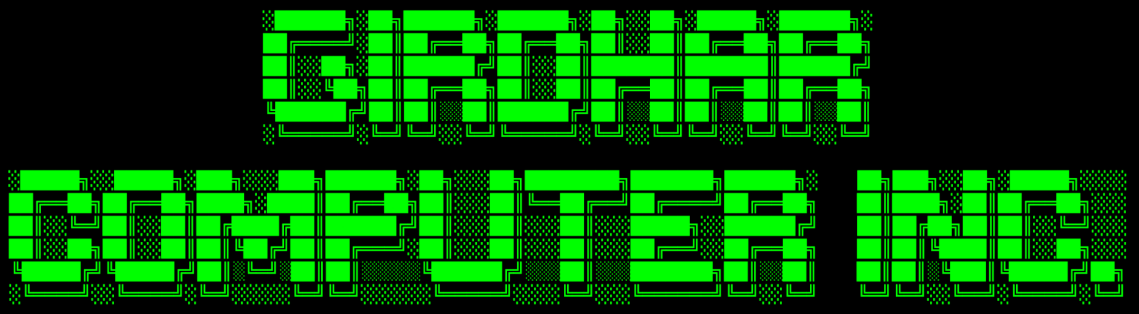

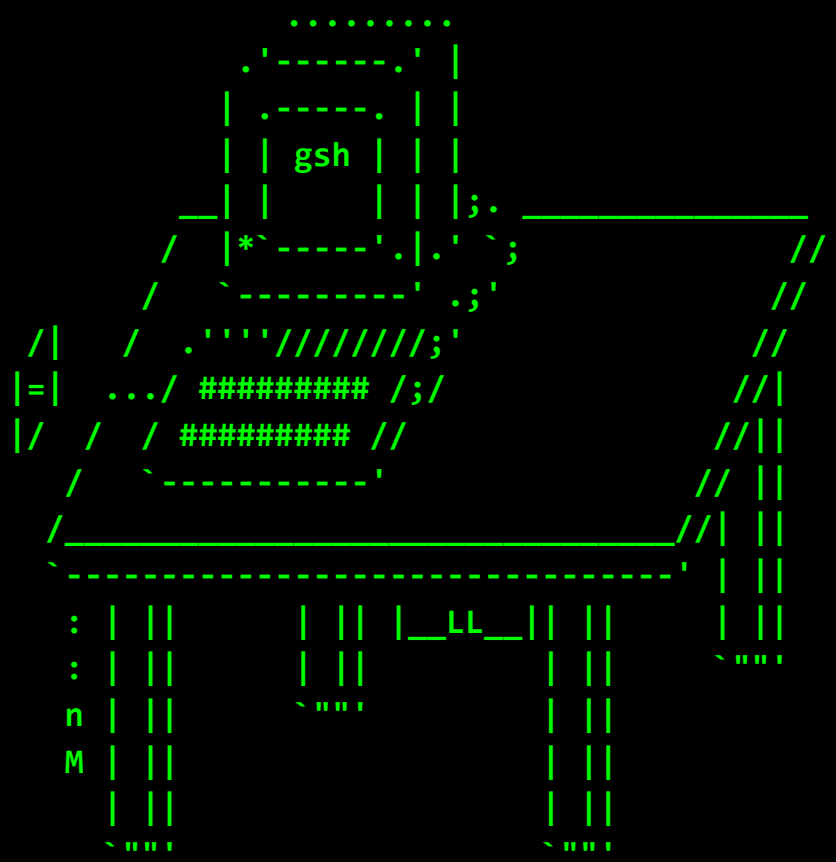

gsh-manpage

v0.5.1

Contents

1. Introduction to gsh

gsh (verbose: The Girdhar Shell) is a state-of-the-art unix based shell designed in a world-class research facility called ‘Old Boys Hostel’ at IIIT-Delhi.

2. Setup using ‘make’

To setup gsh to run on your system, go to the directory where the gsh binaries are located and type

$ sudo make gsh

OR

$ sudo make

(sudo necessary for certain privileged actions during setup)

3. Interface Overview

Users with previous shell experience should feel right at home. For every input cycle, there is persistent ‘gsh:’ branding in green, along with the current directory printed in blue, (colored using UNIX color code conventions) followed by a ‘$’ sign and an input prompt.

4. Supported commands and options

Here’s a list of the supported commands and their valid options on gsh. This list can be accessed anytime by typing

gsh: $ help

The output is

The Girdhar Shell, version 0.5.1; release (x86_64-linux-gnu)
Designed for CSE231: Operating Systems, IIIT-Delhi

Internal Commands

cd [options] [DIR]

Description:

Changes current directory to DIR.

Options:

-L: Force symbolic links to be resolved (default)

-P: Use the physical directory structure without following symbolic links.

echo [options] [args …]

Description:

Displays arguments followed by newline on stdout.

Options:

-n: Omits the newline at the end

--help: Prints the gsh-manpage for ‘echo’

pwd [options]

Description:

Print the current working directory

Options:

-L: Print the value of $PWD

-P: Print the physical directory, without any symbolic links

External Commands

cat [options] [FILE]

Description:

Concatenate FILE to standard output.

Options:

-E: display ‘$’ at the end of each line.

-n: number all output lines

date [options] [args …]

Description:

Print the system time & date.

Options:

-u: Prints the UTC time.

-d: Followed by arguments ‘today’, ‘tomorrow’ & ‘yesterday’, prints the date & time for today, tomorrow and yesterday respectively.

ls [options] [DIR]

Description:

List all contents of directory DIR.

Options:

-a: (all) Do not ignore entries starting with .

-A: (almost-all) Do not list implied . and ..

mkdir [options] [DIR]

Description:

Create directory DIR if it doesn’t already exist.

Options:

-v: (verbose) print confirmation message

-m: (mode) choose directory mode (eg. 777)

rm [options] [FILE]

Description:

Remove file specified in FILE.

Options:

-v: (verbose) print deletion message

-i: (interactive) prompt before deletion

5. POSIX multithreading mode

All the commands mentioned in the previous section can be executed in POSIX multithreading mode by appending the command-name with ‘“&t”. Taking an example, to execute ‘mkdir’ in multi-threaded mode, you type in ‘mkdir&t’, other arguments remaining identical.

6. Documentation of errors and bugs

Defensive programming has been a crucial paradigm I’ve followed throughout the development of gsh. Here are errors/bugs I’ve degraded gracefully such that user experience isn’t compromised.

Internal Commands

cd [options] [DIR]

1. **The user may enter an invalid path**
   The shell prints out “cd: No such file or directory”
2. **The user may enter the wrong flags**
   The shell prints out “cd: Invalid options”
3. **The user may provide insufficient arguments**
   The shell prints out “cd: Too few arguments”

echo [options] [args …]

1. **The user may enter with/without double quotes (“”)**
   Support for both has been added. When double quotes are present, they’re dropped when they are “echoed”.
2. **The user may enter the wrong flags, but it’s alright**
   A wrong flag is treated as a string and printed.
3. **The user may provide insufficient arguments**
   The shell prints out “echo: Too few arguments”

pwd [options]

1. **The user may enter an invalid option**
   The shell prints out “pwd: Invalid option [option-name]”
2. **The system may not have the required privileges**

The shell prints out “pwd: unsuccessful, kindly check permissions”

External Commands

cat [options] [FILE]

1. **The user may provide an invalid file name**
   The shell prints out “cat: No such file or directory”
2. **The user may enter the wrong flags**
   The shell prints out “cat: Invalid arguments”
3. **The user may provide insufficient arguments**
   The shell prints out “cat: Too few arguments”

date [options] [args …]

1. **The user may enter an invalid identifier after ‘-d’ flag**
   The shell prints out “date: invalid identifier ‘[id]’”
2. **The user may enter the wrong flags**
   The shell prints out “date: invalid flags/syntax”

ls [options] [DIR]

1. **The user may enter an invalid path**
   The shell prints “ls: [DIR]: no such file or directory”
2. **The user may use ls without specifying the directory**
   Current directory ‘.’ is loaded as the default argument

mkdir [options] [DIR]

1. **The user may enter an invalid path/invalid flags**
   The shell prints “mkdir: error, check arguments again”
2. **The user may enter invalid mode after ‘-m’ flag**
   The shell prints “mkdir: invalid mode entered”
3. **The user may provide insufficient arguments**
   The shell prints “mkdir: too few arguments”

rm [options] [FILE]

1. **The user may try to remove a protected file**
   The shell prints “rm: [FILE]: file could not be deleted”
2. **The user may enter case-insensitive prompt after ‘-i’**
   The shell can handle both ‘y’, ‘Y’ and ‘n’, ‘N’
3. **The user may provide insufficient arguments**
   The shell prints “rm: too few arguments”

Apart from error-handling for each command, if the user enters and invalid/unrecognized command, the shell prints “[COMMAND]: command not found”

7. Cleaning-up/Reinstalling

To revert back the changes made during gsh setup, type the following command in your main shell:

$ sudo make gsh-clean

This sequence is automatically executed when you type ‘exit’ in gsh. However, the functionality is provided separately in case the system crashes while gsh is operational.

If executing this command results in an error, gsh has already been cleaned during its last run, so don’t worry.

I hope you like this little piece of software I wrote over the weekend.

Thank you for using gsh!

EOF