

UNIX Command Line Interface (CLI)

Amitabha Sanyal

Unix Commands

- Unix has a set of predefined commands that help a user in interacting with the OS:
 - Manage files and directories.
 - Login to a remote system
 - Identify processes taking large CPU time, kill a hung process.
 - Printing a document.
- Variations of the command are introduced by switches or **options**.
- Commands can be combined by using operators.
- The interaction can take two forms:
 - **Command line Interface (CLI)**: Interaction with Unix from the command line in a terminal.
 - **Bash script**: Interaction by placing commands in a file.

Example of a command

`ls -ltr Dropbox`

```
graph TD; A[ls] --> B[command]; C[-ltr] --> D[options]; E[Dropbox] --> F[argument]
```

Meaning: List the contents of the directory Dropbox

- In the long format
- sorted by time
- in reverse order (latest at the end)

General syntax of a command

> `command option(s) argument(s)`

- In the command `ls -ltr Dropbox`, each of `l`, `t` and `r` is an option
- Some (but not all) options can be written in a verbose fashion:

```
ls -lt --reverse Dropbox
```

```
ls -l -t --reverse Dropbox
```

- The command to find information about commands is also a command called `man`
`man ls`: Gives information about `ls`

File management commands

| | |
|----------------------------------|--|
| <code>cat</code> | Concatenate files and displays the result |
| <code>cp, mv</code> | Copy, move (rename) files (or directories) |
| <code>ls, tree</code> | List files and directories. |
| <code>chmod, chgrp, chown</code> | Change access modes, group or owner on files. |
| <code>mkdir</code> | Create a directory. |
| <code>ln, ln -s</code> | Create a hard, symbolic link. |
| <code>pwd</code> | Print working directory |
| <code>rm, rmdir</code> | Remove files, directories |
| <code>less</code> | Display files by screenful, forward and backward |
| <code>wc</code> | Count lines, words, and characters. |
| <code>head, tail</code> | Show the first few lines (last few lines) of a file. |

ls -l

```
sanyal@sanyal:~/Dropbox/cs251/bash$ ls -al
total 120
```

| permission | no of links | owner | group | size | date | name |
|------------|-------------|--------|--------|-------|--------------|--------------------|
| drwxrwxr-x | 4 | sanyal | sanyal | 4096 | Aug 3 21:16 | . |
| drwxrwxr-x | 15 | sanyal | sanyal | 4096 | Jul 31 11:31 | .. |
| -rw-rw-r-- | 1 | sanyal | sanyal | 0 | Aug 3 21:01 | .a.txt |
| -rw-rw-r-- | 1 | sanyal | sanyal | 0 | Aug 2 18:53 | bar.txt |
| -rw-rw-r-- | 1 | sanyal | sanyal | 399 | Aug 3 21:16 | bash-examples.sh |
| -rw-rw-r-- | 1 | sanyal | sanyal | 262 | Aug 3 10:25 | bash-examples.sh~ |
| -rw-rw-r-- | 1 | sanyal | sanyal | 87349 | Aug 1 12:14 | bash-scribbles.xoj |
| -rw-rw-r-- | 1 | sanyal | sanyal | 0 | Aug 2 18:54 | foo.txt |
| -rw----- | 1 | sanyal | sanyal | 136 | Aug 3 21:04 | history |
| drwxrwxr-x | 3 | sanyal | sanyal | 4096 | Aug 2 19:27 | tmp |
| drwxrwxr-x | 3 | sanyal | sanyal | 4096 | Aug 2 19:40 | tmp1 |

Owner and group

- Owner and group
 - Each user is assigned a primary group
 - The system administrator can introduce a new group and populate the group with users
 - When a user creates a file he is the default owner and his primary group is the default group
 - The group can be changed with `chgrp`
 - Every other user in the same group has permission defined by the second triad.

permission

| Three permission triads | |
|-------------------------|---|
| first triad | what the owner can do |
| second triad | what the group members can do |
| third triad | what other users can do |
| Each triad | |
| first character | <input type="checkbox"/> r : readable |
| second character | <input type="checkbox"/> w : writable |
| third character | <input type="checkbox"/> x : executable |

```
-rwx--x--- 1 sanyal testgroup    136 Aug  3 21:04 test.sh
drwxrw-r-x 3 sanyal cs251        4096 Aug  2 19:27 tmp
```


permission

```
-rwx--x--- 1 sanyal testgroup 136 Aug 3 21:04 test.sh
```

- Not a directory (-)
- owner has read, write and execute permissions
- group member has execute permission
- others have no permission

```
drwxrw-r-x 3 sanyal cs251 4096 Aug 2 19:27 tmp
```

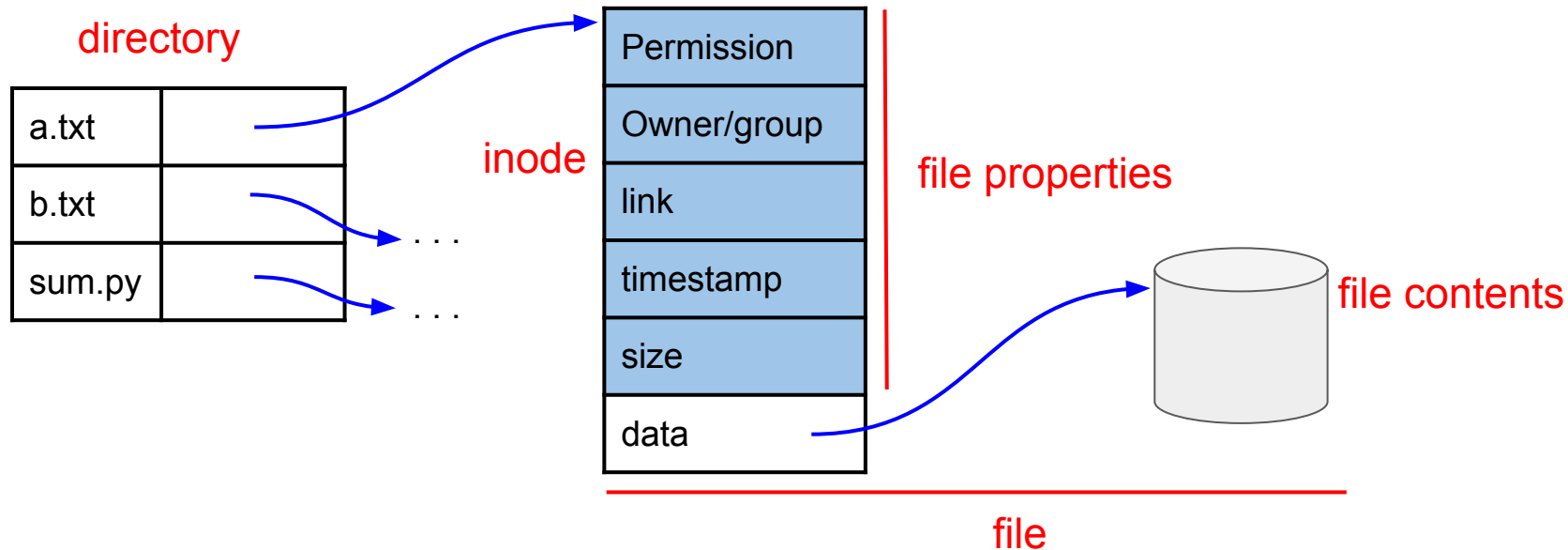
- Is a directory (d)
- owner can list files, change the directory and step into (cd) the directory.
- group members can just list files. The w bit is meaningless if x is not set.
- others can only list files and step into the directory. They can change files but not add or delete files in the directory

changing permission with `chmod`

`chmod NNNN filename`

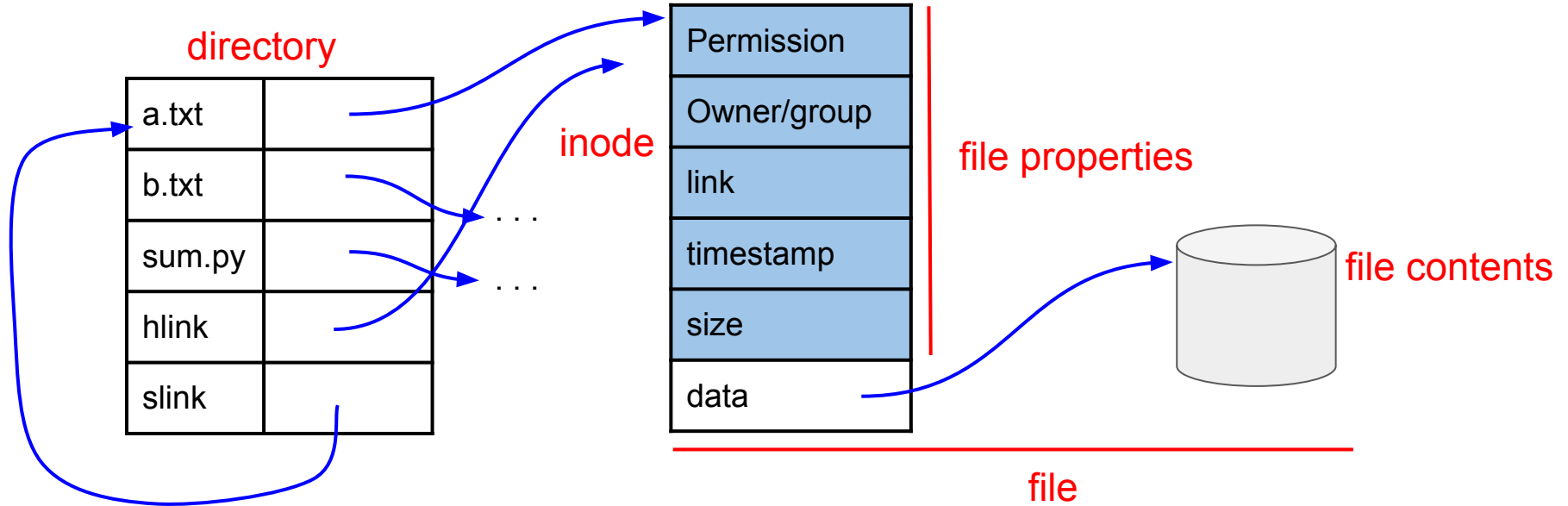
| Symbolic Notation | Numeric Notation | English |
|-------------------------|------------------|---|
| <code>-----</code> | 0000 | no permissions |
| <code>-rwx-----</code> | 0700 | read, write, & execute only for owner |
| <code>-rwxrwx---</code> | 0770 | read, write, & execute for owner and group |
| <code>-rwxrwxrwx</code> | 0777 | read, write, & execute for owner, group and others |
| <code>---x--x--x</code> | 0111 | execute |
| <code>--w--w--w-</code> | 0222 | write |
| <code>--wx-wx-wx</code> | 0333 | write & execute |
| <code>-r--r--r--</code> | 0444 | read |
| <code>-r-xr-xr-x</code> | 0555 | read & execute |
| <code>-rw-rw-rw-</code> | 0666 | read & write |
| <code>-rwxr-----</code> | 0740 | owner can read, write, & execute; group can only read; others have no permissions |

Filesystem representation



- A directory is essentially a table.
- A file contains properties of the file and a pointer to its contents.
- The **size** of a file is the size of the file contents.
- The **size** of a directory is the size of the table. 4096 bytes allocated at a time to store the table.

Hard links, soft links and the `ln` command



- `ln hlink a.txt` -- creates the hard link `hlink`
- `ln -s slink a.txt` -- creates the symbolic link `slink`
- Links are normally used to call a new version of a program with an old name.

File comparison and searching

| | |
|--|---|
| <code>cmp</code> | Compare two files, byte by byte. |
| <code>comm</code> | Compare items in two sorted files. |
| <code>diff</code> , <code>meld</code> , <code>diff3</code> | Compare two (three) files, line by line. |
| <code>find</code> | Search the system for files by name and take a range of possible actions. |
| <code>grep</code> | Search files for text patterns. |
| <code>which</code> | Print pathname of a command. |
| <code>whereis</code> | Find command. |
| <code>locate</code> | Search a pre-existing database to show where files are on the system. |

Comparing files with `diff`

- `diff file1.txt file2.txt`: Compares files `file1.txt` and `file2.txt`. Indicates changes that have to be made to `file1.txt` to make it identical to `file2.txt`.
- Example:

```
This is a first line
This is a second line
This is a third line.
This is the last line.
And that is all
```

```
This is the zeroth line.
This is a first line
This is a second line.
    This is a third line
This is the last line.
```

```
0a1
> This is the zeroeth line.
2,3c3,4
< This is a second line
< This is a third line.
---
> This is a second line.
> This is a third line
5d5
< And that is all
```

Comparing files with `diff`

file1.txt:

```
This is a first line
This is a second line
This is a third line.
This is the last line.
And that is all
```

file2.txt:

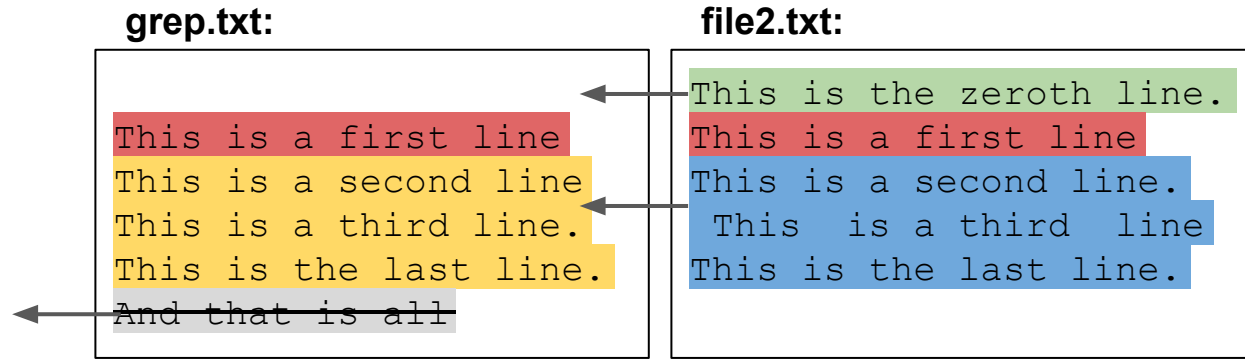
```
This is the zeroth line.
This is a first line
This is a second line.
    This is a third line
This is the last line.
```

- `0a1`: add at position 0 of the first file the line “This is the zeroeth line” to match line 1 of the second file.
- `2,3c3,4`: change lines 2,3 as shown below to match lines 3,4 of the second file.
 - < This is a second line
 - < This is a third line.
 -
 - > This is a second line.
 - > This is a third line
- `5d5`: delete line 5 (shown below) of the first file to match line 5 of the second file.
 - < And that is all

```
diff file1.txt file2.txt
```

```
0a1
> This is the zeroeth line.
2,3c3,4
< This is a second line
< This is a third line.
---
> This is a second line.
> This is a third line
5d5
< And that is all
```

Comparing files with `diff`



All changes are made to the first file.

Searching for patterns with `egrep` or `grep -E`

- `egrep`: Extented global regular expression print
- `egrep <pattern> <filename>`: Prints every line in `<filename>` containing the pattern described by `print`.
- Examples:

| | |
|--|--|
| <code>egrep line grep.txt</code> | Print all lines containing the pattern <code>line</code> |
| <code>egrep 'a f' grep.txt</code> | Print all lines containing <code>a f</code> . |
| <code>egrep 'first second' grep.txt</code> | Print all lines containing <code>first</code> or <code>second</code> |
| <code>egrep '(Th)?is' grep.txt</code> | Print all lines containing <code>This</code> or <code>is</code> |
| <code>egrep '[a-z]{2,5}' grep.txt</code> | ...strings made of characters <code>a to z</code> and length 2,3,4 or 5. |
| <code>egrep '[a-z]{5}' grep.txt</code> | ...strings made of characters <code>a to z</code> and length 5. |

Searching for patterns with `egrep` or `grep -E`

| | |
|--|--|
| <code>egrep '19(D O)050[0-9]{3}' Students.csv</code> | Students who had originally joined CSE |
| <code>egrep -v '19(D O)050[0-9]{3}' Students.csv</code> | Others |
| <code>egrep '\"S' Students.csv</code> | How many names start with S? |
| <code>egrep '\"([[:alpha:]]+){m}([[:alpha:]]+)\"'</code> <code>Students.csv</code> | How many names have <code>m</code> words in them? Notice the space after the + |
| <code>ls -al egrep '^d.*'</code> | Which files are directories |
| <code>egrep</code> <code>'^([a-zA-Z0-9_\\-\\.]+)@([a-zA-Z0-9_\\-\\.]+)\\.([a-zA-Z]{2,5})\$'</code> <code>emails</code> | Lines containing valid email addresses |

Difference between egrep and `grep`

- `grep` recognizes uses Basic Regular Expressions (BRE) which does not recognize the characters `{, }, (,), |` and `+` as special characters. They have to be “escaped” using `\`.

```
> egrep 'a|b' grep.txt
```

```
This is a first line  
This is a second line  
This is a third line.  
This is the last line.  
And that is all  
Funny line a|b
```

```
> grep 'a|b' grep.txt
```

```
Funny line a|b
```

System status

| | |
|-------------|------------------------------------|
| date | Display or set date. |
| df | Show free disk space. |
| du | Show disk usage. |
| env | Environment variables. |
| ps | Show processes. |
| stat | Display file or filesystem status. |
| uname | Display system information |
| kill, pkill | Terminate a running process. |

Disk partitions

- **Disk partitioning** is the creation of one or more regions, so that each region can be managed separately.^[2] These regions are called partitions.
 - On my disk: sda1,..., sda6 and sdb1,...,sdb6
- Different parts of the filesystem can be mounted on different partitions. On my system:
 - `/` is mounted on sdb8
 - `/boot` is mounted on sdb7
 - `/home` is mounted on sda6
- Separating user data from system data can prevent the system partition from becoming full and rendering the system unusable.

The df command

```
$ df -h
```

| Filesystem | Size | Used | Avail | Use% | Mounted on |
|------------|------|------|-------|------|------------|
| /dev/sdb8 | 61G | 28G | 31G | 48% | / |
| /dev/sda3 | 459G | 70M | 435G | 1% | /backup |
| /dev/sdb7 | 3.7G | 147M | 3.3G | 5% | /boot |
| /dev/sda5 | 477M | 3.7M | 473M | 1% | /boot/efi |
| /dev/sda6 | 1.3T | 62G | 1.2T | 6% | /home |

Text processing and others

| | |
|-----------------|--|
| cut | Extracts fields of a line separated by delimiters. |
| fmt | Convert text to specified width by filling lines and removing newlines. |
| paste | Merge corresponding lines of one or more files into tab-separated vertical columns and write to standard output. |
| sort | Sort lines by key |
| tr | Translate characters. |
| expand | Convert tabs in given files to an appropriate number of spaces |
| wget | Access files through the Internet |
| basename | Remove leading directory components from a path. |
| | |

References

[Cheat sheet](#) Just mug it up.

[Advanced bash scripting guide](#) Note, “advanced”.

[All about regular expressions](#) Bookmark this.

[The best that one can find about `find`](#)

[The `tr` command](#)

[Disk partitioning](#)

[Monitoring processes with `ps`](#)

[The `basename` command](#)

[`wget` it?](#)