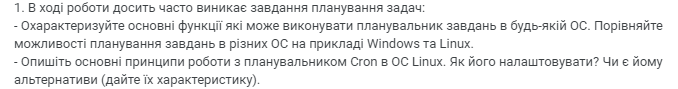
**Wokrcase № 7**

**Виконав Когут Богдан  
**

Task Scheduling in Operating Systems

**Key Functions of a Task Scheduler:**

**Task Automation:**

Allows tasks to be executed at a specific time or periodically, eliminating the need for manual initiation.

**Resource Management:**

Allocates system resources for scheduled tasks with minimal impact on current performance.

**Monitoring and Logging:**

Records task execution details for error analysis or confirmation of successful completion.

**Priority Management:**

Assigns different priority levels to tasks.

**Flexibility and Configuration:**

Enables customization of time, frequency, and conditions for task execution.

**Comparison of Task Scheduling in Windows and Linux:**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Windows Task Scheduler** | **Linux (Cron)** |
| **Interface** | Graphical and CLI | CLI (via crontab file) |
| **Configuration Flexibility** | Advanced settings via GUI/CLI | Simple configuration format |
| **Logging** | Built-in logs (Event Viewer) | Logs accessible via syslog |
| **Task Complexity** | Supports complex scenarios and triggers | Mostly time-based scheduling |
| **Automation** | |  | | --- | | Supports launching programs, scripts, etc. |  |  | | --- | |  | | Executes scripts and commands |
| **Priority Management** | Allows setting task priorities | No direct priority support |
| **Additional Features** | Integration with Windows API | Simplicity and ease of use |
| **Alternatives** | PowerShell, third-party tools | systemd-timers, at, anacron |

**Principles of Cron in Linux**

**How Cron Works:**

**1 Cron Daemon:** A background service that runs tasks based on a defined schedule.

**2 Crontab:** A configuration file where task schedules are defined.

**3 Schedule Format:**

\* \* \* \* \* command

**First** \* **means** **minute** (0–59): Specifies the minute when the task will run. For example, 30 means the task will run at the 30th minute of the hour.

**Second** \* **means** **hour** (0–23): Specifies the hour when the task will run. For example, 5 means the task will run at 5 AM.

**Third** \* **means** **day of the Month** (1–31): Specifies the day of the month when the task will run. For example, 15 means the task will run on the 15th day of the month.

**Fourth** \* **means** **month** (1–12): Specifies the month when the task will run. For example, 7 means the task will run in July.

**Fifth** \* **means** **day of the Week** (0–6): Specifies the day of the week when the task will run. For example, 0 represents Sunday, and 6 represents Saturday.

**Setting Up Cron:**

1. Open the crontab file for editing:

crontab -e

1. Add a task in the appropriate format.
2. Check the list of scheduled taskscrontab -l
3. Task execution logs are stored in syslog or a specified log file.

**Alternatives to Cron and Their Characteristics:**

1. **Anacron:**

Designed for periodic tasks on systems that may not be continuously powered on.

Does not require the system to be running constantly.

Less flexible compared to Cron.

1. **Systemd Timers:**

Part of the Systemd initialization system.

More versatile for event-based scheduling (e.g., task dependencies, system state triggers).

Supports logging and is easy to monitor.

1. **At:**

Used for one-time task execution at a specified time.

Simple usage:

echo "command" | at time

1. **Fcron:**

A hybrid of Cron and Anacron.

Can execute tasks either at a fixed time or after the system starts.

**Виконав Трощинський Ярослав**

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:

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

- Виконання однієї й тієї ж задачі двічі в день (час також визначаєте самостійно).

- 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).

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

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

2. To create a crontab schedule you need to use command *“crontab -e”* after this need to enter a schedule and command in this type:

*Min Hour Day Month Week-day Command*

For example

*\* \* \* \* \* /home/user/greet.sh*

Will do a script greet.sh every minute.

To do something at 8 o'clock you need to type in

*0 8* *\* \* \* command*

To do something twice a day, for example command at 6 and 7 o'clock

*0 6,7 \* \* \* command*

To do something only in working days at 12 o'clock

*0 12 \* \* 1-5 command*

To do something once a year, month, day, every hour, every boot, we need to type in

Run once a year at midnight of 1 January: *@yearly* *command*

Run once a month at midnight of the first day of the month: *@monthly command*

Run once a day at midnight: @daily command

Run once an hour at the beginning of the hour: @hourly command

Run at startup: *@reboot command*

3. For an alternative of cron I chose anacron, to use in we need to do this command

*sudo nano /etc/anacrontab*

Syntax and usage of this planner are a lot different from cron, here we have

*Days Minutes\_from\_boot Name\_of\_job command*

For example:

1 5 Daily /home/user/daily.sh

Will do a script daily.sh every day, 5 mins after the boot

**Виконав Михайленко Олексій**

**Conclusion:**

The workcase demonstrates the basic principles of working with task schedulers in operating systems, comparing the functionalities of Windows Task Scheduler and Linux Cron. The main focus is on task automation, resource management, configuration flexibility, monitoring, and the use of Cron alternatives such as Anacron, systemd-timers, and at. Practical tasks include scheduling activities via Cron, executing them at specific times or under certain conditions, and showcasing alternative approaches. This case highlights the importance of efficient task management for optimal resource utilization and ensuring system continuity.