“Київський фаховий коледж зв’язку”
Циклова комісія Комп’ютерної інженерії

ЗВІТ ПО ВИКОНАННЮ ЛАБОРАТОРНОЇ РОБОТИ №4

з дисципліни: «Операційні системи»

Тема: «Команди Linux для Управління процесами»

Виконала студентка
групи РПЗ-13б
Дімітрова С.П.
Перевірив викладач
Сушанова В.С.

Київ 2024

Робота студентки групи РПЗ-13б Дімітрової Софії

Мета роботи:

1. Отримання практичних навиків роботи з командною оболонкою Bash.
2. Знайомство з базовими командами для управління процесами.

Матеріальне забезпечення занять

1. ЕОМ типу IBM PC.
2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).
3. ОС GNU/Linux (будь-який дистрибутив).
4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

Завдання для попередньої підготовки.

1. *Прочитайте короткі теоретичні відомості до лабораторної роботи та зробіть невеличкий словник базових англійських термінів з питань призначення команд та їх параметрів.

Термін англійською	Термін українською
Process ID (PID)	Ідентифікатор процесу (PID)
PPID (Parent Process ID)	PID батьківського процесу (якщо процес запущено іншим процесом)
Runaway process	Запущений процес програми
Real-time	Інформація, яка оновлюється постійно.
Wildcard character	Символ шаблон (*)
Signal	Повідомлення, яке надсилається процесу.
Swap space	Дисковий простір, який використовується як розширення пам'яті.
Zombie process	Процес, що завершив виконання, але все ще має запис в таблиці процесів.

2. На базі розглянутого матеріалу дайте відповіді на наступні питання:
   2.1. *Які команди для моніторингу стану процесів ви знаєте. Як переглянути їх можливі параметри?
   - *ps:* Displays information about active processes in the system. Each process has an ID associated with it called the process ID (PID). This PID is assigned in the order that processes are created. To view the possible options, use the man ps or ps --help command.
   - *top:* The top command displays information about running processes in real time. You can use the built-in help to find out about the possible options for

the top command. First, type the top command in the terminal and then press the h key. You can also use the man top command.

2.2. *Чи може команда ps у реальному часі відслідковувати стан процесів?

The ps command can display information only for a specific point in time. Instead, the top command can solve this problem. The top command displays process information similarly to the ps command, but it does it in real-time mode.

2.3. **За якими параметрами можливе сортування процесів в команді top? Як переключатись між ними?

By default, when you start top, it sorts the processes based on the %CPU value. You can change the sort order by using one of several interactive commands while top is running. Each interactive command is a single character that you can press while top is running and changes the behavior of the program. Pressing f allows you to select the field to use to sort the output, and pressing d allows you to change the polling interval. Press q to exit the top display.

To switch between the different sorting options in the top, we use the following keys:

- Press Shift + P to sort by %CPU.
- Press Shift + M to sort by %MEM.
- Press Shift + N to sort by PID.
- Press Shift + T to sort by TIME+.

2.4. **Які команди для завершення роботи процесів ви знаєте?

- kill: Sends signals to processes by their Process ID (PID). To send a process signal, you must either be the owner of the process or be logged in as the root user.
- killall: The killall command is used if you want to kill the controlling (parent) process and all children. This command allows you to use wildcards, which makes it a very useful tool when you have a system that is down.

Хід роботи.

1. Початкова робота в CLI-режимі в Linux ОС сімейства Linux:

1.1. Запустіть віртуальну машину VirtualBox, оберіть CentOS та запустіть її. Виконайте вхід в систему під користувачем: CentOS, пароль для входу: reverse ***(якщо виконуєте ЛР у 401 ауд.)*** та запустіть термінал.

1.2. Запустіть віртуальну машину Ubuntu_PC ***(якщо виконуєте завдання ЛР через академію netacad)***

1.3. Запустіть свою операційну систему сімейства Linux ***(якщо працюєте на власному ПК та її встановили)*** та запустіть термінал.

2. Дайте відповіді на наступні питання:

- Як вивести вміст директорії /proc? Де вона знаходиться та для чого призначена? Охарактеризуйте інформацію про її вміст?

The contents of the /proc directory can be displayed with the *ls /proc* command:

```
sysadmin@localhost:~$ ls /proc
1           consoles       irq          mounts        sysrq-trigger
10          cpuinfo        kallsyms     mpt           sysvipc
14          crypto         kcore        mtrr          thread-self
16          devices        key-users    net           timer_list
26          diskstats      keys         pagetypeinfo  tty
47          dma            kmsg         partitions    uptime
62          driver         kpagecgroup  pressure      version
63          dynamic_debug  kpagecount   schedstat     version_signature
7           execdomains    kpageflags   scsi          vmallocinfo
acpi        fb             loadavg      self          vmstat
bootconfig  filesystems    locks        slabinfo      zoneinfo
buddyinfo   fs             mdstat       softirqs
bus         interrupts     meminfo      stat
cgroups     iomem          misc         swaps
cmdline     ioports        modules      sys
```

It’s called a virtual filesystem because it lives inside of the RAM and consumes no physical storage. If the system is shut down or restarted, the RAM is cleared and files of the /proc directory are generated from scratch. Contains information about processes (as the name "proc" implies), also provides information about the system hardware and the current kernel configuration.

- /proc/cmdline - Contains information passed to kernel during boot
- /proc/ meminfo - Contains information about kernel memory usage
- /proc/modules - Contains list of modules loaded into the kernel

The information contained in /proc is dynamic and updated in real time. This means that it changes when the system state changes.

- Як вивести інформацію про поточні сеанси користувачів. Якою командою це можна зробити?

The *who* command displays a list of users who are currently logged into the system, where they are logged in from and when they logged in.

```
sysadmin@localhost:~$ who
sysadmin pts/0        Mar  1 18:28
sysadmin@localhost:~$
```

- Які дії можна зробити в терміналі за допомогою комбінацій Ctrl + C, Ctrl + D та Ctrl + Z?

- *Ctrl + C,* used to force the termination of a running process. When you press Ctrl + C, the terminal sends an interrupt signal (SIGINT) to the process, which causes it to stop immediately. This is a quick way to exit a process or command.
- Pressing *Ctrl + D* tells the terminal to register the so-called EOF (end of file), i.e. the input stream is over. Bash interprets this as a desire to exit the program.
- *Ctrl + Z* is used to suspend a running process in the terminal. This means that you temporarily pause the execution of a program and move it to the background, allowing you to continue using the terminal. When you press CTRL + Z, your terminal registers a "suspend" command, which then sends the SIGTSTP signal to the foreground process. It's useful when you want to halt a process without terminating it.

- *Чим відрізняється фоновий процес від звичайного. Де вони використовуються?
  - *Normal process (Foreground process)* - initialized and managed using a terminal session. When a command/process is running in the foreground, it completely occupies the terminal that started it. Users cannot use other commands because the command prompt will not be available while this process is running in the foreground. They are usually used for user interaction with the system or to run user programs.

  - *Background processes* are processes that are not connected to the terminal; they do not expect user input. Thus, other processes can run in parallel with the process running in the background because they do not have to wait for it to complete. Used for automation, system services, and daemons.

- *Опишіть наступні команди та поясніть що вони виконують – команда jobs, bg, fg.
  - The *jobs* command shows a list of tasks that are currently running or paused in the background. Each task has its own unique number that identifies it in the list.

  - The *bg* command is used to resume a background process. It can be used with or without a job number. If you use it without a job number the default job is brought to the foreground. The process still runs in the background. You cannot send any input to it.

  - The *fg* command will bring a background task into the foreground. Just like the bg command, it can be used with or without a job number. Using

it with a job number means it will operate on a specific job. If it is used without a job number the last command that was sent to the background is used.

- **Якою командою можна переглянути інформацію про запущені в системи фонові процеси та задачі?

To view information about background processes and tasks running in the system, you can use the *jobs* command in the terminal. This command displays a list of all jobs that are currently running in the background or suspended. It provides the job ID, status, and command for each background process or task, allowing you to manage them effectively.

- **Як призупинити фоновий процес, як його потім відновити та при необхідності перезапусти?

To pause a background process, use the keyboard shortcut Ctrl + Z. This will stop the process and put it into a standby state. To resume a paused background process, use the fg command followed by the % sign and the job number. For example, to resume the last paused background process, use the fg %1 command. To restart a background process, you can pause it with the keyboard shortcut Ctrl + Z and then resume it by entering the command 'bg'.

3. Запустіть термінал, та в командному рядку виконайте наступні дії для ознайомлення з роботою з процесами:
   - запустіть команду top, проаналізуйте отриманий в цій команді результат та охарактеризуйте найбільш активні процеси у системі;

```
top - 17:54:23 up 16 days,  2:13,  1 user,  load average: 0.08, 0.20, 0.23
Tasks:   8 total,   1 running,   7 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.7 us,  0.4 sy,  0.0 ni, 98.8 id,  0.0 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem : 65841716 total, 39765688 free, 11515248 used, 14560780 buff/cache
KiB Swap:  8388604 total,  8388604 free,        0 used. 53560276 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0    4388    772    708 S   0.0  0.0   0:00.03 init
    7 root      20   0   78644   3808   3256 S   0.0  0.0   0:00.00 login
   10 syslog    20   0  191336   3736   3232 S   0.0  0.0   0:00.23 rsyslogd
   14 root      20   0   28368   2696   2432 S   0.0  0.0   0:00.00 cron
   16 root      20   0   72312   3360   2616 S   0.0  0.0   0:00.00 sshd
   26 bind      20   0  217060  18300   7152 S   0.0  0.0   0:00.03 named
   47 sysadmin  20   0   19228   4116   2968 S   0.0  0.0   0:00.01 bash
   60 sysadmin  20   0   38716   3300   2852 R   0.0  0.0   0:00.00 top
```

The most active processes can be identified by the amount of CPU time used (%CPU). By default, top sorts processes by this parameter in descending order, which allows you to quickly identify the most active processes. All processes show 0.0% CPU usage and 0.0% MEM. However, if we look at physical memory usage (RES), the process identified as 'named' uses the most physical memory.

- призупинити виконання команди top

треба використати комбінацію клавіш CTRL + Z

```
[1]+  Stopped                 top
sysadmin@localhost:~$
```

- вивести інформацію про процеси за допомогою команди ps:

```
sysadmin@localhost:~$ ps
    PID TTY          TIME CMD
     47 pts/0    00:00:00 bash
     60 pts/0    00:00:00 top
     61 pts/0    00:00:00 ps
```

The command *ps -aux* will display detailed information about all the processes.

```
sysadmin@localhost:~$ ps -aux
USER         PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1  0.0  0.0   4388   772 pts/0    Ss   17:52   0:00 /sbin/init
root           7  0.0  0.0  78644  3808 pts/0    S    17:52   0:00 /bin/login -f
syslog        10  0.0  0.0 191336  3736 ?        Ssl  17:52   0:00 /usr/sbin/rsy
root          14  0.0  0.0  28368  2696 ?        Ss   17:52   0:00 /usr/sbin/cro
root          16  0.0  0.0  72312  3360 ?        Ss   17:52   0:00 /usr/sbin/ssh
bind          26  0.0  0.0 217060 18300 ?        Ssl  17:52   0:00 /usr/sbin/nam
sysadmin      47  0.0  0.0  19228  4324 pts/0    S    17:52   0:00 -bash
sysadmin      60  0.0  0.0  38716  3300 pts/0    T    17:54   0:00 top
sysadmin      63  0.0  0.0  36712  3248 pts/0    R+   17:57   0:00 ps -aux
sysadmin@localhost:~$
```

- *наведіть 5 прикладів з використанням різних параметрів команди ps (наприклад, вивести тільки системні процеси, вивести процеси конкретного користувача, вивести дерево процесів тощо). Опишіть, що саме роблять обрані Вами параметри:

    - **ps -a:** Option -a tells ps to display all processes except session headers and processes without a terminal.

```
sysadmin@localhost:~$ ps -a
    PID TTY          TIME CMD
      7 pts/0    00:00:00 login
     47 pts/0    00:00:00 bash
     60 pts/0    00:00:00 top
     64 pts/0    00:00:00 ps
```

- **ps -f:** Option -f tells ps to display the full format listing.

```
sysadmin@localhost:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
sysadmin      47       7  0 17:52 pts/0    00:00:00 -bash
sysadmin      60      47  0 17:54 pts/0    00:00:00 top
sysadmin      65      47  0 18:01 pts/0    00:00:00 ps -f
sysadmin@localhost:~$
```

- **ps -l:** Option -l tells ps to display a long format of process information.

```
sysadmin@localhost:~$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1001      47       7  0  80   0 -  4807 wait   pts/0    00:00:00 bash
4 T  1001      60      47  0  80   0 -  9679 signal pts/0    00:00:00 top
0 R  1001      66      47  0  80   0 -  6490 -      pts/0    00:00:00 ps
sysadmin@localhost:~$
```

- **ps -u username:** Display processes of a specific user.

```
sysadmin@localhost:~$ ps -u sysadmin
    PID TTY          TIME CMD
     47 pts/0    00:00:00 bash
     60 pts/0    00:00:00 top
     67 pts/0    00:00:00 ps
sysadmin@localhost:~$
```

- **ps axjf:** Display the process tree.

```
sysadmin@localhost:~$ ps axjf
   PPID     PID    PGID     SID TTY        TPGID STAT   UID   TIME COMMAND
      0       1       1       1 pts/0       68 Ss       0   0:00 /sbin/init
      1       7       1       1 pts/0       68 S        0   0:00 /bin/login -f
      7      47      47       1 pts/0       68 S     1001   0:00  \_ -bash
     47      60      60       1 pts/0       68 T     1001   0:00      \_ top
     47      68      68       1 pts/0       68 R+    1001   0:00      \_ ps ax
      1      10      10      10 ?           -1 Ssl    104   0:00 /usr/sbin/rsy
      1      14      14      14 ?           -1 Ss       0   0:00 /usr/sbin/cro
      1      16      16      16 ?           -1 Ss       0   0:00 /usr/sbin/ssh
      1      26      26      26 ?           -1 Ssl    106   0:00 /usr/sbin/nam
sysadmin@localhost:~$
```

- **передивіться чи є у Вас запущені фонові процеси, які саме?

```
sysadmin@localhost:~$ jobs
[1]+  Stopped                 top
[2]-  Running                 sleep 1005 &
sysadmin@localhost:~$
```

The jobs command tells us:

- [1], [2] – job number.
- (+) – identifies the job as the current default. When using commands such as fg or bg, this is the job they will affect when no job id is specified.
  (-) – identifies that the job is next to become the default. When the current default exits, this will become the default job.
- Stopped, Running – the current state of that specific job.
- top, sleep 1005 &– the command line that launched the process.

To show the process IDs of all jobs, I use the *jobs -l* command.

```
sysadmin@localhost:~$ jobs -l
[1]+    60 Stopped (signal)        top
[2]-    61 Running                 sleep 1005 &
sysadmin@localhost:~$
```

- **відновити виконання призупиненого фонового процесу спочатку у позиції “на передньому плані” (foreground), потім ще раз його призупинити, а потім відновити його виконання у позиції “на задньому плані” (background).

1. First, let's resume the process in the foreground position: fg %1

```
sysadmin@localhost:~$ fg %1
```

```
top - 06:33:24 up 16 days, 14:52,  1 user,  load average: 0.66, 0.33, 0.23
Tasks:   9 total,   1 running,   8 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.9 us,  0.4 sy,  0.0 ni, 98.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem : 65841716 total, 41043972 free, 10259836 used, 14537908 buff/cache
KiB Swap:  8388604 total,  8388604 free,        0 used. 54817376 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
      1 root      20   0    4388    864    804 S   0.0  0.0   0:00.02 init
      7 root      20   0   78644   3760   3208 S   0.0  0.0   0:00.00 login
     10 syslog    20   0  191336   3880   3384 S   0.0  0.0   0:00.23 rsyslogd
     14 root      20   0   28368   2792   2524 S   0.0  0.0   0:00.00 cron
     16 root      20   0   72312   3428   2680 S   0.0  0.0   0:00.00 sshd
     26 bind      20   0  217060  18180   7036 S   0.0  0.0   0:00.02 named
     47 sysadmin  20   0   19228   4324   3124 S   0.0  0.0   0:00.03 bash
     60 sysadmin  20   0   38716   3204   2756 R   0.0  0.0   0:00.00 top
     61 sysadmin  20   0    4544    820    760 S   0.0  0.0   0:00.00 sleep
```

2. Then, to pause it, press Ctrl + Z.

```
[1]+  Stopped                 top
sysadmin@localhost:~$ bg
```

3. To resume the process in the "background" position, run the command:

```
sysadmin@localhost:~$ bg
[1]+ top &
sysadmin@localhost:~$
```

- завершити роботу даного фонового процесу.

```
sysadmin@localhost:~$ kill %1

sysadmin@localhost:
[1]-  Done                    top
sysadmin@localhost:~$ jobs
[2]+  Running                 sleep 1005 &
sysadmin@localhost:~$
```

After the *kill* command, I entered the *jobs* command to check if the top process had finished. However, for some reason, the command is not displayed. The screenshot shows that the *top* command no longer appears in the jobs, which indicates that it has stopped.

I repeat this procedure for the second process :

```
sysadmin@localhost:~$ jobs
[2]+  Running                 sleep 1005 &
sysadmin@localhost:~$ fg %2
sleep 1005
^Z
[2]+  Stopped                 sleep 1005
sysadmin@localhost:~$ bg %2
[2]+ sleep 1005 &
sysadmin@localhost:~$ kill %2
sysadmin@localhost:~$ jobs
[2]+  Terminated              sleep 1005
sysadmin@localhost:~$ jobs
sysadmin@localhost:~$ jobs
sysadmin@localhost:~$
```

Відповіді на контрольні запитання:

1. Яке призначення директорії */proc* в системах Linux. Яку інформацію вона зберігає?

The /proc file system serves as an interface to kernel data structures and runtime information, which provides a way for both users and applications to access detailed information about processes, system configuration, hardware, and more, by exposing this data through a hierarchy of virtual files.

2. Як серед будь-яких трьох процесів динамічно визначати, який з них в поточний момент часу використовує найбільший обсяг пам'яті? Який відсоток пам’яті він споживає від загального обсягу?

   To identify the process that is currently consuming the most memory among any three processes, you can use a monitoring tool like the *top* command. This command displays a list of currently running processes and provides information on memory usage. It usually sorts the processes by CPU usage. Processes can be sorted according to their memory usage (%MEM) by pressing the Shift + M key combination. The process that consumes the most memory will be first in the list.

   To calculate the percentage of memory used by this process, refer to the 'RES' column. This column displays the actual amount of memory that the process uses in the physical memory of the system. To determine the percentage, divide the 'RES' value by the total amount of physical memory.

3. Як отримати ієрархію батьківських процесів в системах Linux? Наведіть її структуру та охарактеризуйте.

   Processes can be mapped into a “tree” which can be viewed with the *pstree* command. This command displays a process tree where each process is a child of its parent process.

```
sysadmin@localhost:~$ pstree
init-+-cron
     |-login---bash---pstree
     |-named---3*[{named}]
     |-rsyslogd---2*[{rsyslogd}]
     `-sshd
sysadmin@localhost:~$
```

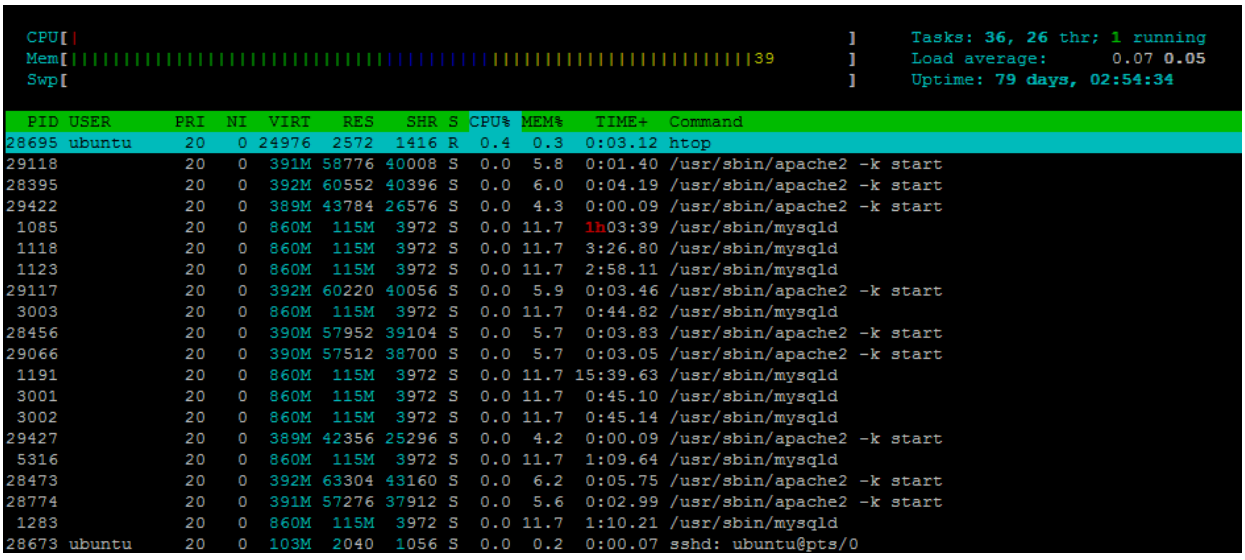
   The root of the tree is either the specified PID or 'init' if no PID is specified. If a username is provided, the tree will only display processes belonging to that user. To improve readability, identical branches are merged and displayed in square brackets with the number of repetitions in front. If child process threads are detected at the parent, they are displayed as the process name in curly braces.

4. *Чим відрізняється команда *top* від *ps*?

- One major difference between ps and top is that ps provides a snapshot of the current processes, while top gives a real-time, continuously updating view. Consequently, this enhances the suitability of top for real-time monitoring of system performance and resource utilization.
- ps allows you to select and display specific processes based on various criteria, such as user, process ID, or terminal. In contrast, the top shows all running processes by default, although you can sort and filter them interactively.
- The top command provides a more detailed and dynamic display, including system load averages, memory usage, and CPU usage. ps, on the other hand, offers a simpler, more static output that focuses on processing information.
- top shows detailed memory usage information, including total, used, free, and cached memory. ps, by default, does not display memory usage, but it can be customized to show this information.
- top also displays the percentage of CPU usage for each process, making it easier to identify resource-hungry processes. While ps can also display CPU usage, it requires additional customization to do so.

5. *Які додаткові можливості реалізує *htop* в порівнянні з *top*?

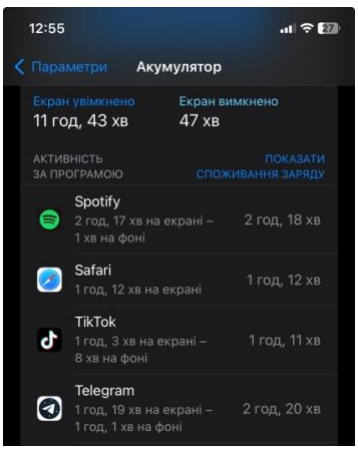
htop is a cross-platform ncurses-based process viewer. It is similar to *top*, but allows you to scroll vertically and horizontally, and interact using a pointing device (mouse). You can observe all processes running on the system, along with their command line arguments, as well as view them in a tree format, select multiple processes and act on them all at once. Tasks related to processes (killing, renicing) can be done without entering their PIDs.

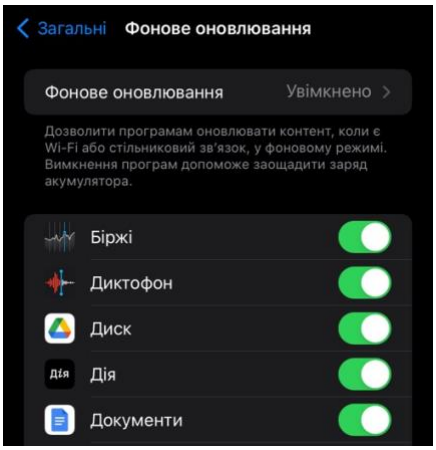


6. **Опишіть компоненти вашої мобільної ОС для здійснення моніторингу запущених в системі процесів?

Task Manager: iOS has its own way of managing and switching between apps. To access the switcher, swipe up from the bottom of the screen and hold it on the screen. Your recently used apps will appear on the screen, allowing you to quickly browse and switch between them, and close apps by swiping up.

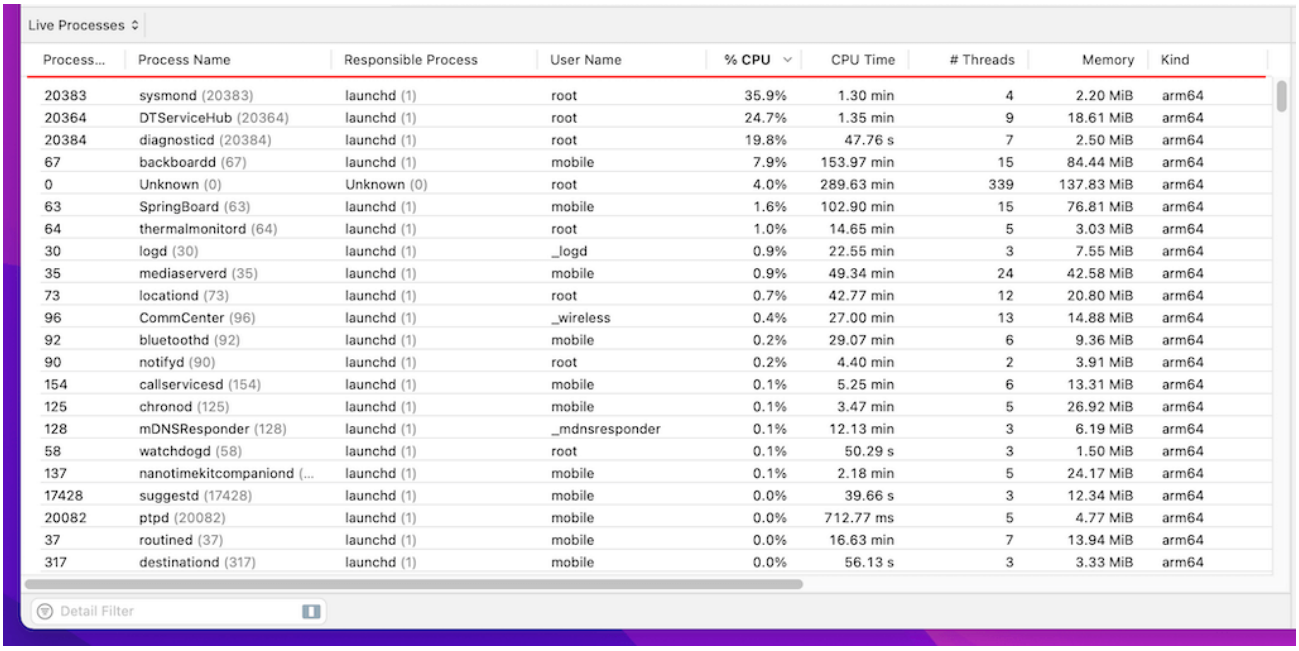
Device settings (Settings): You can control background processes on your iOS device in Settings. For example, you can disable resource-intensive features, such as background app updates or notifications, to reduce the number of processes running in the background. In the settings, you can also see how long the application has been running and how much memory it uses.





However, we will not be able to find out more detailed information because, unlike desktop operating systems, iOS doesn't offer its own ways to look into the processes running on an iPhone. There are also no apps in the App Store that can help you check the inner workings of your device.

Developer system tools: Apple provides developer tools that allow you to monitor the performance of apps on iOS devices in detail. These tools can include Xcode Instruments, which allows you to analyse various process parameters such as memory usage, execution time, networking, and more. However, you need to have a MacBook to use it.



7. **Чи підтримує Ваша мобільна ОС термінальне керування роботою процесів, опишіть як саме.

   No, iOS doesn't support managing workflows from a terminal. iOS uses a sandbox model that limits access to system resources for each application. This makes the system more secure and stable, but also makes it difficult to directly control processes from the terminal. However, users who have jailbroken their iOS devices can access the command line directly, either by using an app like MobileTerminal or by connecting directly to the device via SSH.

8. **Чи можливо поставити сторонні програмні засоби, що дозволяють організувати управління та моніторинг роботою процесів у Вашому мобільному телефоні. Коротко опишіть їх.

   I did some research in the App Store and none of the system monitoring apps worked in the functionality we needed, but they showed some information about battery, memory usage, CPU usage, but not quite the exact information we can view in iPhone settings anyway. Perhaps paid apps would show more parameters, but for security reasons, iOS does not allow third-party apps to control system processes or change system settings. This means that even paid apps are limited in their capabilities when it comes to managing and monitoring system processes on iPhone and other iOS devices.

   Programs:

   - *System Status Pro: hw monitor*

As the developers write System Status provides graphical monitoring of all the device's resources such as CPU, memory, disk, battery and wifi/cell data usage. Further, it shows network-related info such as wifi/cell connection details and the routing table. Finally, System Status displays operating system details such as kernel version, memory page statistics and shows various hardware-related data.

- *Device Monitor*

Displays information about the CPU, memory, temperature, battery and network. Allows you to monitor the CPU and memory usage by current processes.

Висновки:

During the laboratory work, I studied the main aspects of process management in the Linux operating system. Theoretical aspects of the command interface and working with the Bash command shell were covered in detail. Practical skills were gained in working with process monitoring and control commands, such as ps, top, kill, jobs, bg, and fg, along with their parameters. I've had problems with the terminal freezing up and not accepting what I've been typing. The work was delayed because of this problem.