WORK-CASE №7

Виконали студенти групи РПЗ-03:
Кошіль Владислав та Фещенко Эвгеній

1. В ході роботи досить часто виникає завдання планування задач:

а) **Охарактеризуйте основні функції які може виконувати планувальник завдань в будь-якій ОС. Порівняйте можливості планування завдань в різних ОС на прикладі Windows та Linux.**
ГОТУВАВ СТУДЕНТ ФЕЩЕНКО ЭВГЕНІЙ

A task scheduler (also known as a task scheduler or job scheduler) is software that allows you to schedule and automate tasks on your computer according to a specific schedule or timetable. The main functions that a task scheduler can perform in any OS include:

- Running programs or scripts according to a schedule or timetable.
- Execution of tasks at a certain time or after certain events (for example, starting a program after the previous one has finished).
- Automatic execution of tasks with administrator (or other user) rights at a specific time.
- Monitor task execution and notify you of errors or failures.

б) **Опишіть основні принципи роботи з планувальником Cron в ОС Linux. Як його налаштовувати? Чи є йому альтернативи (дайте їх характеристику).**
ГОТУВАВ СТУДЕНТ КОШІЛЬ ВЛАДИСЛАВ

Cron is a Linux task scheduler that allows you to automatically execute commands and scripts according to a specified schedule. The basic principles of working with the Cron scheduler in Linux are as follows:

Cron works with a crontab file that contains a list of tasks and their parameters. Each user has his own crontab file.

Tasks in the crontab are specified using a special syntax consisting of five fields:

- Minutes (0-59)
- Hours (0-23)
- Days of the month (1-31)
- Months (1-12 or abbreviated names)
- Days of the week (0-7 or abbreviated names, 0 and 7 are Sunday)

Tasks can be launched at a specific time or with a specific frequency. For example, a command to be executed every hour can be specified as "0 * * * *".

Task parameters can include commands to execute, environment variables, repeat intervals, and other parameters.

To set up Cron, you need to open the crontab file for the current user using the "crontab -e" command. Then you can add or edit task lines according to the Cron syntax. After saving

the crontab file, Cron automatically supports scheduling tasks according to the specified schedule.

Alternatives to Cron are the following schedulers:

Anacron is a Linux task scheduler that allows you to run tasks based on the time relative to the last execution, rather than the current time. Anacron is useful for systems that do not always run 24/7, such as laptops or servers.

2. Для вашої віртуальної машини зі встановленою ОС Linux здійсніть планування обраних вами задач (запуск додатків, вмикання/вимикання машини, очистка каталогів, видалення файлів, резервне копіювання, архівування тощо на ваш вибір) через планувальник Cron:
ГОТУВАВ СТУДЕНТ КОШІЛЬ ВЛАДИСЛАВ

а) **First, let's download Cron itself**

```
quadfordt@Ubuntu:~$ sudo apt-get install cron
[sudo] password for quadfordt:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
cron is already the newest version (3.0pl1-137ubuntu3).
cron set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 113 not upgraded.
```

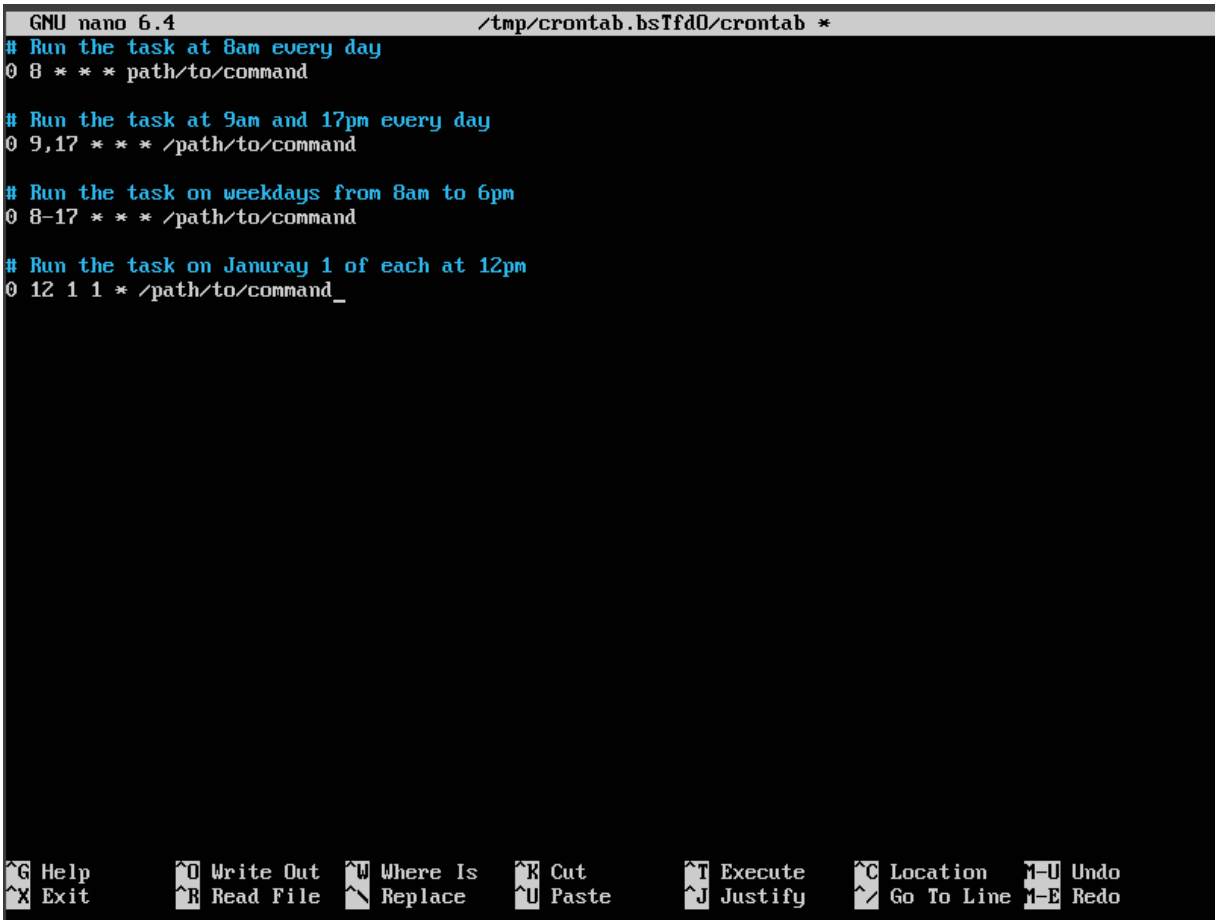
б) **Then open the configuration file**

```
quadfordt@Ubuntu:~$ sudo crontab -e
no crontab for root - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.tiny
  3. /bin/ed

Choose 1-3 [1]: _
```

в) **Now we plan the tasks and launch them.**



3. Встановіть альтернативний Cron'у планувальник задач (на Ваш вибір). Виконані у завданні 2 дії продемонструйте через нього.

ГОТУВАВ СТУДЕНТ ФЕЩЕНКО ЭВГЕНІЙ

systemd-timer

It is a built-in task scheduler in the Linux operating system. It can be used to schedule system tasks and build projects. It is based on systemd, so you may need some knowledge of this framework.

Now you can create your own systemd-timer. For example, to create a timer that runs every hour and displays a message in the terminal, enter the following command:

```
sudo systemctl edit --force --full mytimer.timer
```

A text editor opens where you can enter the configuration for your timer. Below is a sample configuration for a timer:

Now you need to create a service that will be executed by the timer. For example, to create a service that will display a message in the terminal, enter the following command:

```
sudo systemctl edit --force --full mytask.service
```

A text editor opens where you can enter the configuration for your service. Below is a sample configuration for a service:

Now restart systemd so that it reads your configuration files:

```
sudo systemctl daemon-reload
```

Done