Farshad Chowdhury

EECE-4810

Assignment #1

9/28/17

Objective

The objective of this assignment is to work with UNIX processes. This assignment covers how to fork process into a parent and child. This assignment also covers the importance of using the wait() function whilst managing processes to ensure that the child does not die before the parent. Lastly, this assignment also deals with running different code in a new process by calling cd, ls, and pwd kernel commands from a child process.

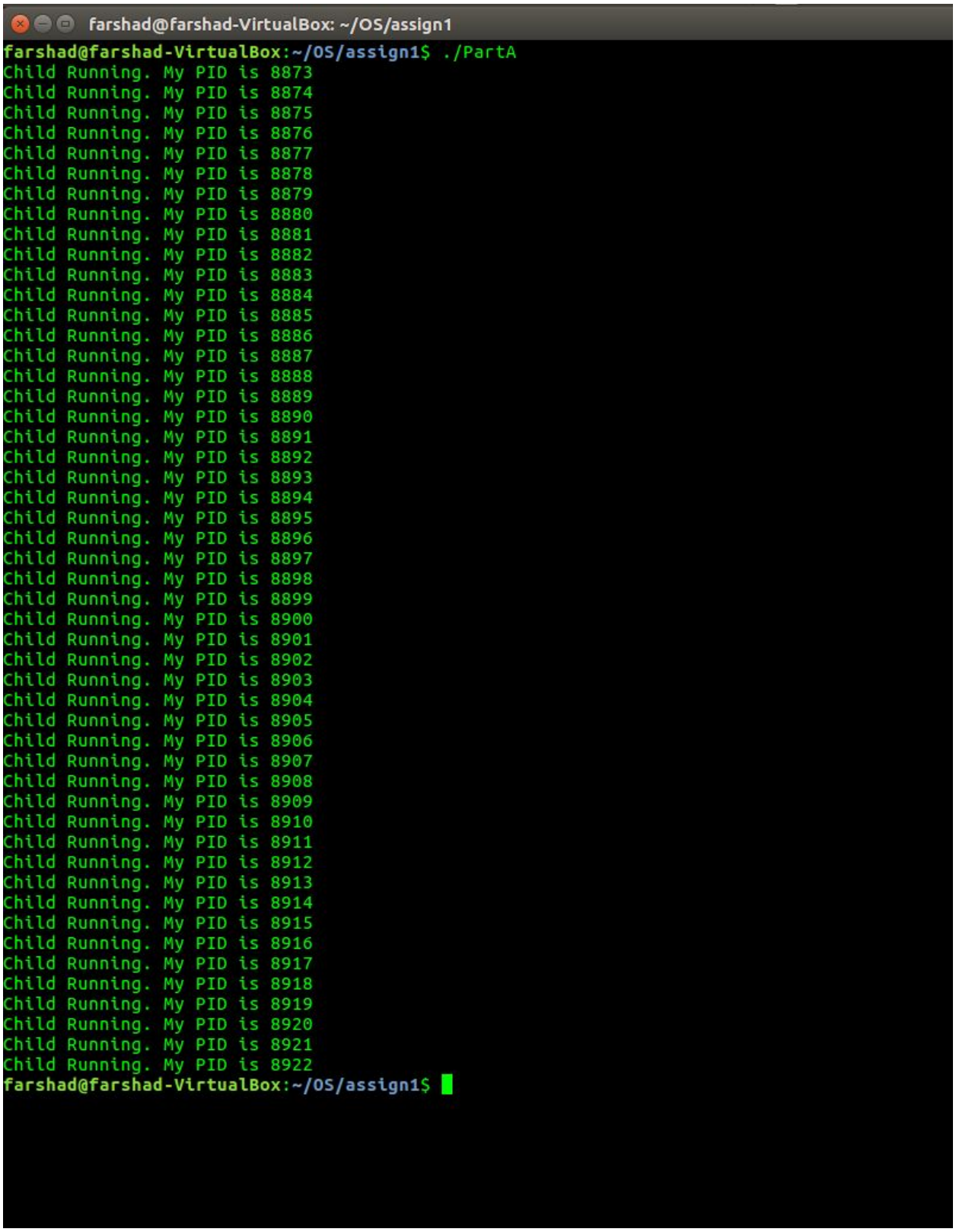
Background

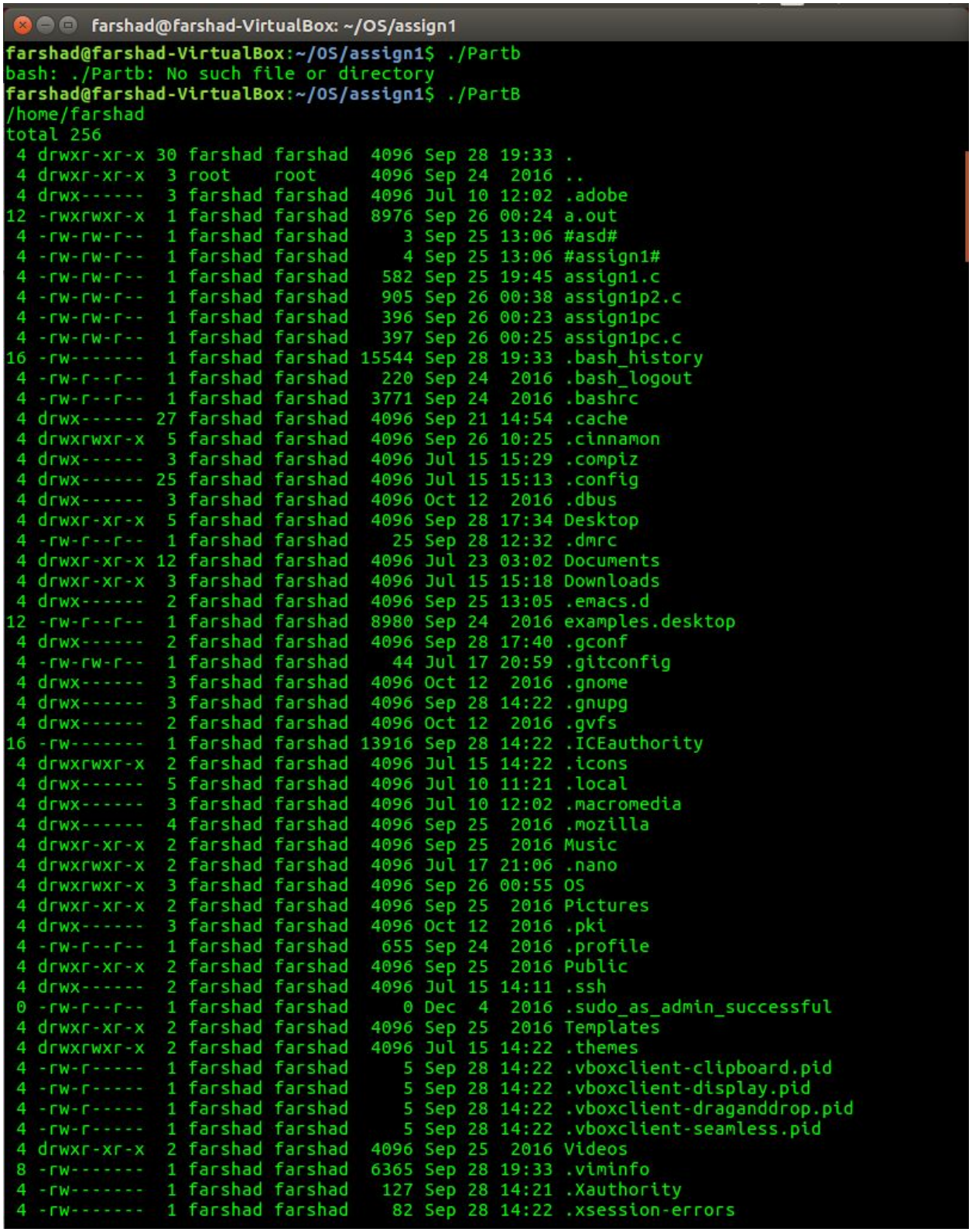
This assignment makes extensive use of the fork command in order to make new processes called children which copy the memory of the parent process. The first part of the assignment uses the fork function to create 50 child processes that display their PID (Process Identification) and the exit. The second part of the assignment creates a child process and has the child process execute code that is different from the parent process. In this case each child calls the kernel commands *cd,* followed by *ls* and *pwd*. The last part of this assignment makes 2*n process, where n is a user-input. Each child then prints out “I am process number #”. The program also ensures that each child process dies before the parent to prevent the formation of zombie processes.

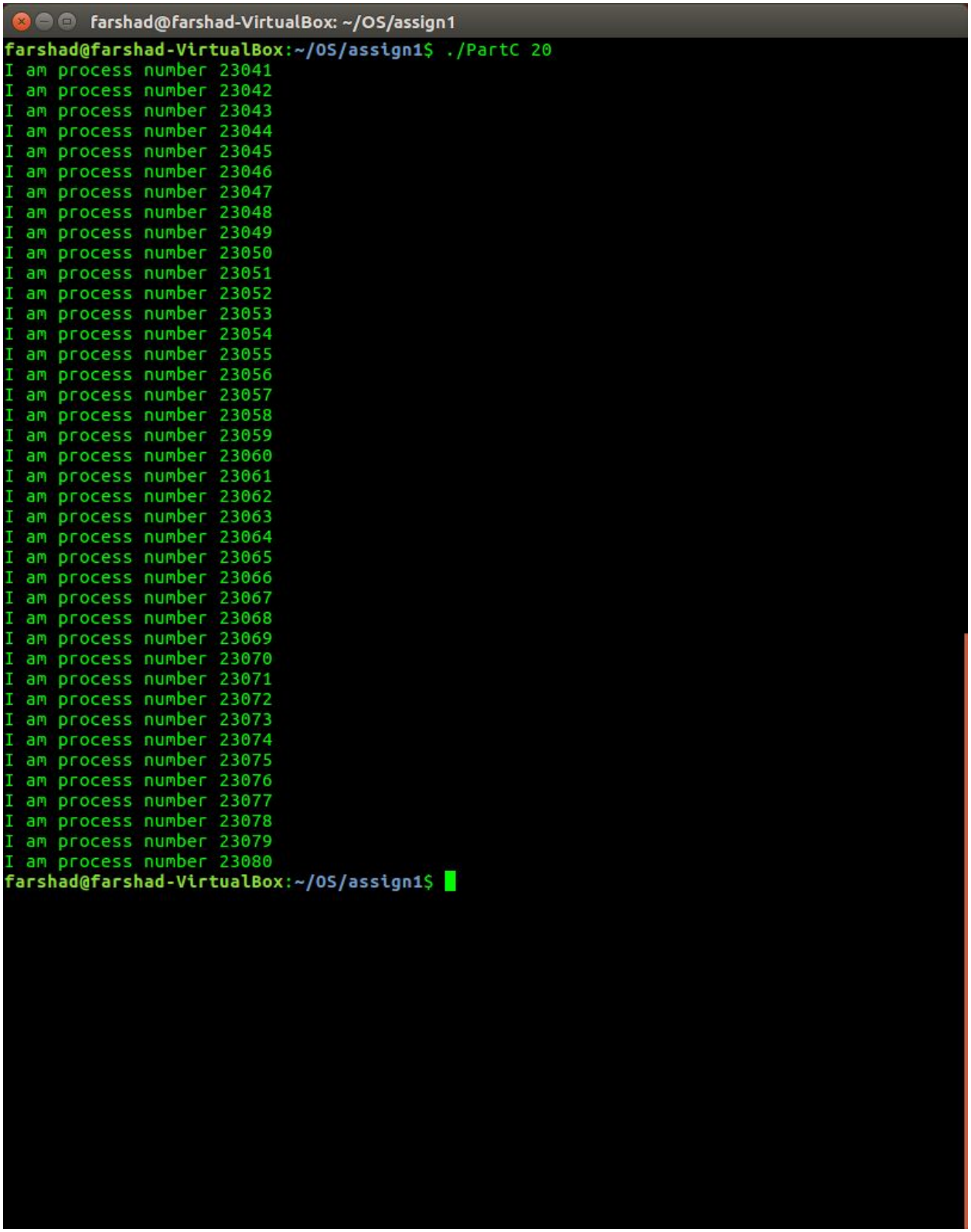
Functions & Algorithms Used

This assignment used several different functions that are included in <sys/types.h>. The first function dealt with is fork(). This is the function used to create a new child process which essentially a clone of the parent as it copies the memory registers. Fork initially returns 0 for the child process and returns the positive integer PID of the the parent process. USing this we can control what each process does. The next important function dealt with in this assignment is the wait() function. In this context, wait is used with the child’s PID to ensure that the child process dies before he parent, essentially making the parent wait for the child to “finish”.

Results


farshad@farshad-VirtualBox: ~/OS/assign1
farshad@farshad-VirtualBox:~/OS/assign1$ ./PartA
Child Running. My PID is 8873
Child Running. My PID is 8874
Child Running. My PID is 8875
Child Running. My PID is 8876
Child Running. My PID is 8877
Child Running. My PID is 8878
Child Running. My PID is 8879
Child Running. My PID is 8880
Child Running. My PID is 8881
Child Running. My PID is 8882
Child Running. My PID is 8883
Child Running. My PID is 8884
Child Running. My PID is 8885
Child Running. My PID is 8886
Child Running. My PID is 8887
Child Running. My PID is 8888
Child Running. My PID is 8889
Child Running. My PID is 8890
Child Running. My PID is 8891
Child Running. My PID is 8892
Child Running. My PID is 8893
Child Running. My PID is 8894
Child Running. My PID is 8895
Child Running. My PID is 8896
Child Running. My PID is 8897
Child Running. My PID is 8898
Child Running. My PID is 8899
Child Running. My PID is 8900
Child Running. My PID is 8901
Child Running. My PID is 8902
Child Running. My PID is 8903
Child Running. My PID is 8904
Child Running. My PID is 8905
Child Running. My PID is 8906
Child Running. My PID is 8907
Child Running. My PID is 8908
Child Running. My PID is 8909
Child Running. My PID is 8910
Child Running. My PID is 8911
Child Running. My PID is 8912
Child Running. My PID is 8913
Child Running. My PID is 8914
Child Running. My PID is 8915
Child Running. My PID is 8916
Child Running. My PID is 8917
Child Running. My PID is 8918
Child Running. My PID is 8919
Child Running. My PID is 8920
Child Running. My PID is 8921
Child Running. My PID is 8922
farshad@farshad-VirtualBox:~/OS/assign1$


farshad@farshad-VirtualBox: ~/OS/assign1
farshad@farshad-VirtualBox:~/OS/assign1$ ./Partb
bash: ./Partb: No such file or directory
farshad@farshad-VirtualBox:~/OS/assign1$ ./PartB
/home/farshad
total 256
4 drwxr-xr-x 30 farshad farshad 4096 Sep 28 19:33 .
4 drwxr-xr-x 3 root root 4096 Sep 24 2016 ..
4 drwx------ 3 farshad farshad 4096 Jul 10 12:02 .adobe
12 -rwxrwxr-x 1 farshad farshad 8976 Sep 26 00:24 a.out
4 -rw-rw-r-- 1 farshad farshad 3 Sep 25 13:06 #asd#
4 -rw-rw-r-- 1 farshad farshad 4 Sep 25 13:06 #assign1#
4 -rw-rw-r-- 1 farshad farshad 582 Sep 25 19:45 assign1.c
4 -rw-rw-r-- 1 farshad farshad 905 Sep 26 00:38 assign1p2.c
4 -rw-rw-r-- 1 farshad farshad 396 Sep 26 00:23 assign1pc
4 -rw-rw-r-- 1 farshad farshad 397 Sep 26 00:25 assign1pc.c
16 -rw------- 1 farshad farshad 15544 Sep 28 19:33 .bash_history
4 -rw-r--r-- 1 farshad farshad 220 Sep 24 2016 .bash_logout
4 -rw-r--r-- 1 farshad farshad 3771 Sep 24 2016 .bashrc
4 drwx------ 27 farshad farshad 4096 Sep 21 14:54 .cache
4 drwxrwxr-x 5 farshad farshad 4096 Sep 26 10:25 .cinnamon
4 drwx------ 3 farshad farshad 4096 Jul 15 15:29 .compiz
4 drwx------ 25 farshad farshad 4096 Jul 15 15:13 .config
4 drwx------ 3 farshad farshad 4096 Oct 12 2016 .dbus
4 drwxr-xr-x 5 farshad farshad 4096 Sep 28 17:34 Desktop
4 -rw-r--r-- 1 farshad farshad 25 Sep 28 12:32 .dmrc
4 drwxr-xr-x 12 farshad farshad 4096 Jul 23 03:02 Documents
4 drwxr-xr-x 3 farshad farshad 4096 Jul 15 15:18 Downloads
4 drwx------ 2 farshad farshad 4096 Sep 25 13:05 .emacs.d
12 -rw-r--r-- 1 farshad farshad 8980 Sep 24 2016 examples.desktop
4 drwx------ 2 farshad farshad 4096 Sep 28 17:40 .gconf
4 -rw-rw-r-- 1 farshad farshad 44 Jul 17 20:59 .gitconfig
4 drwx------ 3 farshad farshad 4096 Oct 12 2016 .gnome
4 drwx------ 3 farshad farshad 4096 Sep 28 14:22 .gnupg
4 drwx------ 2 farshad farshad 4096 Oct 12 2016 .gvfs
16 -rw------- 1 farshad farshad 13916 Sep 28 14:22 .ICEauthority
4 drwxrwxr-x 2 farshad farshad 4096 Jul 15 14:22 .icons
4 drwx------ 5 farshad farshad 4096 Jul 10 11:21 .local
4 drwx------ 3 farshad farshad 4096 Jul 10 12:02 .macromedia
4 drwx------ 4 farshad farshad 4096 Sep 25 2016 .mozilla
4 drwxr-xr-x 2 farshad farshad 4096 Sep 25 2016 Music
4 drwxrwxr-x 2 farshad farshad 4096 Jul 17 21:06 .nano
4 drwxrwxr-x 3 farshad farshad 4096 Sep 26 00:55 OS
4 drwxr-xr-x 2 farshad farshad 4096 Sep 25 2016 Pictures
4 drwx------ 3 farshad farshad 4096 Oct 12 2016 .pki
4 -rw-r--r-- 1 farshad farshad 655 Sep 24 2016 .profile
4 drwxr-xr-x 2 farshad farshad 4096 Sep 25 2016 Public
4 drwx------ 2 farshad farshad 4096 Jul 15 14:11 .ssh
0 -rw-r--r-- 1 farshad farshad 0 Dec 4 2016 .sudo_as_admin_successful
4 drwxr-xr-x 2 farshad farshad 4096 Sep 25 2016 Templates
4 drwxrwxr-x 2 farshad farshad 4096 Jul 15 14:22 .themes
4 -rw-r----- 1 farshad farshad 5 Sep 28 14:22 .vboxclient-clipboard.pid
4 -rw-r----- 1 farshad farshad 5 Sep 28 14:22 .vboxclient-display.pid
4 -rw-r----- 1 farshad farshad 5 Sep 28 14:22 .vboxclient-draganddrop.pid
4 -rw-r----- 1 farshad farshad 5 Sep 28 14:22 .vboxclient-seamless.pid
4 drwxr-xr-x 2 farshad farshad 4096 Sep 25 2016 Videos
8 -rw------- 1 farshad farshad 6365 Sep 28 19:33 .viminfo
4 -rw------- 1 farshad farshad 127 Sep 28 14:21 .Xauthority
4 -rw------- 1 farshad farshad 82 Sep 28 14:22 .xsession-errors


farshad@farshad-VirtualBox: ~/OS/assign1
farshad@farshad-VirtualBox:~/OS/assign1$ ./PartC 20
I am process number 23041
I am process number 23042
I am process number 23043
I am process number 23044
I am process number 23045
I am process number 23046
I am process number 23047
I am process number 23048
I am process number 23049
I am process number 23050
I am process number 23051
I am process number 23052
I am process number 23053
I am process number 23054
I am process number 23055
I am process number 23056
I am process number 23057
I am process number 23058
I am process number 23059
I am process number 23060
I am process number 23061
I am process number 23062
I am process number 23063
I am process number 23064
I am process number 23065
I am process number 23066
I am process number 23067
I am process number 23068
I am process number 23069
I am process number 23070
I am process number 23071
I am process number 23072
I am process number 23073
I am process number 23074
I am process number 23075
I am process number 23076
I am process number 23077
I am process number 23078
I am process number 23079
I am process number 23080
farshad@farshad-VirtualBox:~/OS/assign1$

Observations

The code ran as expected. For the first part the os assigns a pid of 8873 for the first child and increments by 1 for each subsequent child. For the second part, as can be seen by the outputs, the console commands of cd, pwd, and then ls are run successfully as the program changes directory, prints the working directory and then prints the contents of said directory. FOr the last part of the program, the program successfully takes in a user read integer and forks 2*n number of processes and prints out the pid before killing it.

Conclusion

All of the code compiled and ran as expected. Learning how to manage and create processes is very important in understanding how an OS work. This assignment also made very clear that we must be very careful in not letting a parent process die before its child.

Source Code

```
/*#####################################################
## Farshad Chowdhury                                ##
## EECE- 4810 Operating Systems and Kernel Design   ##
## Sept 28, 2017                                    ##
## Assignment 1 Part 1                              ##
##                          ##
##                          ##
#####################################################*/


#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv) {
    int res;
for(int i=0;i<50;i++){

    res = fork();
    if (res < 0) {
        perror("fork");
        exit(0);


    }
    else if (res == 0) { // Child Enters Here
        printf("Child Running. My PID is %d\n",getpid()); //Ouputs child running and pid
        exit(0);

    }else{
        int child_pid=res;
        waitpid(child_pid, NULL, 0); //Parent waits for child to terminate
        }
    }
}
```

```c
/*#####################################################
## Farshad Chowdhury                                ##
## EECE- 4810 Operating Systems and Kernel Design   ##
## Sept 28, 2017                                    ##
## Assignment 1 Part 2                              ##
##                                                   ##
##                                                   ##
#####################################################*/


#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char **argv) {
    int res;
for(int i=0;i<10;i++){

    res = fork();
    if (res < 0) {
        perror("fork");
        exit(0);
        }
    else if (res == 0) {
        chdir("/home/farshad"); //calls chdir/cd
        system("pwd"); // calls pwd
        system("ls -sal"); // calls ls
        exit(0);

    }else{
        int child_pid=res;
        waitpid(child_pid, NULL, 0);
        }
    }
}
```

```c
/*#####################################################
## Farshad Chowdhury                                ##
## EECE- 4810 Operating Systems and Kernel Design   ##
## Sept 28, 2017                                    ##
## Assignment 1 Part 3                              ##
##                                        ##
##                                        ##
#####################################################*/

#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]) {
    if (argc!=2){
        printf("No or Invalid Argument");
        exit(0);
    }else{
    int n= atoi(argv[1]);   //converts ascii to int
    int res;
    for (int i=0; i<2*n;i++){
        res=fork();
        if (res < 0) {
            perror("fork");
            exit(0);
        }else if(res==0){
            printf("I am process number %d\n",getpid());
            exit(0);
        }else{
            int child_pid=res;
            waitpid(child_pid, NULL, 0);

        }

        }
    return 0;
    }


}
```