Linux,the Bash
&
everything

Prelude

A short history of Tux.

“The Linux philosophy is:

‘Laugh in the face of danger’.

Oops. Wrong one.

‘Do it yourself’.

Yes, that’s it.”

(Linus Torwalds)

Linux is...

- ... an implementation of UNIX.
- ... written in the C programming language.
- “De gustibus et coloribus non disputandum est”: ...customizable to anyone’s likings.
- ...using GNU tools, a set of freely available standard tools for handling the operating system.
- ...adhering to POSIX.

Act I

...

Basics of the Linux operating system
aka
“Everything’s a file”

Basics: built-ins vs commands

`command :=` program, i.e. application, e.g. `less, cat, df, ...`

`built-in :=` built in capabilities of your shell, e.g. `cd, let, set, ...`

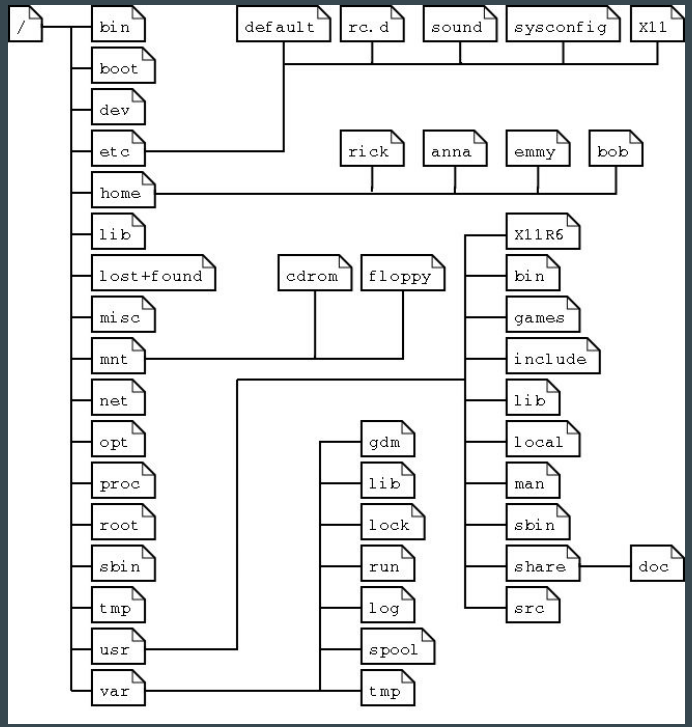
Basics - helping yourself aka RTFM

- `whatis <string>` : short explanation of `<string>`
- `man <command>` : display manual for a command
- `info <command>` : display info for a command (shortened `man`)
- `apropos <string>` : search whatis database for `<string>`
- `<command> --help` : display a short explanation on how to use `<command>`
- `help <builtin>` : display information about shell builtins

Basics: files

- `file <filename>` : display file type
- `stat <filename>` : display detailed file information
- `cat <filename>` : stream content of `<filename>` to `stdout`
- `less <filename>` : view `<filename>` pagewise (-> `less := pager`)
- `df` : reports file system disk space usage

The file system I: layout



The file system II: layout

`/bin` := binaries, i.e. programs

`/boot` := startup files and the kernel live here

`/dev` := references to peripheral hardware

`/etc` := most important config files go here

`/home` := user space

`/lib` := library files live here

`/mnt` := mount points for external file systems

The file system III: layout

`/net` := mount points for remote file systems

`/opt` := third party software lives here

`/proc` := virtual file system w/ information about system resources

`/root` := admin user home

`/sbin` := programs for use by the system and system admin

`/usr` := programs, libraries, docs, etc. for all user-related programs

`/var` := storage for variable files and temporary stuff, e.g. log files, mail queue, ...

The file system IV: inodes

- Every partition has it own file system
- In the file system a file is represented by an `inode`
- `inode :=` kind of serial number containing information about the actual data that makes up the file, e.g. owner, location on hard disk, etc.
- Every partition has its own set of inodes => throughout a system with multiple partitions files with same inode number can exist.
- Each inode describes a data structure on the hard disk, storing the properties of a file including the physical location of the file data.
- When a hard disk is initialized to accept data storage, e.g. on initial system setup, a fixed number of inodes per partition is created. This number represents the maximum amount of files of all types (incl. directories, symlinks, etc. that can exist at the same time on the partition.

The file system V: inodes

touch <file>

Create inode containing

i. Owner & group
ii. File type
iii. Permissions
iv. Date and time of creation, last read and change
v. Date and time this information has been changed in the inode
vi. Number of links to this file
vii. File size
viii. Physical address of file data

The file system VI: permissions

```
ls server.js
```

```
-rw-r--r-- 1 hp hp 1.6K Jul 19 14:22 server.js
```

Permissions: 3 read-write-execute triplets: user - group - others

`0 := - :=` the access right that is supposed to be on this place is not granted.

`4 := r :=` read access

`2 := w :=` write access

`1 := x :=` executable

The file system VII: users & groups

- Configured in `/etc/passwd`
- `id`
- `whoami`
- `chmod`
- `chown`

Interlude

•••

The many faces of Bourne

Shells

- `sh` aka Bourne Shell
  - Original shell used on UNIX systems.
- `bash` aka Bourne Again SHell
  - Standard GNU shell.
  - Intuitive and flexible.
  - De facto standard.
  - Superset of the Bourne Shell.
- `csh` aka C Shell
  - Syntax of this shell resembles C programming language.
- `ksh` aka Korn Shell
  - More UNIX-y superset of Bourne Shell than Bourne Again Shell.
- `zsh` aka Z Shell
  - Superset of the Bourne Shell.
  - Combines cherry picked features from `bash`, `csh` and `ksh`.

Bash and the file system: pathfinding

- `which <string>`
  - Looks through directories in the user's **search path** for files matching string
- `find <path> <expression>`
  - Powerful search tool
  - `find . -size +5000k`
  - `find . -name “*.tmp” -exec rm {} \;`
- `locate <expression>`
  - Easier to use but more restricted find
- `grep <pattern> <glob>`
  - Searches all files matching `<glob>` for `<pattern>`
  - Powerful matcher
  - Often used in combination with `find` and `locate`

Bash and the file system: how to view POIs

- “less is more”
- head
- tail

Act II

•••

Anatomy of a shell script

Recomended literature

- `man bash`
- Bash Programming HOW TO [Mike G Mikkey]
- Bash Cookbook [O'Reilly Media - C. Newham, JP Vossen, C. Albing]