ITCS 3146, Spring 2016

PROGRAMMING ASSIGNMENT #3

Parent-Child Cooperation

Submission Deadline: 11:59 PM, Friday, April 20th, 2016

Write a program, based on parent-child cooperation, where the parent process offers the program user a menu whose options are to:

1. Display the current date and time,
2. Display the calendar for the current month,
3. List files in the current directory,
4. Quit the program.

For any choice other than quit, the parent should create a child process that invokes a call to system(…) to execute the requested Linux command. Linux command 'date' can be used for choice #1, 'cal' for #2, and 'ls -l' for #3. Make sure that upon the execution of a single Linux command, the child process terminates (dies). Before displaying the menu again, the parent should **wait** for the child process to terminate executing the previous command. When the user chooses to quit, the parent process terminates.

When writing the program, make sure that you display the identification number of the process that displays the information. So, for example, a suggested output of a run of your program should look similar to the following:

```
Parent process ID: 2765, Parent's parent ID: 904
Choose the function to be performed by the child:
    (1) Display current date and time
    (2) Display the calendar of the current month
    (3) List the files in the current directory
    (4) Exit from the program

Enter your choice: 1
Child process ID 2766; Parent ID 2765
Fri Nov 26 17:36:25 EST 1999

Current process PID 2765
Choose the function to be performed by the child:
    (1) Display current date and time
    (2) Display the calendar of the current month
    (3) List the files in the current directory
    (4) Exit from the program

Enter your choice: 3
Child process ID 2774; Parent ID 2765
total 18
-rwxr-xr-x   1 cukic    5093        6740 Nov 26 16:58 a.out
-rw-r--r--   1 cukic    5093        1150 Nov 26 16:59 a3.c

Current process PID 2765
Choose the function to be performed by the child:
    (1) Display current date and time
    (2) Display the calendar of the current month
    (3) List the files in the current directory
```

```
    (4) Exit from the program

Enter your choice: 2
Child process ID 2797; Parent ID 2765
   December 1999
 S  M Tu  W Th  F  S
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31


Current process PID 2765
Choose the function to be performed by the child:
    (1) Display current date and time
    (2) Display the calendar of the current month
    (3) List the files in the current directory
    (4) Exit from the program

Enter your choice: 4

Current process PID 2765
Program terminated
```

Of course, every time you run your program, the process ID numbers will be different, reflecting the run-time assignment of process PIDs. Some of the C function calls you will need to use in this program are `fork( ), getpid( ), getppid( ),sprintf( ), system( ), exit( ), wait( ),` etc. The use of these functions is very well described in one of the recommended textbooks ("UNIX Systems Programming, see the reference in the sylabus"). For the description of `exit( )` and `wait( )` system calls, take a look at the descriptions in sections 2.11 and 3.4 of the same textbook. The following header files should most likely be included in your programs:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
```

Submission

In order to be graded, you must submit the following:

1. *Submit all the program files* to Moodle. **Programs submitted through e-mail will be rejected! It is your responsibility to submit programs successfully.**
2. Please don't be late, because grading needs to be completed quickly. If your program does not work properly or if it does not compile, state this clearly in the documentation, to be submitted through Moodle together with the source code.
3. **No programs or program documentation files will be accepted after Monday, April 23rd, due to the grading deadline at the end of the semester.**

Academic Honesty:

Each student is expected to develop his/her assignment alone. **Do not share programs, or program parts, with your colleagues**. Violators of this policy will be held responsible for academic dishonesty, and will bear consequences in accordance to the rules and regulations of University of North Carolina at Charlotte.