

The Unix Shell: Basic Commands¹

Introducing the Shell

- A shell is a program whose primary purpose is to read commands and run other programs.
- The shell's main advantages are its high action-to-keystroke ratio, its support for automating repetitive tasks, and its capacity to access networked machines.
- The shell's main disadvantages are its primarily textual nature and how cryptic its commands and operation can be.

Navigating Files and Directories

- The file system is responsible for managing information on the disk.
- Information is stored in files, which are stored in directories (folders).
- Directories can also store other directories, which forms a directory tree.
- `cd path` changes the current working directory.
- `ls path` prints a listing of a specific file or directory; `ls` on its own lists the current working directory.
- `pwd` prints the user's current working directory.
- `/` on its own is the root directory of the whole file system.
- A relative path specifies a location starting from the current location.
- An absolute path specifies a location from the root of the file system.
- Directory names in a path are separated with `/` on Unix, but `\` on Windows.
- `..` means 'the directory above the current one'; `.` on its own means 'the current directory'.

Working With Files and Directories

- `cp old new` copies a file.
- `mkdir path` creates a new directory.
- `mv old new` moves (renames) a file or directory.
- `rm path` removes (deletes) a file.
- `*` matches zero or more characters in a filename, so `*.txt` matches all files ending in `.txt`.
- `?` matches any single character in a filename, so `?.txt` matches `a.txt` but not `any.txt`.
- Use of the Control key may be described in many ways, including `Ctrl-X`, `Control-X`, and `^X`.
- Most files' names are `something.extension`. The extension isn't required, and doesn't guarantee anything, but is normally used to indicate the type of data in the file.

¹ Based on <https://swcarpentry.github.io/shell-novice/reference/>

Pipes and Filters

- `cat` displays the contents of its inputs.
- `head` displays the first 10 lines of its input.
- `tail` displays the last 10 lines of its input.
- `wc` counts lines, words, and characters in its inputs.
- `command > file` redirects a command's output to a file (overwriting any existing content).
- `command >> file` appends a command's output to a file.
- `<` operator redirects input to a command
- `first | second` is a pipeline: the output of the first command is used as the input to the second.

Process Management

- `ps` lists processes
- `top` dynamic list of processes, shows CPU and memory usage
- `kill` kills a process
- `killall` kills multiple processes
- `&` at the end of a command executes it in the background

Finding Things

- `find` finds files with specific properties that match patterns.
- `grep` selects lines in files that match patterns.
- `--help` is a flag supported by many bash commands, and programs that can be run from within Bash, to display more information on how to use these commands or programs.
- `man command` displays the manual page for a given command.
- `$(command)` inserts a command's output in place.