«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»**

Виконали

Студенти групи КСМ-03Б

Команда: Кучмій-Зікеєв

М.О, Зінченко І.О. та

Звєрьков А.Г.

Перевірив викладач

Сушанова В.С.

Київ 2022

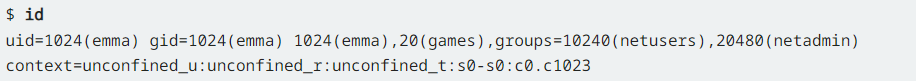
**Завдання для попередньої підготовки.**

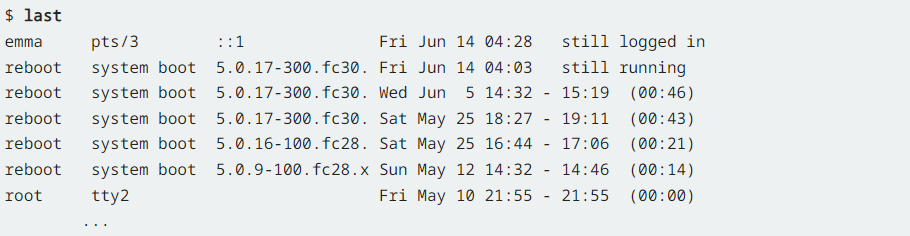
**Виконав студент Кучмій-Зікеєв М.**

**Хід роботи.**

**2. *Виконав студент Зінченко І.***

1) In fact, the proc folder is a very unusual folder. It does not exist on disk or even in RAM, as / tmp does. All subdirectories, files and information stored in them are generated by the kernel on the fly as soon as you request it. But everything works so transparently that you won't notice any difference between a normal file system and a proc if you open it with a file manager. Only the proc linux folder contains files of zero size. Also, each file will have a current creation date. For example, the file / proc / meminfo will contain different data each time it is opened, since memory usage is constantly fluctuating. With the help of such a system, developers adhere to the main concept of Unix - everything is a file. All files are editable with any editor, and all are in plain text format, but you will need superuser rights to analyze the entire directory. Almost all files are read-only, we can only get information from them. But there are also writable ones, in particular / proc / sys, with which you can configure various kernel parameters.

2) It is very easy to display the current user information on the command line using the id command. The output will vary depending on your login ID: 

Information about the last user login to the system is displayed using the last command. 

The information in the columns may vary, but a number of important entries in the list above mean:

User (emma) logged in via network (pseudo TTY pts/3) and is still logged in.

The current boot time is given along with the kernel. In the example above, about 25 minutes before the user logs in.

The superuser (root) logged into the system through a virtual console (TTY tty2) around mid-May.

3) The key combinations Ctrl + C, Ctrl + D and Ctrl + Z are often used in the terminal to exit the foreground program and transfer control to Bash. Ctrl + C terminates the process. It basically kills the ego. The result of the Ctrl + D action is the same. However, there is a difference between these two exit methods, and it lies in the internal mechanism. Pressing Ctrl + C causes the terminal to send a SIGINT signal to the process that is currently controlling it. When the foreground program receives a SIGINT signal, it must stop its work. Pressing Ctrl + D tells the terminal that it is necessary to register the so-called EOF (end of file), that is, the input stream is over. Bash interprets this as a request to exit the program. The Ctrl + Z key combination sends a signal to the process that tells it to stop. This means that the process remains in the system, but somehow freezes. Of course, it recedes into the background. With the help of the bg command, it is possible to start it again, while leaving it in the background mode. The fg command not only resumes a previously suspended process, but also brings it from the background to the foreground.

4) A background process is a computer process that runs off-stage (that is, in the background) and without user intervention. Typical tasks for these processes include logging, system monitoring, scheduling, and user notification. A background process is usually a child process created by a control process to handle a computing task. Once created, the child process will run independently, performing tasks independently of the control process, freeing the control process from performing that task. In Windows, a background process is either a computer program that does not create a user interface or a Windows service. The first ones are launched in the same way as any other program, for example, through the "Start" menu. Windows services, on the other hand, are started by Service Control Manager. In Windows Vista and later, they run in a separate session. There is no limit to how much system resources a system service or background process can use. Indeed, in the Windows Server family of Microsoft operating systems, background processes are expected to be the main consumers of system resources.

5) The jobs, bg (background) and fg (foreground) commands allow you to manage jobs running in the foreground and background:

jobs displays a list of background processes

fg number brings the process to the foreground

bg number puts the process in the background

6) You can use the ps command to list all background processes in Linux. Other Linux commands to obtain what processes are running in the background on Linux.

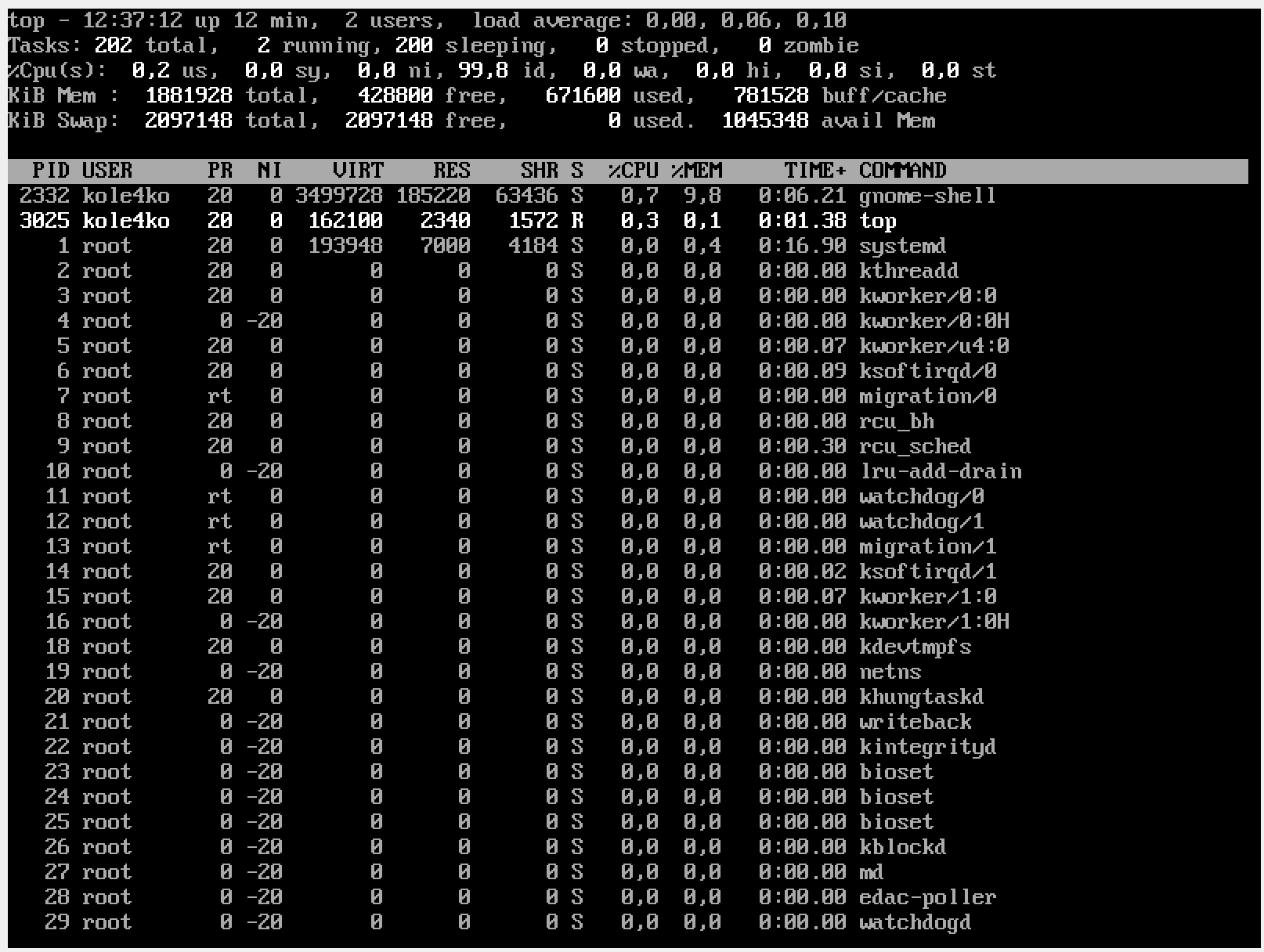
top command – Display your Linux server's resource usage and see the processes that are eating up most system resources such as memory, CPU, disk and more.

htop command – Just like a top command but with an improved user interface.

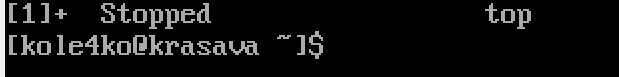
7) Click Start , then Settings > Privacy > Background Apps. Under Background apps, set Allow apps to run in the background to Off. To suspend the process running in the background, use: kill -STOP %job\_id

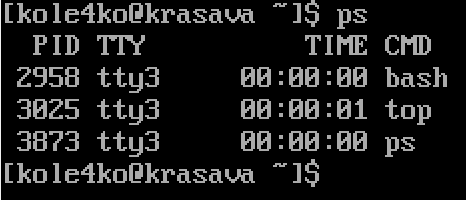
**3. *Виконав студент Звєрьков А.***

1) The *top* command shows most active commands and processes. The most active process is *systemd*, which is related with system running. Next one is top command. It is the second active process, `cause it refreshes every second to show relevant processes. Also there runs *nm-dispatcher* process, which stands for NetworkManager-dispatcher. This service runs user provided scripts upon certain changes in NetworkManager. Sometimes *gnome-shell* and *gnome-apps* appear.



2)

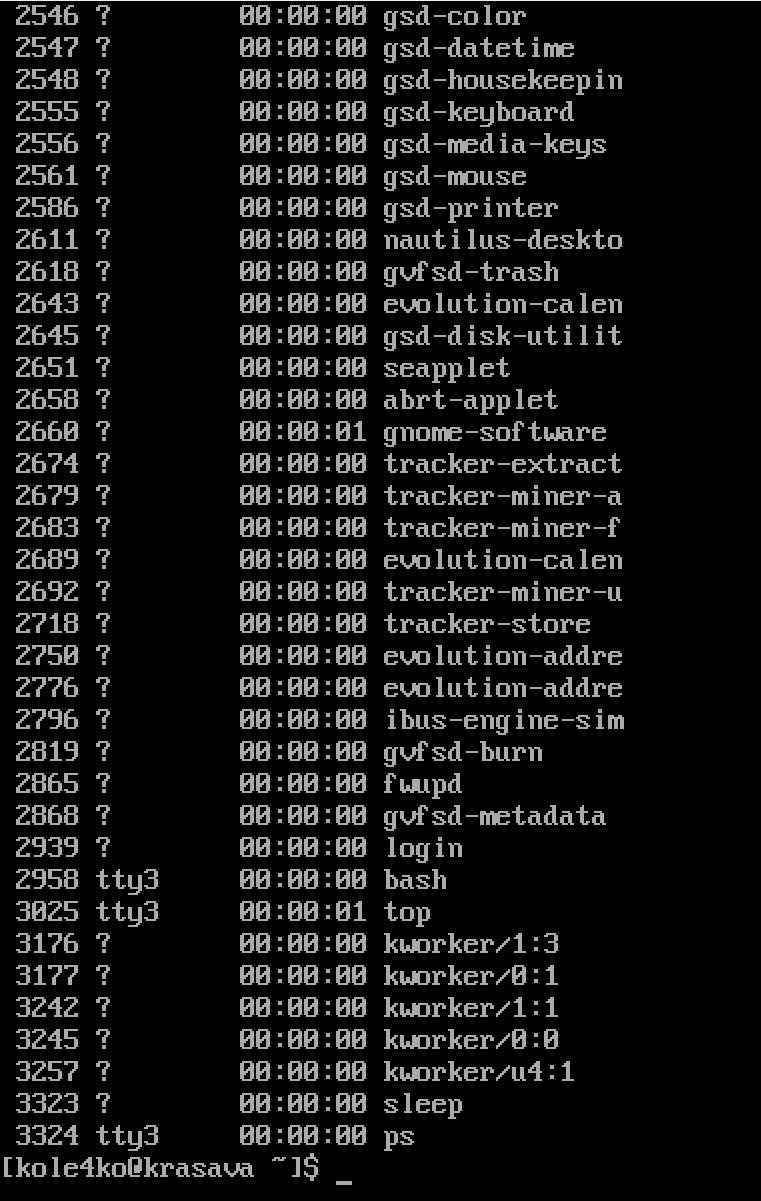


3)

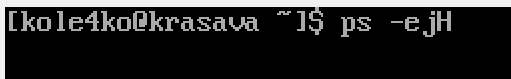
4)

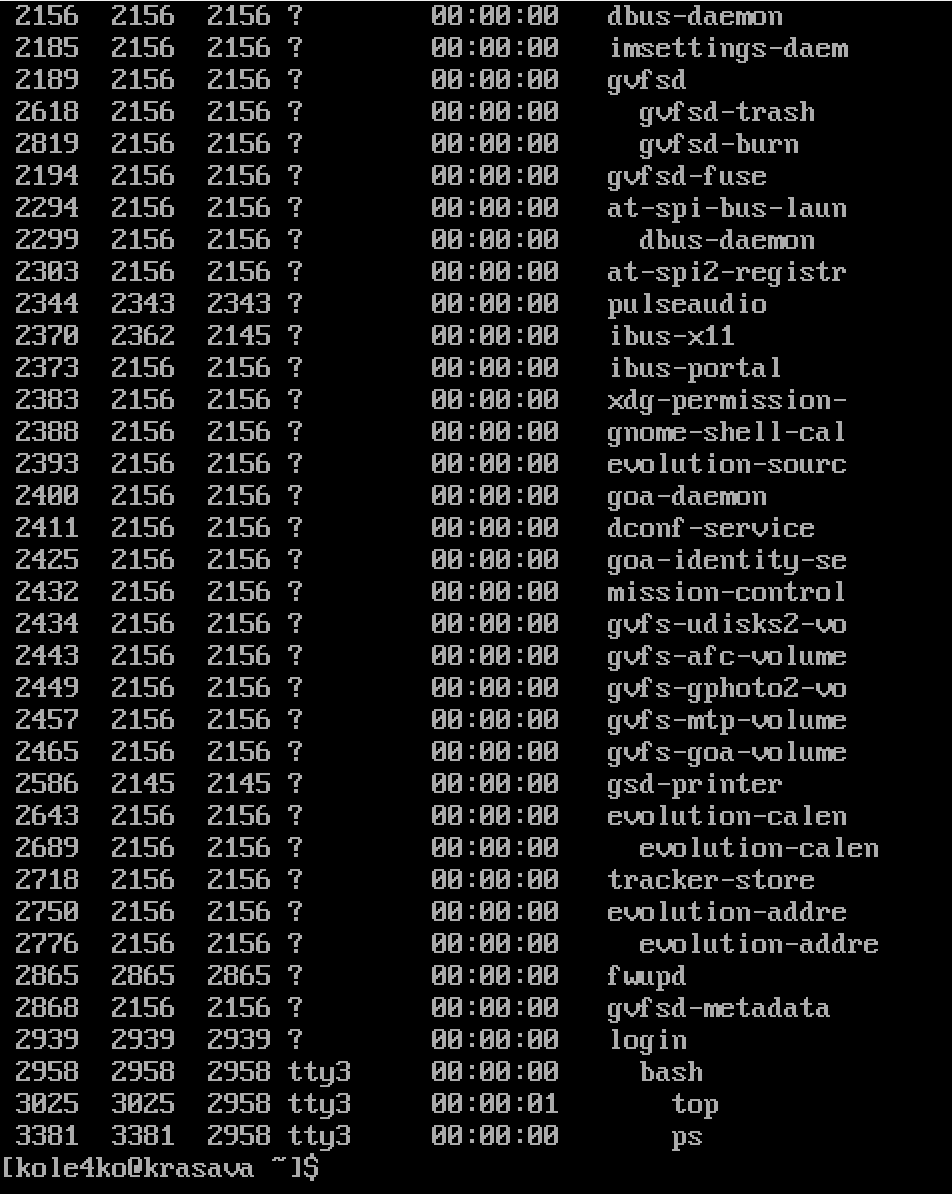


•

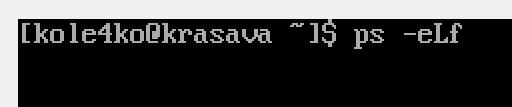


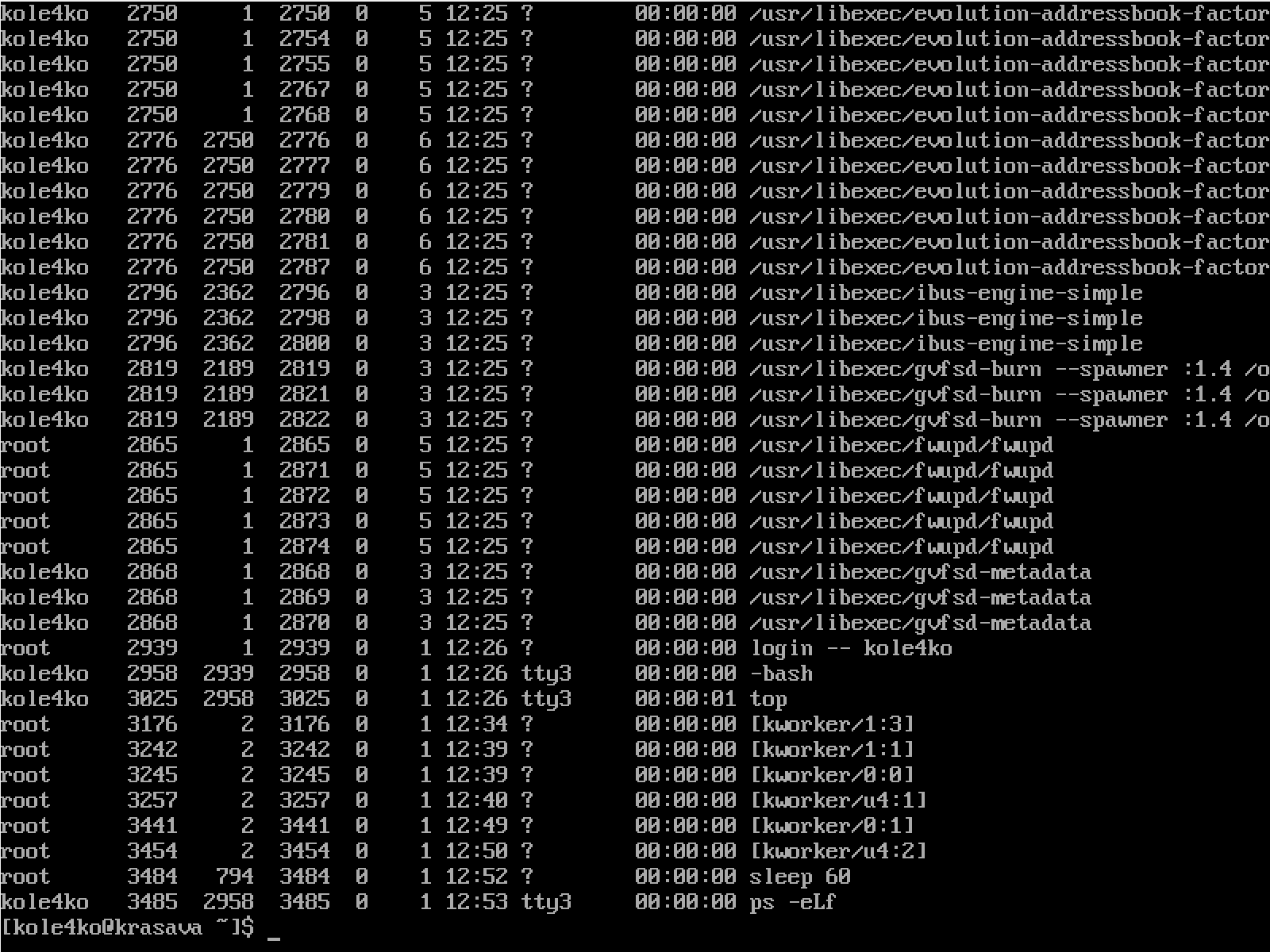
*ps -e* shows every single process running in the system

•

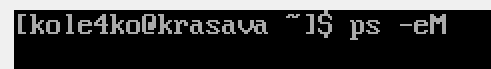


*ps -ejH* prints a process tree. We can see a slightly different structure compared to *ps -e* command.

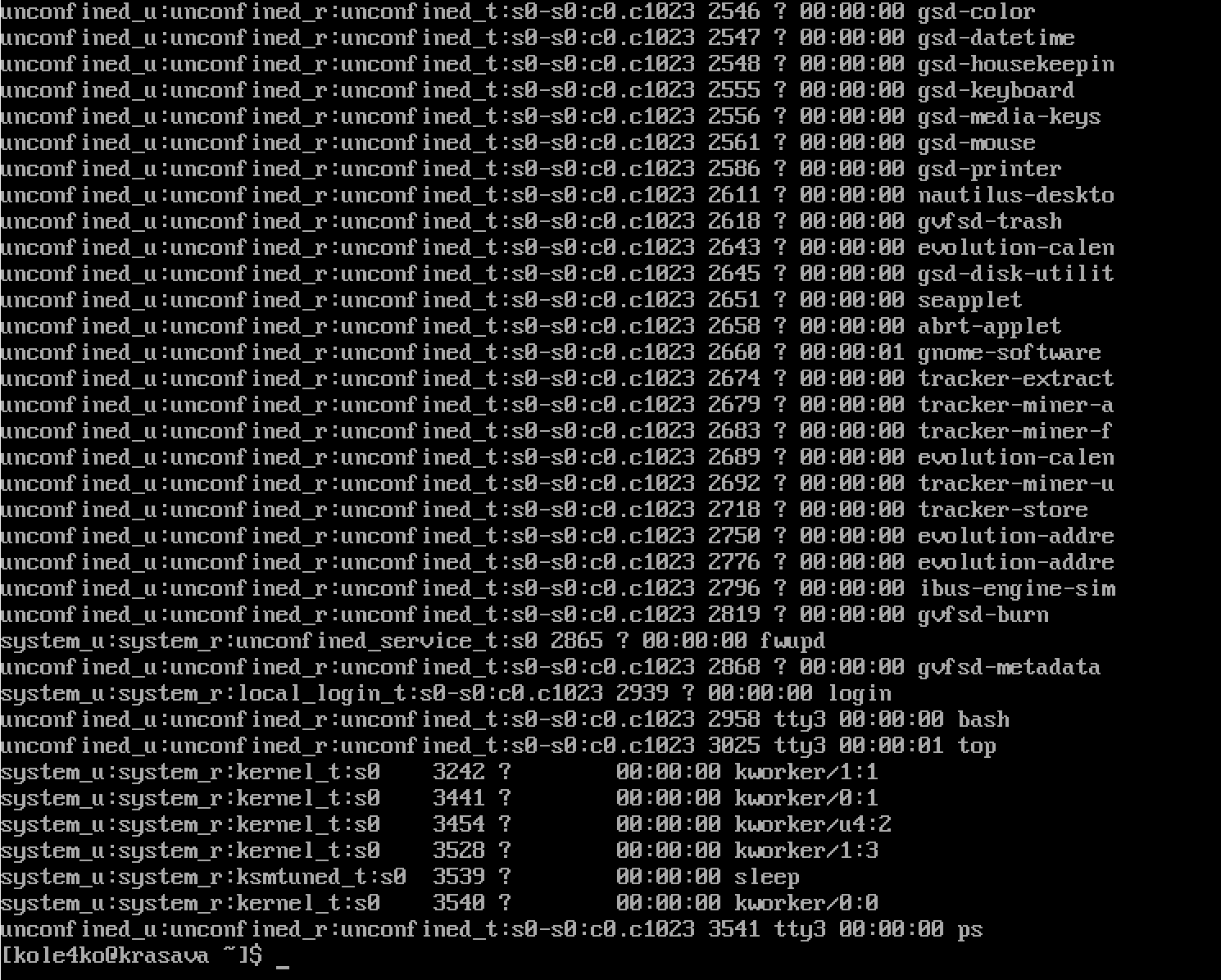
•

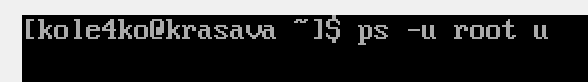


*ps -eLf* command shows information about threads.

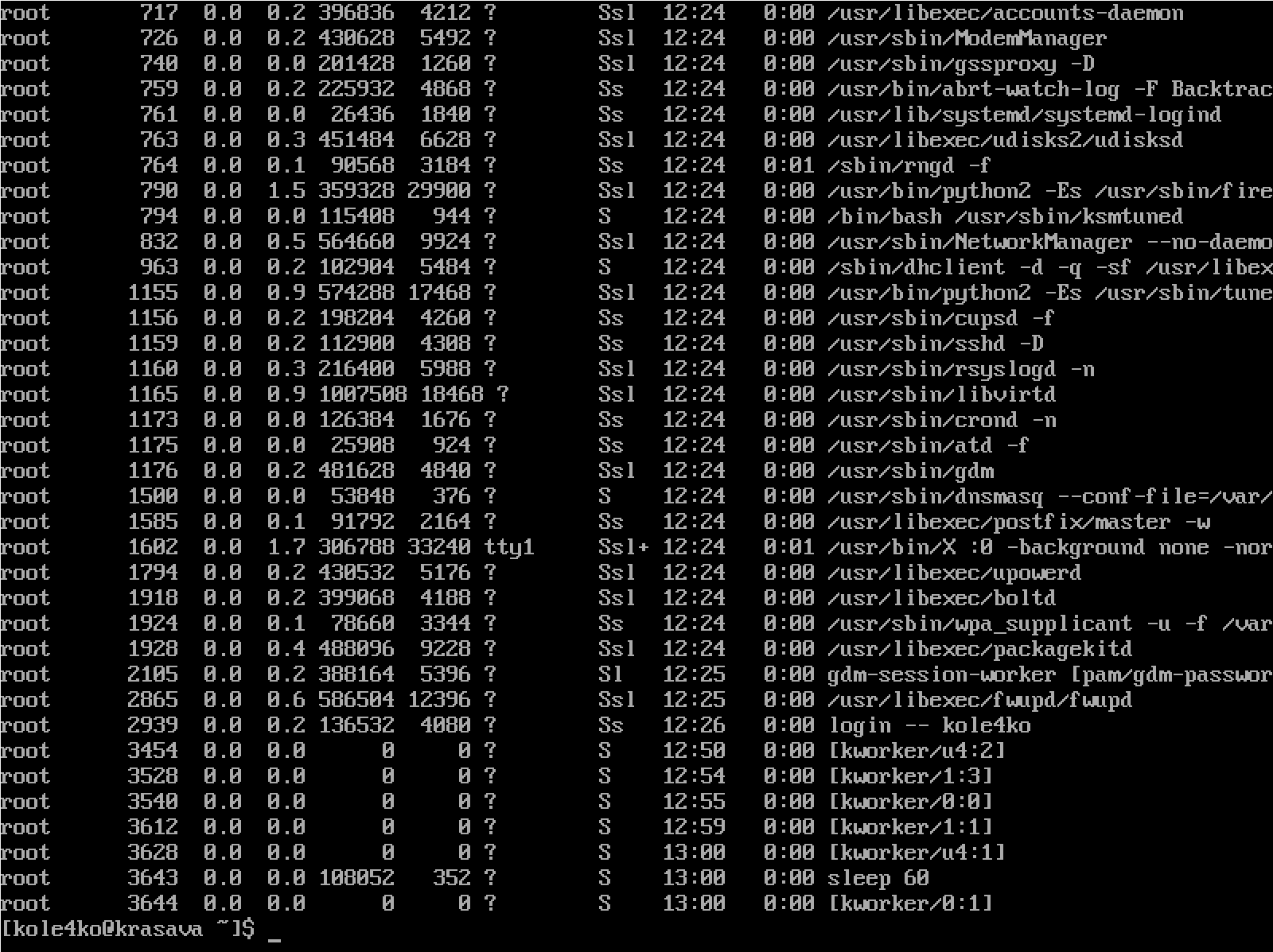
•

*ps -eM* command shows all the security information.



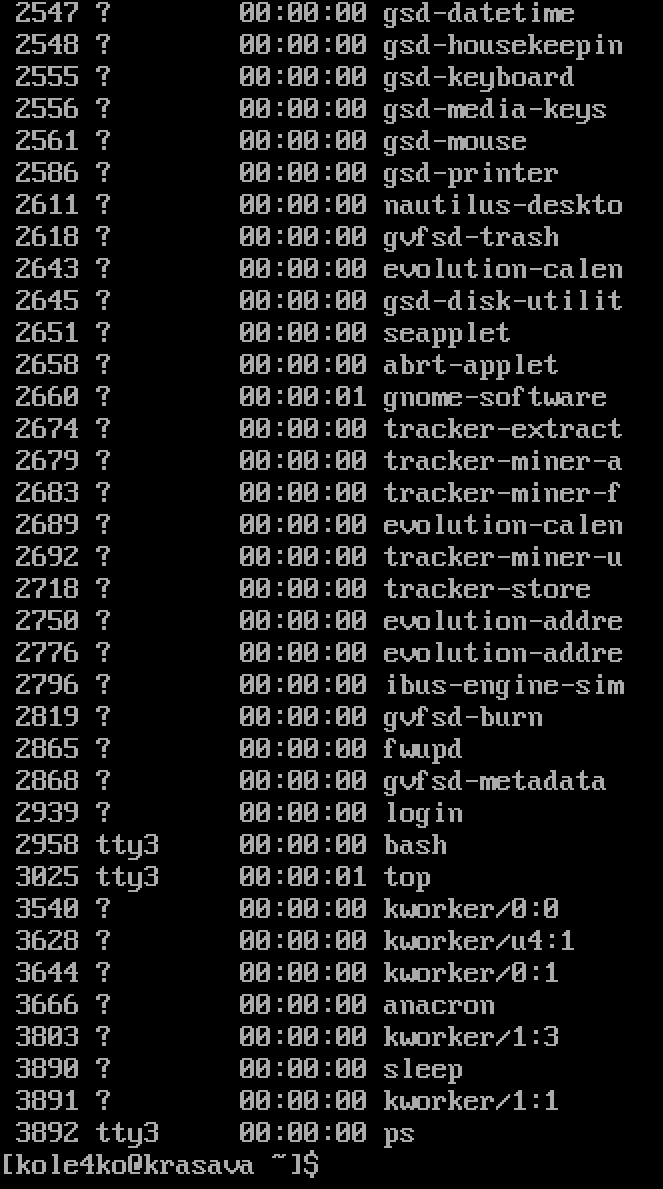


•

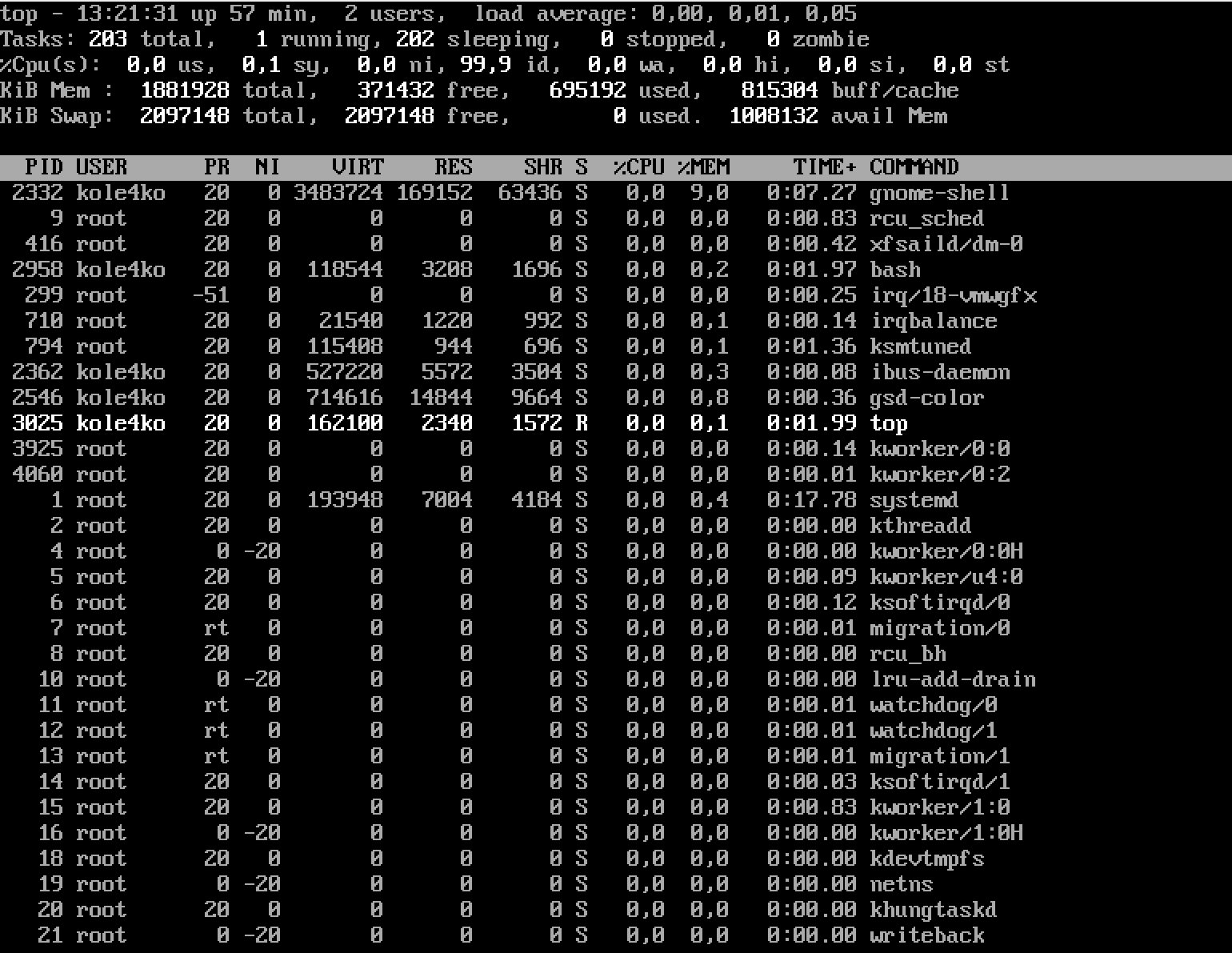


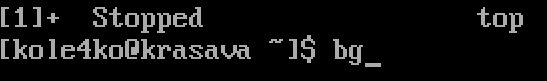
*ps -u root u* command shows every process running as root (real & effective ID) in user format

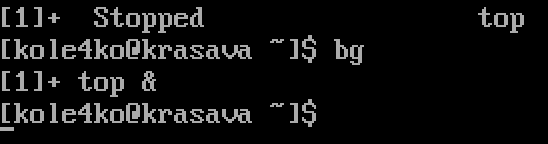
5) I’ve got many processes running in the background:

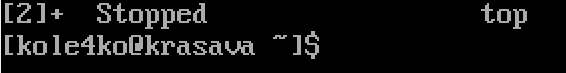


6)







7) 

**Відповіді на контрольні запитання**

***Виконав студент Зінченко І.***

1) User space programs in Linux cannot access the system kernel directly. But in order to obtain information from the kernel, several special directories were created with the help of which any program or user can obtain data about the state of the computer and the kernel. This is the proc and sys file system. From these folders you can get any information about your system. For example, how much swap memory is currently being used, how large is the processor cache, which kernel modules are loaded, how many disks or partitions are there, etc. All this can be obtained in plain text form from the proc linux folder. This guide will look at the proc file system, its structure, the purpose of the files, and where to find the information you need. But first a little theory. In fact, the proc folder is a very unusual folder. It does not exist on disk or even in RAM, as / tmp does. All subdirectories, files and information stored in them are generated by the kernel on the fly as soon as you request it. But everything works so transparently that you won't notice any difference between a normal file system and a proc if you open it with a file manager. Only the proc linux folder contains files of zero size. Also, each file will have a current creation date. For example, the file / proc / meminfo will contain different data each time it is opened, since memory usage is constantly fluctuating. With the help of such a system, developers adhere to the main concept of Unix - everything is a file. All files are editable with any editor, and all are in plain text format, but you will need superuser rights to analyze the entire directory. Almost all files are read-only, we can only get information from them. But there are also writable ones, in particular / proc / sys, with which you can configure various kernel parameters.

2) ps aux --sort=-%mem | head

Using top: when you open top, pressing m will sort processes based on memory usage. But this will not solve your problem, in Linux everything is either file or process. So the files you opened will eating the memory too. So this will not help. lsof will give you all opened files with the size of the file or the file offset in bytes. For interactive analysis, you can use htop and then press function key F6 to select the sort order, for example PERCENT\_MEM(percent memory usage), M\_RESIDENT(resident memory size, real memory used), M\_SIZE( virtual memory size) or M\_SHARE (shared memory size).

3) When a program is running, the kernel creates a process that helps store the details of the program's execution in system memory. When a program is executed, it becomes a process for the system. So we can say that a process is a program until it is executed. A process created by the kernel is known as a "parent process" and all processes derived from the parent process are called "child processes". A single process can consist of multiple child processes that have a unique PID but the same PPID. When working with the Linux system, it is necessary to know about the PPID of the running processes. A problem with a child process can affect other processes. In such cases, we may need to terminate the parent process. PPIDs of running processes can be determined in various ways. The simplest approach is the " ps " command and the " pstree " command. We have seen above how we can find out the PPID of a particular process using these two command tools.

4) top is mostly used interactively (try reading the help page or pressing "h" while top is running), while ps is for non-interactive use (scripting, extracting some information using shell pipelines, etc.). Save this answer.

5) top is a standard command-line utility preinstalled on Linux distributions and Unix-like operating systems. It is used to display information about the system and processes that consume the most CPU, as well as RAM usage. htop is an interactive process viewer and manager for Linux and Unix-like operating systems based on ncurses. If you take top and put it on steroids, you get htop. Htop is becoming increasingly popular among Linux users due to its advanced features and ease of use. In fact, this sparked the " top Vs htop " debate. Below are some features of htop that top doesn't have - why Linux users now prefer htop to its older counterpart top : It has a nicer text-graphics interface with color output. It is easy to use and highly customizable. Allows you to scroll the list of processes vertically and horizontally to see all processes and complete command lines. It also displays a process tree and comes with mouse support. Allows you to easily perform certain functions related to processes (kill, spawn, etc.) that can be performed without entering their PIDs. Htop is also much faster than top. Another important thing to share is that in the latest version of Ubuntu 18.04, the htop package is pre-installed, it is in the list of standard Bionic packages.

6) Processes are programs that run on a Mac. Processes can be applications, system applications used by macOS, or invisible background processes. With the help of System Monitoring, you can get information about these processes, including the amount of memory and processor time they use. In the System Monitor application on Mac, you can do the following. Obtaining information about the process. Select a process, then double-click it or click the information button in the System Monitor window (or use the Touch Bar). Process sorting. To sort the list, click the column header. Sorting column elements in reverse order. Click the arrow in the header of the selected column. View general information about all processes. Click "CPU" in the System Monitor window (or use the Touch Bar). Information, including the total number of open processes and threads, will appear in the lower area of ​​the window. Process search. Enter the name of the process or application in the search field.

7) Probably the most useful tool for examining and terminating processes is the Terminal, which is a program that provides access to the lower levels of the operating system and files of Mac OS X. The Terminal is a text-based tool that allows you to perform a variety of routine tasks, such as browsing directories, copying , move and delete files, and get detailed information about each running process, including: process identifier (PID) time spent running the path to the command or program file. The first step is to open Terminal either from the Applications -> Utilities folder or simply type Terminal into Spotlight. The terminal is always marked with the icon below. When it opens, you will see a standard terminal window as shown below. The first line shows the date and time of the last login. The second line is the command line, where you type the commands you want to run. The command line always starts with your computer name, followed by the local account name. The current directory ("working directory") when you open Terminal always defaults to your Home folder. Before we describe how to check and terminate the process on your computer, it's worth knowing a few basic terminal commands. To display the name of the current folder, type pwd To list the files in the current folder, type ls -l To change to another folder, type cd <folder name> Note that many commands in the terminal can take different parameters (sometimes called switches) that can change their effect. The easiest way to find out what command line options are available is to type the command in a terminal followed by -? for example ls -? Another useful apropos command. Type apropos <command> in a terminal window for a description of this command and its options.

8) Unfortunately, it is very difficult to install programs on the iOS operating system that are not available in the official app store. Therefore, I think that there are such programs, but they are not very suitable for ios, so they do not work correctly.

**Висновок**

Виконуючи лабораторну роботу №4 ми отримали практичні навички роботи з командною оболонкою Bash, ознайомились з базовими командами для управління процесами.