Using the shell

Johan Öhlin

26 October 2021

Yabs

Me

Johan Öhlin

At Yabs since 2017

Primarily running GNU+Linux since 2008

The terminal

Started as a keyboard driven user interface to the operating system's kernel

Grown to not primarily not communicate directly with the kernel

More 'advanced' scripts to automate chores

“Shell” vs. “terminal”

Pretty much used interchangeably

Terminal (emulator) controls few things and executes shell

Shell interprets ‘commands’ and their interfaces

Bash

Bourne-Again Shell (GNU, 1989)

Replacement of Bourne Shell (Bell, 1979)

De-facto standard on GNU+Linux systems

Available on Mac and Windows

Alternatives

`csh`	C shell	1978–
`fish`	Friendly interactive shell	2005–
`ksh`	Korn shell	1983–2012
`zsh`	Z shell	1990–

File system

All directories has itself in it, called `.`

All directories has previous directory in it, called `..`

Your home directory is called `~`

Everything starts at the root directory, called `/`

"Everything is a file". Eg. `/dev/cdrom0`, `/dev/hda0`, etc.

Basics

`$ command [options and arguments]`

`cd`	Change directory	`$ cd .././dir`
`rm`	Remove	`$ rm secrets.gz`
`mkdir`	Create directory	`$ mkdir newdir`
`rmdir`	Remove empty directory	`$ rmdir olddir`
`touch`	Change file timestamps	`$ touch newfile`
`mv`	Move a file (also rename)	`$ mv olddir/file newdir`
`ls`	List files in directory	`$ ls ~`

Note that `cd` is a built-in command and not an external program.

Basics - Example

```
$ mkdir "bash course"
$ mkdir "bash course/day 1"
$ cd "bash course/day 1"
$ cd ..
$ rmdir "day 1"
$ touch "file 1" "file 2"
$ mv "file 2" "file 1"
$ rm "file 1"
$ mv "file 1" "../file"
$ rm "../file"
```

How to

A lot of programs make use of `--help` option. You can usually figure out how a program works by using it. (It can often be shortened to `-h`.)

`man`	Briefer **man**ual page	`$ man ls`
`info`	More extensive **info**rmation	`$ info ls`
`whatis`	Header of `man` page.	`$ whatis ls`
`tldr`	Web client for a project called tldr-pages; `tldr.ostera.io`	`$ tldr ls`

Online search engines.

File content

`cat`	Concatenates and prints files on the standard output	`$ cat file1 file2`
`head`	Outputs the first part of files	`$ head -n 20 file`
`tail`	Outputs the last part of files	`$ tail -n 20 file`
`less`	Outputs the first part of files	`$ head -n 20 file`

File content - Example

```
$ cat "file"
$ head -n 2 /dev/urandom
$ tail -n 2 "some file"
$ less /some/file
```

Finding programs

`command -v`	Finding location of program to run (from $PATH)	`$ command -v ls`
`which`	Finding location of program to run (from $PATH)	`$ which ls`

Piping

Communication between programs.

```
$ ls | wc -l | sort -u
```

Writing to file.

```
$ ls > file
```

```
$ > file ls
```

Reading from file.

```
$ wc < file
```

```
$ < file wc
```

Appending to file.

```
$ echo hello >> file
```

```
$ >> file echo hello
```

Piping

Combining execution.

```
$ ls; echo "hello"
```

First have to succeed.

```
$ ls && echo "hello"
```

Execute second if first fails.

```
$ ls || echo "hello"
```

Combine them.

```
$ (ls || echo "hello") && pwd
```

Core utils

A handful of GNU core utils 8,27.

IEEE Std 1003.1-2008 utilities

`date`	Prints or sets the system date and time	`$ date --date="fortnight"`
`dd`	Copies and converts a file	`$ dd if=/home/ubuntu.iso of=/dev/usb0`
`df`	Shows disk free space on file systems	`$ df -h dir/file.txt`
`du`	Shows disk usage on file systems	`$ du -h dir/file.txt`
`ln`	Creates a link (shortcut) to a file	`$ ln -s existing_file link`
`false`	Does nothing, but exits unsuccessfully	`$ false`

Core utils

`mktemp`	Creates a temporary file or directory	`$ mktemp`
`nl`	Numbers lines of files	`$ nl file`
`printf`	Formats and prints data	`$ printf "0x%x" 20`
`pwd`	Prints the current working directory	`$ pwd`
`seq`	Prints a sequence of numbers	`$ seq 1 7 32`
`shuf`	Generate random permutations	`$ shuf -u fil`
`sleep`	Delays for a specified amount of time	`$ sleep 30 1m`

Core utils

`sort`	Sort lines of text files	`$ sort -u file`
`stat`	Returns data about an inode	`$ stat file`
`tac`	Concatenates and prints files in reverse order line by line	`$ tac file`
`tee`	Sends output to multiple files	`$ tee file`
`tr`	Translates or deletes characters	`$ tr "\n" "\t"`
`true`	Does nothing, but exits successfully	`$ true`

Core utils

`wc`	Prints the number of bytes, words, and lines in files	`$ wc -l file`
`yes`	Prints a string repeatedly	`$ yes no`