MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY

«KHARKIV POLYTECHNIC INSTITUTE»

Department of Software Engineering and Management Information Technologies

List of laboratory reports

discipline « Fundamentals of Operating System »

Executed by: Chukwu Irele omike

Student of group KH-201.8ei.1

Checked by:

Prof. S. L

Gloskakova Anna Alexandrouna

Kharkiv – 2019

MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY

«KHARKIV POLYTECHNIC INSTITUTE»

Department of Software Engineering and Management Information Technologies

Report from lab № 3

discipline « Fundamentals of Operating System »

Kharkiv

2019

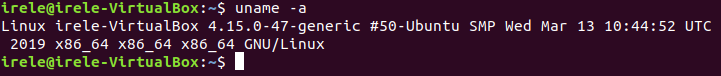
**Theme :**  **Process Management.**

**Objective:** learn is to determine which processes are running in the system.

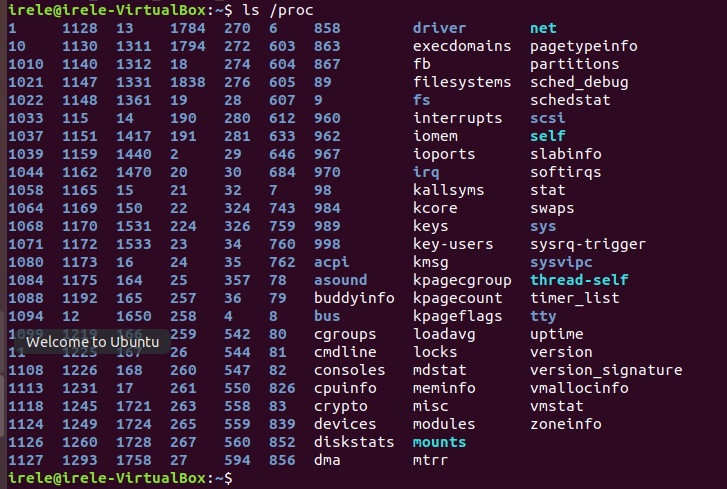
**Scenario 1: System information logging**

In this scenario, the commands that provide information about the system are studied.

Determine the name of the current UNIX system using the ***uname -a*** command.



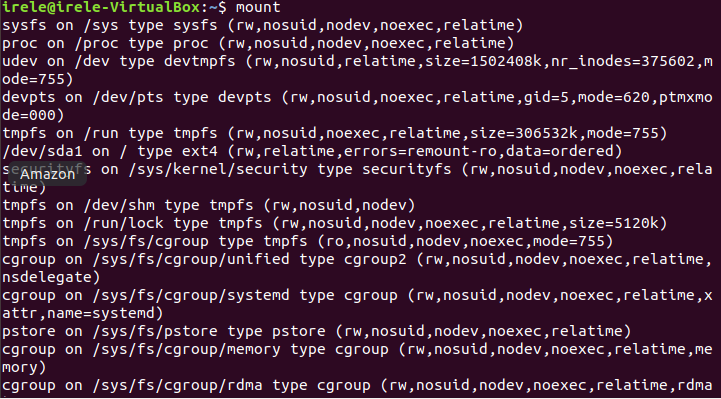
Display the contents of the proc directory with the ***ls*** /proc command:



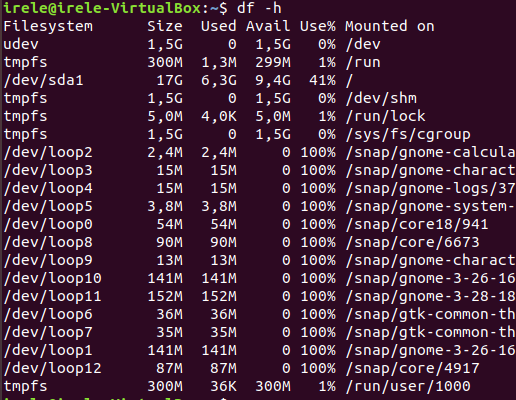
Display the current user sessions using the ***who*** command:



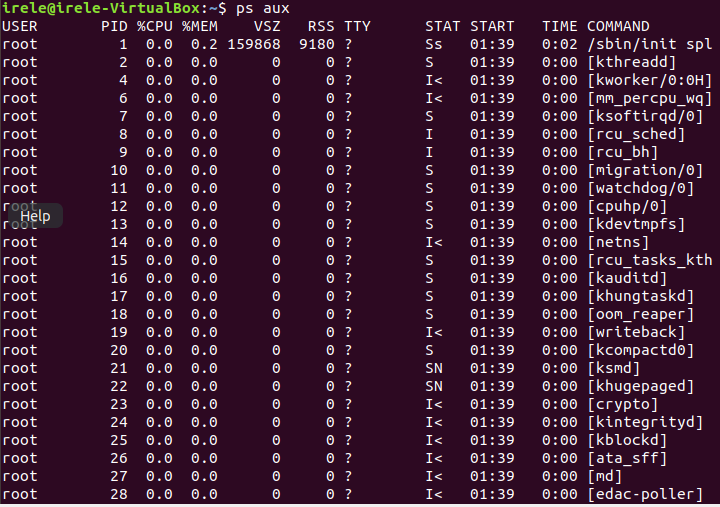
List all mounted devices using the mount command:



Display the load of mounted disks using the ***df -h*** command:



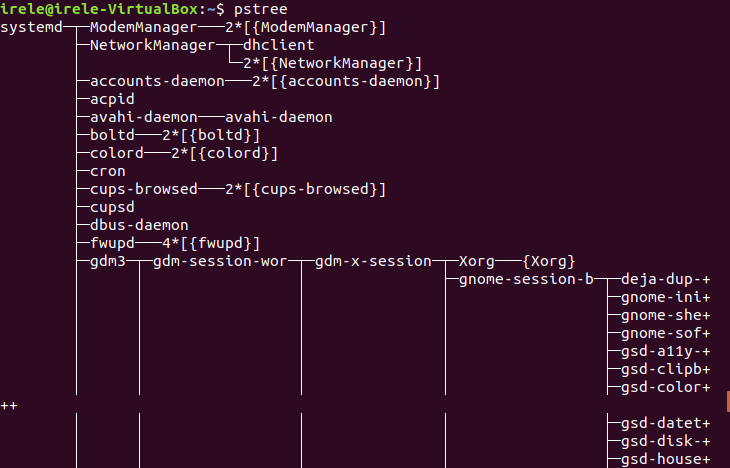
Display information about all running processes using the ***ps aux*** command.



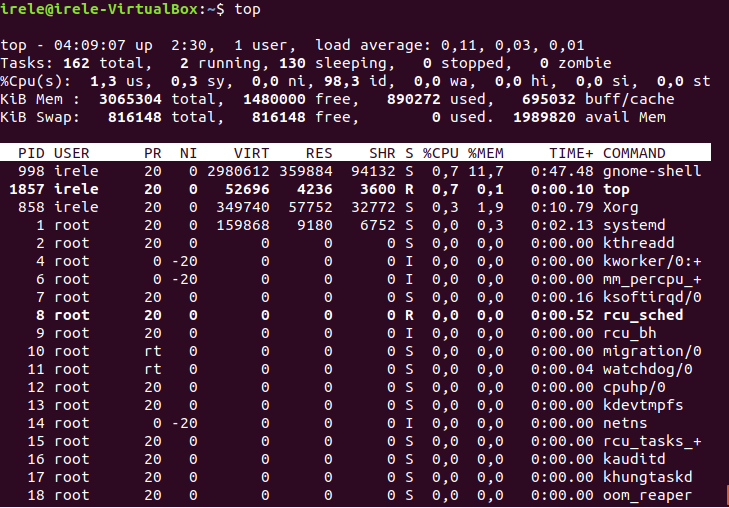
Leave in the output of the previous command only system processes using the command ***ps aux | grep -v user***:



Display the process hierarchy with the command ***pstree***



Consider the behavior of processes interactively with the help of the command ***top***.



**Scenario 2: Managing processes using signals**

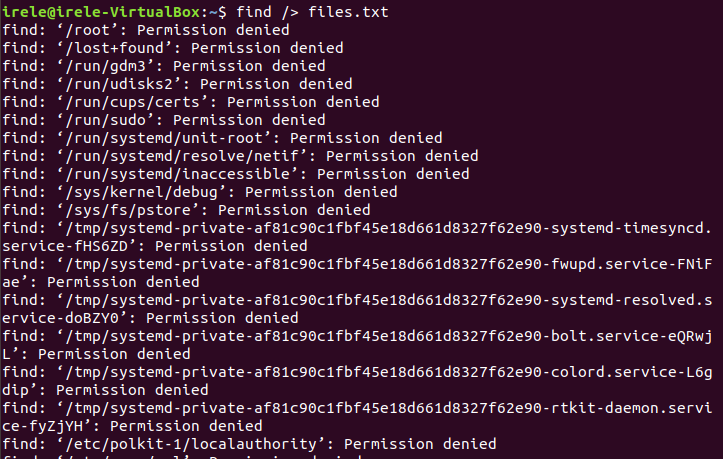
In this scenario, we study signals, control keys for transmitting signals to processes, commands for processes control.

*Initial conditions:* Command line after logging in.

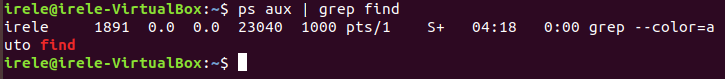
Run the ***yes*** command, which produces an infinite output of the ***y*** character on the screen. Interrupt it by pressing Ctrl-C.



Run the collection of information about all files of the system using the command ***find /> files.txt.***



Find the ***id*** of the process that was just started with the command ***ps aux |grep find*** launched in another terminal.



Send the end signal to this process with the command ***kill 8178***, specifying the process ID as the parameter.



**Scenario 3: Running tasks in the background**

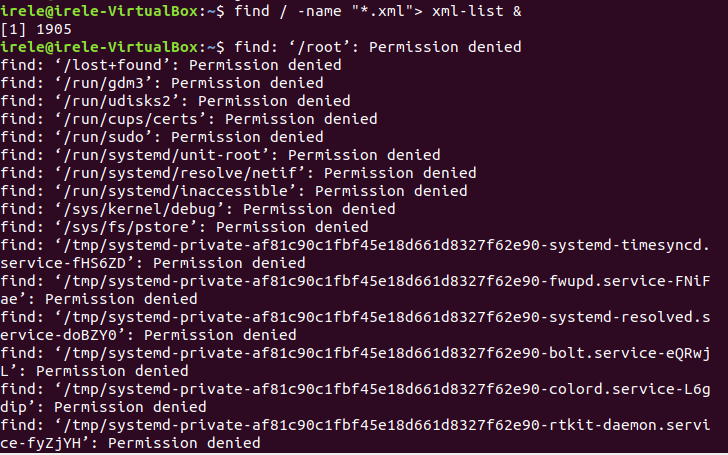
In this scenario, work is being done with the tasks of the command shell, running tasks in the background.

*Initial conditions:* Command line after logging in.

The current list of running tasks in the command shell can be viewed with the ***jobs*** command.



Commands can be started immediately in the background. It is necessary to add the symbol "***&***" (ampersand) to the end of the command line: ***find / -name "\* .xml"> xml-list &***

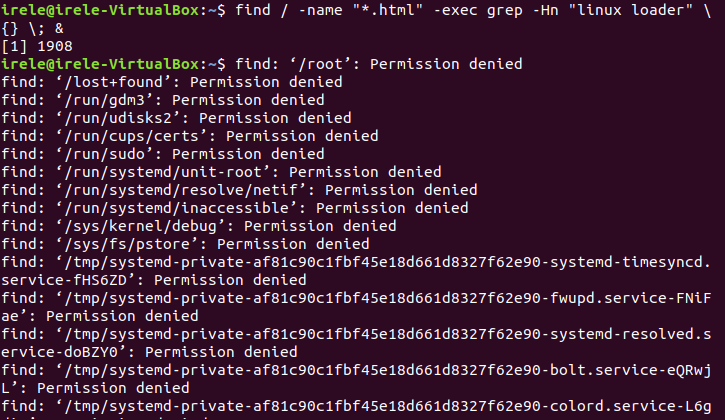


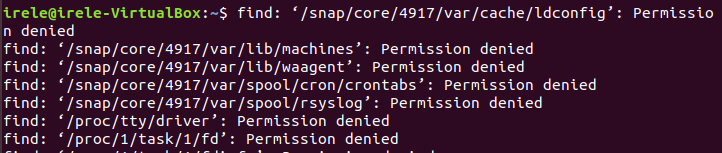
**Scenario 4: Running Daemons**

In this scenario, daemons are considered as processes that are not associated with any terminal.

*Initial conditions:* Command line after logging in.

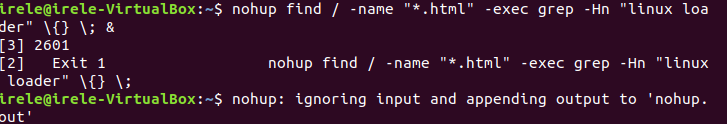
If you run the command in the background and exit the command shell: for example, ***find / -name "\*.html" -exec grep -Hn "linux loader" \{} \; &*** and ***exit***, then the running command will end with the ***SIGHUP*** signal.





To ensure that programs do not receive a ***SIGHUP*** signal, a special command ***nohup*** is used:

***nohup find / -name "\*.txt" -exec grep -Hn "linux loader" \{} \; &***



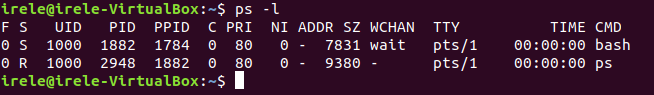
This program will exit correctly after the search is completed.

**Scenario 5: Changing the priorities of running programs**

This scenario examines the UNIX priority mechanism and commands for changing the priorities of the processes being started.

*Initial conditions:* Command line after logging in.

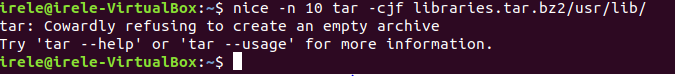
Each process in the system has its own priority level (in UNIX it is called "**nice**"), which can be seen with the ***ps -l*** command in the "**NI**" column.



By default, the priority of simple user processes is zero.

To start the process with a given priority, use the ***nice*** command. For example, let's start creating an archive with a lower priority:

***nice -n 10 tar -cjf libraries.tar.bz2 /usr/lib/***.



To change the priority of an already running program, use the command ***renice***. For example, to lower the priority of a previously running ***backup process tar -cjf libraries.tar.bz2/usr/lib &*** use the command

***renice +10 -p 3442:***



Common users can not increase the priority of processes, only to lower.

**Conclusion:**

In this laboratory the following was considered :

**Changing the priorities of running programs**

**Running Daemons**

**Running tasks in the background**

**Managing processes using signals**

**System information logging**