Lab3-CS 470: (Operating Systems)
Winter 2023
Instructor: Solmaz Seyed Monir

Lab3-Instructions for Running a C Program in Ubuntu and System Calls

Scoring: 20 pts

Please complete Tutorial 5 before starting the lab assignment.
This assignment is an individual submission.

Activity Results: This lab teaches you the following topics:

- ❖ Instructions for Running a C Program in Ubuntu.
- ❖ Learn about Linux system calls and how they are used in processes.

The fork system call creates a new process by duplicating the calling process. The new process, called the child, is an exact copy of the calling process, called the parent. The fork call returns a value of 0 in the child process and the process ID of the child process in the parent process. This allows the parent and child processes to differentiate between themselves and to communicate with each other if necessary.
Consider the following piece of C code:

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

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

There are several process management tools in Linux that can be used to manage and monitor processes on a system. Some of the most commonly used tools include:

1. ps: Displays information about currently running processes.
2. top: Displays real-time information about running processes and system usage.
3. htop: An interactive version of top, that provides a more user-friendly interface.
4. kill: Sends a signal to a process to terminate it.

5. pkill: Sends a signal to all processes with a specific name.
6. pgrep: Displays the process IDs of processes with a specific name.
7. pstree: Displays a tree-like view of the running processes.
8. lsof: Lists open files and the processes that have them open.
9. pidof: Displays the process ID of a running program.
10. nice and renice: Modifies the priority of a running process.

These tools provide a variety of information and options to manage, monitor and control processes on a Linux system.

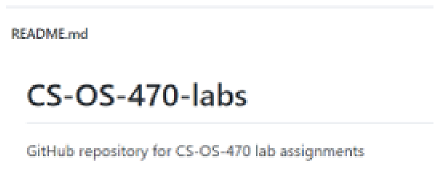
When you run the top command in the terminal, it will display real-time information about running processes and system usage, such as CPU usage, memory usage, and process information.

2. When you start a browser, you will notice the browser process appear in the top display. What does it consume?
3. How much memory is available in the system?
4. Which process consumes the most CPU?
5. Which process has the most memory?
6. Could you please explain the following commands?
7. apt-get, yum, wget, gzip, tar, rar
8. 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.
9. 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.
10. 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.

How do I submit my assignment?

1. **Answer all questions** in the word document, then take screenshots of your terminal and paste them into a word document, then save it as a pdf.

2. Make a folder in your host operating system called OS470 lab-assignments and save your lab 3 in that folder.
3. Upload your pdf file to your repository "CS-OS-470-labs" from Tutorial 1 is in our repository for this class.



README.md

CS-OS-470-labs

GitHub repository for CS-OS-470 lab assignments

4. Upload a URL repository to the canvas.