

Introduction to Linux/Unix Commands

What is Unix?

- A multi-task and multi-user Operating System.
- Developed in 1969 at AT&T's Bell Labs.
- Some other variants: System V, Solaris, SCO Unix, SunOS, 4.4BSD, FreeBSD, NetBSD, OpenBSD, etc.

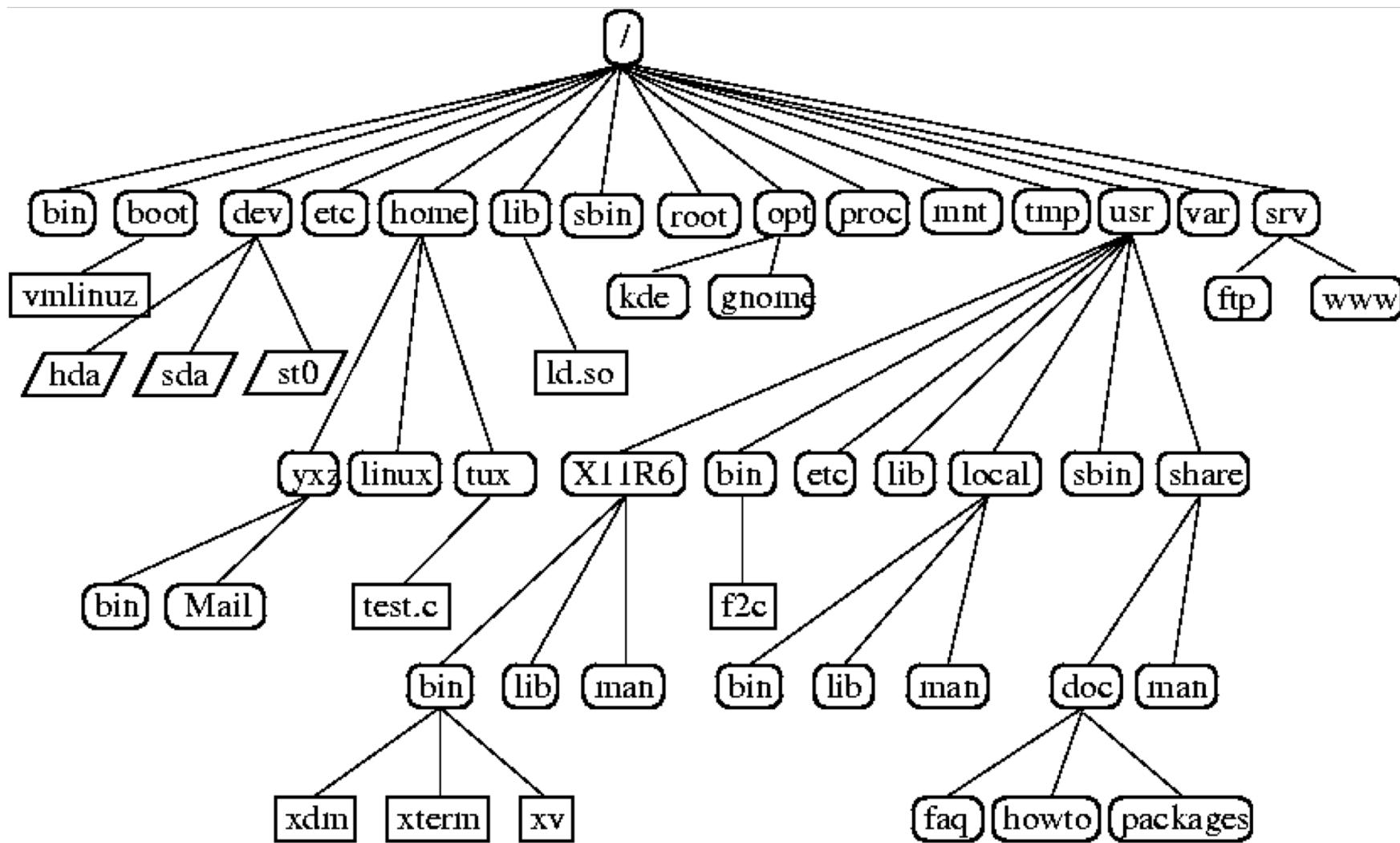
What is Linux

- A clone of Unix.
- Developed (Linux kernel) in 1991 by Linus Torvalds, who was a Finnish graduate student.
- Inspired by and a replacement of Minix.
- Linus' Minix became Linux.
- Originally developed for 32-bit x86-based PC and then was ported to other architectures.

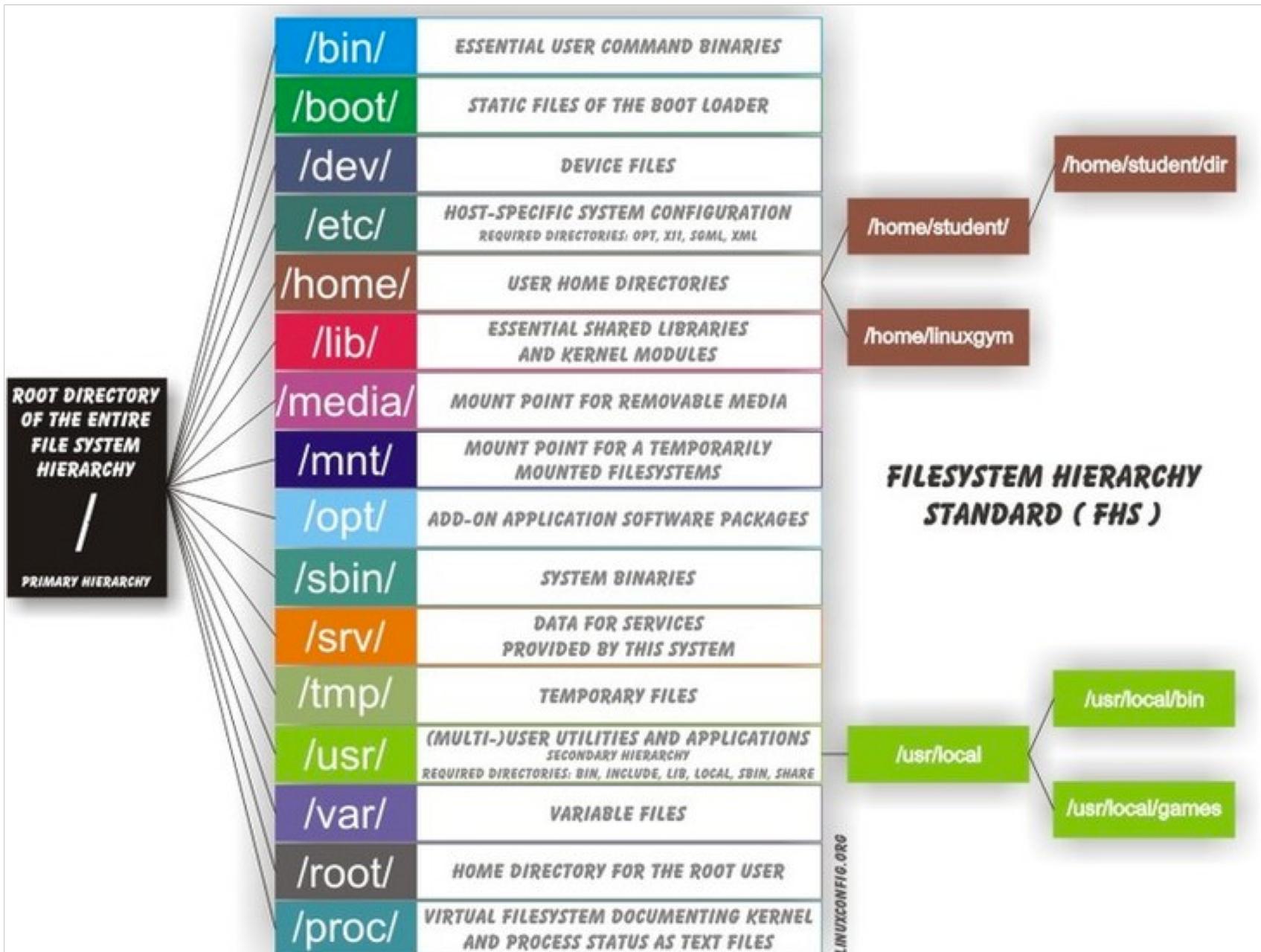
Many Linux Distributions Exist

- More than 300 Linux distributions
 - Redhat
 - RHEL (commercially supported)
 - Fedora (free)
 - CentOS (free RHEL, England)
 - Ubuntu (South Africa)
 - Mint (from Ubuntu)
 - Debian (GNU/Linux)
 - SuSe (German)
 - ...

Linux File Tree



Linux File Tree (cont'd)



The **man** Command

- Large number of Linux commands.
- Use the **man** command to display the **manual pages**, which come in handy since multiple flavors of Unix/Linux distributions' options and parameters may vary.
- Usage: **man \$command**
man followed by a **command** (for which you want help) and start reading. Press **q** to quit
- Example: **man whoami**

```
WHOAMI(1)          BSD General Commands Manual        WHOAMI(1)

NAME
  whoami -- display effective user id

SYNOPSIS
  whoami
```

Working with Directories

- **pwd**
 - print Working Directory.
- **cd**
 - change directory.
- **ls**
 - list directory contents.
- **mkdir**
 - make directories.
- **rmdir**
 - remove directories.

The `cd` Command

- `cd` (Change Directory) changes current directory to a different one.
- `cd /etc` changes directory to `/etc`
- `cd ~` goes to your home directory, e.g., `/Users/xy44`
- `cd` the same as `cd ~`
- `cd ..` goes to the parent directory, the one right above the current directory
- `cd -` goes to the previous directory

Absolute and Relative Paths

- When you type a path starting with a **slash** (/), then the **root** of the file tree is assumed as starting point.
- If you don't start your path with a slash, then the current directory is the assumed as the starting point.

Path Completion

- The **tab** key can help you in typing a path without errors.
- If your current directory contains a directory called Applications, typing **cd Ap** followed by the **tab** key will expand the command line to **cd /Applications/**.

The **ls** Command

- **ls** lists (current) directory contents.
- **ls -a** shows all files, including the **hidden files** (starting with a dot).
- **ls -l** displays the contents of the directory in long listing format, which contains file mode, number of links, owner name, group name, number of bytes in the file, time of file last modified, minute of file last modified, and file name.
- **ls -al** shows all files with long listing
- **ls -lh** shows all files with long listing in a more human readable format
- **ls -alh** three options (-a, -l, -h) applied together

The **mkdir** and Related Directory Commands

- **mkdir newDir** makes a new directory called newDir.
- **rmdir newDir** removes the newDir directory if it is empty.
- **rmdir -p newDir/newDir2/newDir3** recursively removes the directories if they are all empty.

Working with Files

Everything is a file in Linux. A directory is a special kind of file. The following commands are for file handling:

- **file** displays the file type.
- **rm** removes directory entries.
- **cp** copies files.
- **mv** moves files.
- **head** displays first lines of a file.
- **tail** displays the last lines of a file.
- **cat** concatenates and/or prints files.
- **touch** creates new empty files and change the timestamps on existing files and directories.

The **file** Command

- **file myfile** displays the file type of myfile.
- Linux does not use extensions to determine the file type. Your editor does not care whether a file ends in .TXT or .DOC.

The **rm** Command

- **rm myfile** removes a file called myfile and it is not recoverable.
- **rm -i myfile** requests confirmation before attempting to remove myfile.
- **rm -r myDir1** recursively force removes everything rooted at myDir1 (directories and files). Be careful!

The **cp** Command

- **cp source target** copies the source file to the target. If the target is a directory, then the source file(s) is copied to that target directory.
- **cp -r dir1 dir2** recursively copies all the files in all subdirectories of dir1 to dir2.
- **cp -n** prevents overwriting existing files.
- **cp -i** prompts a confirmation (Y/N) before copying a file that would overwrite an existing file.

The **mv** Command

- **mv old_file_name new_file_name** renames the file called **old_file_name** to a new name called **new_file_name**.
- **mv aFile dir** moves **aFile** from the current directory to another directory **dir**.

The **head** and **tail** Commands

- **head /ect/myfile.txt** displays the first 10 lines of the file myfile.txt.
- **head -n 20 /ect/myfile.txt** displays the first 20 lines of the file myfile.txt.
- **head -20 /ect/myfile.txt** displays the first 20 lines of the file myfile.txt.
- **tail /ect/myfile.txt** displays the last 10 lines of the file myfile.txt.
- **tail -n 20 /ect/myfile.txt** displays the last 20 lines of the file myfile.txt.
- **tail -20 /ect/myfile.txt** displays the last 20 lines of the file myfile.txt.

The **cat** Command

- The **cat** command copies standard input to standard output. It can be used to display a file or to concatenate files into a bigger one.
- **cat myfile** displays the contents of myfile.
- Concatenation example:

echo Cornell > f1

echo University > f2

cat f1 f2 >> f3 (concatenate f1 and f2 into bigger f3)

- **cat > summer.txt**

It's very hot! (followed by Ctrl-d to send EOF) creates a file called summer.txt containing “It's very hot!”

- **cat file1 > file2** creates file2 and copies file1 to file2 .

The **cat** Command

- Combine all the files in a directory into a single file called, newSingleFile:

cat * > newSingleFile

- Combine all the files whose names start with "ab" in a directory into a single file called, ab_all:

cat ab* > ab_all

- Combine all the files whose names contain "ab" in a directory into a single file called, al_ab_all:

cat *ab* > all_ab_all

The **touch** Command

- **touch file1 file2 file3** creates three new, empty files named file1, file2 and file3. If any of these files already exists, it merely changes the last access times for such file to the current time without overwriting.
- **touch -a file3** changes the access time on file3 to the current time. (CentOS)
- **touch -m file3** changes the modification time on file3 to the current time. (CentOS)
- **touch -am file3** changes both the access and modification times to the current time. (CentOS)
- **touch -t 201405011022 file1** changes the last access time of file1 to 10:22 a.m. May 1, 2014. (MacOS and CentOS)
- **touch -d '1 May 2014 10:20' file1** changes the last access time of file1 to 10:22 a.m. May 1, 2014. (CentOS)

The **find** and **grep** Commands

- **find**
 - Search for files in a directory hierarchy:
`find myfile`
`find Doc*`
`find *conf`
- **grep**
 - Searches an input file(s) in a directory for lines containing a match to a given pattern.

```
grep ty f3
```

`School: Cornell University`
`City: Ithaca`

```
grep -n Uni f3
```

```
1:School: Cornell University
```

`4:Country: United States`

Network Related

- **wget**
 - download a file from a website.
- **ping**
 - ping a host and display output the ping results.
- **whois**
 - get the domain information of site.

Network Related: **wget**

- **wget URL** downloads a file from the URL with its original file name.
- **wget -O new_file_name URL** downloads the file from the URL and write it to the new_file_name.
- **wget --output-document=new_file_name URL** (the same as above)

wget -o log_file_name URL downloads a file from the URL with its original file name and write log messages to log_file_name.

- **wget --output-file=log_file_name URL** (the same as above)

Network Related: **wget** (cont'd)

- **wget http://www.gutenberg.org/files/100/100-0.txt**
- **wget -O shakespeare.txt**
`http://www.gutenberg.org/files/100/100-0.txt`
- **wget --output-document=shakespeare.txt**
`http://www.gutenberg.org/files/100/100-0.txt`
- **wget -o log_shakespeare**
`http://www.gutenberg.org/files/100/100-0.txt`
- **wget --output-file=log_shakespeare**
`http://www.gutenberg.org/files/100/100-0.txt`

Network Related: ping

- ping cornell.edu

```
EN-SS-AXY44-E:~ xy44$ ping cornell.edu
PING cornell.edu (128.253.173.241): 56 data bytes
64 bytes from 128.253.173.241: icmp_seq=0 ttl=124 time=0.853 ms
64 bytes from 128.253.173.241: icmp_seq=1 ttl=124 time=1.396 ms
64 bytes from 128.253.173.241: icmp_seq=2 ttl=124 time=0.745 ms
64 bytes from 128.253.173.241: icmp_seq=3 ttl=124 time=0.623 ms
^C
--- cornell.edu ping statistics ---
4 packets transmitted, 4 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.623/0.904/1.396/0.295 ms
```

- ping 128.253.173.241

```
PING 128.253.173.241 (128.253.173.241): 56 data bytes
64 bytes from 128.253.173.241: icmp_seq=0 ttl=124 time=0.984 ms
64 bytes from 128.253.173.241: icmp_seq=1 ttl=124 time=0.973 ms
64 bytes from 128.253.173.241: icmp_seq=2 ttl=124 time=0.732 ms
64 bytes from 128.253.173.241: icmp_seq=3 ttl=124 time=0.625 ms
64 bytes from 128.253.173.241: icmp_seq=4 ttl=124 time=0.721 ms
64 bytes from 128.253.173.241: icmp_seq=5 ttl=124 time=0.698 ms
^C
--- 128.253.173.241 ping statistics ---
6 packets transmitted, 6 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.625/0.789/0.984/0.138 ms
```

Network Related: whois

- **whois cornell.edu**

```
EN-SS-AXY44-E:~ xy44$ whois domain cornell.edu

Whois Server Version 2.0

Domain names in the .com and .net domains can now be registered
with many different competing registrars. Go to http://www.internic.net
for detailed information.

Aborting search 50 records found .....
```

•
•
•

Domain Name: CORNELL.EDU

Registrant:

Cornell University
Cornell Information Technologies
Network Operations Center 729 Rhodes Hall
Ithaca, NY 14853
UNITED STATES

Name Servers:

BIGRED.CIT.CORNELL.EDU	128.253.180.35
DNS.CIT.CORNELL.EDU	192.35.82.53

Domain record activated: 15-Jul-1985

Domain record last updated: 27-Oct-2014

Domain expires: 31-Jul-2015

File Permission Related

- Permissions are managed in three distinct scopes or classes, which are known as **owner**, **group**, and **others**.
- Three type of permissions: **read (r)**, **write (w)**, and **execute (x)**.

rwxr-Xr--

owner group others

Three Permission Triads (modes)

1 st triad	what the owner can do
2 nd triad	what the group members can do
3 rd triad	what others can do

Each Triad

1 st character	r: readable
2 nd character	w: writable
3 rd character	x: executable

#	Permission	rwX
7	read, write and execute	111
6	read and write	110
5	read and execute	101
4	read only	100
3	write and execute	011
2	write only	010
1	execute only	001
0	none	000

The **chmod** Command

- It changes file modes or Access Control Lists.
- **chmod octal myfile** changes the permissions of file to octal, which can be found separately for user, group, and others.
- **chmod 777 myfile** changes myfile permissions to read, write, execute for all.
- **chmod 777 count1.txt** changes count1.txt's permissions from the current ones to rwxrwxrwx
- **chmod 755 myfile** rwx for owner, r-x for group and others.

File Compression Related

- **tar**
 - manipulates tape archives; creates and manipulates streaming archive files.
- **gzip**
 - conducts compression/decompression using Lempel-Ziv coding (LZ77)

The **tar** Command

- **tar -cf *file.tar* *files***
 - creates a tar named *file.tar* containing *files*
- **tar -xf *file.tar***
 - extracts the files from *file.tar*
- **tar -czf *file.tar.gz* *files***
 - creates a tar with Gzip compression
- **tar -xzf *file.tar.gz***
 - extracts a tar using Gzip
- **tar -cjf *file.tar.bz2***
 - creates a tar with Bzip2 compression
- **tar -xjf *file.tar.bz2***
 - extracts a tar using Bzip2

The **gzip** Command

- **gzip myfile**
 - compresses *myfile* and renames it to *myfile.gz*
- **gzip -d myfile.gz**
 - decompresses *myfile.gz* back to *myfile*

Other Useful Commands or Key Combinations

- **clear** clears the terminal screen
- **exit** logs out the current session
- **Ctrl-c** halts the current command
- **Ctrl-d** similar to **exit**
- **Ctrl-w** erase one word in the current line
- **Ctrl-u** erase the whole current line
- **!!** repeats the last command
- **history** lists command history
- **!history#** execute the #*-th* command in the command history.