“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**WORK-CASE №4**

з дисципліни “Операційні системи”:

Виконали студенти

групи РПЗ-13А та РПЗ-13Б

Eleven Two Zeroes:

Vlad Sapozhnyk

Max Karpenko

Dmytro Onufriiev

Київ 2024

Робота студентів групи РПЗ-13А та РПЗ-13Б Eleven Two Zeroes: Vlad Sapozhnyk, Max Karpenko and Dmytro Onufriiev.

**Завдання:**

**1. В ході роботи досить часто виникає необхідність встановлювати нові програми та додатки. Для цього необхідно в терміналі вміти працювати з менеджерами пакетів:**

- Дайте розгорнуте визначення таким поняттям як «пакет» та «репозиторій».

*У контекстi Arch Linux.*

In Arch Linux, **packages** are compressed tar.xz or tar.zst archives that contain all the files of a program, the scripts for installing it, and metadata describing how these files should be placed on the user's file system. In addition, the metadata includes information about dependencies that are necessary for the program to work correctly. Packages in Arch Linux are designed to be easy to create and manage, reflecting the distribution's philosophy of simplicity and brevity.

**Arch Linux uses several core repositories, each with a specific purpose:**

* **Core**: Contains the essential packages needed to run the system.
* **Extra**: Includes additional programs and tools that are not included in the main system.
* **Community**: Contains packages maintained by the Arch Linux community and moved here from the AUR (Arch User Repository) by vote.
* **Multilib**: Designed to support 32-bit programs on 64-bit systems.
* **Testing**: Used to test new packages before moving them to other repositories.

In addition to the official repositories, there is the Arch User Repository (**AUR**), a community repository containing thousands of **PKGBUILD** scripts created by users. These scripts allow you to automate the process of building a package from source code. AUR is an extremely valuable resource that allows users to get the latest versions of programs or those that are not included in official repositories.

- Надайте короткий огляд існуючих менеджерів пакетів у Linux. Охарактеризуйте їх основні можливості.

1. **APT (Advanced Package Tool)**

* **Distributions**: Debian, Ubuntu and derivatives.
* **Features**: APT allows you to easily install, update, and uninstall packages using commands from the terminal. It automatically resolves package dependencies, which greatly simplifies the software management process. APT uses .deb packages.

1. **DNF (Dandified YUM)**

* **Distributions**: Fedora, RHEL, CentOS and derivatives.
* **Features**: DNF is the successor to YUM and offers improved dependency resolution, metadata caching, and functionality extensions through plugins. It uses .rpm packages.

1. **Pacman.**

* **Distributions**: Arch Linux, Manjaro, and other Arch derivatives.
* **Features**: Pacman is famous for its speed and efficiency. It performs the functions of installing, updating, and uninstalling packages, and provides easy dependency management. It uses the proprietary Arch package format.

1. **Zypper package format.**

* **Distributions:** openSUSE, SUSE Linux Enterprise.
* **Features:** Zypper provides powerful package management capabilities, including installing, updating, searching, and uninstalling packages. It supports a wide range of options for managing repositories and dependencies.

1. **Portage**

* **Distributions**: Gentoo.
* **Features**: Portage is more than just a package manager; it is a complete software management system that allows users to build applications from source code, customizing them for specific needs. It uses ebuild scripts to automate compilation and installation.

1. **RPM (Red Hat Package Manager)**

* **Distributions:** Fedora, RHEL, CentOS, and others that use .rpm packages.
* **Features:** RPM is a low-level package manager on its own, allowing you to install, upgrade, and remove packages. However, it is often used in conjunction with tools such as YUM or DNF to resolve dependencies.

**2. Визначте який менеджер пакетів використовує ваш дистрибутив Linux. Опишіть основні команди для роботи з ним:**

**- Пошук, скачування та установка необхідних пакетів, яких у Вашій системі немає (зі сховища по замовчуванню, з нового репозиторію тощо).**

**Пошук пакетів:** To find a package in the repositories, use the command: *pacman -Ss package\_name*

**Установка пакетів:** To install a package (or packages), use: *pacman -S package\_name*

**- Перегляд інформації про встановлені та доступні пакети.**

To view **details about the installed package:** *pacman -Qi package\_name*

If you want a list of **all installed packages**: *pacman -Q*

- Видалення непотрібних або застарілих пакетів.

**To remove a package** along with its dependencies that are no longer used by other packages: *pacman -Rs package\_name*

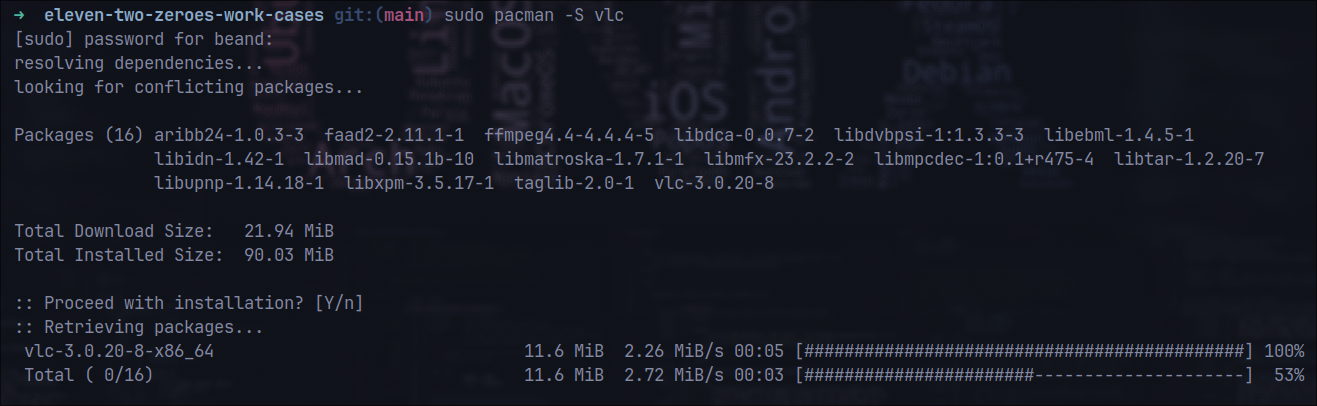
**- Оновлення менеджера пакетів.**

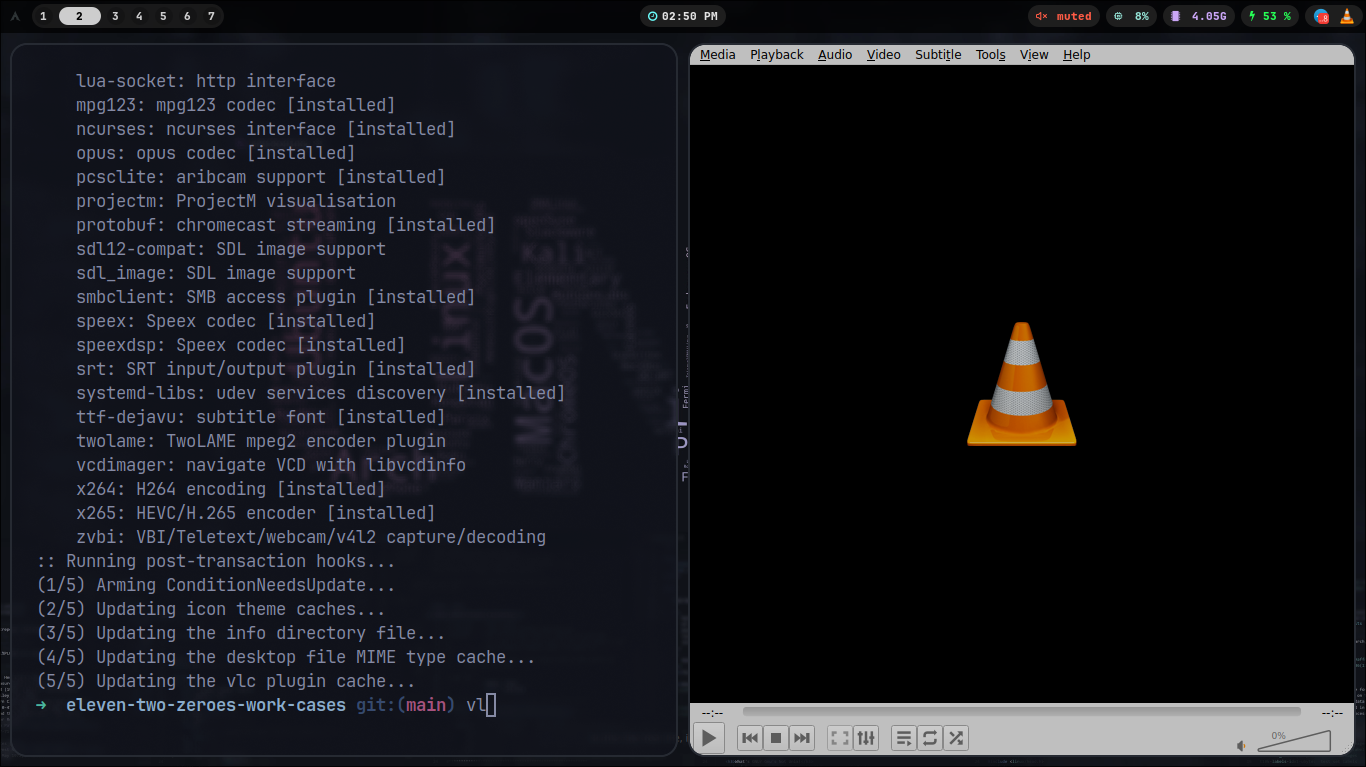
**To update** all installed packages to the latest versions: *pacman -Syu*

**3. Встановіть у терміналі через менеджер пакетів на свою систему:**

Я використовую менеджер пакетiв pacman.

**- Новий відео- чи аудіоплейер.**

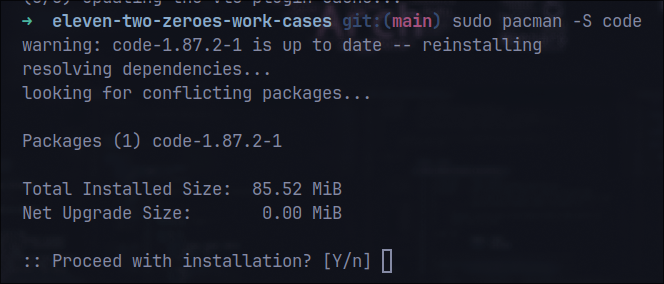
****

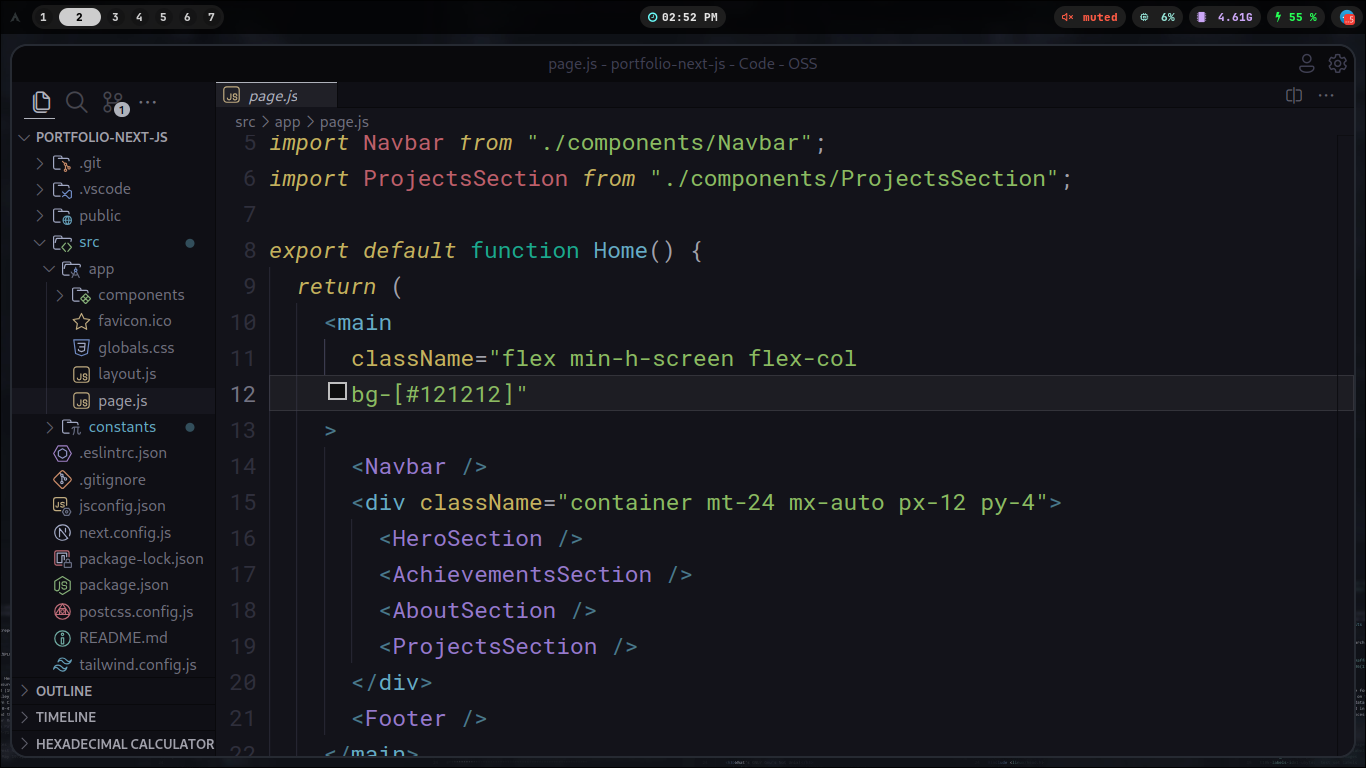
****

It’s works!

**- Середовище для мови програмування, що ви вивчаєте.**

Я обрав vscode:

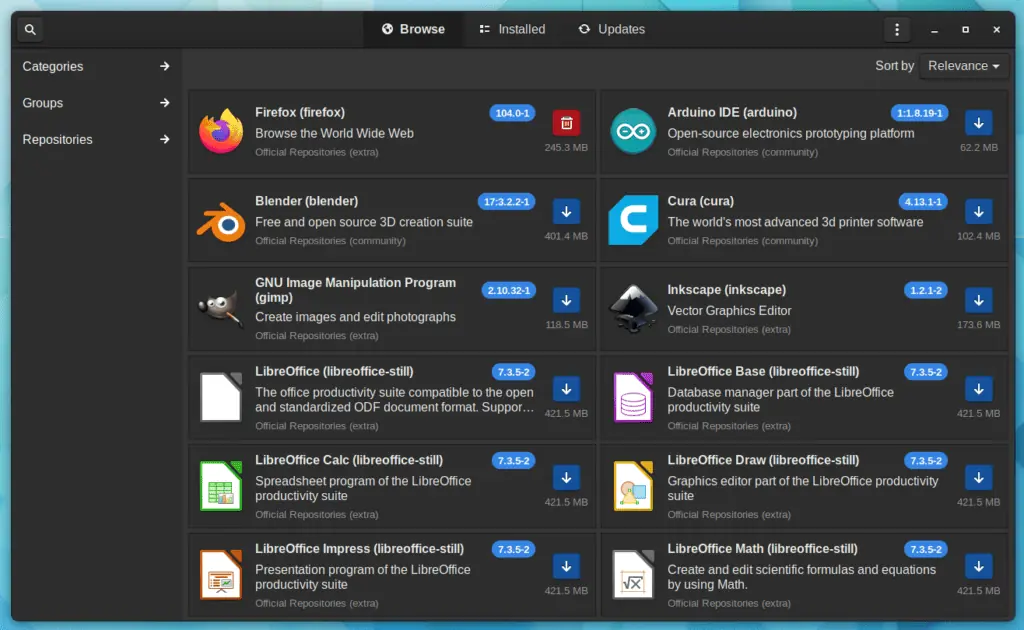




*It’s works!*

**4. Яким чином можна встановити нові програми через магазини додатків та менеджери пакетів у графічному середовищі. Наведіть свої приклади.**

I don't use any frontend for pacman and do everything through the terminal, so I won't be able to show you how to do it.

There is Pamac - Pamac is a GTK3 graphical frontend for Pacman based on libalpm (a package management library) built by Guillaume Benoit, a software engineer, and Manjaro’s dev team member. It allows users to search for, install, update, and remove applications from their computers using simple steps.

Here you don’t even have to think about how to install anything, you just click on the buttons and it does everything for you :D