Lab 3

1. How many child processes are created upon execution of this program?

Three child processes will be created upon execution of this program.

2. When you start a browser, you will notice the browser process appear in the top display. What does it consume?

When you start the browser it consumes 46.7% CPU power.

3. How much memory is available in the system?

9737004 KB of memory is available in the system.

4. Which process consumes the most CPU?

The firefox browser is currently taking up the most CPU.

5. Which process has the most memory?

The process that uses the most memory is the gnome-+ process.

6. Could you please explain the following commands? apt-get, yum, wget, gzip, tar, rar

apt-get starts a command line process where you can install packages as well as update and remove these packages

yum is used for getting to install and deleting Red Hat Enterprises packages.

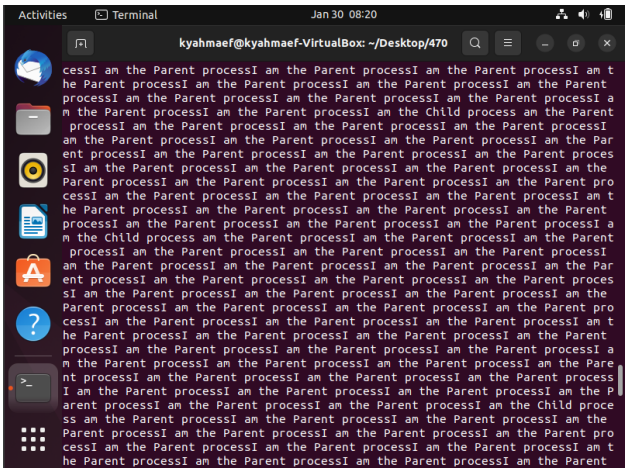
wget is used to retrieving content and files from various web servers.

The gzip command compresses files.

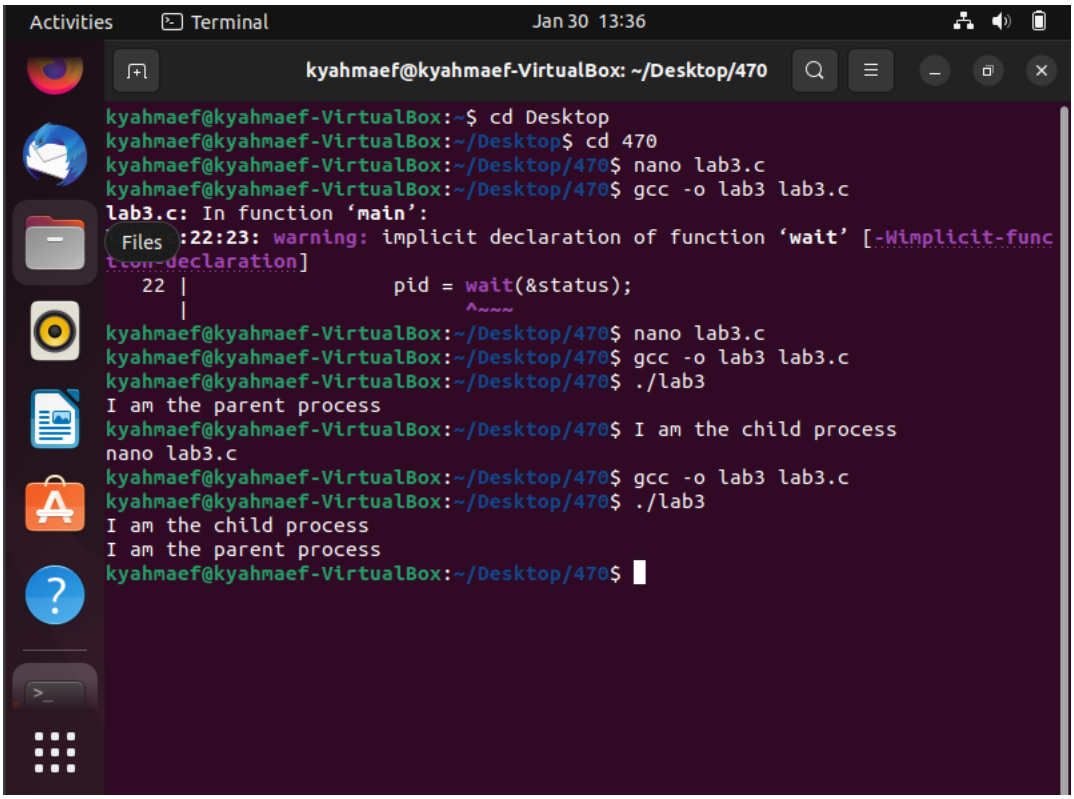
The tar command helps create, extract, and list items from the archive.

The rar command also is used for compressing and archiving files.

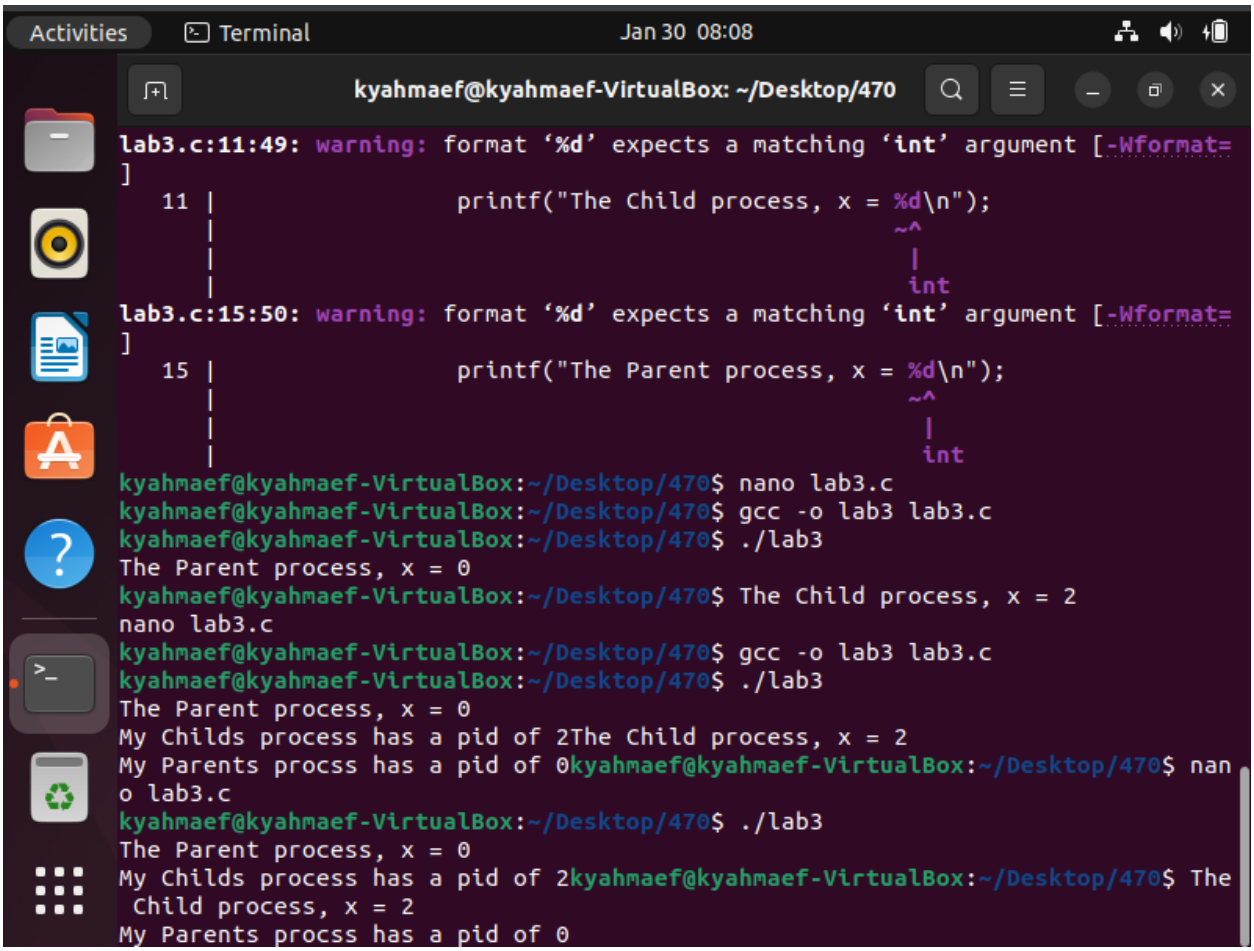
7. Write a program that will generate a child process. In a loop, the child process writes "I am a child process" 200 times and the parent process repeatedly prints "I am a parent process" in a loop.



8. Write a program that create a child process with the fork () system call. The parent process waits for the child process to finish before printing the contents of the current directory.



9. Write a program that create a child process with the fork () system call and print its PID. Following a fork () system call, both parent and child processes print their process type and PID. Additionally, the parent process prints the PID of its child, and the child process prints the PID of its parent.


Activities
Terminal
Jan 30 08:08
kyahmaef@kyahmaef-VirtualBox: ~/Desktop/470
lab3.c:11:49: warning: format ‘%d’ expects a matching ‘int’ argument [-Wformat=
]
11 | printf("The Child process, x = %d\n");
| ~^
| |
| int
lab3.c:15:50: warning: format ‘%d’ expects a matching ‘int’ argument [-Wformat=
]
15 | printf("The Parent process, x = %d\n");
| ~^
| |
| int
kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ nano lab3.c
kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ gcc -o lab3 lab3.c
kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ ./lab3
The Parent process, x = 0
kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ The Child process, x = 2
nano lab3.c
kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ gcc -o lab3 lab3.c
kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ ./lab3
The Parent process, x = 0
My Childs process has a pid of 2The Child process, x = 2
My Parents procss has a pid of 0kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ nan
o lab3.c
kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ ./lab3
The Parent process, x = 0
My Childs process has a pid of 2kyahmaef@kyahmaef-VirtualBox:~/Desktop/470$ The
Child process, x = 2
My Parents procss has a pid of 0