«Київський фаховий коледж зв’язку»

Циклова комісія комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**СРС\_WORK-CASE №5**

з дисципліни: «Операційні системи»

Виконали студенти

групи РПЗ-23Б

Команда Booblik-Team:

Михальов В.,

Мірошніченко А.,

Білоус М.

Перевірила викладач

Сушанова В.С.

Київ 2025

1. (Білоус) Describe the main functions that a task scheduler can perform in any OS. Compare the capabilities of task scheduling in different OSes using Windows and Linux as an example.

A task scheduler is a system component of an OS that automatically executes certain commands or programs at a scheduled time or under certain conditions.

**Main functions of the planner:**

* Automatic task execution

Launch tasks at a specified time (once or periodically).

* Support for various types of triggers

Ability to run on a schedule, on events (e.g., system startup, user login), on a condition (e.g., system inactivity).

* Task priority management

Defining which tasks have priority when executed.

* Resource management

Limiting the use of CPU, memory, and other resources for running tasks.

* Logging and notifications

Maintaining execution logs, reporting errors or successful completion.

* Re-execution and error handling

Ability to re-run tasks in case of failure, setting actions for errors.

* Security

Execution of tasks with appropriate access rights, authentication.

**Comparing task scheduling capabilities in Windows and Linux.**

|  |  |  |
| --- | --- | --- |
| **Function** | **Windows** | **Linux** |
| **Management Interface** | Graphical UI, Command Line | Text-based config files , CLI tools, systemd utilities |
| **Trigger Types** | Time-based, system events, logon, idle time | Mainly time-based (cron); systemd supports events and conditions |
| **Scheduling Flexibility** | Highly flexible: daily, weekly, at logon/startup, on idle | Cron: fixed intervals; systemd timers: more flexible |
| **Execution Rights** | Run as specific user/system, supports role-based access | Run as user or root; access managed via crontab or systemd |
| **Logging** | Windows Event Log, per-task logs | Syslog/journal (systemd), separate logs for cron if configured |
| **Error Handling** | Retry on failure, custom error actions | Scripts can handle errors; systemd supports retries and actions |
| **Extensibility** | Multiple triggers/actions per task, complex workflows | Cron: simple; systemd timers: more advanced with dependencies |
| **Conditional Execution** | Conditions based on power state, network availability, etc. | systemd supports conditions; cron does not |

(Білоус) Describe the basic principles of working with the Cron scheduler in Linux. How to configure it? Are there alternatives to it (describe them).

Cron is a system daemon in Linux that allows users to schedule commands or scripts to run automatically at specified times or intervals.

**Basic principles of work:**

* Crontab files are tables that store rules (schedules) for executing tasks.
* Types of crontab:

User (crontab -e) — for a specific user.

System (/etc/crontab, /etc/cron.d/) — for system processes.

* View and edit:

crontab -l — view jobs.

crontab -e — edit.

crontab -r — delete.

**Cron Alternatives:**

* Anacron

Runs tasks with a delay if the PC was turned off at the scheduled startup time.

Suitable for laptops or systems that do not run 24/7.

Less flexible, but reliable for daily/weekly tasks.

* systemd timers

A modern alternative for systems using systemd.

Supports:

Flexible scheduling (times, intervals, event-based).

Dependencies between services.

System states (battery, network).

Configurable via .timer and .service files.

* at

Runs tasks once at a specified time.

Example: echo "backup.sh" | at 03:00

1. (Михальов) For your virtual machine with the Linux OS installed, schedule the tasks of your choice (launching applications, turning the machine on/off, cleaning directories, deleting files, backing up, archiving, etc. of your choice) via the Cron scheduler:

- Performing a scheduled task at a time clearly defined by you (for example, at 8 am, 6:30 pm, etc.).

Task: clear the temporary folder /tmp/mytemp/



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

Task: create a text file with the current time in /home/user/logs/timestamp.txt



- Performing the same task only on weekdays (or only on weekends) in a clearly defined time period (for example, from 8 am to 6 pm).

Task: clear the temporary folder /tmp/mytemp/ every hour on weekdays



- Performing tasks only once a year, once a month, once a day, every hour, when turned on (after reboot).

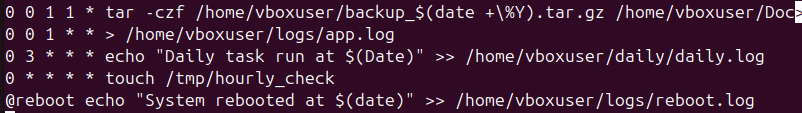
Task: create an archive of the /home/user/Documents folder in /home/user/backups/backup\_YEAR.tar.gz

Task: clear the log file /home/user/logs/app.log

Task: create a text file with the date in the /home/user/daily/ folder

Task: create an empty file /tmp/hourly\_check

Task: log that the system has started



1. (Мірошніченко) Install an alternative task scheduler to Cron (of your choice). Demonstrate the actions performed in task 2 through it.

Аlternative task scheduler **systemd timers**

1. Clean up the /tmp/mytemp/ directory every day at 8:00 AM
   * Create the systemd service sudo nano /etc/systemd/system/clean-temp.service
   * Create a timer

* Activate the timer

1. Recording time twice a day (9:00 and 21:00) to the file /home/user/logs/timestamp.txt

* Create the script /home/user/scripts/log\_time.sh
* Create the service /etc/systemd/system/log-time.service
* Create a timer
* Activate the timer

Висновок: In this work, we worked with the Cron scheduler, configuring the execution of various tasks at a given time. We also got acquainted with the alternative to Cron — systemd timers