CS 341 Assignment 1

Implementing a basic Shell

Due on 30 September 2017 (Saturday) 5:30 PM

No late submissions will be accepted. The assignment is group assignment. The groups that have been made for CS 342 are valid for this assignment.

In the first week lectures, we have seen the system call interface that an Operating System provides to the user programs to interact with the hardware resources. The objective of this assignment is to familiarize you with the system calls and the interface.

In this assignment, you are required to implement a shell (or command line interpreter) in C. The shell interface should provide the user a prompt after which he types in a command, and then the shell creates a separate child process that executes the command. After execution, it prompts for more user inputs until the user enters `exit` at the prompt. Your shell should be able to execute the following commands:

Command	Description
`cd <directory_name>`	Changes current directory if user has appropriate permissions
`ls`	Lists information about files I the current directory
`rm`	Deletes indicated files. Supports options –r, -f, -v
`history n`	Prints the most recent n commands issued by the numbers. If n is omitted, prints all commands issued by the user.
`issue n`	Issues the nth command in the history once again.
`<program_name>`	Creates a child process to run `<program_name>`. Supports the redirection operators > and < to redirect the input and ouput of the program to indicated files.
`exit`	Exits the shell

After implementing the basic shell that supports the above commands. You should add two advanced features to the shell:

1. Design a new command that provides a useful facility. As an example, consider a command `rmexcept <list_of_files>` which removes all files except those in `<list_of_files>` from the current directory.

2. Support a command `<program_name> m` that creates a child process to execute `program_name`, but aborts the process if it does not complete its operation in `m` seconds. (Hint: Use an appropriate routine from the library to deliver a SIGALRM signal after `m` seconds, and use a signal handler to perform appropriate actions.)

Grading related:

For this assignment you need to hand in the following:

- Your Source code and a Makefile for compiling your source code.

- A README file with documentation about your code. Give a brief overview on the overall structure of your code and any other specific feature that you would like to highlight. Don't miss to mention the problems and bugs in your code (if any). It should also contain the details of the team members in the beginning along with a brief description on the contribution of each of the team member.