cs201 – sum 2020

Lab 7 – (15 points)

For information on how to do this lab, see the page on "Exceptional Flow Control Part II Summary: Signals" in the week 7 module.

Submit your code to the assignment folder for lab 7 by the due date.

"Family Feud" Shootout with Signals

Were going to program a shootout between child and parent processes which basically goes like this:

1. For 10 rounds, both child and parent will be sending each other signals. After sending a signal, each process will sleep for a random number of secs (from 1 to 3 seconds is about right).
2. When a process gets a signal, inside the signal handler generate a random number (I used values between 0 and 49) as the amount of damage inflicted. NOTE: Your signal handler should be very short. I set a global variable and let the main code handle the signal inside the loop (just check for damage not being zero). So your handler should be very short (only one or two lines).
3. To avoid getting hit again while processing a previous hit; block the signal first and restore it afterwards. See sigblock.c on how to block a signal. To restore the signal use at the end of processing:
   sigprocmask(SIG_SETMASK,&prev_mask,&mask);
   Remember to set the global variable back to 0 after the processing is done. NOTE: Only block the signal while processing a hit: check for a hit, then block signal, process the hit, then unblock signal.
4. To process a hit, keep track in each process how much damage has been done (this can be a local variable outside of the loop). If a process exceeds a maximum damage (I used 200) then they have 'died'. Have them either quit using _exit or just break out of the loop (the parent should break out and then wait for the child to end so it can be 'reaped').
5. If a process manages to get through all rounds, then it survives. Note that either process may both live or both die as well as one living and one dying.

HINT: Look at the sigchild.c example as a starting point. Note you'll need to put in a loop so you can get several rounds in.

Also, look at assignment 3 for information on using rand(). You will need to seed each process separately with a different value. Try using (time(NULL) % parentid) as a seed for the child process and just time(NULL) for the parent (seems to work for me).

Here is some output from an example program I wrote. You can make the output more fun by using a theme ("Star Wars", "Star Trek", etc) instead of using "parent" and "child". You can also use some cool ASCII art or some interesting dialog between the parent and child.

```
====================
OUTPUT ONE
====================
child 12692 got hit with +45. damage is now 45
parent 12691 got hit with +36. damage is now 36
child 12692 got hit with +43. damage is now 88
parent 12691 got hit with +40. damage is now 76
parent 12691 got hit with +14. damage is now 90
child 12692 got hit with +25. damage is now 113
parent 12691 got hit with +43. damage is now 133
child 12692 got hit with +25. damage is now 138
parent 12691 got hit with +24. damage is now 157
child 12692 got hit with +12. damage is now 150
parent 12691 got hit with +39. damage is now 196
child 12692 got hit with +45. damage is now 195
parent 12691 got hit with +44. damage is now 240
parent has died!
child 12692 got hit with +31. damage is now 226
child has died!

======================
OUTPUT TWO
======================
child 12720 got hit with +3. damage is now 3
parent 12719 got hit with +25. damage is now 25
child 12720 got hit with +29. damage is now 32
parent 12719 got hit with +26. damage is now 51
child 12720 got hit with +41. damage is now 73
parent 12719 got hit with +30. damage is now 81
child 12720 got hit with +45. damage is now 118
parent 12719 got hit with +18. damage is now 99
child 12720 got hit with +47. damage is now 165
parent 12719 got hit with +4. damage is now 103
child 12720 got hit with +32. damage is now 197
parent 12719 got hit with +48. damage is now 151
child 12720 got hit with +20. damage is now 217
child has died!
parent 12719 got hit with +1. damage is now 152
parent has survived!

======================
```

OUTPUT THREE

==========

```
parent 12855 got hit with +44. damage is now 44
child 12856 got hit with +42. damage is now 42
child 12856 got hit with +46. damage is now 88
parent 12855 got hit with +29. damage is now 73
child 12856 got hit with +36. damage is now 124
child 12856 got hit with +9. damage is now 133
parent 12855 got hit with +30. damage is now 103
parent 12855 got hit with +36. damage is now 139
child 12856 got hit with +24. damage is now 157
parent 12855 got hit with +41. damage is now 180
child 12856 got hit with +9. damage is now 166
parent 12855 got hit with +18. damage is now 198
child 12856 got hit with +18. damage is now 184
parent 12855 got hit with +29. damage is now 227
child 12856 got hit with +34. damage is now 218
parent has died!
child has died!
```