Real Time Systems, January 2024

Child Processes execute Programs with command line arguments (say, with switches)

Opened: Monday, 22 January 2024, 9:00 AM
Due: Friday, 2 February 2024, 11:59 PM

✔ Done

In the previous assignment we have developed a C program which when executed as "**./a.out executable1 executable2 ... executableN**" creates N child processes where the 1st child process 1st executes the 1st executable file (given by **executable1**). 2nd executes the 2nd executable file (given by **executable2**) and so on.

Please note that in this assignment no scope was there to pass command line arguments to the executables. Extend the assignment so that this can be done now. That is the program now can be called as "./a.out ls -l ps -ef" such that 1st child process executes the command "ls -l" and the second one executes "ps -ef"**!**

Please note that "*ls*" or "*ps*" are executable files in "*/bin*" folder. That is the absolute path for those files are "*/bin/ls*" and "*/bin/ps*". That is, the command line for testing for your program could be "./a.out /bin/ls -l /bin/ps -ef", as well.

Also please note that, here, a.out will receive 4 command line arguments ("/bin/ls", "-l", "/bin/ps", "-ef", *argc* being 5). Whereas, "ls" and "ps" will be executed with 1 command line argument, "-l" and "-ef", respectevily.

"ls" and "ps" have been mentioned here just as examples. Your program should work for any executable file for child processes.

Submission status

Attempt number	This is attempt 1.
Submission status	Submitted for grading
Grading status	Not graded
Time remaining	Assignment was submitted 2 days 23 hours early
Last modified	Wednesday, 31 January 2024, 12:31 AM

Online text	➕ (127 words)
	
	 #include <stdio.h>
	 #include <stdlib.h>
	 #include <unistd.h>
	 #include <sys/wait.h>
	
	 int main(int argc, char *argv[]) {
	    // Checking if the ...

File submissions

⚙ [program6.c](#) 31 January 2024, 12:31 AM

Submission comments

▶ [Comments (0)](#)

◄ [Child Processes execute different Programs](#)

Jump to...

[Synchronizing termination of processes](#) ►