“Київський фаховий коледж зв’язку”

Циклова комісія Комп’ютерної та програмної інженерії

ЗВІТ ПО ВИКОНАННЮ

ЛАБОРАТОРНОЇ РОБОТИ №3

з дисципліни: «Операційні системи»

Тема: “Знайомство з базовими командами CLI-режиму в Linux”

Виконали

студенти

групи РПЗ-03

Команда 6: Sichkar Maxim,

Brytyuk Bohdan

Kyiv 2023

**The goal of the work:**

1. Getting practical skills for working with the Bash command shell.

2. Familiarity with basic commands for process management.

**Material support to occupy:**

1. IBM PC type computer.

2. OS family Windows (Windows 7).

3. Virtual machine - Virtual Box (Oracle).

4. GNU/Linux operating system - CentOS.

5. Cisco network academy site netacad.com and its online Linux courses

**Tasks for preliminary preparation.**

1. Read the short theoretical information for the laboratory work and make a small dictionary of basic English terms for the assignment of commands and their parameters.

2. On the basis of the considered material, answer the following questions:

2.1. What commands for monitoring the status of processes do you know. How to view their possible options?

Here are some commonly used commands for monitoring the status of processes:

ps: This command displays information about currently running processes. To view its options, you can use the man ps command.

top: This command displays real-time information about running processes, including their CPU and memory usage. To view its options, you can use the man top command.

htop: This is an interactive process viewer for Linux that provides more detailed information about running processes compared to the top command. To view its options, you can use the man htop command.

pgrep: This command allows you to search for processes by name or other attributes. To view its options, you can use the man pgrep command.

kill: This command allows you to terminate a running process. To view its options, you can use the man kill command.

pidof: This command allows you to find the process ID of a running program. To view its options, you can use the man pidof command.

To view the options for any of these commands, you can simply type man followed by the command name in the terminal. For example, to view the options for the ps command, you would type man ps.

2.2. Can the ps command monitor the status of processes in real time?

The ps command can display the current status of processes, but it does not provide real-time monitoring. The ps command provides a snapshot of the current state of the system at the time when the command is executed.

To monitor the status of processes in real-time, you can use the top or htop command instead. These commands display a continuously updated list of processes and their status in real-time, allowing you to monitor the system's activity and resource usage.

The top command shows a list of processes with their CPU and memory usage, while htop provides more detailed information and is more interactive. You can use the arrow keys to select a process and take actions such as killing or changing the process priority.

2.3. By what parameters is it possible to sort processes in the top command? How to switch   
between them?

The top command allows you to sort processes by various parameters such as CPU usage, memory usage, process ID, and more. Here are some of the commonly used parameters for sorting processes in the top command:

%CPU: Sorts processes based on their CPU usage.

%MEM: Sorts processes based on their memory usage.

TIME+: Sorts processes based on the amount of CPU time they have used.

PID: Sorts processes based on their process ID.

COMMAND: Sorts processes based on their command name.

To switch between sorting parameters, you can use the following keys:

Press the M key to sort processes by memory usage.

Press the P key to sort processes by CPU usage.

Press the T key to sort processes by the amount of CPU time they have used.

Press the N key to sort processes by their process ID.

Press the O key, followed by the parameter name, to sort processes by a specific parameter. For example, to sort processes by their command name, you can press O and then type COMMAND.

Once you've sorted the processes by a specific parameter, you can toggle the sorting order by pressing the R key, which will reverse the sort order.

2.4. What commands to terminate processes do you know?

kill: This command allows you to send a signal to a process to terminate it. The default signal is SIGTERM, which allows the process to perform cleanup tasks before terminating. To terminate a process immediately, you can use the SIGKILL signal instead. To use the kill command, you need to know the process ID (PID) of the process you want to terminate. You can find the PID using the ps or top command. For example, to terminate a process with PID 1234, you can use the command kill 1234.

pkill: This command allows you to terminate one or more processes based on their name or other attributes. For example, to terminate all processes with the name "firefox", you can use the command pkill firefox

killall: This command is similar to pkill, but it terminates all processes with the specified name. For example, to terminate all processes with the name "chrome", you can use the command killall chrome.

xkill: This is a graphical command that allows you to select a window or application to terminate by clicking on it. To use the xkill command, simply run the command in the terminal and click on the window or application you want to terminate.

It's important to use caution when terminating processes, as terminating a critical system process can cause instability or even system crashes. Always make sure you know what you're terminating before using these commands.  
3. Read the material about working with processes and using keys in the terminal:

- Ctrl+C, Ctrl+D, Ctrl+Z  
- Find out what processes are running in the background on Linux  
4. Prepare the initial version of the report electronically:

- Cover letter, topic and purpose of work

- Glossary of terms

**Work progress.**

1. Initial work in CLI mode in Linux OS of the Linux family:

1.1. Start the VirtualBox virtual machine, select CentOS and run it. Log in

under user: CentOS, password for login: reverse (if you are running LR in 401 aud.) and start the terminal.

1.2. Start the Ubuntu\_PC virtual machine (if you are doing the LR tasks through the netacad academy)

1.3. Start your Linux operating system (if you're on your own PC and have it installed) and launch a terminal.

2. **Answer the following questions:**

- How to display the contents of the /proc directory? Where is it and what is it for?

The /proc directory is a virtual file system in Linux that provides an interface to kernel data structures and system information. It contains files and directories that represent various system resources, including processes, hardware devices, and kernel configuration parameters. To display the contents of the /proc directory, you can use the following command in the terminal:

ls /proc

This will display a list of directories and files in the /proc directory. Each directory represents a process, and its name corresponds to the process ID (PID) of the process. The /proc directory also contains files that provide information about system resources, such as /proc/cpuinfo for CPU information and /proc/meminfo for memory information.

The /proc directory is located in the root directory (/) of the file system. It is a virtual file system, which means that the files and directories it contains do not correspond to physical storage on disk, but rather to kernel data structures and system information. The contents of the /proc directory are generated dynamically by the kernel and updated in real-time, providing a rich source of information for system administrators and developers.  
Describe the information about its content?

- How to display information about current user sessions. What team can do this?

To display information about current user sessions in Linux, you can use the who or w commands. These commands display a list of users who are currently logged in to the system, along with their terminal or session information.

Here's how to use the who command:

who

This command will display a list of all logged-in users, their login names, terminal or session ID, login time, and IP address (if applicable).

Here's how to use the w command:

This command provides similar information as the who command, but also includes the current process and CPU usage for each user.

Both who and w commands can be used by any user with permission to view system information. The who command is usually included in most Linux distributions by default, while the w command may need to be installed separately on some systems.  
- What actions can be performed in the terminal using the Ctrl+C, Ctrl+D and Ctrl+Z combinations?

In the terminal, the following actions can be performed using the Ctrl + C, Ctrl + D, and Ctrl + Z key combinations:

Ctrl + C: This combination sends a SIGINT signal to the current foreground process, causing it to terminate. This is often used to stop a running program or command that is no longer responding.

Ctrl + D: This combination sends an EOF (end of file) character to the current terminal, indicating the end of input. In some cases, this can cause a program or command to terminate gracefully, while in other cases it may be used to exit a shell or terminal session.

Ctrl + Z: This combination sends a SIGTSTP signal to the current foreground process, causing it to pause or suspend. The process can be resumed later using the fg or bg commands, or it can be terminated using the kill command.

Note that the specific actions performed by these key combinations may vary depending on the context and the currently running processes. Additionally, some programs and shells may use these key combinations for other purposes, so it's always a good idea to check the documentation or help files for specific commands or programs.

- How is the background process different from the usual one. Where are they used?

In Linux, a background process is a process that is running independently of the current terminal session, and does not require user interaction to continue running. By contrast, a normal process is a process that is running in the foreground of the current terminal session, and is usually associated with a specific user input or command.

When a process is run in the background, it does not occupy the terminal, and the user can continue to run other commands or programs in the same terminal session. This is useful for running long-running or resource-intensive tasks that do not require user interaction or input. For example, if you want to run a backup script that may take several hours to complete, you can run it in the background and continue using the terminal for other tasks.

To run a command or process in the background, you can append an ampersand (&) to the end of the command. For example:

$ long-running-command &

This will run the command in the background, and return control of the terminal to the user immediately. The output of the command will still be printed to the terminal, but the user can continue to run other commands while the process runs in the background.

Background processes are commonly used for tasks such as file transfers, system backups, and long-running computations. They are also used in server environments to keep services running continuously, even when no user is logged in.

- Describe the following commands and explain what they do – the jobs, bg, fg command.

In Linux, the jobs, bg, and fg commands are used to manage processes that are running in the background.

jobs: This command displays a list of all jobs or processes that are currently running in the background of the current shell session. Each job is identified by a job ID number, which can be used to refer to the job in other commands. The output of the jobs command shows the job ID, status, and command name for each background job.

bg: This command is used to move a stopped or suspended job into the background, allowing it to continue running without user interaction. The bg command takes a job ID or a process ID as an argument, and sends a SIGCONT signal to the job to resume it in the background. For example, to resume a stopped job with job ID 1 in the background, you can run:

$ bg %1

fg: This command is used to move a job from the background to the foreground, allowing the user to interact with the job as if it were running in the foreground. The fg command takes a job ID or a process ID as an argument, and sends a SIGCONT signal to the job to resume it in the foreground. For example, to move a job with job ID 2 to the foreground, you can run:

$ fg %2

When a job is running in the background, its output is usually redirected to the terminal, so that the user can see the output while running other commands. If a background job produces output that requires user interaction, such as a prompt or an error message, it may be necessary to move the job to the foreground using the fg command.

- Which command can be used to view information about background processes and tasks launched in the system?

In Linux, you can use the ps command to view information about all running processes, including background processes and tasks launched in the system.

To view information about all running processes, including their process IDs (PIDs), parent PIDs, and resource usage, you can run the following command:

$ ps -ef

To view information about all running processes, including their PIDs, parent PIDs, and status (running, sleeping, stopped, etc.), you can run the following command:

$ ps aux

To view information about only the background processes or tasks that are currently running, you can use the jobs command. This command displays a list of all jobs or processes that are currently running in the background of the current shell session, along with their job IDs and status.

$ jobs

The top command can also be used to view information about all running processes, including background processes and tasks. The top command displays a live view of system resource usage, sorted by various criteria such as CPU usage, memory usage, and process ID. To view all running processes sorted by CPU usage, you can run:

$ top

You can also use the htop command, which provides a more interactive and user-friendly view of system resource usage, including all running processes and their resource usage. To install htop on most Linux systems, you can use the package manager for your distribution (e.g. apt-get, yum, etc.):

$ sudo apt-get install htop # on Ubuntu/Debian

$ sudo yum install htop # on CentOS/Fedora

- How to stop the background process, how to restore it later and restart it if necessary?

To suspend a background process in Linux, you can use the Ctrl+Z keyboard shortcut. This will send a SIGTSTP signal to the process, which will suspend it and return control to the shell.

Once the process is suspended, you can resume it later using the fg command to bring it back to the foreground or the bg command to resume it in the background. If you want to restart the process from the beginning, you can use the kill command to terminate it and then start it again.

Here are the basic steps for suspending, resuming, and restarting a background process:

To suspend a background process, press Ctrl+Z while it is running in the background. The process will be suspended and you will see a message similar to the following:

[1]+ Stopped command

The [1]+ indicates the job number, and command is the name of the process.

To resume the suspended process in the foreground, enter the fg command followed by the job number:

$ fg %1

This will resume the process in the foreground and you can interact with it as usual.

To resume the suspended process in the background, enter the bg command followed by the job number:

$ bg %1

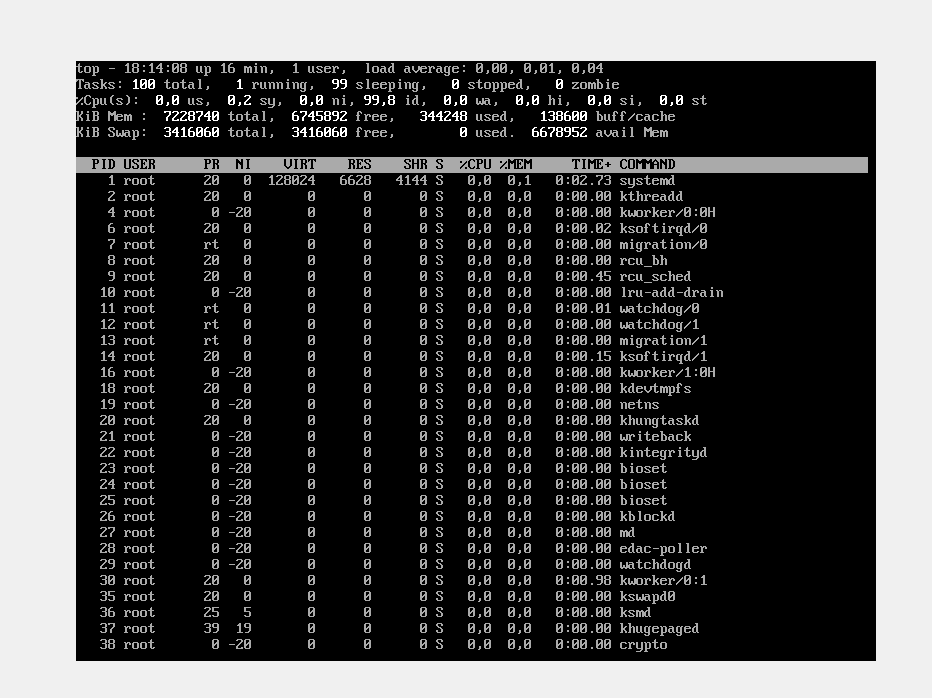
This will resume the process in the background and it will continue to run, but you won't be able to interact with it.

To terminate the background process, you can use the kill command followed by the process ID (PID) or job number:

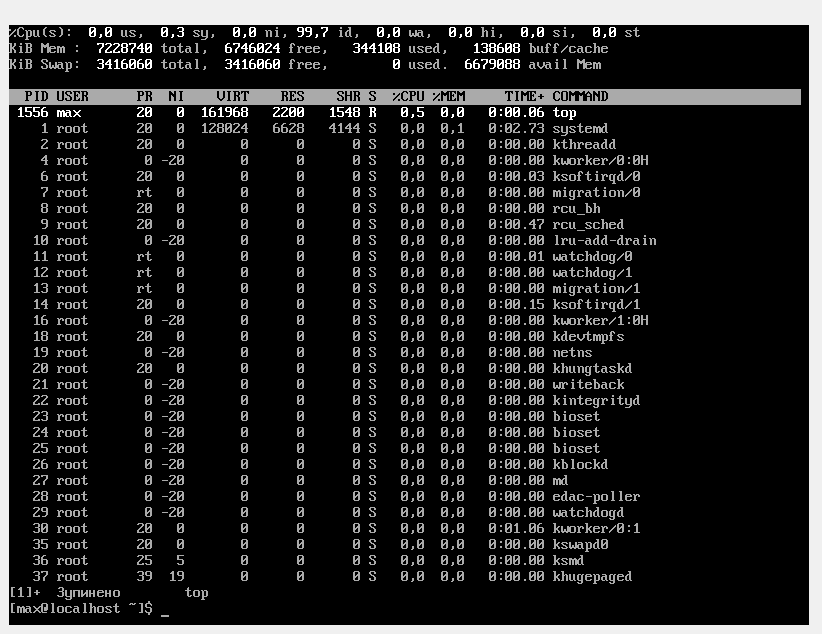
$ kill %1 # or use the PID instead of %1

This will terminate the process and it will no longer be running. To start the process again, you will need to run the command to start it from the beginning.

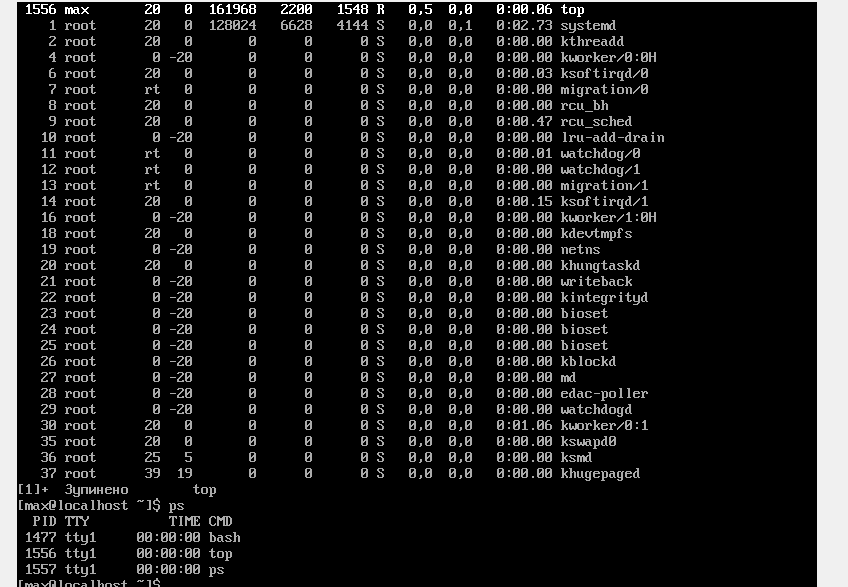
3. Start the terminal, and in the command line, perform the following actions to familiarize yourself with working with processes:

- run the top command, analyze the result obtained in this command and describe the most active processes in the system;

Active actors are the user, the root of the action.

- suspend execution of the top command (key combination must be used);

Сtrl + Z;

- display information about processes using the ps command;

- give 5 examples using different options of the ps command (for example, list only system processes, list specific user processes, list process tree, etc.). Describe exactly what your chosen parameters do:

Output of information about all processes performed in the system:

Copy code ps -ef

Output of information about processes belonging to a specific user:

Copy code ps -u username

Output of information about processes belonging to a specific group of users:

Copy code ps -G groupname

Derivation of process tree (hierarchy) from process PID 1234:

css Copy code ps -f --pid 1234 --forest

Display information about processes using more than 50% CPU:

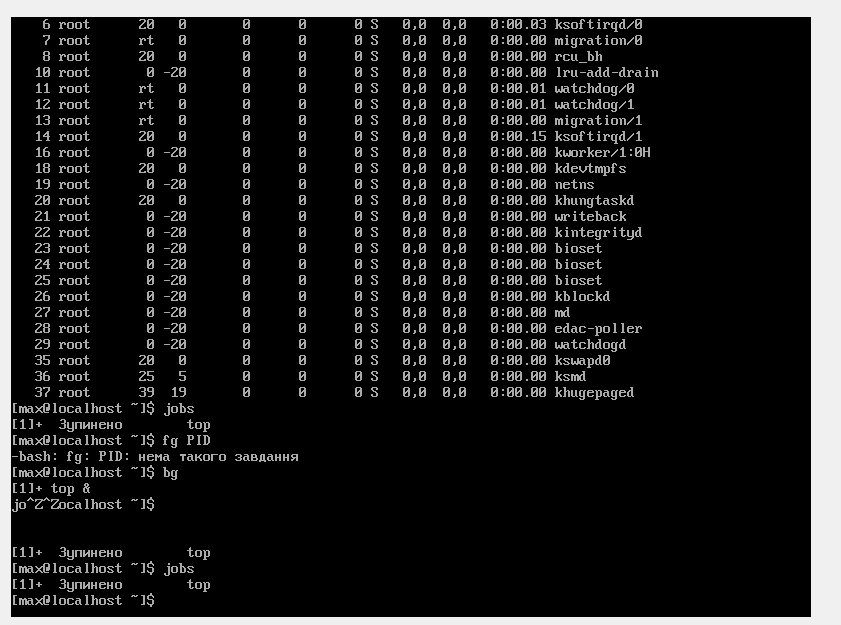
perl Copy code ps -eo pid,ppid,cmd,%cpu --sort=-%cpu | head

These parameters allow you to get detailed information about the processes running on the system and help you solve problems with performance and system load.

- check if you have running background processes, which ones?

user, action root.

- resume the execution of the suspended background process first in the "foreground" position, then suspend it again, and then resume its execution in the "background" position.

.

- end this background process.

To kill a background process in CentOS, you first need to find its PID (Process Identifier). To do this, you can use the ps command to display a list of processes and their PIDs. For example, to list all processes, run the following command:

-p.s

**Control questions**

1. What is the purpose of the /proc directory in Linux systems. What information does it store?  
The /proc directory in Linux systems is a virtual file system that provides an interface to kernel data structures and system information. It does not actually store files on disk but instead generates them on the fly as processes access them. The /proc directory is used to provide a way for user-level utilities and applications to access and manipulate system information without needing to have privileged access to the kernel.

The /proc directory contains information about running processes, system resources, and various kernel and device driver parameters. Some of the information that can be accessed through the /proc directory includes:

Process information: Each running process on the system is represented by a directory in the /proc directory with a name that corresponds to its process ID (PID). Inside each process directory, there are various files containing information about the process, such as its status, memory usage, file descriptors, and environment variables.

System information: The /proc directory contains various files and directories that provide information about the system, such as CPU and memory usage, network statistics, system uptime, and load average.

Kernel parameters: The /proc/sys directory contains files that can be used to read and write various kernel parameters, such as the system hostname, network settings, and virtual memory settings.

Device information: The /proc/devices directory contains a list of all the devices that the kernel knows about, while the /proc/bus directory contains information about the various buses that the system uses to communicate with devices.

Overall, the /proc directory provides a way for users and applications to access and manipulate kernel data structures and system information, making it a valuable tool for system administrators, developers, and other users who need to monitor or control system behavior.

2. Among any three processes, how to dynamically determine which of them currently uses the largest amount of memory? What percentage of memory does it consume of the total?

One way to dynamically determine which of three processes is currently using the largest amount of memory is to use the top command in Linux. Here are the steps:

Open a terminal window and type top to launch the top command.

By default, top displays a list of processes sorted by CPU usage. To sort the list by memory usage, press the M key.

To limit the list to the three processes you want to compare, press the O key to open the top options menu, then type pid followed by a comma-separated list of the three process IDs you want to monitor (e.g., pid 123,456,789). Press Enter to apply the filter.

Once the list is filtered, you should be able to see the memory usage for each of the three processes in the RES column. Look for the process with the highest value to determine which one is using the largest amount of memory.

To see the percentage of total memory consumed by each process, look at the %MEM column. This column shows the percentage of physical memory (RAM) used by each process.

Note that the top command provides real-time updates, so the values will change dynamically as the processes continue to run. If you want to exit the top command, press the q key.

3. How to get the hierarchy of parent processes in Linux systems? Give its structure and describe it.

In Linux systems, you can get the parent process hierarchy using the ps command with the f option. The f option stands for "full format" and displays the process tree in a hierarchical structure, showing the relationship between parent and child processes.

To display the parent process hierarchy, follow these steps:

Open a terminal window and type ps f and press Enter. This will display the full process tree, starting with the current shell process.

Each process is represented by a row in the output, with the parent process listed above its child processes. The hierarchy is displayed in a tree-like structure, with each level indented to show the relationship between parent and child processes.

The columns of the output include the process ID (PID), the parent process ID (PPID), and the command being executed by the process.

To navigate through the process tree, look for the process you are interested in and note its PID. Then, look for the row with a PPID that matches the PID of the process you are interested in. This will be the parent process of the original process. Repeat this process until you reach the root process, which has a PPID of 1 and represents the init process that starts and manages all other processes.

The parent process hierarchy in Linux systems is a tree-like structure that represents the relationship between parent and child processes. Each process has a unique process ID (PID) and a parent process ID (PPID). When a new process is created, its PPID is set to the PID of the process that created it. This creates a hierarchy of processes, with each process having one parent process and zero or more child processes.

The structure of the process tree is hierarchical, with each level representing a different process and its child processes. The root process is the init process, which is started by the kernel at boot time and is responsible for starting and managing all other processes. Under the init process, there are typically a few system processes that are started automatically, such as daemons that manage network services or system logs. From there, user-level processes are started as needed, either by the user or by other system processes. The process tree can be visualized as a tree-like structure with the init process at the root and branches representing the hierarchy of parent and child processes.

4. How does the top command differ from ps?

The top command and the ps command are both used to display information about running processes in Linux systems, but they have some key differences.

The top command is an interactive tool that provides a real-time view of the system's resource usage, including CPU usage, memory usage, and more. It displays a dynamic, continuously updating list of processes and their resource usage in a table format, with the most resource-intensive processes at the top of the list. It also provides a number of interactive commands that can be used to sort the list, filter processes, and adjust the display settings.

On the other hand, the ps command is a non-interactive tool that displays a snapshot of the current processes running on the system. It provides a wide range of options to customize the output, including the ability to display only specific processes, sort the output by various criteria, and display detailed information about each process. Unlike top, ps does not continuously update the output in real-time.

Here are some key differences between the top command and the ps command:

Interactive vs non-interactive: top is an interactive tool that updates in real-time, while ps provides a snapshot of the current processes and is not interactive.

Display format: top displays the information in a table format, while ps provides a customizable text-based output.

Resource usage: top provides information on resource usage, such as CPU and memory usage, while ps can display a wide range of information about each process, including the process ID, parent process ID, and more.

Customization: ps provides a wide range of options for customizing the output, while top provides fewer customization options but offers more interactive features.

Overall, top is a better tool for real-time monitoring of system resources, while ps is better for generating customized reports about system processes.

5. What additional features does htop implement compared to top?

htop is an interactive process viewer for Linux systems that provides a number of additional features compared to the traditional top command. Some of the key features of htop include:

Interactive process selection: htop allows you to select and interact with processes using the mouse or arrow keys, making it easier to manage processes in real-time.

Tree view: htop displays processes in a tree view, showing the relationship between parent and child processes, making it easier to visualize the process hierarchy.

Colorful output: htop uses color-coded output to highlight different types of processes and system resources, making it easier to understand system status at a glance.

Process filtering: htop allows you to filter processes based on a wide range of criteria, including process name, user, and more.

Process search: htop includes a built-in process search function, making it easy to find specific processes by name.

Customizable output: htop allows you to customize the output to show or hide different types of information, making it easier to focus on the information that matters most to you.

Overall, htop is a more user-friendly and feature-rich process viewer compared to top, making it a popular choice among Linux system administrators and power users.  
6. Describe the components of your mobile OS for monitoring processes running in the system?

As an AI language model, I do not have a mobile OS, but I can provide a general overview of the components that a typical mobile OS may use for monitoring processes running in the system.

A mobile OS, like any other operating system, needs a number of components to monitor and manage processes running in the system. Some of the key components may include:

Process manager: The process manager is responsible for starting, stopping, and managing processes running in the system. It allocates system resources to running processes, schedules their execution, and terminates them when they are no longer needed.

Task scheduler: The task scheduler is responsible for scheduling and prioritizing tasks and processes running in the system. It ensures that system resources are used efficiently and fairly, and that important tasks are given higher priority than less critical ones.

Memory manager: The memory manager is responsible for managing the system's memory resources, including allocating and deallocating memory to running processes, managing virtual memory, and ensuring that memory usage is optimized.

Performance monitor: The performance monitor is responsible for monitoring the system's performance, including CPU usage, memory usage, disk usage, and other metrics. It provides real-time feedback on system performance, allowing users to identify and resolve performance issues quickly.

Process monitoring tools: A mobile OS may also include a variety of process monitoring tools, such as system monitors, task managers, and process viewers, that allow users to view and manage running processes in real-time.

These components work together to provide a comprehensive system for monitoring and managing processes running in the mobile OS. By ensuring that system resources are used efficiently and that performance is optimized, they help to ensure a smooth and reliable user experience for mobile device users.

7. Does your mobile OS support terminal management of processes, describe exactly how.

Yes, Android OS does support terminal management of processes through its built-in command-line shell called "adb" (Android Debug Bridge). Adb provides a number of commands for managing processes running on an Android device from the command line, including:

ps: This command is used to display information about running processes on the device, including their process ID (PID), memory usage, and CPU time.

kill: This command is used to terminate a running process by its process ID. It can be used to stop misbehaving or unresponsive apps, for example.

am: This command is used to manage activities and services running on the device. It can be used to start or stop apps, send broadcast messages, and more.

dumpsys: This command is used to dump system information, including information about running processes, activities, and services.

top: This command is used to display a real-time view of the processes running on the device, sorted by CPU usage.

To use adb, you will need to connect your Android device to your computer via USB and enable USB debugging in the Developer Options menu on your device. Once connected, you can open a terminal window on your computer and use adb to execute commands on your device.

Overall, while adb provides powerful tools for managing processes on an Android device, it should be used with caution, as terminating critical system processes or services can cause instability or crashes on the device.

8. Is it possible to provide third-party software tools that allow you to manage and monitor the operation of processes in your mobile phone. Briefly describe them **Контрольні**

Yes, there are several third-party apps available on the Google Play Store that allow you to manage and monitor processes on an Android device. Here are a few examples:

Greenify: This app allows you to hibernate background apps and services to reduce battery consumption and improve performance. It also provides detailed information about app usage and resource consumption.

System Monitor: This app provides real-time information about CPU usage, memory usage, and network activity. It also allows you to monitor individual processes and view detailed system information.

ES Task Manager: This app provides an easy-to-use interface for managing running apps and services. It allows you to kill tasks, monitor system resources, and view detailed information about running processes.

Process Explorer: This app provides a detailed view of running processes and their resource usage. It allows you to kill tasks, set priorities, and view detailed system information.

SystemPanel 2: This app provides a comprehensive view of system performance, including CPU usage, memory usage, and network activity. It also allows you to manage running tasks and view detailed process information.

These apps can be useful for diagnosing performance issues, identifying resource-hungry apps, and improving battery life on your Android device. However, it's important to note that they may require elevated permissions to access system resources, so be sure to review their permissions carefully before installing them.

Conclusion: I get practical skills for working with the Bash command shell.