)

```
wget -O- https://www.virtualbox.org/download/oracle_vbox_2016.asc \
  | gpg --dearmor \
  | sudo tee /usr/share/keyrings/oracle-virtualbox-2016.gpg > /dev/null
```

b. Dodaj repozytorium VirtualBox dla Ubuntu 24.04 (Noble)

```
echo "deb [arch=amd64 signed-by=/usr/share/keyrings/oracle-virtualbox-2016.gpg] \
https://download.virtualbox.org/virtualbox/debian noble contrib" \
  | sudo tee /etc/apt/sources.list.d/virtualbox.list
```

c. Zainstaluj VirtualBox

```
sudo apt update
sudo apt install virtualbox-7.2
```

d. Sprawdź wersji VirtualBox:

```
vboxmanage --version
```

e. Sprawdź, czy VirtualBox działa

```
virtualbox
```

10.2 WERSJA W FORMACIE .sources (nowy format Ubuntu 24.04)

Ubuntu 24.04 wprowadziło nową strukturę repozytoriów APT - VirtualBox wymaga ręcznej konwersji wg dokumentacji.

a. Dodaj klucza GPG

```
wget -O- https://www.virtualbox.org/download/oracle_vbox_2016.asc \
  | sudo gpg --dearmor --yes \
  --output /usr/share/keyrings/oracle-virtualbox-2016.gpg
```

b. Dodaj repozytorium VirtualBox w formacie .sources

Utwórz plik:

```
sudo nano /etc/apt/sources.list.d/virtualbox.sources
```

Wklej:

```
Types: deb
URIs: https://download.virtualbox.org/virtualbox/debian
Suites: noble
Components: contrib
Architectures: amd64
Signed-By: /usr/share/keyrings/oracle-virtualbox-2016.gpg
```

Zapisz CTRL+O, zamknij CTRL+X.

c. Zainstaluj VirtualBox

```
sudo apt update
sudo apt install virtualbox-7.2
```

d. Usuń stare, niepotrzebne zależności

```
sudo apt autoremove
```

e. Sprawdź wersji VirtualBox:

```
vboxmanage --version
```

f. Sprawdź, czy VirtualBox działa

```
virtualbox
```

SEKCJA 11 – Przywrócenie migawki

Po sprawdzeniu przez prowadzącego: Przywróć migawkę „Stan początkowy”.

Podsumowanie

Uczeń potrafi instalować i usuwać programy różnymi metodami, pracować bezpiecznie, korzystać z migawek, rozumie aktualizacje systemu i menedżery pakietów.