**Work-Case №7**

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

*Готували матеріал студенти Усенко С. і Титов О.*

*- Охарактеризуйте основні функції які може виконувати планувальник*

*завдань в будь-якій ОС. Порівняйте можливості планування завдань в*

*різних ОС на прикладі Windows та Linux.*

The Task Scheduler is a tool that allows predefined actions to be automatically executed whenever a certain set of conditions is met. For example, you can schedule a task to run a backup script every night, or send you an e-mail whenever a certain system event occurs.

On both Windows and Linux, task schedulers has somewhat similar capabilities, although Cron (linux scheduler) is more flexible but lacks graphical interface.

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

1) Use the crontab -e command to open your user account’s crontab file. Commands in this file run with your user account’s permissions. If you want a command to run with system permissions, use the sudo crontab -e command to open the root account’s crontab file. Use the su -c “crontab -e” command instead if your Linux distribution doesn’t use sudo.

Use the arrow keys or the page down key to scroll to the bottom of the crontab file in Nano. The lines starting with # are comment lines, which means that cron ignores them. Comments just provide information to people editing the file.

Lines in the crontab file are written in the following sequence, with the following acceptable values:

minute(0-59) hour(0-23) day(1-31) month(1-12) weekday(0-6) command

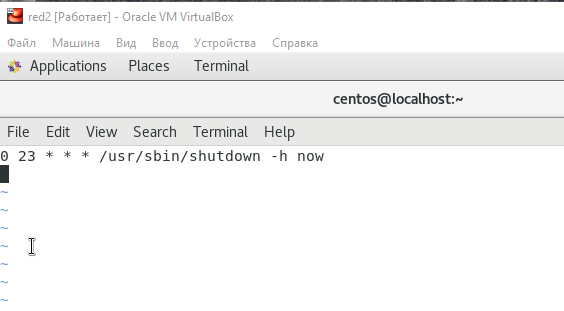
You can use an asterisk (\*) character to match any value. For example, using a asterisk for the month would cause the command to run every month.

2) Yes, there are many different alternatives to Cron in Linux like Anacron, Cronie, fcron, bcron and others. They have some additional features, like running task at any time, work with task on daily and minute basis and others, run jobs one after another and more.

***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:***

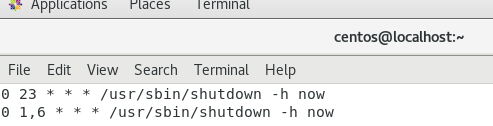
*Готував матеріал студент Нестолій Н.*

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

First of all, you need to launch terminal and enter **crontab -e** command. After that you can use Cron to do any operation you want at any time. For example, I want to shutdown my PC(Virtual machine) at 23:00 every day, so I enter **0 23 \* \* \* /usr/sbin/shutdown -h now**

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

Procces is same, but you should write time with a comma - **0 1,6 \* \* \* /usr/sbin/shutdown -h now**



* *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).*

To specify a command that runs only on weekdays, we set the Day-of-week to 1-5 with no spaces. To specify a time, use same algorithm. This will look like that:



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

1. Once per year (1 month 1 day 00:00):



1. Once per month:



1. Once per day:

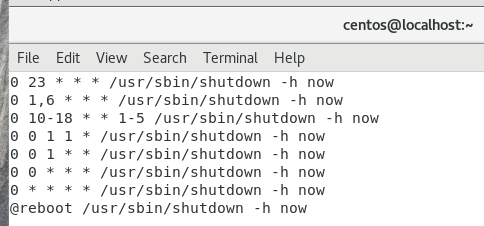


1. Every hour:



1. On reboot:



So, your Cron scheduler will look like that:   


(I used **/usr/sbin/shutdown -h now** as an example command to shutdown PC, don’t even try to use is like that!**)**

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

I have used fcron scheduler – more modern version of Cron, but its look is similar to the cron – it even uses same syntaxis:  
**minute hour day-of-month month day of week command-to-be-executed**

So I will not repeat same steps as in second task – because it will look similar. Thought, I will write about some additional features of **fcron**:

- run jobs one after the other (not in parallel),

- set the max system load average value under which the job should be run

- run jobs at fcron's startup if they should have been run during system down time

- mail user to tell him a job has not run and why

- a better management of the mailing of outputs