CS 47200 Homework 2 Mark Erickson

Submission: in MS Word, as an attachment on Blackboard.

These homework assignments are from Chapters 3 and 4.

1. Including the initial parent process, how many processes are created by the following program?

#include <stdio.h>

#include <unistd.h>

Int main()

{

int i;

for (i = 0; i < 5; i++)

fork();

return 0;

}

2^5=32

2. a. Using the program 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()

{

pid t pid, pid1;

/\* 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 \*/  0

printf("child: pid1 = %d",pid1); /\* B \*/  2603

}

else { /\* parent process \*/

pid1 = getpid();

printf("parent: pid = %d",pid); /\* C \*/  2603

printf("parent: pid1 = %d",pid1); /\* D \*/  2600

wait(NULL);

}

return 0;

}

b. Using the program below, explain what the output will be at lines X and Y.

#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 \*/  0, -1, -4, -9, -16

}

}

else if (pid > 0) {

wait(NULL);

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

printf("PARENT: %d ",nums[i]); /\* LINE Y \*/  0, 1, 2, 3, 4

}

return 0;

}

1. Which of the following components of program state are shared across threads in a multithreaded process?
2. Register values
3. Heap memory 
4. Global variables 
5. Stack memory
6. Consider the following code segment:

pid\_t pid;

pid = fork();

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

fork();

thread create( . . .);

}

fork();

1. How many unique processes are created?  6
2. How many unique threads are created?  3
3. The program shown in figure below uses the Pthreads API. What would be the output from the program at LINE C and LINE P?

#include <pthread.h>

#include <stdio.h>

#include <types.h>

int value = 0;

void \*runner(void \*param); /\* the thread \*/

int main(int argc, char \*argv[])

{

pid\_t pid;

pthread\_t tid;

pthread \_attr\_t attr;

pid = fork();

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

pthread\_attr\_init(&attr);

pthread\_create(&tid,&attr,runner,NULL);

pthread\_join(tid,NULL);

printf("CHILD: value = %d",value); /\* LINE C \*/  5

}

else if (pid > 0) { /\* parent process \*/

wait(NULL);

printf("PARENT: value = %d", value); /\* LINE P \*/  0

}

}

void \*runner(void \*param) {

value = 5;

pthread exit(0);

}