Department of Computer Applications

Programme: MCA

Course: CA719-OPERATING SYSTEMS LAB

Odd semester (I-SEM) [July 2025-2026]

PHASE -I

UNIX/LINUX Basic Commands

Login and authentication

login	access computer; start interactive session
logout	disconnect terminal session
passwd	change local login password; you must set a strong password that is not easily guessed
kinit	obtain kerberos ticket for connections to other kerberized computers
kdestroy	destroy kerberos tickets (authorizations)
keeptoken	request longer lifetime for kerberos ticket so long-running jobs can access AFS files

Information

date	show date and time
history	list of previously executed commands
pine	read or send email messages or networks news groups
msgs	display system messages
man	show online documentation by program name
info	online documentation for GNU programs
w, who	who is on the system and what they are doing

whoami	who is logged onto this terminal
tpo	show system stats and top CPU using processes
uptime	show one-line summary of system status
finger	find out info about a user@system
whois	look up information in the Stanford Directory

File management

cat	combine files
cp	copy files
ls	list files in a directory and their attributes
mv	change file name or directory location
rm	remove files
ln	create another link (name) to a file
chmod	set file permissions
crypt	encode/decode a file with a private key
gzip, gunzip	compress/decompress a file
find	find files that match specific criteria

Display contents of files

cat	copy files to display device
more	show text file on display terminal with paging control
head	show first few lines of a file(s)
tail	show last few lines of a file; or reverse line order
vi	full-featured screen editor for modifying text files
pico	simple screen editor for modifying text files
grep	display lines that match a pattern
lpr	send file to line printer
pr	format file with page headers, multiple columns, etc.
diff	compare two files and show differences

cmp	compare two binary files and report if different
comm	compare two files; show common or unique lines
od	display binary files as equivalent octal/hex codes
strings	show printable text embedded in binary files
file	examine file(s) and guess type: text, data, program, etc.
wc	count characters, words, and lines in a file

Directories

cd	change to new directory
mkdir	create new directory
rmdir	remove empty directory (remove files first)
mv	change name of directory
pwd	show current directory

Devices

df	summarise free space on disk drive
du	show disk space used by files or directories

Special character handling for C-shell (See man cs)

*	match any characters in a file name
~user	shorthand for home directory of user
$name	substitute value of variable name
\	turn off special meaning of character that follows
'	in pairs, quote string with special chars, except !
"	in pairs, quote string with special chars, except !, $
`	in pairs, substitute output from enclosed command

Controlling program execution for C-shell (See man csh)

&	run job in background
DEL, ^c	kill job in foreground
^z	suspend job in foreground
fg	restart suspended job in foreground
bg	run suspended job in background
;	delimit commands on same line
()	group commands on same line
!	re-run earlier commands from history list
jobs	list current jobs
ps	print process screen
kill	kill background job or previous process
nice	run program at lower priority
at	run program at a later time
crontab	run program at specified intervals
limit	see or set resource limits for programs
alias	create alias name for program (in .login)
sh, csh	execute command file

Controlling program input/output for C-shell (See man csh)

|	pipe output to input
>	redirect output to a storage file
<	redirect input from a storage file
>>	append redirected output to a storage file
tee	copy input to both file and next program in pipe
script	make file record of all terminal activity

Editors and formatting utilities

sed	programmable text editor for data streams
vi	full-featured editor for character terminals
emacs	GNU emacs editor for character terminals
xemacs	GNU emacs editory for X-Windows terminals
pico	very simple text editor, same as pine Compose screen
fmt	fill and break lines to make all same length
fold	break long lines to specified length

X-Window client programs (output to X terminal or server)

xterm	provide login shell window
xauth	manipulate authorization files
xload	show system load
xman	full screen online manual viewer
pinex	send or recieve mail messages
xemacs	GNU emacs editor
gv	interface to contol gs to display PostScript or PDF files on screen
xdvi	display DVI files on X-Window (screen preview)
netscape	web browser
gnuplot	interactive data plotting on screen

TeX typesetting system

tex	process TeX files to DVI (device independent) output
latex	process LaTeX files to DVI
texpr	process and print TeX and LaTeX in one step
dvips	print DVI files on Postscript laser printer
xdvi	display DVI files on X-Window (screen preview)
latex2html	translate LaTeX files to HTML (for web pages)

Printing

lpr	send file to print queue
lpq	examine status of files in print queue
lprm	remove a file from print queue
op lpc abort qname	abort print queue qname to clear printer (all files saved)
op lpc start qname	restart print queue qname
enscript	convert text files to PostScript format for printing

Interpreted languages and data manipulation utilities

sed	programmable text editor for data streams
awk	pattern scanning and processing language; 1985 vers.
perl	Practical Extraction and Report Language
sort	sort or merge lines in a file(s) by specified fields
tr	translate characters
cut	cut out columns from a file
paste	paste columns into a file
dd	copy data between devices; reblock; convert EBCDIC

Graphics and mapping

gnuplot	interactive data plotting; outputs to PostScript or X-windows
GMT	general 2D and 3D data processing and plotting software package; also plots maps
gs	"ghostscript" converter displays PostScript files on X-window displays or other devices

Networking/communications

klogin	remote login to kerberized computer; encrypted
krsh	execute single command on remote kerberized computer; encrypted
krcp	remote file copy to/from kerberized computer; encrypted
ssh	remote login/command execution; encrypted
scp	remote non-interactive file copy; encrypted
sftp	remote interactive file copy; encrypted
telnet	remote network login - plain text password
ftp	network file transfer program - plain text passwords
rlogin	remote login to "trusted" computer that is not kerberized
rsh	execute single command on remote "trusted" computer
rcp	remote file copy to/from "trusted" computer
host	find IP address for given host name, or vice versa
netscape	web browser for X-window terminals/servers
lynx	web browser for character based (text-only) terminals
kermit	transfer files over modem connections
gzip, gunzip	compress/decompress a file
tar	combine multiple files/dirs into single archive
uuencode, uudecode	encode/decode a binary file for transmission via email

Compilers, interpreters and programming tools

csh	command language interpreter (C-shell scripts)
ksh	command language interpreter (Korn-shell scripts)
sh	command language interpreter (Borne-shell scripts)
f77	Compaq(HP) Fortran 77 compiler
f95	Compaq(HP) Fortran 90/95 compiler

f2c	convert fortran source code to C source code
cc, c89	Compaq(HP) ANSI 89 standard C compiler
cxx	Compaq(HP) C++ compiler
gcc	GNU C compiler
g++	GNU C++ compiler
pc	Compaq(HP) Pascal compiler
dbx	command-line symbolic debugger for compiled C or Fortran
ladebug	X-window symbolic debugger for compiled C or Fortran
make	recompile programs from modified source
gmake	GNU version of make utility
cflow	generate C flow graph
error	analyze and disperse compiler error messages

- **Programming libraries (see man library name)**

lapack	Fortran 77 routines for numerical linear algebra (supersedes LINPACK and EISPACK)
X	routines to interface with X window system (no man page -- get the X Toolkit book)
dbm	database routines
xdr	library routines for external data representation
netcdf	routines for machine independent data representation

- **Tape manipulation and archiving**

mt	manipulate tape drive and position tape
dd	unformatted tape read and write; file conversion
tar	archive disk files on tape or disk
ltf	read/write ANSI standard label tapes

Geology programs

supcrt92	thermodynamic properties of high P/T reactions
diagram	calculate activity phase diagrams from log K values

PHASE –II

VI COMMANDS

Cursor Movement Commands:

Esc h	Move the cursor left by one character
Esc j	Move the cursor down by one line
Esc k	Move the cursor up by one line
Esc l	Move the cursor right by one character
Esc ^ d	Scroll the screen down by 12 lines
Esc ^ u	Scroll the screen up by 12 lines
Esc ^ b	Scroll the screen up by 24 lines
Esc ^ f	Scroll the screen down by 24 lines
Esc $	Move the cursor to the end of a line

Insert and Replace Commands:

Esc a	Append to the right of the cursor
Esc A	Append at the end of the line
Esc I	Insert at cursor position
Esc o	Insert a blank line below the current
Esc O	Insert a blank line above the current
Esc rx	Replace character at current cursor

Word Movement Commands:

Esc w	Move to next word
Esc b	Move backward to previous word
Esc e	Move to the end of the word

Deletion and Modification Commands:

Esc dw	Delete Word
Esc dd	Delete line
Esc cw	Change word
Esc cc	Change line
Esc C	Change rest of the line from current

Esc D	Delete rest of the line from current cursor
Esc J	Join lines
Esc x	Delete char at current cursor position
Esc u	Undo last change
Esc U	Restore last change

Pattern Finding Commands:

/pattern	Find the next line containing the pattern
Esc n	Find the next occurrence of the pattern
?pattern	Find the previous line containing the pattern
Esc :line number	Go to line number
Esc : se nu	To see line numbers

Block Operation Commands:

Esc yy	Yank current line into buffer
Esc nyy	Yank n lines into buffer
Esc p	Put back yanked text after the cursor
Esc P	Put back yanked text before cursor
Esc ndd	Delete n lines from current cursor
Esc n>>	Shift n lines by one tab position to right
Esc n<<	Shift n lines by one tab position to left

Saving and Exiting From Vi:

ESC : w	Save all changes made
: wq	Save all changes and quit from vi
: ®	Quit
: q!	Quit without saving
: !command	Execute Unix command

UTILITY COMMANDS

- CAL

Print the calendar

cal 1998	all months of the year
cal 12 1998	December month of 1998

- DATE

Print System date (date can be changed by system administrator)

Options:- d (day of a month), y(2 digits of an year), H,M,S (hour ,minutes and seconds)

date	system date with time
date +%m	print month number
date +%h	print month name
date +"%m %h"	name and number
date –u	universal time
date +%r	12 hr time

- WHO

Details about login users

Options:- am I (about user), -Hu (more information with header details)

who am i

who –Hu

- TTY

Know the terminal name. Unix treats terminals in multiuser as files in device directory.

tty

- UNAME

Print system information

uname –a all

options –s (print kernel name), -n (network name), -e (hardware platform) ,-o (os)

- TPUT

Various displays

```
tput 12,30
echo "hh"          position cursor
tput smso          highlight text
echo "ddd"
tput rmso          disable highlight
```

- BC

Binary calculator
Provides all type of calculations

```
bc
12.3 + 9.9
<ctrl + d>
sacle =2           truncate to 2 decimal places
ibase=2            to set input bae
obase=16           set output base
c=a+b
c                  to print c
```

- TIME

To calculate the time taken by a command

```
time sort –o 1.c 2.c
```

time shows real(elapsed time),user(execution time), sys time(time by os to do the operation)

- SCRIPT

To record the user activities for further use .Doesn't record commands in vi editor

```
script             write to typescript file
script –a          appends to typescript file
script logfile     writes to logfile
```

- SPELL

Spell check of a file

```
spell  filename    displays incorrect words
ispell filename    displays lines and suggestions
```

FILE SYSTEM COMMANDS

- LS

  Listing files

  ls -x (display files in many columns)
  ls -a hidden files
  ls -f shows / for directory, * for exe file
  ls -r list in reverse order
  ls -R recursive listing
  l (long listing) i (inode number) t (modification time) u (access time)

- SPLIT

split filename splits file into many of 1000 lines each in name fna, fnb, fnc…

split –34 filename splits after 34

- CMP

  Compare two files

cmp filename1 filename2 displays difference between 2 files
cmp –l filename1 filename2 displays line numbers

- COMM

  Displays common in 2 files
  comm f1 f2

- DIFF

Says the difference between 2 files, suggests which line to be changed

- **CHMOD**

  Change permission mode of a file

  chmod 744 filename

rwxrwxrwx -this combination stands for read, write, execute permissions of user ,group ,public.
Either a number can be given or abbreviations can be given
chmod ugo+rwx filename r,w,x perm for u , g , o
chmod u-x,g+r filename remove exec for user add read for group
chmod –r u+x shell assign permissions for all files in shell directory

- CHOWN

  Change owner and/or group of a file

chown owner_name filename
chown owner:groupname filename
chown :groupname filename

- CHGRP

Change group of a file
chgrp <name> filename

- CP

Copy the contents of one file to another
cp t1.txt t2.txt
cp –i s1.doc s2.doc prompts an alert message
cp –r copying directory structures

- RM

Remove a file from the directory
rm filename
rm –i filename interactive remove
rm –f filename forcibly remove a file
rm –r directory name delete a directory with contents

- MV

Move a file to another(rename)
Move a file from one directory to another
mv d f
mv d ipc/shell/d

- WC

Count of lines ,words, bytes in file
cat <filename> |wc (all)
cat <filename> |wc –l (only lines)
cat <filename> |wc -c

FILTERS

- PAGINATING FILES

pr filename formats to print the file
pr –d filename double spaces the file

pr –n filename number lines
pr –3 filename filename in 3 columns

- HEAD

Display the beginning of the file

head fn first 10 lines of a file
head –3 fn cut first 3 lines of a file

- TAIL

Display the end of the file

tail –3 fn extract last 3 lines of a file
tail +11 fn from 11th line

- CUT

Extract columns and fields

cut –c1 cut column one
cut –c1-11 form one to eleven
cut –c1,4,5 cut column 1,4,5
cut –f1-5 cut field one to five
cut -d " " –f5 cut field 5 by delimiter space
d can be ; / , " " :

- TR

Translate characters

tr [options] < filename

cat filename | tr [a-z] [A-Z] translate lower to upper case
tr –d " " delete blank spaces
tr –s " " suppress many spaces to one
tr '|/' '~-' change | to / ~ to –
tr –cd '/' delete except / (c for complement)

- SORT

Sort the contents of file

sort –t \ : +1 <filename> sort by : separated field by col 2
sort -t \ : +2 -3 +1 secondary key 2 ,primary key 3,resume after first field(sort from 2nd field to 3rd field
sort –u unique contents in a file
sort –m <fn1> <fn2> sorts by merging 2 files

PATTERN MATCHING

➢ GREP

Globally search for a regular expression and print it.
grep sales filename prints all lines containing sales
grep 'components number' filename match phrase
options

-c count of match
-n line numbers of matching lines
-v inVerse match for other than pattern
-l filenames which are matching
grep –l 'ipc' *.c
-i ignore case
-e multiple options
-o give matched pattern
grep –e 'agar' –e 'Agar' –e ' Aagar' filename

grep –5 "do loop" update.sql 5 lines above and below do loop
gr.ep –A 5 …. (below)
grep –B 5 … (above)
grep –N 2 'bill' accounts.c two occurrences of bill

Regular Expressions	
*	Zero or more occurrences of a character
.	Matches a single character
[pqr]	Matches p or q or r
[c1-c2]	Matches between c1 to c2
[^ pqr]	Matches other than p q r
^ pattern	Matches pattern at the beginning of line
Pattern$	Matches pattern at the end of line

grep [rr][aa][nn][ii] os.txt matches for lower or upper case
grep "7…$" emp.txt matches 7000 to 7999

- EGREP

  Extending grep

ch+	One /more of character
ch?	Zero/more of character
exp1|exp1	Match exp1 or 2
(x1|x2)x3	Match x1x3 or x2x3

egrep '(lalitha|meena)kumari' student.txt
egrep –f pat.txt filename pat.txt should contain patterns seperated by '|'

- FGREP

  faster grep
  only patterns used regular expressions cannot be used
  fgrep –f <pattern file > targetfile

- SED

Stream Editor
sed 'ed commands' <filename>
sed –n '20,30p' print lines from 20 to 30
sed '1,10d' delete 1 to 10 lines
sed 's/old/new/g' search for old and replace by new
sed 'y/str1/str2/ replace each in str1 by str2

PROCESS COMMANDS

echo $$
process id of current shell
ps
process status (pid, name, tty, time)
ps –f
full information(uid, ppid etc)
ps –f –u cra
full information of user cra
ps –a
processes of all users
ps –x
system process
nohup < any command> & makes to run a command without *hangup* even after logout

kill pid	to terminate a process
kill $!	Kill last background process
kill –9 121	Sure kill of process 121
kill –9 0	kills process including login shell

- NICE

  Job execution with low priority
  nice wc –l unix.txt

- AT

  Execute a command or file at a given time
  at 14.00
  sam.sh
  <ctrl –d>

- BATCH

  Schedule the process for later execution
  batch < filename

- CRON

  ChRONograph. It is a daemon process which ticks every minute.

It looks for the file in /var/spool/cron.
User can insert their file in cron directory by
cron cron.txt
crontab –l shows the contents of cron file

SCRIPT STATEMENTS

- Declaration -not needed,no datatypes,identifier names

- I/O
  - echo name list /string
  - echo $a,$b
  - echo “enter a variable”
  - echo “value of “ $a
  - echo -n “give two int”
  - read variables

- Operators
  - relational(numbers) -gt, -lt , -ge, -le ,-eq, -ne
  - relational (strings) = , != , -n string,-Z string $a -gt $b , $abc -ne $ghj

-n $a ($a > 0)
-z $a ($a equal to zero)

- **Logical operators**

and -a $a -gt $b -a $c -gt $f
or -o
not !

- **File operators**

-r , -w , -x , -c , -b , -d
-f , -s ,-k

```
if [ -r $fn ]
then
echo "file has read"
fi
```

- **Control Statements --IF**

```
if [ $a -gt $b ]
then
echo " hello"
fi

if [ $a -ne $b]
then
echo "hello"
elif [ ]
echo
else
echo "hai"
fi
```

```
if [ $a = $b ]
--
--
If [ -z $a ]
then
echo "null string"
fi

if [ -n $a ]
then
echo "not null"
fi
```

- **For**

```
for i in a b c
do
echo $i
done
```

for i in *
do
echo $i
done
$* parameters in command line

- **While**

```
while [ $i -lt $n ]
do
..
..
expr ` $i + i `
Done
```

- **Case**

```
case  expr  in    case  $i  in
1)    ;;                1)
2)    ;;                a)
*) default;;            asd)
esac
```

- **Arithmetic Statements**

```
+ - * / %
expr 1 + 2
expr 1 \* 2
expr $a + $b
expr $a \* $b
```

REFERENCES

TEXT BOOK

- Eric Foster-Jonson, John C. Welch and Micah Anderson, “Beginning Shell Scripting”, Wiley-India
- Introduction to Linux – A Hands on Guide
- Ashley J.S Mills, "Unix Shell Scripting Tutorial", Copyright © 2005 The University Of Birmingham

WEB REFERENCES

- SHELLdorado (http://www.oase-shareware.org/shell/)
- Shell Information from CUED (http://www-h.eng.cam.ac.uk/help/unix.html#Shell)
- Linux Shell Scripting Tutorial - A Beginner’s handbook (http://www.freeos.com/guides/lsst/)
- http://www.japarker.btinternet.co.uk/sh/
- http://linux-training.be