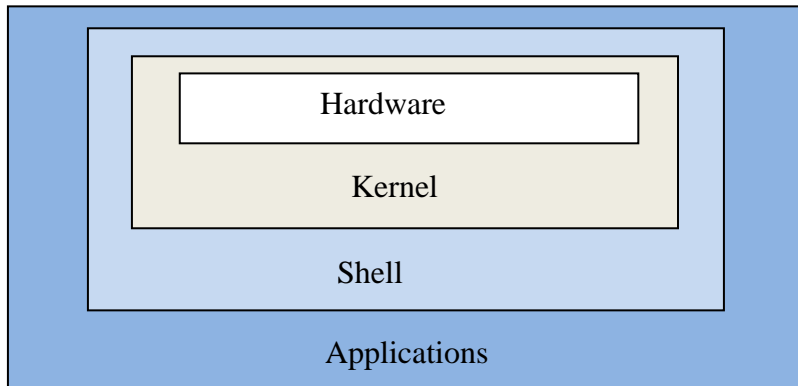


## Review chapter 1-3

- Unix/Linux Architecture



- Using the Shell

- 1) What is the shell
- 2) Program & process
- 3) Support different shells, shortcuts, **redirections**, **pipes**, alias, **wildcards** (\*, ?, [], [! ]), processes
- 4) The first touch: telnet(PuTTY), ftp, *login*, using commands(*date*, *cat*, *passwd*, *man*, ...), *logout*  
*man* sort | *less*  
*man* -k sort  
*info* sort  
*sort* --help
- 5) Working with files
  - ✓ Browsing files and directories (*ls*, *pwd*, *mkdir*,...)
  - ✓ File attributese.g.  
drwx---r-x 2 cscqxcx cscqxcx 4096 Feb 25 2008 ch08  
1|-----2---| 3|---4 ---||---5----| |- 6 -| |----- 7 ---| |- 8 -|
  1. Type
  2. Access modes
  3. Number of links
  4. Owner (*chown* user file)
  5. Group (*chgrp* group file)
  6. Size in Bytes
  7. Modification Date and Time
  8. Name
  - ✓ Special files : . and ..
  - ✓ Protecting files and directories  
Permissions for directories: r, w, x  
Permissions for files: r, w, x

e.g. - rwx rwx rwx  
- rw- r-- r--  
d rw- r-x r-x

Changing file attributes (**chmod** mode file: set up a mode; add / remove a permission to owner; group members or other users; number code of permission)

e.g. **chmod** go=rw filename  
**chmod** g-w,o-r,a+x filename  
**chmod** 777 file1 file2 file3

- ✓ Copy/move/remove files (**cp**, **mv**, **rm**, **rmdir**,...)
- ✓ File archives ( tar
- ✓ File compression

Command	Extension(s)	uncompression
<b>zip</b>	.zip	<b>unzip</b>
<b>compress</b>	.tar.z .taz .tz	<b>uncompress</b>
<b>gzip</b>	.tar.gz .tgz	<b>gunzip</b>

e.g. **tar** cvf tarfile.tar file1, file2,...  
**zip** zipfile.zip file(s)  
**gzip** tarfile.tar  
**compress** tarfile.tar

- ✓ Text editor **vi**  
Command mode, text mode, insert / remove word/lines, search/replace text, read another file, copy /paste blocks,....
- ✓ ...

## 6) Data processing

- ✓ Sort (sort -tc -r [+pos1 [-pos2]] {sortfield -bfMn}\*)
- ✓ Compare files(cmp, diff)
- ✓ Scheduling (crontab -ler [username], at -csm time [date [,year]] [+increment] [script])
- ✓ Find utility (find pathList expression)  
name pattern  
perm oct  
type ch  
user userid  
group groupid  
atime count  
ctime count  
exec command

e.g. find . -name "\*.c" -exec chmod o+r '{}' \;  
find . -exec grep "public" '{}' \; -print

- ✓ Regular expression (see handouts)
- ✓ Filter patterns (grep, egrep, fgrep) **grep** -hilnvw pattern {fileName}\* (see Regular expression examples section & homework2)
- ✓ Pattern scanning and processing (see awk handout & homework2)
  - awk -Fc -f awk.code InputFile
  - awk -Fc 'condition action' InputFile
  - e.g. awk -F: '/cscqxcx/ {print \$1," uses ",\$7}' /etc/passwd
- ✓ Stream Editor (sed)
  - e.g. sed 's/^/ /' file > file.new
  - indents each line in the file by 2 spaces
    - sed 's/^ \*//' file > file.new
  - removes all leading spaces from each line of the file
    - sed '/a/d' file > file.new
  - deletes all lines containing 'a'
    - sed -e 's/^<</' -e 's/\$/ >>/' file

Example: add two lines in the beginning of the file

```
% cat sed1
1i\
abcd\
efg
% sed -f sed1 file > file.new
% cat sed2
1,3c\
Lines 1-3 are censored
```

```
% cat dummy2
one
two
three
four
five
six
```

```
% sed -f sed2 dummy2
Lines 1-3 are censored
four
five
six
```

- ✓ Translate utility(tr -cds string1 string2)
- ✓ ...