Steven Tang

CSE 460

1/20/2017

Homework 1

1. ( 10 points )  
   How many processes does the following piece of code create? Why?
2. int main()
3. {
4. fork();
5. fork();
6. fork();
7. return 0;
8. }

* This code creates 8 process. If we were to plug in the number of fork() methods that are called, we can plug it into 2n. So 23 = 8 processes running.

1. ( 20 points )  
   a) Write a C-program that creates a chain of 10 processes and prints out their process ids and relationships. For example, process 1 is the parent of process 2, process 2 is the parent of process 3, process 3 is the parent of 4 and so on. Each child has to print out all her ancestors identified by the process ids.
2. /\*
3. \*   Steven Tang
4. \*   CSE 460 Homework 2A
5. \*
6. \*   Create program that creates a chain of 10 processes and prints out
7. \*   process ids and relationships. process 1 is parent of process 2,
8. \*   process 2 is parent of process 3... etc
9. \*
10. \*   Child = 0
11. \*   Parent = Anything but 0
12. \*/
14. #include <iostream>
15. #include <unistd.h>
16. **using** **namespace** std;
18. **int** main()
19. {
20. cout << endl;
21. **int** pid = fork();
22. **for**(**int** i=0; i < 10; i++) {
23. **if**(pid == 0) {
24. cout << "This is a child with ID: " << getpid()
25. << ". My parent ID is: " << getppid() << endl;
26. pid = fork();
27. } **else**
28. wait(0);
29. }
30. }

TYPESCRIPT FOR 2A

1. Script started on Sun Jan 22 13:40:19 2017
2. [?1034hbash-3.2$ c++ 2a.cpp
3. bash-3.2$ ./a.out
5. This is a child with ID: 6260. My parent ID is: 6259
6. This is a child with ID: 6261. My parent ID is: 6260
7. This is a child with ID: 6262. My parent ID is: 6261
8. This is a child with ID: 6263. My parent ID is: 6262
9. This is a child with ID: 6264. My parent ID is: 6263
10. This is a child with ID: 6265. My parent ID is: 6264
11. This is a child with ID: 6266. My parent ID is: 6265
12. This is a child with ID: 6267. My parent ID is: 6266
13. This is a child with ID: 6268. My parent ID is: 6267
14. This is a child with ID: 6269. My parent ID is: 6268
15. bash-3.2$ exit
16. exit
18. Script done on Sun Jan 22 13:40:25 2017

b) Write a C-program that creates a fan of 10 processes. That is, process 1 is the parent of prcocesses 2, 3, 4, 5, 6 and so on.

1. /\*
2. \*   Steven Tang
3. \*   CSE 460 Homework 2b
4. \*
5. \*   Write a C-program that creates a fan of 10 processes.
6. \*   That is, process 1 is the parent of prcocesses 2, 3, 4, 5, 6 and so on.
7. \*
8. \*/
9. #include <iostream>
10. #include <unistd.h>
12. **using** **namespace** std;
14. **int** main()
15. {
16. **int** pid;
17. **int** parent = getpid();
19. cout << "\n\t\t\tParent id is: " << parent << endl;
20. pid = fork();
22. **for**(**int** i=0; i < 10; ++i) {
23. **if**(pid > 0) {
24. pid = fork(); //We don't want parents, we want child so we fork
25. **if**(pid == 0) {
26. cout << "This is a child process: " << getpid() << ", and the parent ID is: " << getppid() << endl;
27. }
28. }
29. }
30. }

TYPESCRIPT FOR 2B

1. Script started on Sun Jan 22 13:41:25 2017
2. [?1034hbash-3.2$ c++ 2b.cpp
3. bash-3.2$ ./a.out
5. Parent id is: 6347
6. This is a child process: 6349, and the parent ID is: 6347
7. This is a child process: 6350, and the parent ID is: 6347
8. This is a child process: 6351, and the parent ID is: 6347
9. This is a child process: 6352, and the parent ID is: 6347
10. This is a child process: 6353, and the parent ID is: 6347
11. This is a child process: 6354, and the parent ID is: 6347
12. This is a child process: 6355, and the parent ID is: 6347
13. This is a child process: 6356, and the parent ID is: 6347
14. This is a child process: 6357, and the parent ID is: 6347
15. This is a child process: 6358, and the parent ID is: 6347
16. bash-3.2$ exit
17. exit
19. Script done on Sun Jan 22 13:41:33 2017
20. (10 points)  
    a) Write a simple program named **test1.cpp**, which contains an infinite **while** loop. Compile the program to an executable named **test1** and run it in the background.
21. //test1.cpp
22. #include <iostream>
24. **using** **namespace** std;
26. **int** main(){
27. **int** x;
28. **while** (1){
29. x++;
30. }
31. }

b) Write a shell script that searches for whether the process **test1** is in the system. If it is not, your script displays the message 'Process test1 not running!'. If it is running, your script kills the process, and displays the message 'Process test1 killed!'.

1. #
2. #testProcess
3. #This script will find if the user's program is running
4. #If it's runnning the script will kill the program, if not then the program
5. #will print out that their process is not running.
6. #
7. process=$(pgrep $1)
9. **if** [ "$process" == "" ]
10. then
11. echo "Process $1 not running!"
12. **else**
13. pkill $1
14. echo "Process $1 killed!"
15. fi

TYPESCRIPT FOR 3B

1. Script started on Sun Jan 22 12:36:15 2017
2. [?1034hbash-3.2$ p[Kls
3. 2a.cpp      3b          [31mtest1[39;49m[0m       [31mtestProcess[39;49m[0m
4. 2b.cpp      [31ma.out[39;49m[0m       test1.cpp
5. bash-3.2$ ./test1 &
6. [1] 2327
7. bash-3.2$ ./testProcess test1
8. Process test1 killed!
9. [1]+  Terminated: 15          ./test1
10. bash-3.2$ ps
11. PID TTY           TIME CMD
12. 1428 ttys000    0:00.57 -bash
13. 2324 ttys000    0:00.01 script 3b
14. 2325 ttys001    0:00.01 /bin/bash -i
15. bash-3.2$ ps./testProcess test1
16. [C[C[C[C[C[C[C[C[C[Cps[K./testProcess test1
17. Process test1 not running!
18. bash-3.2$ exit
19. exit
21. Script done on Sun Jan 22 12:36:38 2017