“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.**

**Завдання №3**

**виконав Панчук О.С.**

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.