CSSE332 Exam 1 Part 1

December 15, 2023

CSSE332 Exam 1 Part 1

Please write your name here:

Please write your section number here:

This exam has two problems, located in `problem1.c` and `problem2.c`. In addition, there is one optional bonus problem located in `problem3.c`. You will have a total of 50 minutes to complete all parts.

Accessing the exam

1. ssh into the class server.
2. You will find a new file, exam1.zip. Unzip it with:

   > `unzip -P enter_exam1_password_here exam1.zip`

   Your instructor will tell you what the exam password is.
3. `cd exam1` to access the exam code.

Allowed Resources

This exam is **closed book**. You are allowed to use one 8.5" by 11" single sided page of notes that you develop yourself. You may not use a web browser or any means of communication (e.g., cell phones, MS Teams, Discord, etc.)

Problem 1 (70 points)

Modify `program1.c` to generate `NUM_CHILDREN` (defined by default to 10) child processes, each of which should execute `./pretty.bin` with the following arguments:

1. The first argument is the parent's process ID.
2. The second argument is the sentence `"Be brave and never give up!"`.

The parent must detect if the child has crashed or if it has exited successfully, and if so, print its PID and exit status. Expect `./pretty.bin` to crash if its PID is a multiple of three.

Additional requirements

- All of the children must be able to start simultaneously, i.e., the parent should not block waiting for a child, until all children have started.
- After creating all of the children, the parent should wait for them **in the order in which they were created**. In other words, the parent must wait for child 0, then child 1, then child 2, and so on.
- The order in which the parent waits for its children and prints whether they crashed or their exit status **does matter**.
- The order in which the children print their message **does not matter**.
- All children will have the same parent process id.
- **Your code should not generate any zombies or orphan children**.

Sample output

As usual, build your code using `make` and then run it. Here's my output:

```
$ ./problem1.bin
./pretty.bin    2627413 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627409 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627410 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627411 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627412 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627414 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627415 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627416 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627417 My parent (2627408) told me: Be brave and never give up!
./pretty.bin    2627418 My parent (2627408) told me: Be brave and never give up!
Child 0 with PID 2627409 crashed!
Child 1 with PID 2627410 finished with exit code 1!
Child 2 with PID 2627411 finished with exit code 2!
Child 3 with PID 2627412 crashed!
Child 4 with PID 2627413 finished with exit code 1!
Child 5 with PID 2627414 finished with exit code 2!
Child 6 with PID 2627415 crashed!
Child 7 with PID 2627416 finished with exit code 1!
Child 8 with PID 2627417 finished with exit code 2!
Child 9 with PID 2627418 crashed!
Parent 2627408 finished ....
```

Problem 2 (30 points)

This is a slight variation of Problem 1. Here are the differences:

- We want our children to execute `prettier.bin` instead of `pretty.bin`.
- `prettier.bin` will crash if its PID is a **multiple of 4**.
- `prettier.bin` might enter an infinite loop. **However**, `prettier.c` is equipped to catch alarm signals and exit with code `99` if such a thing happens, but it does not set such timers itself.

Write your code in `problem2.c`.

In this variation of Problem 1, we would like to do two things:

1. Avoid running `./prettier.bin` if you **foresee** that it will crash.
2. Catch children that run for more than 5 seconds.

If you foresee that a child will crash, then:

1. print `Child <PID> is bad, will not execute`.
2. `exit`

Critically: the child process must do this check, **without executing `./prettier.bin`.**

For children that run for more than 5 seconds, you must:

1. Use appropriate mechanisms to force that child to exit with 99 if it loops for more than 5 seconds. In other words, you must force the child to trigger the handling of alarm signals if at least 5 seconds have passed.
2. The parent must detect those children and print `Child <num> with <PID> timed out!`

Additional requirements

- If the parent detects that a child might crash, it **must directly wait for it** and not fork any additional children until that child has exited.
- The order in which the parent waits for the **successful children** must be in the order in which they are created.
- The parent must print the timed out message for all children that do so.

Sample output

```
$ ./problem2.bin
Child 2681000 is bad, will not execute.
Child 2681004 is bad, will not execute.
./prettier.bin  2681001 My parent (2680999) told me: Be brave and never give up!
Child 2681008 is bad, will not execute.
./prettier.bin  2681002 My parent (2680999) told me: Be brave and never give up!
./prettier.bin  2681003 My parent (2680999) told me: Be brave and never give up!
./prettier.bin  2681005 My parent (2680999) told me: Be brave and never give up!
Child 2681012 is bad, will not execute.
./prettier.bin  2681006 My parent (2680999) told me: Be brave and never give up!
./prettier.bin  2681007 My parent (2680999) told me: Be brave and never give up!
./prettier.bin  2681009 My parent (2680999) told me: Be brave and never give up!
./prettier.bin  2681013 My parent (2680999) told me: Be brave and never give up!
./prettier.bin  2681011 My parent (2680999) told me: Be brave and never give up!
./prettier.bin  2681010 My parent (2680999) told me: Be brave and never give up!
Child 0 with PID 2681001 finished with exit code 1!
Child 1 with PID 2681002 finished with exit code 2!
Child 2 with PID 2681003 timed out!
Child 3 with PID 2681005 finished with exit code 1!
Child 4 with PID 2681006 finished with exit code 2!
Child 5 with PID 2681007 timed out!
Child 6 with PID 2681009 finished with exit code 1!
Child 7 with PID 2681010 finished with exit code 2!
Child 8 with PID 2681011 timed out!
Child 9 with PID 2681013 finished with exit code 1!
Parent 2680999 finished ....
```

(BONUS) Problem 3 (10 points)

In this part, we would like to drop the assumption we made in problem 2, specifically, that `prettier.c` is already equipped to catch alarm signals. `bonus.c` is very much the same behavior as `prettier.c`, except that it does not catch alarm signals, which makes it impossible for us to distinguish between a crash and time out.

Your job in this bonus problem is to solve this issue without modifying the code in `bonus.c`. Naturally, the children you created must execute `bonus.bin` instead of `prettier.bin`.

Write your code in `problem3.c`.

> *Hint*: You might need to manually kill a process, you can do so using the `kill` system call. You can find its documentation using `man 2 kill`.

Sample output

```
$ ./problem3.bin
Child 2682364 is bad, will not execute.
Child 2682368 is bad, will not execute.
./bonus.bin     2682367 My parent (2682363) told me: Be brave and never give up!
Child 2682372 is bad, will not execute.
./bonus.bin     2682373 My parent (2682363) told me: Be brave and never give up!
./bonus.bin     2682369 My parent (2682363) told me: Be brave and never give up!
Child 2682376 is bad, will not execute.
./bonus.bin     2682374 My parent (2682363) told me: Be brave and never give up!
Child 2682384 is bad, will not execute.
./bonus.bin     2682377 My parent (2682363) told me: Be brave and never give up!
./bonus.bin     2682380 My parent (2682363) told me: Be brave and never give up!
./bonus.bin     2682386 My parent (2682363) told me: Be brave and never give up!
./bonus.bin     2682383 My parent (2682363) told me: Be brave and never give up!
./bonus.bin     2682385 My parent (2682363) told me: Be brave and never give up!
./bonus.bin     2682388 My parent (2682363) told me: Be brave and never give up!
Child 0 with PID 2682365 timed out!
Child 1 with PID 2682366 finished with exit code 1!
Child 2 with PID 2682370 finished with exit code 1!
Child 3 with PID 2682371 finished with exit code 2!
Child 4 with PID 2682375 finished with exit code 1!
Child 5 with PID 2682378 crashed!
Child 6 with PID 2682379 timed out!
Child 7 with PID 2682381 finished with exit code 1!
Child 8 with PID 2682382 finished with exit code 2!
Child 9 with PID 2682387 crashed!
Parent 2682363 finished ....
```

Submission instructions

Leave your code in your exam folder and exit your ssh session.