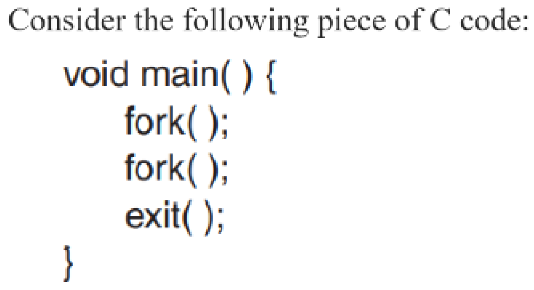
1. How many child processes are created upon execution of this program?



Consider the following piece of C code:

```
void main( ) {
     fork( );
     fork( );
     exit( );
}
```

The number of child processes created upon execution of this program is three.

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

It consumes both CPU and memory form the system.

3. How much memory is available in the system?

My system has roughly 20 gb of total memory available, with about 16 gb free

4. Which process consumes the most CPU?
5. Which process has the most memory?

For my system, gnome-shell is using the most CPU and memory.

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

Apt-get is a utility system for linux, apt stands for advanced package tool, its primary for installing, updating, and uninstalling programs.

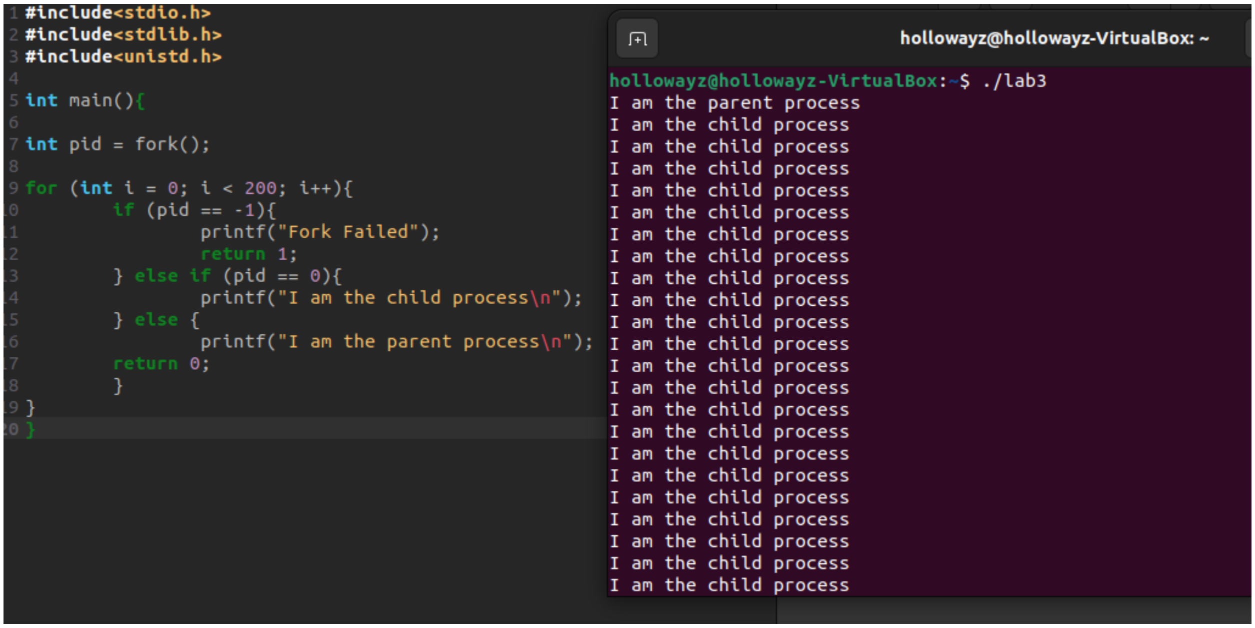
Yum stands for Yellowdog update modifier, it's similar to apt-get.

Wget lets you download files form the internet.

Tar is used to archive files in linux, these files can be compressed into many formats

Rar is like tar, although it only compresses files into a rar file, which can be highly compressed but is rather slow.

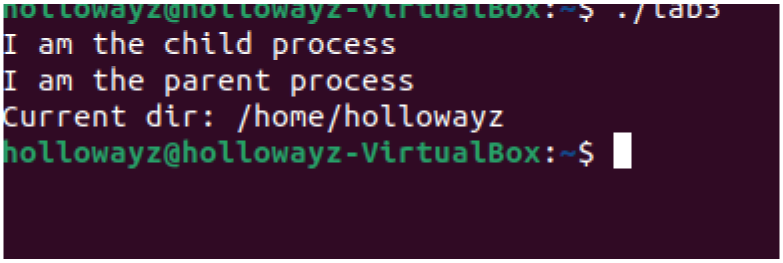
7. Write a program that will generate a child process. In a loop, the child process writes "I am a child process" 200 times and the parent process repeatedly prints "I am a parent process" in a loop.



```c
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main(){

int pid = fork();

for (int i = 0; i < 200; i++){
        if (pid == -1){
                printf("Fork Failed");
                return 1;
        } else if (pid == 0){
                printf("I am the child process\n");
        } else {
                printf("I am the parent process\n");
        return 0;
        }
}
}
```

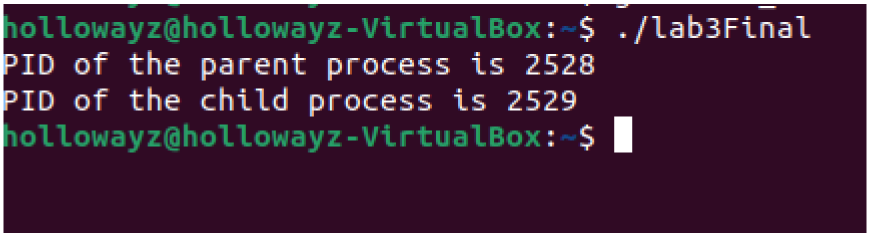
```
hollowayz@hollowayz-VirtualBox:~$ ./lab3
I am the parent process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
I am the child process
```

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.



```
hollowayz@hollowayz-VirtualBox:~$ ./lab3
I am the child process
I am the parent process
Current dir: /home/hollowayz
hollowayz@hollowayz-VirtualBox:~$ 
```

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.



```
hollowayz@hollowayz-VirtualBox:~$ ./lab3Final
PID of the parent process is 2528
PID of the child process is 2529
hollowayz@hollowayz-VirtualBox:~$ 
```