CS347(M): Operating Systems
Autumn 2017
Dept. of Computer Science & Engineering, IIT Bombay

Assignment – 1

- *Due Date: 11ᵗʰ August, 2017*
- *Question-4 will be anounced later*

`proc` file system

/proc file system is a mechanism provided, so that kernel can send information to processes. This is an interface provided to the user, to interact with the kernel and get the required information about processes running on the system. The /proc file system is nicely documented in the proc man page. You can access this document by running the command man proc on a linux system. Understand systemwide proc files such as memory, cpuinfo, etc and process related files such as status, stat, limits, etc.

Exercises

1. Collect the following basic information about your machine using proc file system. Write the required commands and answer the following questions:

- **CPU:**
  - How many CPU cores does the machine have?
  - What is the frequency of CPU?
- **Memory:**
  - How much memory does your system have?
  - How memory is divided and what does it mean?
- **Process Centric:**
  - How many context switches has the system performed since bootup?
  - How much memory is free and available?

To Submit: Write the answer in a file named proc.pdf.

2. Shell Script

A. Write a script "copy.sh" which will create 'N' copies of a file. Filename and N are given as arguments to the script. Copies should be created in a folder named 'files'. Copied file should be named as Filename1, Filename2,... FilesnameN. It should output the time(in seconds) required to create the copies.

Sample Command:
./copy.sh README 5000

Sample Output:
17

Hint: For experiment purpose use README file. Keep README file, copy.sh and a folder named 'files' in same folder and execute the above sample command. (file directory and README file is provided in zip file of assignment)

B. Try to find the 'pid' of the process created by copy.sh. (This will not be evaluated)

Hint: Use 'lsof' command to find pid. To fetch required pid, pipe the output to 'grep' command.

To Submit: Write the script in copy.sh file for Part-A

3. The OS command interpreter is the program that people interact with in order to launch and control programs. On UNIX systems, the command interpreter is often called **shell**: a user-level program that gives people a command-line interface to launching, suspending, and killing other programs. sh, ksh, csh, tcsh, bash, ... are all examples of UNIX shells.

You required to create a shell by writing a program in C language. It should be able to execute all UNIX commands (which are present in /bin directory). Use fork(), wait(), exec() functions.

1. Print out a prompt "CS347(M)$ "
2. Read a line from the user;
3. Parse the line into the program name and an array of parameters;
4. Use the fork() system call to spawn a new child process;
   - The child process then uses the exec() system call (or one of its variants) to launch the specified program;
   - The parent process (the shell) uses the wait() system call (or one of its variants) to wait for the child to terminate;
5. Once the child (the launched program) finishes, the shell repeats the loop by jumping to 1.
6. "exit" command should terminate the shell.
7. "name" command should print your Name.
8. "roll" command should print your Roll Number.

To Submit: C program written in file named "shell.c"

Submission:
This assignment is to be done in groups of 3 students. Only 1 of the member should submit the assignment on moodle. Include a README file having name and roll numbers of the 3 group members.
Zip all the files: a) pro.pdf, b) copy.sh, c) shell.c, d) README in a tar.gz file.
Name the zipped file as: *<your_roll_number>_assignment1.tar.gz*
Upload it on Moodle.