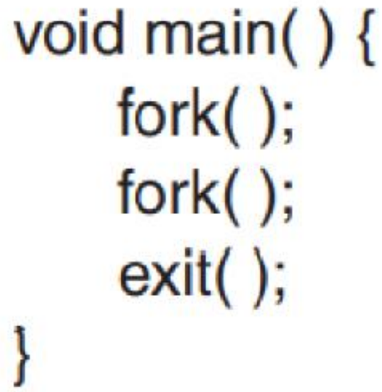
Nick Elias
2.1.2023
CS 470 – Lab 3

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



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

   3 child processes are created. (2^2 - 1)

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

   Firefox is listed as consuming between anywhere from 70% to 140% of CPU on startup. During normal operations it appears to use around 30% CPU.

3. How much memory is available in the system?

   My system has allocated 5187 MB of memory.

4. Which process consumes the most CPU?

   Firefox, if open, otherwise Gnome-+.

5. Which process has the most memory?

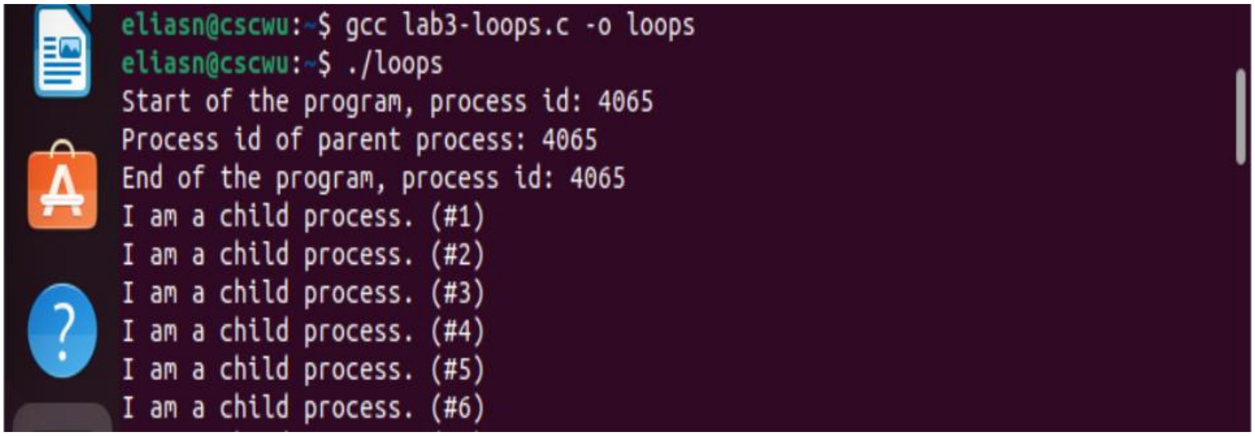
   Gnome-+

6. Explain apt-get, yum, wget, gzip, tar, and rar.

   Apt-get is a package handler function, basically a “back-end grabber” for other libraries.
   Yum is a package handler for Red Hat Enterprise packages.
   Wget is a web server file downloader.
   Gzip is a command that zips files into a zipped folder/directory.
   Tar is a command that compresses files into a tape archive.
   Rar is a command that compresses into a compression archive.

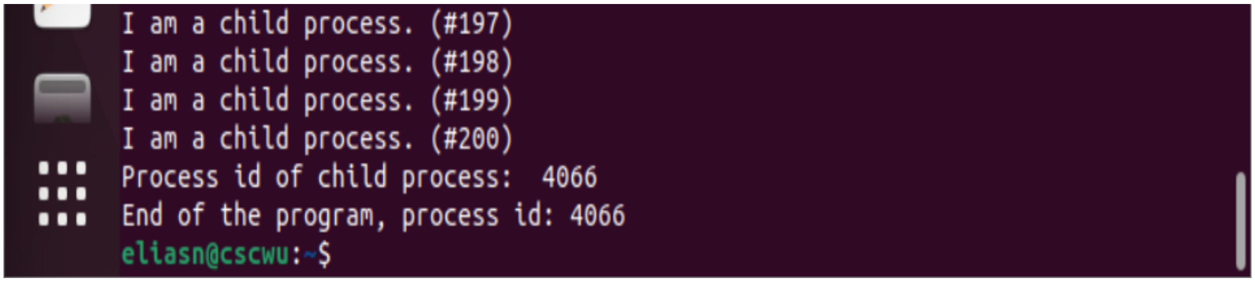
Nick Elias
2.1.2023
CS 470 – Lab 3

7. Write a program that generates a child process. Loop the creation so that the child process prints "I am a child process\n" 200 times, and the parent process writes "I am a parent process\n".



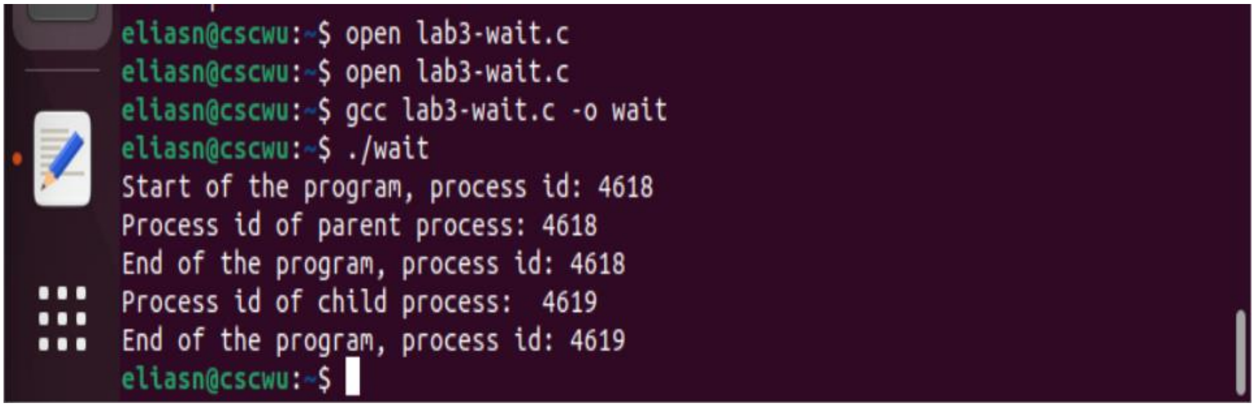
```
eliasn@cscwu:~$ gcc lab3-loops.c -o loops
eliasn@cscwu:~$ ./loops
Start of the program, process id: 4065
Process id of parent process: 4065
End of the program, process id: 4065
I am a child process. (#1)
I am a child process. (#2)
I am a child process. (#3)
I am a child process. (#4)
I am a child process. (#5)
I am a child process. (#6)
```

…



```
I am a child process. (#197)
I am a child process. (#198)
I am a child process. (#199)
I am a child process. (#200)
Process id of child process:  4066
End of the program, process id: 4066
eliasn@cscwu:~$
```

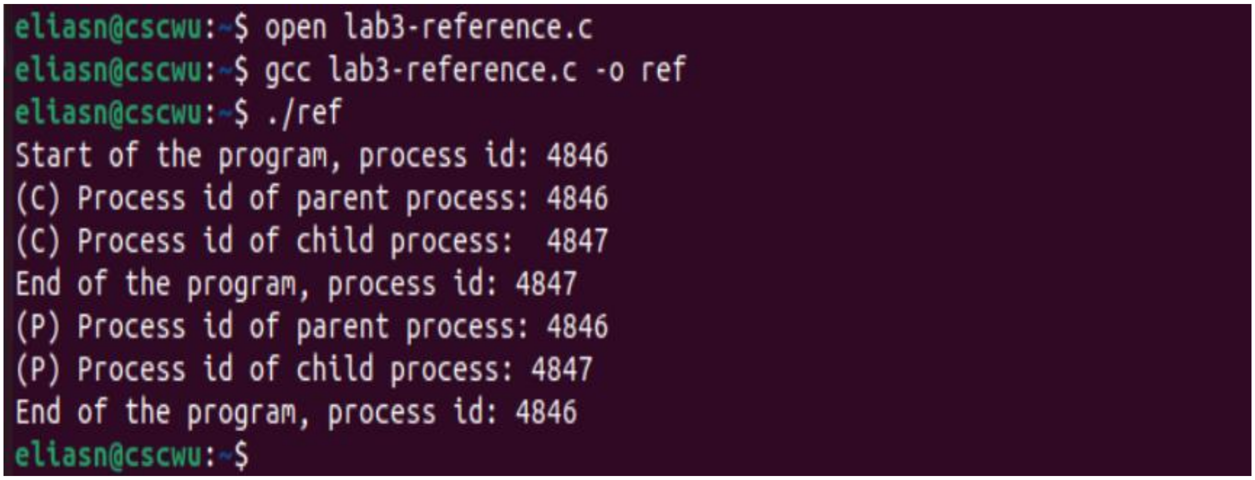
8. Write a program that uses fork() to create a child process. Let the parent process wait for the child process to finish before printing the contents of the current directory.



```
eliasn@cscwu:~$ open lab3-wait.c
eliasn@cscwu:~$ open lab3-wait.c
eliasn@cscwu:~$ gcc lab3-wait.c -o wait
eliasn@cscwu:~$ ./wait
Start of the program, process id: 4618
Process id of parent process: 4618
End of the program, process id: 4618
Process id of child process:  4619
End of the program, process id: 4619
eliasn@cscwu:~$
```

Nick Elias
2.1.2023
CS 470 – Lab 3

9. Write a program that uses fork() to create a child process and prints its PID. Following the fork(), both parent and child print their process type and PID. Additionally, the parent prints its child PID and the child prints its parent PID.



```
eliasn@cscwu:~$ open lab3-reference.c
eliasn@cscwu:~$ gcc lab3-reference.c -o ref
eliasn@cscwu:~$ ./ref
Start of the program, process id: 4846
(C) Process id of parent process: 4846
(C) Process id of child process:  4847
End of the program, process id: 4847
(P) Process id of parent process: 4846
(P) Process id of child process: 4847
End of the program, process id: 4846
eliasn@cscwu:~$
```