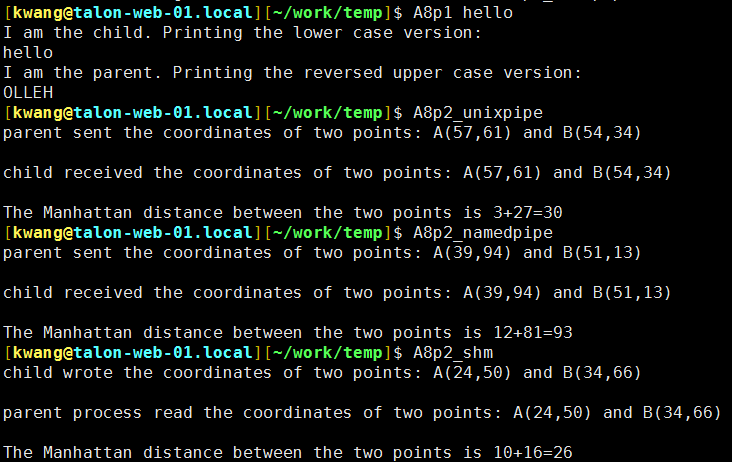
CSCI 3232 Systems Software Assignment 8

Upload all your files to the correct dropbox in Folio before the deadline --- **11:30PM Apr 29, Monday, 2019.**

**Note: Code you wrote for this homework may NOT compile or run under MAC. Do not start to attempt this homework until you have practiced and understood all 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.**

1. (15 pts) Write a C or C++ program A8p1.c(pp) that accepts one command line string parameter. Use the ***fork*** function to produce two processes. In the child process print out the lower case version of the string. In the parent process print out the reversed upper case version of the string. You may call the *toupper* and *tolower* functions in the header <ctype.h> if you wish. Specify in the output whether the parent or child process is printing. Submit code 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 or 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 0 to 99 inclusive (that are interpreted as the coordinates of two points *A,B* in the plane A(a,b) and B(c,d)) to the other process. The sending process should print out the coordinates of the two points A(a,b) and B(c,d). The receiving process should print out the Manhattan distance between the two points A(a,b) and B(c,d). Refer to *https://en.wiktionary.org/wiki/Manhattan\_distance* for the definition of Manhattan distance d(A,B)=|a-c|+|b-d|. Sample runs of the programs in 1 & 2 are shown below. You do NOT need to submit screen shots. Instead submit source code files.



1. (5 pts) You need to write a single makefile to compile all of your programs in 1 and 2. (Up to 5 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 by first-come-first-served). You only need to make it clear for what time periods each process runs and 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 | 14 |
| P2 | 2 | 11 |
| P3 | 4 | 8 |
| P4 | 6 | 20 |

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+1makefile+1solution file) should be submitted.

Solutions:

Question 4:



Question 5:



Turnaround time:

P1: 14

P2: 15

P3: 11