CSC415 OPERATING SYSTEM PRINCIPLES

Assignment 2
List of Functions

For this assignment, you will write a program to offer a list of several functions. When your program is started, show a menu of three functions as follows:

```
1. Hello World!
2. List files
3. Exit
Please select:
```

After the user makes his choice by typing the index of the function, your program should read the user input into a 1024-byte buffer, and then execute the corresponding function.

1. If the user input is 1, create a new process with `fork()` and print out `Hello World!` in the child process. The parent process should wait for its child process to complete. Then reprint the menu and ask the user for choice.
2. If the user input is 2, create a new process with `fork()` and run "`ls`" in the child process to list all the files in the current directory. The parent process should wait for its child process to complete. Then reprint the menu and ask the user for choice.
3. If the user input is 3, the program terminates.
4. If the user input is not 1, 2, or 3, print out an error message, reprint the menu and ask the user for choice.

Hints:

1. You can use the function `fgets()` to read user input into a buffer. The declaration of `fgets()` is

   ```
   char *fgets(char *str, int n, FILE *stream)
   ```

   When `FILE *stream` is `stdin`, `fgets()` reads user input from standard input. You can find more description and examples of `fgets()` here
   http://www.tutorialspoint.com/c_standard_library/c_function_fgets.htm

2. After your program reads user input into the buffer, the user input is stored as a string. When you compare the user input with the indexs of the functions on the list, you may need to covert the user input from a string to an integer. You can use the function `atoi()`. The declaration of `atoi()` is

```
int atoi(const char *str)
```

You can find more description and examples of `atoi()` here
https://www.tutorialspoint.com/c_standard_library/c_function_atoi.htm

3. How to use `execlp()` to list all the files in the current directory can be found in the example in our lecture note. You can also find how to print out an error message there.

4. Remember to include necessary header files.

Here is a sample execution on a Linux machine:

```
1. Hello World!
2. List files
3. Exit
Please select: 1
Hello World!
1. Hello World!
2. List files
3. Exit
Please select: 2
file1       file2         file3
1. Hello World!
2. List files
3. Exit
Please select: 4
Invalid choice!
1. Hello World!
2. List files
3. Exit
Please select: 3
```

Name your code file as “functionlist.c”. Submit the source code file to the regular submission link on iLearn.