QUASH

Quite A Shell

Blake Morrell

Matthew Felsen

**Abstract**

Quash (Quite a shell) is a Unix based text interface. Its commands and functionality are modeled after popular Linux programs like Bash and Shell.

**Features**

|  |  |
| --- | --- |
| Command Name | Description |
| cd | Changes the current working directory |
| echo | Prints out the value of environment variables or any string following command |
| exit | Leaves the Quash interface |
| jobs | Lists all the processes running in the background |
| pwd | Displays absolute path of the current working directory |
| set | Sets the value of a environment variable |
| quit | Leaves the Quash interface |

|  |  |
| --- | --- |
| Symbol | Description |
| & | Background execution |
| < | Redirect standard input from a file |
| > | Redirect standard output to a file |
| | | Pipe - Unilateral communication between processes |

**Implementation**

int main(int argc, char\*\* argv, char\*\* envp)

Responsible for setting up environment. main() begins by masking signals and setting up necessary dependencies for redirecting file I/O. Also passes arguments like environment variables to Quash.

void quash\_run(m\_command\* cmd, char\*\* envp)

Handles commands and symbols passed in from main and directs the proper function to execute for the given command or symbol.

bool m\_get\_command(m\_command\* cmd, FILE\* in)

Takes command cmd and splits it into tokens to be easier for other implementations. The tokens are stored in a structure called m\_command which contains the raw string, the length of the string, an array of the tokens, and the length of the array.

int prim\_cmd(m\_command\* cmd, char\* envp[])

This function runs commands stated in the table that are free of symbols. This is done using the execvpe() system call. This function also uses execvpe() to pass variables into the environment.

int input\_io\_cmd(m\_command\* cmd, bool io, char\* envp[])

Same implementation as prim\_cmd(), but has additional handling for file I/O redirection symbols. (<, >)

int exec\_backg\_command(m\_command\* cmd, char\* envp[])

This function executes commands with the background execution symbol (&). This is done by creating a child process which inherits the same environment as its parent.

int m\_cmd\_background(m\_command\* cmd, char\* envp[])

This function runs commands with the pipe symbol ( | ). In this function, a pipe is created to unilaterally communicate from one process to another.

**Testing**

­ To test functionality, the given test cases were used. This was done by using file redirection of the .txt files into quash. All test cases passed except command7.txt. New variables could be added to the environment but said variables could not be inherited by the child process.

|  |  |
| --- | --- |
| Test Case | Output |
| Test1 | Linux |
| Test2 | Linux matthew-VirtualBox 5.3.0-45-generic #37~18.04.1-Ubuntu SMP Fri Mar 27 15:58:10 UTC 2020 x86\_64 x86\_64 x86\_64 GNU/Linux |
| Test3 | ./  /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:./ |
| Test5 | /home/matthew/Documents/EECS678\_ProjectQUASH  /usr |
| Test6 | bash6.txt command3.txt command7.txt ls NOTES quash.h  command1.txt command5.txt expected\_output ls.c quash quash\_test.h  command2.txt command6.txt file7.c Makefile quash.cREADME.md  My Dummy program is run |
| Test7 | Command: "./file7" not found. |