OS Assignment 2
System Calls

Here is your first programming assignment.

The program you write should use only system calls (read, write etc...). Use of any library calls like (printf, scanf..) will directly fetch you zero marks. To check whether you are using a system call or not you can simply check if it is present in man 2 or not.
Eg .
$ man 2 printf
No manual entry for printf in section 2

This implies printf is not a system call.

Use the man pages extensively.

Q1.

The task is to write your own ls program. You will be given directory via Command Line argument and are required to print directory's content to stdout. You can expect the input directory name to be accompanied by these 3 option(-l,-a,-h) in any combination in which case output should reflect the effect of these options.

Eg. Test cases
$ ./myls dirname
dircontent

$ ./myls dirname -lh
dircontent with effects of these options

$./myls -lah dirname

Notes:

1. dirname can be absolute or relative path, it can also be a filename, in which case you output that file only.
2. A symlink should display the file it points to like in the case of ls -l. (try out in /proc to see what it means, if in doubt.)
3. Options will be given together (like -lah) and not separately (like -a -l) to make parsing easier.
4. Extra marks if you can color the output on stdout according to file type.
5. Options can be present before or after the dirname
6. Anyone caught using exec to run ls will be awarded zero marks straight.
7. Anyone using the “system()” library call in stdlib.h will be awarded straight zero.

Q2.

Write a program which outputs its running status (sleep, stopped ..etc), the parent ID of that process, and number of threads a program is using. Input will be a pid through command line. You will need to get this information from the file /proc/<pid>/status.

Eg:
$ ./foo <pid>

If there is no process which has this pid, then your program should output an error message conveying the same.

Write two codes:

- Library calls allowed
- Only system calls

Run ltrace and strace on these programs and store the output in files using the -f option. Examine the files and see how the system calls and library calls are executed.

Submissions:

1. Deliverables:
    1. All the c codes. (myls.c, status_library.c, status_system.c)
    2. Output of ltrace and strace (library.txt, system.txt)

2. Deadline : 4th September 2014 9PM (no extension requests entertained)

OS TAs