

CSE344 – System Programming - Homework #2 Report

Signal Handling

Student Name: Ozan Şelte

Student Number: 161044061

In this homework, I cannot block SIGSTOP because it is unblockable. Sometimes I used shorter system calls/functions which are not in our slides but course book to escape longer and hard to read functions. (For example: pread, pwrite instead of longer lseek lines; dup instead of fcntl). Finally, to be POSIX compatible I decided to define “_POSIX_C_SOURCE” as “200809” which is the latest POSIX standard.

Problem #1:

The processes must be able to write their common output file at the same time.

Solution #1:

File locks are useful to solve this problem. F_SETLKW parameter of fcntl did well its job, it waits the appropriate moment by self.

Problem #2:

Deleting a line of the temporary file without Standard C Library.

Solution #2:

I did it with a buffer which has a size(funcs.h, MOVE_BUF_SIZE). Now It is much more faster.

Problem #3:

When one of the processes gets SIGTERM the other one will not be terminated.

Solution #3:

I did use SIGUSR2 for this problem. The process which gets SIGTERM sends the other one SIGUSR2. SIGUSR2 is same in my signal handlers except it does not sending other one any signal.

Problem #4:

Process-1 should give information about new added lines to the temporary file.

Solution #4:

I have used SIGCONT for this mission.

Problem #5:

We should print pending signals which are came in the middle of critical section of Process-1.

Solution #5:

We can only block SIGINT, for this reason I counted only this signal with sigpending.

Problem #6:

Process-2 must be aware of the ending of Process-1.

Solution #6:

I sent SIGUSR1 to solve this problem.

Problem #7:

Testing. There is not any given files.

Solution #7:

I have used random binary files generated from /dev/urandom.

Problem #8:

Program-Bs must be aware of each others status as running or closed.

Solution #8:

I have used the same method as Problem #5. When output file's size does not change for some time, the program understands that the faster twin was already closed.

Problem #9:

In Process-2 it is hard to know the files including new lines or not.

Solution #9:

sigsuspend and stat(to determine size of the file) helped for this.