**CS4414 (F2016) Operating Systems**

**Homework 1**

**Points: 4**

**Due: February 9, 2016 (12:30 p.m.)**

Solve the following problems. Each problem is worth 1 point.

(1) Including the initial parent process, how many processes are created by the program shown below? Show the result for each iteration in the for loop. (You will receive no point without the intermediate steps)

#include <stdio.h>

#include <unistd.h>

int main() {

int i;

for (i = 0; i < 4; i++)

fork();

return 0;

}

*1 for the original process (A)*

*1 for the child created for i=0 (B)*

*1 for the child created for i=0 for B (C)*

*1 for the child created for i=0 for C (D)*

*…*

*An infinite number of processes is created. Notably, the fork() would automatically create a new process at the i=0 iteration of each loop thus you wouldn't really ever reach the end of the loop.*

(2) Explain the circumstances under which the line of code marked printf("LINE J") in the following program will be reached.

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

int main() {

/\* fork a child process \*/

pid = fork();

if (pid < 0) { /\* error occurred \*/

fprintf(stderr, "Fork Failed");

return 1;

}

else if (pid == 0) { /\* child process \*/

execlp("/bin/ls", "ls", NULL);

printf ("LINE J");

}

else { /\* parent process \*/

/\* parent will wait for the child to complete \*/

wait(NULL);

printf("Child Complete");

}

return 0;

}

(3) Using the program shown below, identify the values of pid at lines A, B, C, and D. (Assume that the actual pids of the parent and child are 2600 and 2603, respectively)

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

int main() {

/\* fork a child process \*/

pid = fork();

if (pid < 0) { /\* error occurred \*/

fprintf(stderr, "Fork Failed");

return 1;

}

else if (pid == 0) { /\* child process \*/

pid1 = getpid();

printf ("child: pid = %d", pid); /\* A \*/

printf("child: pid1 = %d", pid1); /\* B \*/

}

else { /\* parent process \*/

pid1 = getpid();

printf ("parent: pid = %d", pid); /\* C \*/

printf("parent: pid1 = %d", pid1); /\* D \*/

wait(NULL);

}

return 0;

}

(4) Using the program shown below, explain what the output will be at lines X and Y. (You will get no point if you just write down the result without explanation)

#include <sys/types.h>

#include <stdio.h>

#include <unistd.h>

#define SIZE 5

int nums[SIZE] = {0, 1, 2, 3, 4};

int main() {

int i;

pid\_t pid;

pid = fork();

if (pid ==0) {

for (i = 0; i < SIZE; i++) {

nums[i] \*= -i;

printf("CHILD: %d ", nums[i]); /\* LINE X \*/

}

}

else if (pid > 0) {

wait(NULL);

for (i = 0; i < SIZE; i++)

printf("PARENT: %d ", nums[i]); /\* LINE Y \*/

}

return 0;

}

Use the following criteria to prepare and hand your homework in.

(1) Your homework can be generated using a word processor or by handwriting (but legible).

(2) Your homework should be submitted in class unless it is turned in late, which should be submitted via **collab** within three days from the due date in order to receive partial credit.

(3) Write down the following information on the first page of your homework: your name, homework number (i.e. homework 1), course number (i.e. CS4414), course title (i.e. Operating Systems), semester of the course (i.e. Spring, 2016), and your honor pledge.