“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

# **WORK-CASE №7**

з дисципліни: «Операційні системи»

**Тема: «Планування задач в Linux ОС»**

Виконали студенти

групи КСМ-13а

Команда BBC:

Петрик С.С,

Панчук О.С.

Перевірив викладач

Сушанова В.С.

Київ 2023

**Завдання №1  
Виконав Петрик С.С**

**1.1**

Task schedulers in any operating system (OS) perform several basic functions:

Run programs and scripts at a specific time: The ability to automatically trigger the execution of programs or scripts on a schedule or at a specific time.

Regular execution of tasks: The ability to configure tasks to run repeatedly on a specific schedule (daily, weekly, every hour, etc.).

Execution conditions: Configure the conditions under which a task should run (for example, only at certain times of the day or on certain events).

Execution according to the calendar: Schedule tasks for specific dates or periods (annually, monthly, etc.).

Comparing Windows and Linux task schedulers:

|  |  |
| --- | --- |
| Windows Task Scheduler | Cron on Linux |
| Graphical interface: Has an intuitive graphical interface that allows users to create and customize tasks without using the command line. | Command line: Does not have a graphical interface, but uses a text file to edit the task schedule, which allows for greater flexibility in customization. |
| Triggers and events: The ability to configure triggers (events that cause a task to run) and responses to certain events in the system. | Availability of conditional statements: The ability to use more complex conditions to run tasks (for example, conditions based on the presence of a file or system state). |
| Task repetition: Easily configure tasks to run repeatedly on different schedules. | System control: Provides more control over the system through extensive configuration options via a text file. |

In general, Windows Task Scheduler has a user-friendly graphical interface and an easy way to configure. While Cron on Linux provides more flexibility through the command line and allows you to create more complex conditions for running tasks.

1.2

The Cron scheduler in Linux is a powerful tool for automating tasks. The basic principles of working with Cron:

Crontab file: The Cron job schedule is stored in a crontab file. Each user can have their own crontab file that contains a list of tasks to be executed.

Schedule syntax: A Cron task schedule is defined using five fields: minutes, hours, days of the month, months, and days of the week. Each field accepts a value or a range of values (for example, 0-59 for minutes or \* for all possible values).

Running commands: Cron runs commands or scripts at specified times on a specified schedule. This can be done with the crontab -e command, which opens the crontab file in a text editor for editing.

Setting up Cron:

To customize Cron, you need to edit the crontab file. This can be done with the crontab -e command, where you can add new lines with the schedule of tasks.

Alternatives to Cron in Linux:

|  |
| --- |
| systemd Timers: This is an alternative mechanism for scheduling tasks on a system that is based on systemd, the main initialization system in many modern Linux distributions. It provides greater integration with the system and the ability to create timers directly. |
| At: This is another standard task scheduler that works on the basis of timestamps. It is easier to use for one-time tasks, but less flexible compared to Cron. |
| Anacron: This is a tool for running recurring tasks that differs from Cron in that it can run tasks even if the system is off at the specified start time. It is used for systems that are not always on. |

**Завдання №2**

**виконав Панчук О.С.**

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:  
- Виконання спланованої задачі в чітко визначений Вами час (наприклад о 8 ранку, 18.30 і т.д.).

* Open the Cron scheduler:

In the terminal, enter the command: crontab –е

* Add a task:

Add a line for each task you want to perform at a specific time. For example:

"0 8 \* \* \* /way\_to\_your\_team/your\_team"

Here "0 8 \* \* \*" means "at 8 o'clock every day". Replace /path\_to\_your\_command/your\_command with the path and the command you want to execute.

Save and exit:

* Save the file and exit the editor.

2.1 Performing the same task twice a day (you also determine the time yourself).

To schedule the same task to run twice a day at 8:00 a.m. and 5:00 p.m.

Open the crontab editor:

crontab –e

Add the following line to execute the task at 8:00:

0 8 \* \* \* /way\_to\_your\_team/your\_team

Add the following line to execute the task at 17:00:

0 17 \* \* \* /way\_to\_your\_team/your\_team

Save and close the editor.

2.2 Виконання однієї й тієї ж задачі тільки в будні (або тільки у вихідні дні) у чітко визначений проміжок часу (наприклад з 8 до 18 години).

For execution only on weekdays (Monday through Friday) from 8:00 to 18:00:

0 8-18 \* \* 1-5 /way\_to\_your\_team/your\_team

For execution only on weekends (Saturday and Sunday) from 8:00 to 18:00:

0 8-18 \* \* 6,7 /way\_to\_your\_team/your\_team

Replace /path\_to\_your\_command/your\_command with your path and command. Thus, the specified commands will be executed only on weekdays or only on weekends

- Виконання задач тільки раз у рік, раз у місяць, раз у день, щогодини, при вмиканні (після перезавантаження).

Only once a year (for example, on January 1):

0 0 1 1 \* /way\_to\_your\_team/your\_team

Only once a month (for example, on the first day of each month):

0 0 1 \* \* /way\_to\_your\_team/your\_team

**Only once a day (for example, at 8:00 am):**

0 8 \* \* \* /way\_to\_your\_team/your\_team

Every hour (for example, at every full hour):

0 \* \* \* \* /way\_to\_your\_team/your\_team

At power on (after reboot):

@reboot /path\_to\_your\_command/your\_command

**Завдання №3**

**виконав Панчук О.С.**

Install and configure systemd:

Many Linux distributions already have systemd installed by default. You can check its presence by entering the command:

systemctl --version

Create a service to perform the task:

Create a service file (for example, mytask.service) in the /etc/systemd/system/ or /etc/systemd/user/ directory. Enter the command:

sudo nano /etc/systemd/system/mytask.service

Update the systemd configuration:

Update the systemd configuration by entering the commands:

sudo systemctl daemon-reload

Launch the service and start it up:

Start the service and configure autorun by entering the commands:

sudo systemctl start mytask.service

sudo systemctl enable mytask.service

Висновки:

We got acquainted with schedulers in different operating systems including Cron in Linux OS, systemd Timers, At, Anacron. We also learned how to customize them.