CSC 407: Computer Systems II: 2019 Spring,

Assignment #2

Purpose:

To practice creating processes and signal handling.

Overview:

We will have 2 programs `launcher.c` and `signaller.c`. `launcher` launches 3 child processes, all of which run `signaller`. The `signaller` processes send `SIGUSR1` back to the `launcher` program at random times. `launcher` keeps track of time in intervals of `resetTimeInSec` seconds with `alarm()`. For each interval, it keeps track of the number of times it receives `SIGUSR1` with `counter`. If it receives `SIGUSR1` 3 or more times during one interval then the game is over. It sends `SIGINT` to all 3 children to stop them.

launcher.c:

1. It should have 4 global integers:
    - `counter` with initial value 0
    - `shouldContinue` with initial value 1
    - `resetTimeInSec` read from the command line
    - `signalTimeInSec` read from the command line
2. `main()` should verify that there are two arguments on the command line, and it should verify that their values are at least `MIN_TIME_IN_SECS` (a constant, the integer 2).
3. `main()` should install 2 signal handlers.
    - The one for `SIGALRM` should set `counter` to 0, and reset to receive `SIGALRM` `resetTimeInSec` seconds in the future.
    - The one for `SIGUSR1` should increment (add one to) `counter`. If `counter` is greater than or equal to 3, then it should set `shouldContinue` to 0.
4. `main()` should make 3 child processes that then run `"signaller"`. The `execl()` line should specify the `signalTimeInSec` as a command line argument, but in a string. Do:
5. `const int BUFFER_LEN = 64;`

```
6.
7. char  buffer[BUFFER_LEN];
8.
9. snprintf(buffer,BUFFER_LEN,"%d",signalTimeInSec);
```

Store the 3 child process ids in an array.

10. `main()` should do

```
alarm(resetTimeInSec);
```

This gives it its initial reset time.

11. `main()` should do

```
12.  while  (shouldContinue)
13.  {
14.    sleep(1);
15.  }
```

This makes it hang out while the game is still being played.

16. When the game is over it send `SIGINT` to all children. It also `wait()`s for them to finish
17. `main()` should

```
return(EXIT_SUCCESS);
```

signaller.c:

1. It should have global integers:
    - `maxSignalTime`
    - `shouldContinue` with initial value 1
2. `main()` should get 1 extra command line argument, an integer `maxSignalTime`. It should verify that this argument exists, and it should verify that it is at least `MIN_TIME_IN_SECS`.
3. `main()` should install 2 signal handlers.
    - The one for `SIGALRM` send `SIGUSR1` to its parent and reset the alarm with:

```
alarm(rand() % maxSignalTime + 1);
```

- The one for `SIGINT` should set `shouldContinue` to 0.

4. `main()` should do

   ```
   srand(getpid());
   ```

   which randomizes its random generator.

5. `main()` should do

   ```
   alarm(rand() % maxSignalTime + 1);
   ```

   which sets the initial timer.

6. `main()` so

```
7.   while (shouldContinue)
8.   {
9.     sleep(1);
10.  }
```

   This makes it hang out while the game is still being played.

11. `main()` should quit with

    ```
    return(EXIT_SUCCESS);
    ```

Sample output:

```
$ ./launcher
Usage: launcher <resetTime> <signalTime>
$ ./launcher -10 10
resetTime must be 2 or greater.
$ ./launcher 10 -10
resetTime must be 2 or greater.
$ ./launcher 10 10
signaller 9429 signalling parent
Got 1
signaller 9431 signalling parent
signaller 9430 signalling parent
Got 2
Resetting
signaller 9429 signalling parent
```

```
Got 1
signaller 9431 signalling parent
Got 2
signaller 9431 signalling parent
Got 3  Have reached the limit!
signaller 9429 stopping
signaller 9430 stopping
signaller 9431 stopping
$ ./launcher 10 12
signaller 9874 signalling parent
Got 1
signaller 9873 signalling parent
Got 2
Resetting
signaller 9875 signalling parent
Got 1
signaller 9874 signalling parent
Got 2
signaller 9873 signalling parent
Got 3  Have reached the limit!
signaller 9873 stopping
signaller 9874 stopping
signaller 9875 stopping
$ ./launcher 8 20
signaller 9977 signalling parent
Got 1
signaller 9975 signalling parent
Got 2
Resetting
signaller 9977 signalling parent
Got 1
Resetting
signaller 9976 signalling parent
Got 1
signaller 9977 signalling parent
Got 2
Resetting
signaller 9977 signalling parent
```

```
Got 1
signaller 9975 signalling parent
Got 2
signaller 9976 signalling parent
Got 3  Have reached the limit!
signaller 9975 stopping
signaller 9976 stopping
signaller 9977 stopping
```