Assignment 1

Operating Systems, Fall 2017
Time: One week

Instructions: Do not copy material from other sources, if it is necessary, then provide the references. Plagiarized assignment will get negative marks, and can be called for DC action.

This is about using the `fork()` system call, and inter process communication (IPC). As discussed in the lectures, you are given an input file, and you have to produce some output files based in the inputs you read.
Your parent process is created by default and it reads the input file. The input file is a comma separated file, named `InputAssign-1.txt`. First row of the file lists the names of the processes you have to create. Each process name can be considered a column name of the table given to you in the file. The parent process will create all those processes. Then it will send messages to the created child processes. The contents of the messages are given in the second row of the file, under the column of each process as shown in the table below. The child processes will read the messages and save them in a file. The file's name will be same as process's name followed by '.txt'.

Proces_A	Process_B	Process_C
"Hello from parent"	"Hello process B"	"Hello C"
"Hello A"	"Hello 1"	"Hello 2"

This means that if the file contains the same data as shown in the table, then you will have three files created named **Process_A.txt**, **Process_B.txt** and **Process_C.txt**. The **Process_A.txt** will have two lines in it, the first line will say "Hello from parent" and second line will say "Hello A". Similarly the **Process_B.txt** will have two lines in it, the first line will say "Hello process B" and second line will say "Hello 1". Similar is the case for **Process_C.txt**.
Please note that you can be given any number of columns and rows in the file with arbitrary names. For example if a file has four columns then you will create four processes according to the names of columns. If you have 10 columns then you will create 10 processes, so on so forth.
The rows in the file can also be arbitrary. If there are 10 rows, then this means that each child process will receive 10 messages, and it will output 10 lines in its output file.

Submission The submission should contain only three following things. If there is one missing then the assignment will be given 0 marks. Additional things will also be penalized.

1. `Assign-1.cpp` the code of the assignment
2. `InputAssign-1.txt`
3. `Makefile` which makes executing `make` command in the respective directory compile the code into an executable.

After executing `make` an executable should be created named `assign-1.o`. Executing that executable should generate all the respective output files.

Further Information: For helping material and the syntax of the related APIs, refer to the lab material.

Example of forking and creating child processes is as follows.

```
# include <iostream>
using namespace std;
int main(void)
{
        int pid=0;
        pid = fork();
        if ( pid == 0)
        {
        // This is the child process;
        }
        else
        {
        // This is the Parent process.
        }
        return 0;
}
```

Example of Inter Process Communication using Pipes between Processes is as follows.

```
{
        int pdes[2];
        pipe(pdes);

        if ( pid == 0) // This is the child process;
        {
        close(pdes[1]);
        read(pdes[0]);  // read from parent
        }

        else // This is the Parent process.
        {
        close(pdes[0]);
        write(pdes[1]); // write to child
        }
        return 0;
}
```