“Київський фаховий коледж зв’язку”

Циклова комісія комп’ютерної та програмної інженерії

**ЗВІТ ПО ВИКОНАННЮ**

**ЛАБОРАТОРНОЇ РОБОТИ №4**

з дисципліни: «Операційні системи»

**Тема: «Команди Linux для управління процесами»**

Виконав(ла/ли) студент(ка/и)

групи: КСМ-23а

Папснази

Юхимець Д.О.Любежанін М.С

Перевірила викладач

Сушанова В.С.

Київ 2024

**Лабораторна робота №4   
 Тема:** “Знайомство з базовими командами CLI-режиму в Linux”

**Мета роботи:**

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

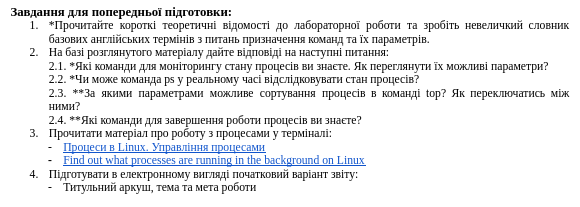
**Матеріальне забезпечення занять:**

1. ЕОМ типу IBM PC.

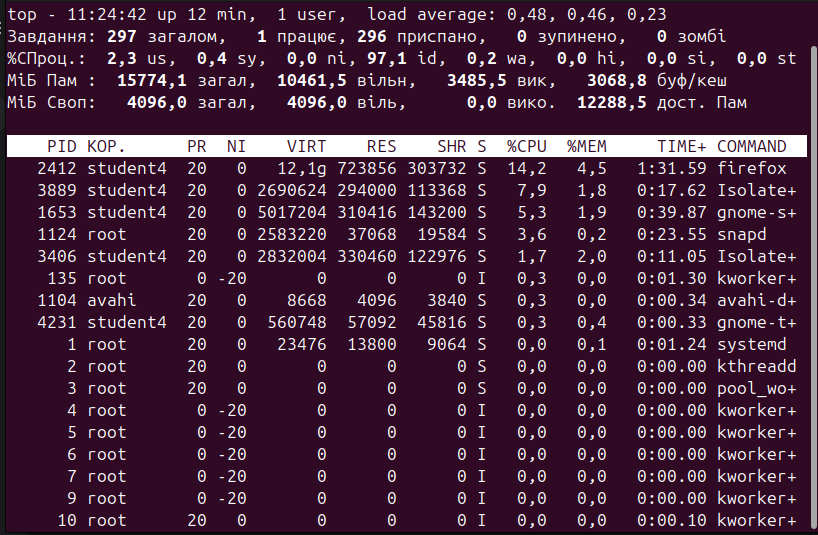
2. ОС сімейства Windows та віртуальна машина Virtual Box (Oracle).

3. ОС GNU/Linux (будь-який дистрибутив).

4. Сайт мережевої академії Cisco netacad.com та його онлайн курси по Linux

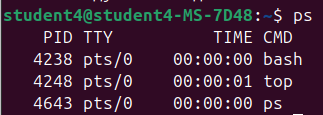


**Виконав Юхимець Дмитро**

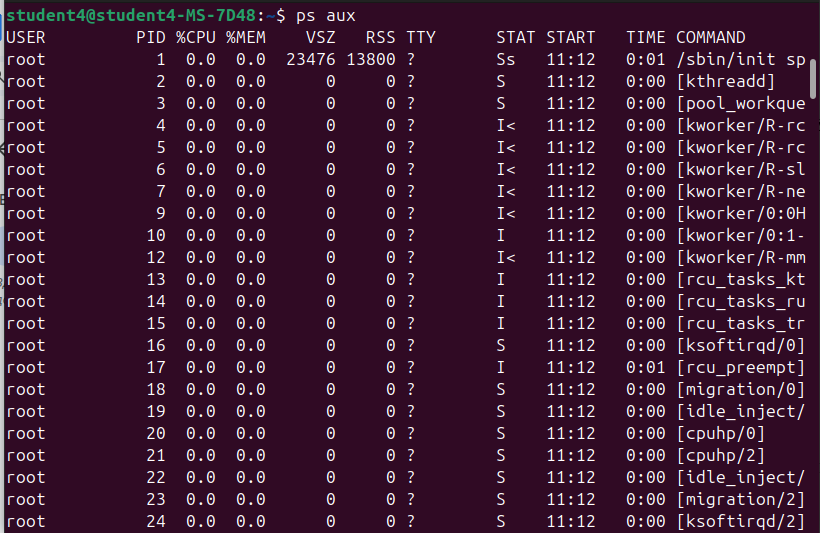
****

The top command displays information about the computer: time, number of users, CPU and memory utilization. It also displays information about all processes running in the system, including PID. It shows the priorities of CPU and memory usage. Using the top command, you can track the processes that load the system the most (games, browsers, daemons, and 3D editors).

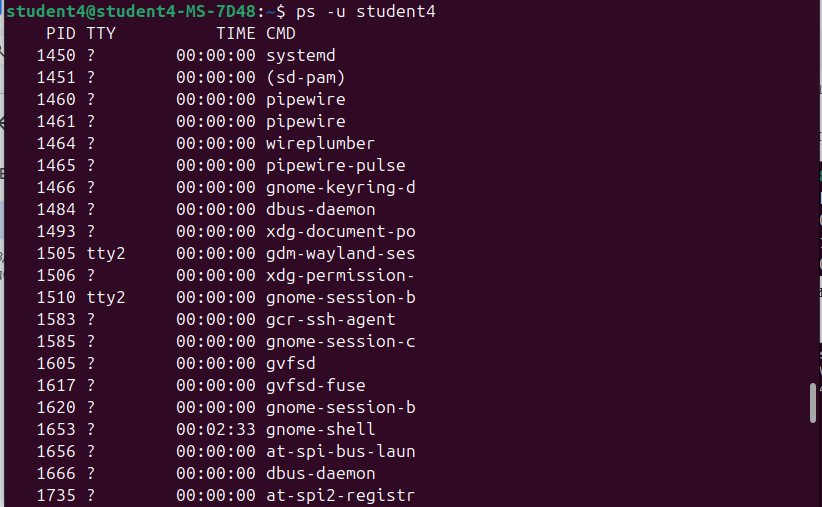
The ps command displays a list of processes.



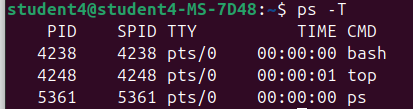
The ps aux(ps -ef) command displays detailed information about all users, including processes.

****

The ps -u “username” command displays information about a specific user.



The ps -T command displays a process tree showing parent and child processes.



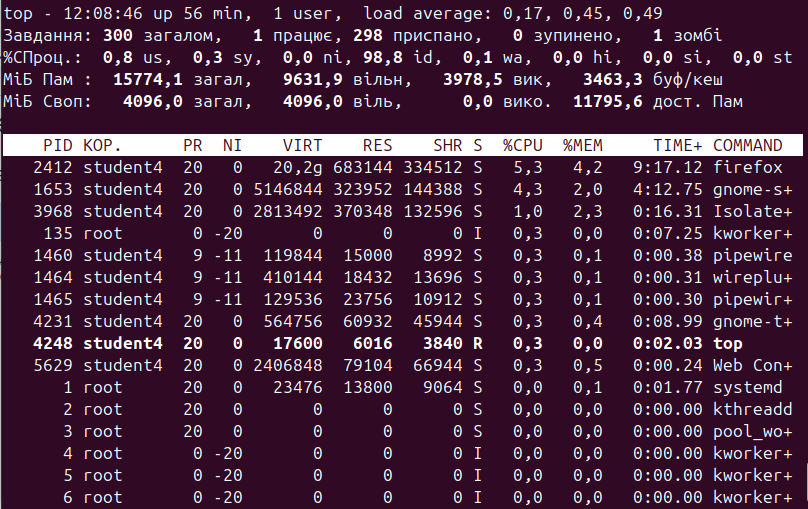
The ps -e | grep firefox command displays all processes related to the Chrome browser.

****

The jobs command view a list of suspended background processes.

з

Using fg %1, the top process was resumed.



Using bg %1, the top process was resumed.



kill %1 stop this process



**Відповіді на контрольні запитання**

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

The /proc directory in Linux is a special place, it's not an ordinary folder. It is part of the so-called virtual file system, where information about the current state of the system is stored.

The main idea of /proc is to provide access to the operating system and process current information in the form of files. For example, here you can find information about the processor, memory, system load, connected devices, and even individual running processes. Thanks to this, we can read the file to quickly understand important system parameters without running special commands.

**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 determine which of the three processes uses the most memory, you can write a script that will read information from the /proc directory and analyze memory consumption.  
  
1.**Check the memory consumption of each process** by its PID. Each process has a subdirectory in /proc, and the status file in that directory contains memory information. The VmRSS field in the /proc/[PID]/status file shows the amount of memory the process is currently using (in KB).

2.**Compare these values between processes** to determine which process has the largest volume in VmRSSs.  
  
**3.Як отримати ієрархію батьківських процесів в системах Linux? Наведіть її структуру та охарактеризуйте.**1. The pstree command The easiest way to see the hierarchy of processes is to use the pstree command, which visually shows the relationships between processes in the form of a tree. Each process in this tree is displayed taking into account its parent, that is, the dependency of each process is displayed. This command displays the process structure in the form of a tree, where the main process (often systemb or init) is at the top. Branches extend from it - child processes that can have their own children**.**The structure of processes in Linux

Themain process (often called systemb or init with PID 1): runs first after the system kernel starts. It is the parent process for many other processes.

Service processes: processes that are run by systemb to support system services (e.g. networking, file systems, user management).

User processes: processes that run user programs (e.g., bash, text editors, browsers). Their parents are often other processes, such as the shell that runs these programs.

**4.Чим відрізняється команда *top* від *ps*?  
  
1. The top command**

Dynamic monitoring: top updates information about processes in real time, usually every few seconds. This allows you to observe how the CPU and memory load changes, which is useful for tracking instant changes.

Interactive mode: top allows you to interact with the data. For example, you can sort processes by various parameters (e.g., CPU or memory usage), terminate processes directly from the top interface, set the refresh interval, etc.

Display of system resources: in addition to information about processes, top also shows the overall system load, CPU, memory, swap usage, and the average system load (load average).

### 2. The ps command

Static snapshot: unlike top, the ps command displays information about processes at the moment the command is run, without further updates. This is convenient for getting an instant snapshot of the state of processes without constant refreshing.

Flexible parameters: ps supports many options for filtering, sorting, and formatting the output. For example, you can display information about the processes of a specific user (ps -u username), sort processes by a certain criterion, display a process tree (ps axjf), and so on.

Convenience for scripts: since ps outputs static data, it is often used in scripts to collect information about the state of processes in an automated way, for example, for monitoring or creating reports**.**

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

htop is an improved version of top that provides a more user-friendly interface and additional features for monitoring Linux processes. Here are the main differences and advantages of htop over top :

### 1. User-friendly interface and color scheme

htop has an intuitive interface with the use of colors, which makes it easier to navigate the CPU, memory, and swap load. This allows you to quickly assess the state of the system.

At the top of the window, htop shows dynamically changing graphs of resource utilization.

### 2. Interactivity and flexible management

In htop, you can easily scroll through the list of processes (horizontally and vertically), which allows you to view all running processes and details without any restrictions on the screen size.

htop allows you to select and terminate multiple processes at the same time, as well as send various signals to processes directly from the interface.

### 3. Filtering and sorting processes

htop allows you to quickly sort processes by various parameters such as CPU usage, memory usage, PID, etc. just by pressing a key.

A search mode is available to quickly find a process by name or PID.

### 4. CPU information on multi-core systems

In htop, information about the utilization of each CPU core is displayed separately, while top usually shows the total load. This is especially useful for monitoring load balancing in multi-core systems.

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

In a mobile operating system, the monitoring of running processes is carried out through several main components, each of which performs its own role in collecting, processing, and displaying information about processes. Here are the main components of such an OS required for process monitoring:

### 1. Operating system kernel

Process management: The kernel is the backbone of monitoring because it is what creates, terminates, schedules, and controls processes. It also provides the basic interfaces (system calls) for obtaining information about processes, such as their status, priorities, memory and CPU usage.

Kernel modules: The kernel can have a special module for collecting metrics (like Linux /proc), where each process has its own PID and can learn about its status, resource usage, etc. This can also be implemented through an API to get the data.

### 2. Metrics collection and storage system

Resource monitoring: Runs in the background and continuously collects data on memory, CPU, battery, and network activity. In mobile OSes such as Android and iOS, there are special system services that record these parameters in real time.

System logs: Important information about process startups, terminations, and crashes is stored in logs that are accessible to administrators or developers. This makes it easier to diagnose and identify problematic processes.

**3. Task and process manager**

Process management APIs: The operating system provides APIs for managing and monitoring processes (for example, ActivityManager in Android) that allow you to track the status of active applications, terminate them, or launch new ones.Visual interface (task manager): Mobile OSes have an interface for viewing running applications and terminating them (if they hang or take up too many resources).

**7. Чи підтримує Ваша мобільна ОС термінальне керування роботою процесів, опишіть як саме.**The iOS operating system, unlike traditional desktop systems such as Linux or macOS, does not have direct terminal access for ordinary users and does not allow process control through a standard command line interface. This limitation is due to the iOS security architecture, which provides tight control over applications and processes, limiting user access to system resources and the kernel.

However, there are some opportunities and tools for developers and technicians:

### 1. Developer mode and Xcode

Developers can get limited access to application processes and resources through the Xcode environment. Xcode Instruments provides tools for tracking and optimizing application resource utilization, but does not allow you to terminate or reprioritize processes.

Thelldb tool (integrated into Xcode) can run commands to debug and execute applications, but again is limited to applications under development.

### 2. Access via Apple Configurator and iTunes Console

For administrators and IT professionals, Apple Configurator provides device management capabilities (configuration, deployment of security profiles), but it does not allow full access to processes through the command line.

The Xcode or iTunes console allows you to view application and system logs, but does not provide tools for direct process control.  
  
**3. Jailbreak to increase access**

**Jailbreaking** is an unofficial method that allows users to gain root access to the iOS file system. After jailbreaking, you can install terminal emulators (for example, MobileTerminal) that allow you to work with processes through the command line interface.

After jailbreaking, MobileTerminal or other similar applications can be used to execute commands similar to ps, top, or kill commands on Unix-like systems. However, jailbreaking significantly reduces the security of the device and is not supported by Apple.

### 4. MDM systems (Mobile Device Management)

Companies can use MDM systems to remotely manage devices, including installing and terminating applications, and monitoring updates. However, MDM systems do not provide direct access to the command line and processes  
  
  
  
**Висновки**  
  
Під час Лабораторної роботи ми отримали практичні навички з оболонкою Bash. А також ознайомились з базовими командами для управління процесами.