Indian Statistical Institute

Operating Systems

**Lab 1: Processes**

> Make sure man pages for system calls are installed. If you are using a Fedora based distribution, you will need to install `man-pages`; for an Ubuntu based distribution, please install `manpages-dev`.

**Man pages:** `fork(2)`, `execve(2)`, `exit(2)`, `wait(2)`, `sleep(3)`, `pstree(1)`, `env(1)`, `getenv(3)`, `setenv(3)`.

1. (a) Run the `env` command.
   (b) Download the programs available via
      (i) `https://www.dropbox.com/s/z3xl9zaqyfv1cu5/env.c?dl=1`
      (ii) `https://www.dropbox.com/s/93fsnvfb9sdb9ga/fork.c?dl=1`

   (these links are also available via the course page). Run them, and explain the output of each program.

2. Run `pstree` and study the output. In particular, try the `-T` and `-h` options, and specify your username, as in the following example.

   ```
   pstree -T -h mandar
   ```

3. Write a (single) C program that does the following:
   (a) creates a child process;
   (b) parent process prints the child's process ID and exits;
   (c) child executes `/bin/ls` and exits.

4. Write a program to fork 10 children. Each process should print its own pid and exit.

5. Write a program that prints a) `"Hello World "`, and b) `"Hello World\n"` and then forks. Also, try using `fprintf(stderr, ...)` instead of `printf`. Explain the output.

   See `http://stackoverflow.com/questions/1716296/why-does-printf-not-flush-after-the-call-unless-a-newline-is-in-the-format-strin` for the answer.

6. Consider a process that forks $n$ children. The children exit one by one, but not in the order in which they were created. The parent eventually calls `wait()` for each of these children. In what order are the zombie children now cleaned up: based on the order in which they were created, or based on the order in which they terminate? Write a program to empirically find the answer to this question.

7. Download, study and understand the source for a very rudimentary shell that is available from the course page.