Network Programming Assignment 1

Design Document

This is a bash created using system calls (pipes, exec, etc) which can replace the traditional linux shell.

Date Submitted:
October 10, 2019

Vidit Jain (2016A7PS0064P)
Jitvan Shukla (2016A7PS0083P)

Deliverables included: shell.c, makefile, functions.c, functions.h

How to run:

```
$ make
$ ./a.out
$ prompt>       Now you can start using the shell
$ prompt> exit    This command exits the prompt
```

Constraints:

1. Maximum size of input can be 100.
2. daemonize command runs c programs only.

Working of the code:

Once the prompt starts, it will work like a normal shell. Type ‘exit’ to return to your normal terminal. While writing commands, you can use as many whitespaces as you want. Writing an ‘&’ at the end of the command will send it to background.

Implementation:

The code is implemented as a parent-child relationship. The parent takes input from stdin and then after finding the path to it and finding the number of arguments, it passes it to the child.
Parent – has the job of taking input, parsing it (removing spaces, etc), calculating number of arguments.
Child – does the actual execution of the commands .

Getfile() – This functions finds the command you entered in the paths mentioned in PATH enviroment variable.

Parsecommand() - This   function is backbone of the code, this code redirects all commands to their respective functions.

Pipecommand() - This function uses n-1 pipes (if there are n ‘|’s) and joins them linearly so that multiple pipes are supported. Then it sends the command in between pipes to parsecommand, which handles the rest.

shmPipe() - This function implements the pipeline with shared memory. ‘S’. Uses the POSIX shared memory descriptor to transfer output from one command to another

doubleshm() - This function handles double pipe with shared memory. ‘SS’

Redirection implemented for operators ‘<’ , ‘>’ and ‘>>’

Daemonize takes a c program and runs it as a daemon. i.e. It creates a child process, kills the parent process, make a new session using setsid and then runs the program without a controlling terminal.

messageQueuePipe()- This function implements the pipeline with message queues.’#’.

doubleMessageQueue()-This function implements double pipe with message queues. ‘##’