Quash (v.1)

Author(s)

Mezisashe Ojuba, 2023.

Description

Quash is a simple implementation of command shell in Linux environment. Quash is implemented in C and can execute some in-build commands as well as commands which require that a corresponding program be loaded and executed from the file system. Quash can also perform basic file output redirection.

Design Choices

Quash is implemented completely in C.

Quash loops repeatedly, printing the current directory to prompt the user, taking a text input, and executing the command based off the input if applicable until the exit command is run or the terminal is closed. When the user types in a command, the input string is tokenized by separating the values by spaces, and the tokens are stored in an string array. The first token is the command while following tokens, if any, are arguments.

There are both in-built commands and commands which require that a corresponding program be loaded and executed from the file system.

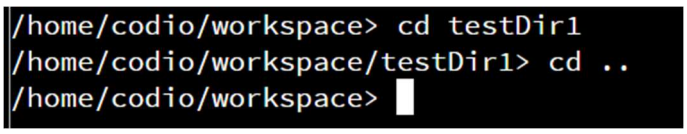
In-built Commands

In-built commands are directly executed in the process hosting Quash in most cases. In this version of Quash, flags are not implemented for in built commands.

There are six in-built commands in Quash as follows:

- cd <dir>: This changes the current working directory to the directory specified in <dir>.

Example:



- pwd: This prints the current working directory to standard output.

Example:

```
/home/codio/workspace> pwd
/home/codio/workspace
/home/codio/workspace>
```

- echo <string >: This prints the string as-is including any quotation marks to standard output, except for environment variables within the string which are indicated by starting with a $ symbol. In case of environment variables, the corresponding environment variable value replaces the variable in the printed string if it exists. If there is no environment variable named such, a null string is printed in its place.

Example:

```
/home/codio/workspace> echo "This is in $LANG language"
"This is in en_US.UTF-8 language"
/home/codio/workspace>
```

- env OR env <variable name>: When not followed by a variable name, env commands prints out all environment variables in order to standard output. When followed by a variable name, env command prints only the corresponding value to that variable to standard output.

Example

```
codio@monacolaptop-heavenbrenda:~/workspace$ ./shell
/home/codio/workspace> env PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:
/usr/local/games:/usr/share/codio-tools
/home/codio/workspace> env
0: CODIO_HOSTNAME=monacolaptop-heavenbrenda
1: TERM=xterm
2: SHELL=/bin/bash
3: SSH_CLIENT=192.168.10.226 50162 22
4: OLDPWD=/home/codio
5: SSH_TTY=/dev/pts/2
6: USER=codio
7: LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=
40;33;01:cd=40;33;01:or=40;31;01:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=
34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arj=01;31:*.taz=01;31:
*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.zip=01;31:*.z=01;31:*
.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01
;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01
;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.ace=01;31:*.zoo=01
;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01
;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01
;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=
01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v
=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp
4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm
=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl
=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=
01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=0
0;36:*.flac=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc
=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.axa=00;36:*.oga=00;36:*.spx=
00;36:*.xspf=00;36:
```

- export <variable name>=<value> OR setenv <variable name>=<value>: export and setenv commands add a new environment variable with the given name and corresponding value if no such variable exists and overwrites the value if the variable already exists.

Example

```
/home/codio/workspace> export time=evening
/home/codio/workspace> env time
evening
```

- exit: This closes the Quash shell.

Program Commands

Program commands are executed by forking off a process and delegates the execution of the command to the child process. The child process passes in the command and arguments array to the execvp command. This loads and executes the corresponding program from the file system.

The parent process (in which the Quash loop is being executed) waits for the completion of the command before printing the prompt and taking in another instruction. However, if a command ends with the & token, Quash executes it in the background – the parent process does not wait for the completion of the execution of the command before printing the prompt, taking and executing other commands.

Commands that run for more than 10 seconds are automatically terminated. The message "Terminated process <process ID> took too long to finish" is printed to standard out.

Output Redirection

In this version of Quash, the > redirection is implemented.

In its most basic form, it is used in this format:

command > *file_name* OR *command* > *file_descriptor*

This redirects the output which would have been printed to the standard output (or command line) to be written to the specified file with the given name or to the stream specified by the file descriptor. If such a file already exists, its content is overwritten.

This version of Quash does not support 1>, 2>, >>, or &> redirections.

Citations

[1] Inge, Anike (Mar 20, 2013) StartsWith [func]. *https://stackoverflow.com/questions/15515088/how-to-check-if-string-starts-with-certain-string-in-c*.

[2] The Test Coder (Oct 27, 2021) How to Slice String in C Language? [article]. *https://medium.com/@kkhicher1/how-to-slice-string-in-c-language-7a5fd3a5db46*.