CSCI 3232 Systems Software Assignment 8

Upload all your files to the correct dropbox in Folio before the deadline --- **11:30PM Apr 26, Sunday, 2020.**

**Note: Do not start to attempt this homework until you have practiced and understood all relevant sample codes in slides and Folio’s example codes. About makefile submission issue in Folio, see last slide in 3\_Pointers\_Functions.pptx. Be sure your makefile works with your own tests under Ubuntu or other Linux variants.**

1. (18 pts) Write a C or C++ program A8p1.c(pp) that accepts one command line argument that is assumed to a string of lower case English characters. Call the ***fork*** function to produce two processes. In the child it should print out the original version of the string. In the parent it should print out the complementary upper case version of the string, i.e. ‘a’→’Z’, ‘b’→’Y’, ‘c’→’X’,…, ‘z’→’A’ (e.g. child: “abc”; parent: “ZYX”). Specify in the output whether the parent or child process is printing. Submit source file, not screen shots.
2. (40 pts) Follow the example programs unix\_pipe.c, named\_pipe.c and shm-posix-combined.c to write **three** versions (two pipe versions and one shared memory version) of an interprocess communication program (A8p2\_unixpipe.c(pp), A8p2\_namedpipe.c(pp) and A8p2\_shm.c(pp)) in C/C++. Each version should create two processes using ***fork***. One of the two processes should send or share four random intergers *a,b,c,d* in the range from -49 to 49 inclusive to the other process. The sending process should print out the values of these four integers. The receiving process should print out the inner product of the two vectors (a,b) and (c,d). Either parent or child can act as the sending process or receiving process. Sample runs of the programs in 1 & 2 are shown below. You do NOT need to submit screen shots. Instead submit source code files.

[kwang@computer][~/temp]$ ./A8p1 abc

I am the child. Printing the original version:

abc

I am the parent. Printing the complementary upper case version:

ZYX

[kwang@computer][~/temp]$ ./A8p2\_shm

child wrote into the shared memory: a=-2, b=-33, c=-21, d=3

parent read from the shared memory: a=-2, b=-33, c=-21, d=3

The inner product between (a,b) and (c,d) is -57

[kwang@computer][~/temp]$ ./A8p2\_unixpipe

parent sent into the unnamed pipe: a=-40, b=5, c=29, d=-13

child received from the unnamed pipe: a=-40, b=5, c=29, d=-13

The inner product between (a,b) and (c,d) is -1225

[kwang@computer][~/temp]$ ./A8p2\_namedpipe

child sent into the named pipe: a=38, b=-12, c=-28, d=-20

parent received from the named pipe: a=38, b=-12, c=-28, d=-20

The inner product between (a,b) and (c,d) is -824

1. (2 pts) You need to write a single makefile to compile all of your programs in 1 and 2. (Up to 2 points could be deducted if you do not provide a working makefile.)
2. (25 pts) Determine the scheduling results of the following four processes P1, P2, P3, P4 using the shortest-remaining-time-first scheduling algorithm by **drawing a Gantt chart** for the scheduling results (break tie, if any, by first-come-first-served). What is the average waiting time? For the Gantt chart, you only need to make it clear for what time periods each process runs and you are not required to use advanced tools to draw the chart in a similar fashion as in the slides.

|  |  |  |
| --- | --- | --- |
| Process | Arrival Time | Burst Time |
| P1 | 0 | 7 |
| P2 | 2 | 4 |
| P3 | 4 | 8 |
| P4 | 6 | 6 |

1. (15 pts) Suppose we have a queue of three processes P1, P2, P3 with burst time 7, 5, 3 respectively and a scheduler uses the Round Robin algorithm to schedule these three processes with time quantum 4. **Draw a Gantt chart** for the scheduling results. What is the turnaround time of P1, P2, P3 respectively?

A total of six files (4 source files +1 makefile +1 solution file) should be submitted.