Steven Tang

CSE 460

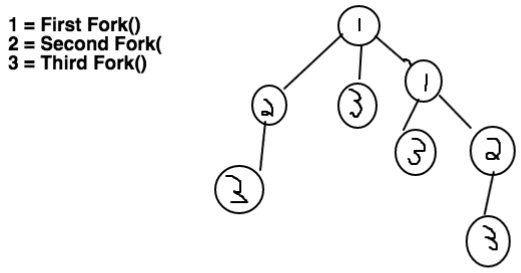
1/20/2017

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 2^n. So $2^3 = 8$ processes running.



2. (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.

```
3. /*
4. *   Steven Tang
5. *   CSE 460 Homework 2A
6. *
7. *   Create program that creates a chain of 10 processes and prints out
8. *   process ids and relationships. process 1 is parent of process 2,
9. *   process 2 is parent of process 3... etc
10.*   Child = 0
11.*   Parent = Anything but 0
```

```
*/

#include <iostream>
#include <unistd.h>
using namespace std;

int main()
{
    cout << endl;
    int pid = fork();
    for(int i=0; i < 10; i++) {
        if(pid == 0) {
            cout << "This is a child with ID: " << getpid()
                << ". My parent ID is: " << getppid() << endl;
            pid = fork(); // Fork again
        } else
            wait(0);
    }
}
```

TYPESCRIPT FOR 2A

```
Script started on Sun Jan 22 13:40:19 2017
[?1034hbash-3.2$ c++ 2a.cpp
bash-3.2$ ./a.out

This is a child with ID: 6260. My parent ID is: 6259
This is a child with ID: 6261. My parent ID is: 6260
This is a child with ID: 6262. My parent ID is: 6261
This is a child with ID: 6263. My parent ID is: 6262
This is a child with ID: 6264. My parent ID is: 6263
This is a child with ID: 6265. My parent ID is: 6264
This is a child with ID: 6266. My parent ID is: 6265
This is a child with ID: 6267. My parent ID is: 6266
This is a child with ID: 6268. My parent ID is: 6267
This is a child with ID: 6269. My parent ID is: 6268
bash-3.2$ exit
exit

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.

```
/*
*   Steven Tang
*   CSE 460 Homework 2b
*
*   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.
*
*/
#include <iostream>
#include <unistd.h>

using namespace std;

int main()
{
    int pid;
    int parent = getpid();

    cout << "\n\t\t\tParent id is: " << parent << endl;
    pid = fork();

    for(int i=0; i < 10; ++i) {
        if(pid > 0) {
            pid = fork(); //We don't want parents, we want child so we fork
            if(pid == 0) {
            cout << "This is a child process: " << getpid() << ", and the parent ID is:
 " << getppid() << endl;
            }
        }
     }
}
```

TYPESCRIPT FOR 2B

```
Script started on Sun Jan 22 13:41:25 2017
[?1034hbash-3.2$ c++ 2b.cpp
bash-3.2$ ./a.out

            Parent id is: 6347
This is a child process: 6349, and the parent ID is: 6347
This is a child process: 6350, and the parent ID is: 6347
This is a child process: 6351, and the parent ID is: 6347
This is a child process: 6352, and the parent ID is: 6347
This is a child process: 6353, and the parent ID is: 6347
This is a child process: 6354, and the parent ID is: 6347
This is a child process: 6355, and the parent ID is: 6347
This is a child process: 6356, and the parent ID is: 6347
This is a child process: 6357, and the parent ID is: 6347
This is a child process: 6358, and the parent ID is: 6347
bash-3.2$ exit
exit

Script done on Sun Jan 22 13:41:33 2017
```

3. (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.

```
//test1.cpp
#include <iostream>
```

```
3.
4.  using namespace std;
5.
6.  int main(){
7.      int x;
8.      while (1){
9.          x++;
10.     }
11. }
```

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 $1)
8.
9.  if [ "$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
20.
21. Script done on Sun Jan 22 12:36:38 2017
```