Edgar Cobos

Lab 5

Waiting on processes

**Issue:** The parent’s initial main parameter was set to void which was causing an issue with trying to pass the parent program any arguments.

**Solution:** Just change the main parameter from ‘void’ to ‘int argc, char \*argv[]’. This corrected the issue

**Issue:** The execl function takes char arrays arguments. The sleep time that needs to be passed onto the child is of data type integer.

**Solution:** Instead of converting the sleep time from an integer to a char array, I can just pass the original argument with ‘argv[2]’ to the execl function.

**Issue:** Sleep time being of value zero causes issues when that value is used to modulo with the random number generated. This is because a number modulo 0 is undefined in most systems, as it is in mine.

**Solution:** Since a sleep time of zero doesn’t make much sense for this program, we can just not allow the sleep time to be zero. So the sleep time range will now be [1, 49], inclusive.

**C Code**

/\*\*

\* File: parent.c

\*/

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <sys/wait.h>

#include <stdlib.h>

#include <ctype.h>

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

pid\_t pid, w;

int k, status, N, T;

char value[3];

if (argc != 3) {

printf("\n Usage: %s <Number of Processes [1,19]> <sleeptime[1,49]>\n", argv[0]);

exit(1);

}

N = atoi(argv[1]); //Number of processes

T = atoi(argv[2]); //Sleeptime

if ( T < 1 || T > 49 ){

printf("\n Sleeptime must be an integer between 1 and 49, inclusive.\n");

exit(1);

}

if( N < 1 || N > 19){

printf("\n Number of processes must be an integer between 1 and 19, inclusive.\n");

exit(1);

}

for (k = 0;k < N; ++k) {

if ((pid = fork()) == 0) {

sprintf(value, "%d", k);

execl("child", "child", value, argv[2], (char \*) 0);

}

else printf ("Forked child %d\n", pid);

}

/\* Wait for children \*/

while ((w = wait(&status)) && w != - 1) {

if (w != - 1) printf ("Wait on PID: %d returns status of: %04X\n", w, status);

}

exit(0);

}

/\*\*

\* File: child.c

\*/

#include <stdio.h>

#include <sys/types.h>

#include <unistd.h>

#include <sys/wait.h>

#include <stdlib.h>

#include <signal.h>

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

pid\_t pid;

int ret\_value;

if (argc != 3) {

printf("\n Usage: %s <value> <sleeptime[1,49]>\n", argv[0]);

exit(1);

}

int T = atoi(argv[2]); //Sleeptime

if ( T < 1 || T > 49 ){

printf("\n Sleeptime must be between 1 and 49, inclusive.\n");

exit(1);

}

pid = getpid();

ret\_value = (int) (pid % 256);

srand((unsigned) pid);

sleep(rand() % T);

if (atoi(\*(argv + 1)) % 2) {

printf("Child %d is terminating with signal 009\n", pid);

kill(pid, 9);

} else {

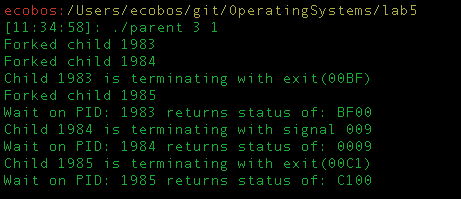
printf("Child %d is terminating with exit(%04X)\n", pid, ret\_value);

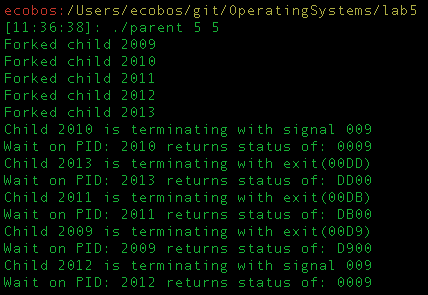
exit(ret\_value);

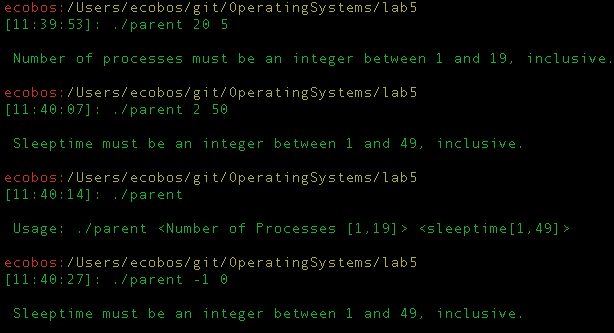
}

}

**Output**

****

****

****