



## Life on The Command Line

Home Page

Title Page



Page 1 of 24

Go Back

Full Screen

Close

Quit

**Satyajit Rai**  
**Chetan Vaity**

Kanwal Rekhi School of Information Technology,  
Indian Institute of Technology,  
Bombay.

<http://www.it.iitb.ac.in/~satyajit/>  
<http://www.it.iitb.ac.in/~chetanv/>



## Some Basic Commands

- `pwd` - show present working directory
- `ls` - list files in a directory
- `cd` - change directory
- `mkdir` - create new directory
- `clear` - clear screen
- `who` - people logged in
- `whoami` - I am logged in as ???
- `exit`, `Ctrl-D` - log out

[Home Page](#)

[Title Page](#)



Page 2 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## Commands and options

What are commands, their options, their arguments.

- `command [options] [arguments]`

Options are usually preceded by “-” or “--”. A command can have many options.

- `ls -F`  
here `-F` is an option
- `ls -F /etc`  
here `/etc` is a argument
- `ls -lF /etc`  
multiple options. This is same as: `ls -l -F /etc`
- `alias`  
personalise commands with your favourite options

[Home Page](#)

[Title Page](#)



Page 3 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



[Home Page](#)

## Files and Permissions

[Title Page](#)

Look at the output of these commands

- `ls -a`
- `ls -l`

users, groups and others

drwx concepts

permissions on directories

understanding file mode numbers: 644 , 777

- `chmod`



[Page 4 of 24](#)

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## Copying and moving files

A command can have many options.

- `cp file1 file2`
- `cp -r`  
recurse into directories
- `mv`  
also used for renaming files
- `rm`  
remove(delete) files
- `rm -r`
- `rm -f`  
No warnings, no prompts
- `rm -i`

[Home Page](#)

[Title Page](#)



Page 5 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## The shell and its environment

- hardware - kernel - shell - user programs
- sh, csh, ksh, bash
- Environment variables
- env and set commands
- The PATH environment variable
- Startup files: `.bash_profile`, `.bashrc`

Home Page

Title Page



Page 6 of 24

Go Back

Full Screen

Close

Quit



## Some more commands

- `echo`
- `cat` : without options
- `cat file` : print the file
- `cat -n file` : numbered lines
- `more`
- `less`
- `head`
- `tail`, `tail -f`
- `wc`, `wc -l`, `wc -w`

[Home Page](#)

[Title Page](#)



Page 7 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## Redirection and piping

Concept of standard input, standard output and standard error

Possible sources/sinks of each stream - console, file, another program

<, >, >>, 2>, | : interpreted by the shell

- `ls > outputfile`
- Let's see what's in outputfile  
`cat outputfile`
- Produce an error message  
`ls outputfile abc`
- Separating the 2 streams  
`ls outputfile abc > a 2> b`
- A simple pipeline  
`ls | wc -l`

Home Page

Title Page



Page 8 of 24

Go Back

Full Screen

Close

Quit





## Useful commands

- `date, date -s`
- `split -b 1.4M bigfile`
- `(sleep 180; echo Reminder: Call Mr. Gupta) &`  
`(date; who) | wc -l`
- `ispell, aspell`
- `which ls`  
`cat `which ispell``
- `file *`
- `find . -name "s*.txt" -print`  
`find . -name core -exec rm -i {} \;`  
`find $HOME -perm 777 -print`
- `locate fortune`

[Home Page](#)

[Title Page](#)



Page 9 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## Filters

- `grep linux readme`  
`grep -i linux readme`  
`ls -l | grep "^d"`
- `sort readme`  
`ls -l | sort +4 -n`
- `cut, paste`  
`cut participants -d"|" -f3 | paste - pdetails`
- `tr "[a-z]" "[A-Z]" < readme`  
Capitalize the contents of the file
- `sed 's/linux/GNULinux/gi' readme`  
Replace “linux” by “GNU/Linux”
- `awk`

Home Page

Title Page

◀ ▶

◀ ▶

Page 10 of 24

Go Back

Full Screen

Close

Quit



Home Page

Title Page



Page 11 of 24

Go Back

Full Screen

Close

Quit

## Wildcards

- `ls *.dat, ls r*`
- `cat ?.txt`
- `ls *.*[ch]`

Escaping to prevent shell from interpreting special characters

- `touch a, touch '?'`
- `rm ? or rm '?'`
- `mpg321 Fear\ of\ the\ Dark.mp3`



Home Page

Title Page



Page 12 of 24

Go Back

Full Screen

Close

Quit

## Examples

- One of the ways to create a file (Ctrl-D to end input)  
`cat > f1`
- `cat f1 f2 > bigf`
- `sort < names > sortednames`
- `who | wc -l`
- `who | cut -d" " -f1 | sort | uniq | wc -l`
- `du -ks * | sort -rn | head -n 3`



[Home Page](#)

[Title Page](#)



Page 13 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Documentation

- `man`
- manual sections  
`man 3 printf`
- `man -k`, `apropos`
- `info`
- HOWTOs, FAQs, [www.tldp.org](http://www.tldp.org)



## Processes and Job Control

- process listing  
`ps, ps -ef`
- continuously updated listing  
`top`
- Send signals to processes  
`kill, kill -9, Ctrl-C`
- Start process in background  
`updatedb &`
- Job control  
`Ctrl-Z, bg, fg, jobs`

[Home Page](#)

[Title Page](#)



Page 14 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)



## Fun Commands

- `cal`  
A cute calender program
  - `cal 1996`
  - `cal 8 1977`
- `units`  
Convert between various standard units of measure
- `bc`, `bc -l`  
An arbitrary precision calculator (“Serious” fun)
  - `2+3/5` : Simple expressions
  - `x=4.34/45` : Variables
  - `s(x)`, `c(x)`, `l(x)`, `e(x)` : Math functions

Home Page

Title Page



Page 15 of 24

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 16 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Shell Scripting

- Shell is not just a command line interpreter
- Has support for variables (remember PATH)
- Offers host of programming constructs
  - Conditional constructs (`if`, `if-else`, `if-elif`, `case`)
  - Looping constructs (`for`, `while`, `until`, `select`)
  - functoins





## Programming Constructs

- `if` construct  
`if condition then; commands; fi`
- *test* ing the conditions  
`test -r file`  
`test -w file`  
`test "$val" -eq 100`  
`test "$LOGNAME" == "guest"`
- Example:  
`if ! test -r "$outfile" then`  
`echo "Can't read from $outfile"`  
`fi`

Home Page

Title Page

◀ ▶

◀ ▶

Page 17 of 24

Go Back

Full Screen

Close

Quit



## Programming Constructs (continued...)

- for loop

```
for var in list; do commands; done
for ((expr1; expr2; expr3)); do commands;
done
```

- while loop

```
while condition; do commands; done
```

- case construct

```
case var in
    pattern1 ) commands1;;
    pattern2 ) commands2;;
esac
```

Home Page

Title Page



Page 18 of 24

Go Back

Full Screen

Close

Quit



## Simple Script 1

```
#!/bin/bash
echo -n "Please enter your name: "
read name
TIME=`date +%H`
case $TIME in
0[6-9] | 1[01] ) echo -n "Good morning" ;;
                12 ) echo -n "Good Noon" ;;
                1[2-6] ) echo -n "Good Afternoon" ;;
                1[7-9] ) echo -n "Good Evening" ;;
                *) echo -n "Good Night" ;;
esac
echo " $name, Nice to meet you!"
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 19 of 24

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 20 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

## Simple Script 2

```
#!/bin/bash
for i in *.cc
do
    mv $i `echo $i | cut -d"." -f1`.cpp
done
```



## Special variables

- `$*` : all the arguments to the shell script
- `$#` : number of arguments passed to the script
- `$?` : exit status of the previous command
- `$$` : PID of the parent process

```
#!/bin/bash
```

```
# store in file ex.sh
```

```
echo "Args($#):  $* "
```

```
$ex.sh earth is round
```

```
Args(3):  earth is round
```

Home Page

Title Page

◀ ▶

◀ ▶

Page 21 of 24

Go Back

Full Screen

Close

Quit



## Introduction to vi

- Making new files, opening existing files
- The 3 modes in vi
- h, j, k, l, Ctrl-B, Ctrl-F - moving around
- :26 - Go to 26th line
- :%s/i/count/gc - Search and replace
- yy, p - Copy, Paste
- dd, dw, cw - Delete, Change
- ... a thousand other things

Home Page

Title Page



Page 22 of 24

Go Back

Full Screen

Close

Quit



[Home Page](#)

[Title Page](#)



Page 23 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)

Thanks..



[Home Page](#)

[Title Page](#)



Page 24 of 24

[Go Back](#)

[Full Screen](#)

[Close](#)

[Quit](#)