

CLI Basics

Help and Documentation



Basic Help

- Cli commands are generally typed in this format: <command> <options> <arguments>
- **Some commands** have **built-in help** “options”, usually -h, or -help, or --help
- There are also Comments in config and example configuration files
- For more detailed help **read the manual**

Example:

```
user@system: ls --help
```

```
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
-a, --all                  do not ignore entries starting with .
-A, --almost-all          do not list implied . and ..
    --author                with -l, print the author of each file
-b, --escape                print C-style escapes for nongraphic characters
    --block-size=SIZE       scale sizes by SIZE before printing them; e.g.,
                           '--block-size=M' prints sizes in units of
                           1,048,576 bytes; see SIZE format below
-B, --ignore-backups        do not list implied entries ending with ~
-c                           with -lt: sort by, and show, ctime (time of last
```

Manual (man pages)

Section	Description
1	General commands
2	System calls
3	Library functions , covering in particular the C standard library
4	Special files (usually devices, those found in /dev) and drivers
5	File formats and conventions
6	Games and screensavers
7	Miscellanea
8	System administration commands and daemons

- Unix manuals were originally printed in multiple volumes and provided to customer
 - Broken down into sections
 - For example User commands are in section 1
- “**man pages**” are now software documentation included in the OS
 - **Preferred way** of looking up commands and documentation

```
user@hostmane: man <command>
user@hostname: whatis <command>
Example:
user@hostname: whatis ls
ls (1)          - list directory contents
user@hostname: whatis grep
grep (1)        - print lines matching a pattern
```

Online Documentations

- **Info** was to be an easier-to-navigate version of **man-pages**
 - Try the “info” command in your VM to see what it does
 - Since the web and internet are widely available the project has stalled
- Many Unix/Linux vendors and GNU/Linux communities maintain **web-based documentation** for their distributions
 - Example: <https://manpages.ubuntu.com>
- Specific applications may have their own manual
 - Example: Ubuntu Server Guide at <https://ubuntu.com/server/docs>
 - Provides guides on how to set up various services, tools, and various configurations
 - Apache Documentation: <http://httpd.apache.org/docs/>

File system



Files and paths

- Everything in the file system is under the “root” /
- typical types of files: **regular file**, **directory**, **symbolic link**
 - See: <https://linuxconfig.org/identifying-file-types-in-linux>
 - **Remember**: file types in Linux are NOT images, documents, dll, etc...
 - A directory is just a special file
 - Contains a list of files that are inside the directory
- Two ways of addressing a file:
 - **Absolute Path**
 - /home/User/Documents/file1
 - **Relative Path** (relative to current working directory)
 - ./Documents/file1

File and Directory Permissions

- Read (**r**), Write (**w**), Execute (**x**) permissions can be specified for each file in the file system
- The r, w, and x can also be assigned for the **owner**, **group**, and **others** for each file
- If your user account does not have permission to read, write, or execute a file, the system will **deny access**.
- **Root** account has access to **all files**!
 - **Limit the use of your root account**, use your regular user for all regular tasks, do not “sudo” or “su” unnecessarily

Common commands

- List directory content: **ls**
- Change Directory: **cd**
- Make directory: **mkdir**
- Remove files and directories: **rm**
- Create a new empty file or change timestamp on existing file: **touch**
- Copy: **cp**
- Move: **mv**

Navigation using shortcuts

- ~ (tilde) refers to user's home directory
 - This refers to a variable called \$HOME in bash
 - Example: `cd ~`
 - ~<username> to get to that users' home directory
- . (dot) refers to current directory
 - Example: `cd ./Documents`
- .. (dot dot) refers to one directory above
 - Example: `cd ..`

More file commands

- more and less!
- cat
- grep
- echo

```
user@hostname:~$ whatis more
more (1)          - file perusal filter for crt viewing
user@hostname:~$ whatis less
less (1)          - opposite of more
user@hostname:~$ whatis grep
grep (1)          - print lines matching a pattern
user@hostname:~$ whatis cat
cat (1)           - concatenate files and print on the standard output
user@hostname:~$ whatis echo
echo (1)          - display a line of text
```

Working in bash



Bash Trickery!

- Tab key to **auto-complete** command, path, or filename
- Up and down arrow keys for **command history**
- Special characters:
 - ***** (asterisk) Matches zero or more characters
 - `ls *.txt`
 - **?** (question mark) Matches any single character
 - `ls ????`
 - **[]** (square brackets) Matches a single character in a range
 - `ls [a][0-9]*`
 - **\$** (dollar sign) Identifies a variable
 - **~** (tilde character) Represents the user home directory
 - `cd ~/Documents`

```
user@hostname:~$ touch a1.txt
user@hostname:~$ touch a874.txt
user@hostname:~$ ls [a-z][0-9]*
a1.txt  a874.txt
user@hostname:~$ rm a*
user@hostname:~$ ls [a-z][0-9]*
ls: ... No such file or directory
```


Bash Trickery!

- **Quoting special characters**

- `\` (backslash): turns off the special meaning of its following character
 - `echo \SPATH` outputs `$PATH` and will ignore function of `$`
- `"` (double quotes): cause most metacharacter special meanings to be ignored
 - `echo "date"`
- `'` (single quotes): negate the translation of all special characters
 - `ls '?'` outputs actual filename matching `"?"`
- ``` (backquote): enable command substitution to occur
 - Exactly opposite of `'quotes'`! it will take the input as commands
 - `echo `date`` outputs the date

Command Redirection

- Redirecting input from or output to other sources such as file, network, device, etc. (other than Standard input or keyboard)
- redirection Symbols
 - < (take input from ...)
 - > (output to ...)
 - >> (output to and append to an existing file)
 - Pipe: (|) takes the output of a command and connects it to the input of another command
 - ex. `ls -la | more`