

Author: Zhixin Liu
UO ID: 951452405
Date: 5/1/2016

Project_1 Report

Notice:

- my project covers uspsv1, uspsv2, uspsv3, uspsv4, uspsv5
- In my Makefile, 5 represents uspsv5, and “make all” command would build all the 5 versions of USPS

uspsv1.c

- State:

- successfully compiles
- compiles with no warnings
- successfully links
- links with no warnings
- works correctly
- no memory leaks

- Description:

- use wait() at end of main function to wait all child processes terminated.

uspsv2.c

- State:

- successfully compiles
- compiles with no warnings
- successfully links
- links with no warnings
- works correctly
- no memory leaks

- Description:

- use wait() at end of main function to wait all child processes terminated.

uspsv3.c

- State:

- successfully compiles
- compiles with no warnings
- successfully links
- links with no warnings
- works correctly
- no memory leaks

- Description:

- use alarm() to build timer and send SIGALRM, the time interval is 1 second.
- add variable “status” into process structure Proc to record the process status, which helps on skipping terminated child process, here -1 denotes process never start, 0 represents process has started executing but not terminated yet, 1 means process terminated.
- build child handler function to handle SIGCHLD signal, mark the receiving child process as terminated in its status variable
- instead of using wait(), change into using waitpid() to deal with child processes.
- adding variable fprocs to record terminated child process.

uspsv4.c

- State:

- successfully compiles
- compiles with no warnings
- successfully links
- links with no warnings
- works correctly
- no memory leaks

- Description:

- implement own strcpy function to copy string
- implement own itoa function itoa() to convert number into string
- implement own atoi function myatoi() to convert string into number
- build function accessProcess() to access to proc file including: stat, statm, io
- extract, calculate and display information including: pid, status, size, user time, kernel time, total time, number of read bytes, number of write bytes

uspsv5.c:

- State:

- successfully compiles
- compiles with no warnings
- successfully links
- links with no warnings
- seems work correctly
- no memory leaks

- Description:

- add int variable “priority” into process structure Proc to record the process priority, the initial value of “priority” is 10
- each time when access to process file “io” and “stat”, calculate the total time and total number bytes for both writing and reading, then based on the result to determine if the specific process is IO-bound or CPU-bound

- for IO-bound, divided the process's priority number by 10 to substantially reduce its priority, for CPU-bound, increase the process's priority number until reach the limitation.
- for IO-bound process, set the alarm(1), for CPU-bound process, set the alarm(3), for other process, set the alarm(2)
- in my test of the workload only contains tow process, where one is the given iotest one and another is the given cputest one, the equal time slice version of uspsv4 would terminated the cputest process around 90 seconds, while the modified version of uspsv5 would terminated the cputest process around 68 seconds, which seems show that uspsv5 is more efficient than uspsv4