

Purpose: To learn Unix programming using system calls for process creation and manipulation. Except for printing, use only basic Unix process related system calls: fork, exec (any of 6 versions), exit, and wait, plus sleep, kill and getpid, and file system calls from Lab1.

Assignment: Write six C (or C++) programs AA.c, BB.c, CC.c, DD.c, EE.c and FF.c, with executable codes in files AA, BB, CC, DD, EE and FF, that function as follows:

1. First you run you code AA (as a process, call it Process A). Process A creates three processes: a process with code BB (call it Process B), a process with code CC (call it Process C1) and another process with code CC (call it C1a). Then Process A waits one of its children to terminate.
2. Process B creates file "XYZ.txt" and writes the last 200 characters from file /usr/class/cis660/x.y into it, and then it terminates.
3. **Process C1 enters infinite loop (include "sleep(1)" in each loop iteration).**
4. Process C1a enters infinite loop.
5. When process A continues, it creates three new processes: a process with code DD, call it Process D, a process with code CC, call it Process C2, and a process with code EE, call it Process E. Then Process A waits for Process D to terminate.
6. Process E reads file XYZ.txt in reverse order, character by character, and each character is printed as read. Then Process E deletes the file and terminates.
7. Process D creates a process with code CC, call it Process C3, and then Process D waits for 4 seconds (use system call "sleep").
8. When Process D continues, it creates new process which executes Unix command: ps -u user-name
user-name = your user name
9. Process D waits for ps process to terminate, then it kills its second child (Process C3) and then it exits.
10. Process A now continues. It kills its remaining three child processes (C1, C1a and C2) and sleeps for 3 sec.
11. When is awoken, Process A prompts for an arbitrary string of characters and you are to provides it from a keyboard.
12. Then, Process A creates new process from code FF (call it Process F). Then Process A terminates.
13. In the exec system call that loads code FF into process F, two strings of characters one from step 11 and one containing your name are passed to a new code. Process F prints two strings and then it terminates.

When a process prints any message, the process ID and code (file) have to be printed first. After each wait system call, a parent process should print ID of a terminated child process.

Your output would look as follows (although the order of printing and some details can be different):

Pid __ Code AA: Created process Pid __ (codeBB), process Pid __ (codeCC) and process Pid__ (codeCC)
Pid __ Code BB: Creating file XYZ.txt and write 200 characters from file x.y

Pid __ Code BB: Done and terminating

Pid __ Code CC: Entering an infinite loop.
 Pid __ Code CC: Entering an infinite loop.

Pid __ Code AA: Process Pid __ terminated.

Pid __ Code AA: Created process Pid __ (code DD), process Pid __ (code CC) and process Pid _ (code EE)

Pid __ Code DD: Created process Pid __ (code CC) and waiting for 4 seconds

Pid __ Code CC: Entering an infinite loop.

Pid __ Code EE: This is what I have read from XYZ.txt: -----

Pid __ Code EE: Deleted XYZ.txt and I am terminating

Pid __ Code CC: Entering an infinite loop.

Pid __ Code AA: Process Pid __ terminated (not D process)

Pid __ Code DD: 4 sec elapsed. Created process Pid __ (ps), and waiting for it to terminate

-----output from ps -----

Pid __ Code DD: Process Pid __ terminated.

Pid __ Code DD: Killing process Pid __

Pid __ Code DD: Process Pid __ terminated.

Pid __ DD: Terminating

Pid __ Code AA: Process Pid __ terminated.

Pid __ Code AA: Killing process Pid __ , Process Pid__ and Process Pid__.

Pid __ Code AA: Process Pid __ terminated.
 Pid __ Code AA: Process Pid __ terminated.
 Pid __ Code AA: Process Pid __ terminated.

Pid __ Code AA: Input the character string: *here you type something*

Pid __ Code AA: Created process Pid __ (code FF) and terminating

Pid __ Code FF: Character string 1: _____. Character string 2 _____. Terminating

Submissions:

- your source code files using command: submit c660a? lab2 AA.c BB.c CC.c DD.c EE.c FF.c
- a hard copy of your source codes (indicate your compilation commands) and a hard copy of the output (printing) of your programs from the last executions.

Due date: Wednesday, January 26, 2011.