
# ECE437/CS481 Fall 2016
# Programming Assignment #01

**Due**: Wednesday, 09/07/2016, by 11:00pm

UNIX/LINUX systems usually include commands that allow you to trace system calls made by a process. Under Linux, this command is strace. For example, to trace all the system calls made during execution of "ls" you would type "strace ls". If the display is too long to fit your screen, you can use "strace –o out4ls ls" to write your output into file out4ls. Check "man strace" (traditional UNIX manual pages) or "info strace" (GNU style manual pages) in your Linux system for details. It is a good tool for learning, either as a light-weight debugger, or as a primitive profiler.

1. (25%) **Use strace with a small program**. Create your own program in C (named as checkfile.c) as follows:

```
#include <stdio.h>

int main(void) {
FILE *fd;
If ((fd=fopen("OSclass","rw"))==NULL)
   printf("\n Program Failed, figure out why...\n");
else {
   fclose(fd);
   printf("\n Program ran successfully \n");
   }
}
```

   a) Before creating the file OSclass (no such file in your working directory), run "strace ./checkfile". From your output, list five major system calls made from your program and explain why the program failed.

   b) Now, create a file OSclass (a dummy file) and run "strace ./checkfile" again. From your output, explain the major differences from the output you got in part a).

      1) Is "fopen" a system call? If not, which system call does it mainly correlate with?
      2) Is"printf" a system call? If not,which system call does it mainly correlate with?

2. (25%) **Use *strace* with a Linux utility command.** Run "strace" with a command of your choice (e.g. pwd, date, ps, jobs and so on), capture the output, and then pick at least four different system calls and briefly describe their function and input/output parameters.

3. (25%) **Use *strace* with the Linux utility command "ls".** Run "strace –c ls", capture the output, and identify five of the most frequently used system calls for "ls". Next, for the identified system calls, use "strace –T –e trace=xxxx1,xxx2,xxx3,xxx4,xxx5 ls" to report their average execution time in microseconds, as well as their min and max execution time. (Note that xxxx1 corresponds to the name of the most used system call, xxx2, the second, and so on.)

4. (25%) **Use *strace* and *ltrace* with the Linux utility command "ls".** Command ltrace is another tracing tool used for tracing the library function calls. Use both strace and ltrace to Linux command ls to report which library functions and system calls are used to
    a. a) Open the current directory
    b. b) Get the list of directory entries
    c. c) Print the output to your screen

You are going to submit a single PDF or MS WORD file to answer all questions asked. Please show your name in every page of your submission, and name your file as "YourLastName_YourFirstNameInitial_PA1"