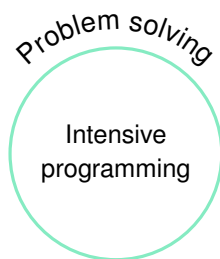


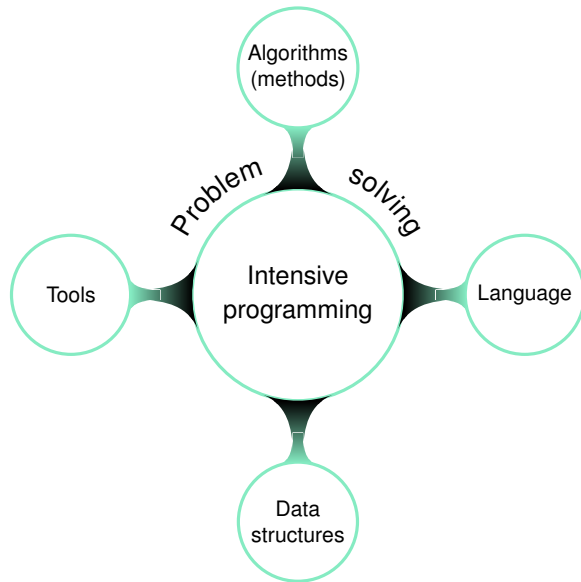
Technical details + Introduction to UNIX-like systems

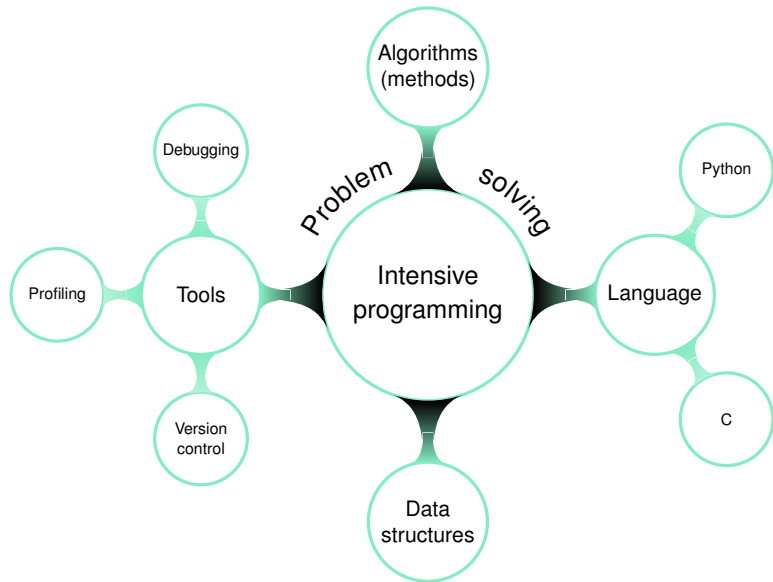
Computing Laboratory

<http://www.isical.ac.in/~dfslab/2020/index.html>

- 1 Course overview
- 2 Technical details
- 3 UNIX-like systems
 - File system hierarchy
 - Commands
- 4 Computing Lab conventions







- **Question:** How to store your data?

- **Question:** How to store your data?
- **Counter-question:** What kind of operations will be needed?

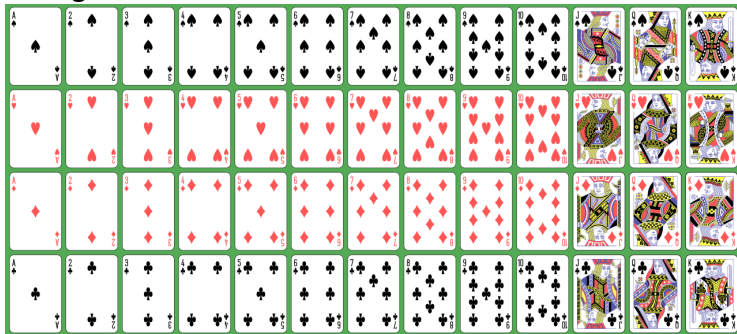
- **Question:** How to store your data?
- **Counter-question:** What kind of operations will be needed?
- **Answer:** Access the i -th element of a list.

- **Question:** How to store your data?
 - **Counter-question:** What kind of operations will be needed?
 - **Answer:** Access the i -th element of a list.
 - **Options: unnumbered list (harder) vs. numbered list (easier)**
 - Ashoka
 - Iltutmish
 - Rajaraja Chola
 - Kanishka
 - Shah Jahan
 - Harshavardhan
1. Ashoka
 2. Iltutmish
 3. Rajaraja Chola
 4. Kanishka
 5. Shah Jahan
 6. Harshavardhan

What method to use?

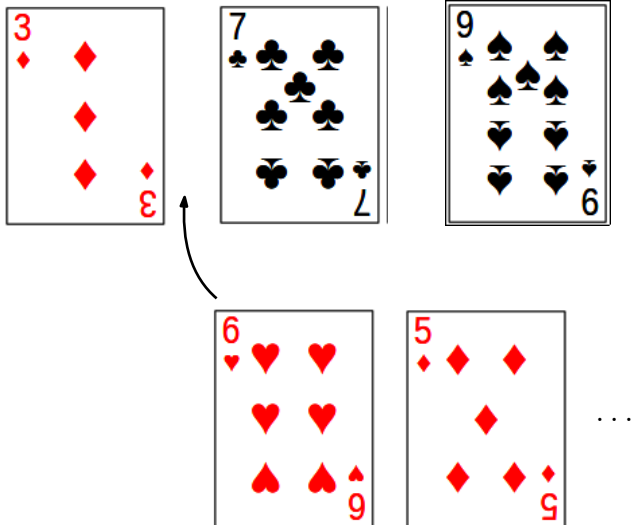
What method to use?

- **Problem:** Given a set of 20 playing cards, arrange them in order.
- **Background:**

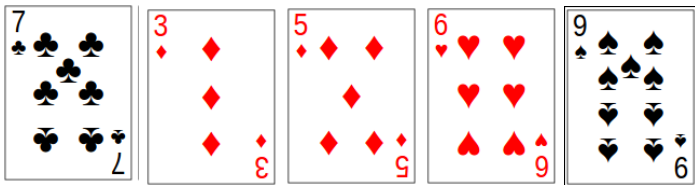


- 4 suits: **Spades**, **Hearts**, **Diamonds**, **Clubs**
- 13 values: 2, 3, . . . , 10, **J** (Jack), **Q** (Queen), **K** (King), **A** (Ace)

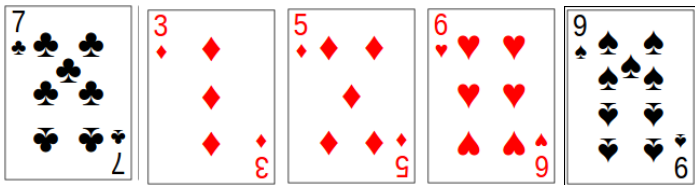
Method 1:



Method 2:

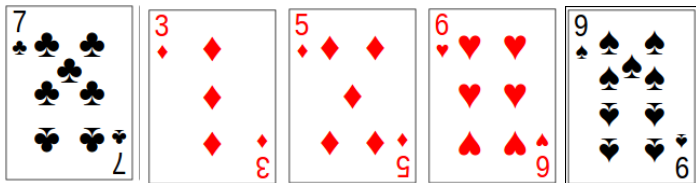


Method 2:



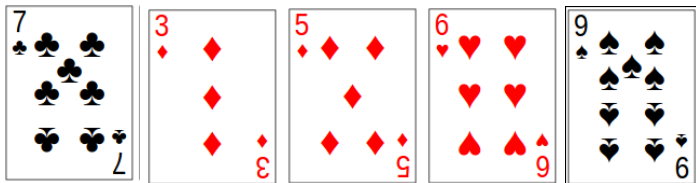
Spades				Hearts				Diamonds				Clubs			
2	3	...	Ace	2	3	...	Ace	2	3	...	Ace	2	3	...	Ace

Method 2:



Spades				Hearts				Diamonds				Clubs			
2	3	...	Ace	2	3	...	Ace	2	3	...	Ace	2	3	...	Ace

Method 2:



Spades				Hearts				Diamonds				Clubs			
2	3	...	Ace	2	3	...	Ace	2	3	...	Ace	2	3	...	Ace



If you are not convinced that Method 2 is easier, try actually physically sorting a bunch of 40 pages, one per student, in roll number order.

- 1 Course overview
- 2 Technical details**
- 3 UNIX-like systems
 - File system hierarchy
 - Commands
- 4 Computing Lab conventions

Some SSH clients for Windows

- **MobaXterm** <https://mobaxterm.mobatek.net/>
- **Solar Putty**
<https://www.solarwinds.com/free-tools/solar-putty>

Connecting to ISI

1. Login to ISI's gateway server.

- **host IP address:** 14.139.222.87, port 2222
- **userid:** mtcxxxx (instead of csxxxx where xxxx is your 4-digit roll number)
- **password:** same as your ISI Webmail/email password

Try running: `ssh mtcxxxx@14.139.222.87 -p 2222`

Connecting to ISI (contd.)

```
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.4.0-145-generic x86_64)
```

```
* Documentation: https://help.ubuntu.com/
```

```
154 packages can be updated.
```

```
10 updates are security updates.
```

```
New release '18.04.5 LTS' available.
```

```
Run 'do-release-upgrade' to upgrade to it.
```

```
Last login: Mon Dec 7 13:17:51 2020 from 42.110.139.124
```

```
...
```

```
mandar@gateway:~$
```

Connecting to ISI (contd.)

2. Now run: `ssh csxxxx@192.168.64.35`

Enter your ISI Webmail/email password when prompted.

Connecting to ISI (contd.)

2. Now run: `ssh csxxxx@192.168.64.35`

Enter your ISI Webmail/email password when prompted.

```
mtcxxxx@gateway:~$ ssh csxxxx@192.168.64.35
The authenticity of host '192.168.64.35 (192.168.64.35)' can't be established.
ECDSA key fingerprint is SHA256:ADmA7Y6bCVJYyVpF/mLI3F8nCM5cIKO+kAN0ahSjmOM.
Are you sure you want to continue connecting (yes/no)? yes
Failed to add the host to the list of known hosts ...
csxxxx@192.168.64.35's password:
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-141-generic x86_64)
...
Last login: Mon Dec  7 15:45:46 2020 from 192.168.143.29
csxxxx@student:~$
```

Transferring files to the server

1. You may use FileZilla <https://filezilla-project.org/>.
2. In a terminal *on your computer at home*, run

```
ssh -L 1234:192.168.64.35:22 mtcxxxx@14.139.222.87 -p 2222
```

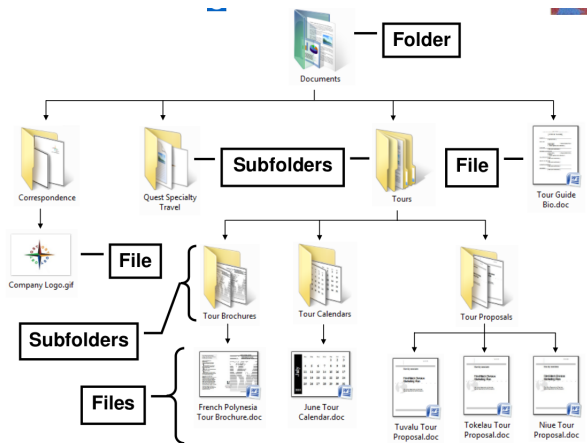
Keep this terminal open and go to next step.

3. Use the following information to connect via filezilla or from command line:
 - **host:** sftp://localhost
 - **port:** 1234
 - **username, password:** csxxxx, and your email password

- 1 Course overview
- 2 Technical details
- 3 UNIX-like systems**
 - File system hierarchy
 - Commands
- 4 Computing Lab conventions

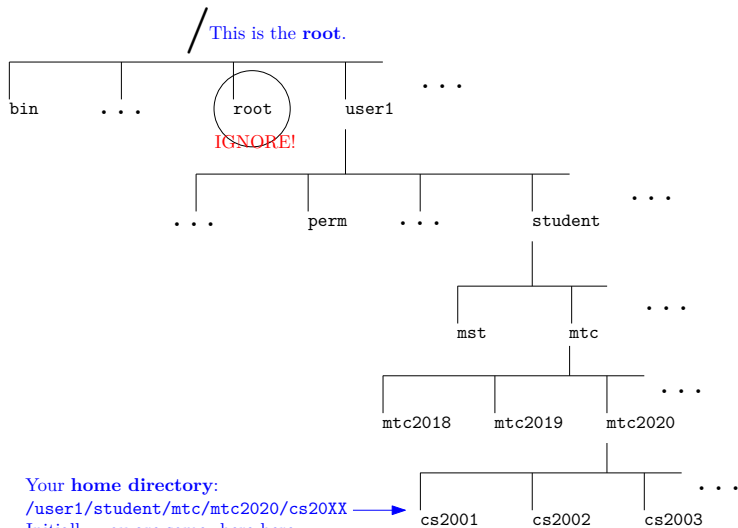
File system structure

Files are organised in a hierarchical structure of folders, sub-folders, and files.



Courtesy: <https://www.slideshare.net/okmomwalking/windows-7-unit-b-ppt>

File system hierarchy



File system structure: terminology

- Folders \equiv *directories*
- Top of the hierarchy: *root directory* (`/`)
- Location of a file or directory: specified by *path*
- Current location in terminal or file browser: *current working directory*
- Normal (or default) start location: *home directory*
- Paths: *absolute* or *relative*
 - absolute path: from root (starts with `/`)
Example: `/usr/bin/firefox`, `/tmp`, `/user1/student`
 - relative path: from current working directory (does not start with `/`)
Example: `clab/assignment1/hello.c`

Note the difference between (forward) slash (`/`, used in Unix-like systems) and backslash (`\`, used in Windows-like systems) !

Essential commands: directories

- `mkdir` : create a directory

Examples:

```
mkdir clab
```

```
mkdir clab/assignment1
```

Create directories as appropriate.

OR

```
mkdir -p clab/assignment1 clab/assignment2
```

- `rmdir` : remove an (empty) directory

Example: `rmdir assignment2, rmdir clab/programs`

Essential commands: navigating the file system

- **cd** : change directory

Example:

```
cd /user1/student/mtc1999
```

```
cd clab/assignment1/
```

```
cd ← go to home directory
```

- **pwd** : print current working directory

Essential commands: navigating the file system

- **cd** : change directory

Example:

```
cd /user1/student/mtc1999
```

```
cd clab/assignment1/
```

```
cd ← go to home directory
```

- **pwd** : print current working directory

Special directory names

- **~** : home directory

Example: `cd ~/clab`

- **.** : current working directory

Example: `./program1`

- **..** : parent directory (one level up)

Example: `cd ..`, `cd ../assignment2`

Essential commands: file listing

- `ls` : view list of files in current directory
- `ls <path>` : view list of files in specified path
- `ls -l` : view detailed list of files
- `ls -lt` : view detailed list of files sorted by modification time

Essential commands: file listing

- `ls` : view list of files in current directory
- `ls <path>` : view list of files in specified path
- `ls -l` : view detailed list of files
- `ls -lt` : view detailed list of files sorted by modification time

Example:

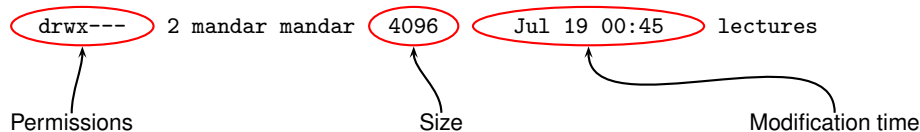
```
$ /bin/ls -l
total 68
drwx----- 2 mandar mandar 4096 Jul 19 00:45 assignments
drwx----- 2 mandar mandar 4096 Jul 22 2016 exams
-rw-r--r-- 1 mandar mandar 13521 Jul 19 00:41 index.html
drwx----- 2 mandar mandar 4096 Jul 19 00:45 lectures
```

Essential commands: permissions

```
drwx--- 2 mandar mandar 4096 Jul 19 00:45 lectures
```

Permissions Size Modification time

Essential commands: permissions



Permissions:

- 9 possible permissions:
{ **r**ead, **w**rite, **e**xecute } × { **u**ser (owner), **g**roup, **o**ther (everyone else) }
- 9 bits (1 ≡ permission granted)

ur	uw	ux	gr	gw	gx	or	ow	ox
----	----	----	----	----	----	----	----	----

- `chmod`: changing permissions

Example:

```
chmod g+wx <path>  
chmod og-wx <path>  
chmod 644 <path>  
chmod 700 <path>
```

Essential commands: files

- `cp` : copy a file

Example:

```
cp program1.c program2.c
```

```
cp -i source-file target-file
```

```
cp -i source-file target-directory
```

Essential commands: files

- **cp** : copy a file

Example:

```
cp program1.c program2.c
```

```
cp -i source-file target-file
```

```
cp -i source-file target-directory
```

- **mv** : rename (move) a file

Example:

```
mv program1.c program2.c
```

```
mv -i source-file target-file
```

```
mv -i source-file target-directory
```

Essential commands: files

- **cp** : copy a file

Example:

```
cp program1.c program2.c
cp -i source-file target-file
cp -i source-file target-directory
```

- **mv** : rename (move) a file

Example:

```
mv program1.c program2.c
mv -i source-file target-file
mv -i source-file target-directory
```

-i ≡ interactive
(asks for confirmation)

- **rm** : remove (delete) a file

Example:

```
rm program1.c
rm -i file1 file2.c *.bak
rm -r some-directory (remove directory and everything inside it)
```

- Useful for quickly viewing a file (not editing)
- Use `less`

Example: `less cs19xx-day0-prog1.c`

- space: move forward one page
- backspace or b: move backward one page
- q : exit the pager
- / : search for a string in the file
- run `man less` for more information

Choose any one that you like. Some options:

- emacs, nano, vim
 - terminal based
 - can use them after logging in to the ISI server
- atom, geany, sublime text, vscode, . . .

If you expect to do a substantial amount of programming, take the time and trouble to get familiar with a powerful editor / IDE. It'll probably save you time and trouble in the long run.

Choose any one that you like. Some options:

- emacs, nano, vim
 - terminal based
 - can use them after logging in to the ISI server
- atom, geany, sublime text, vscode, ...

If you expect to do a substantial amount of programming, take the time and trouble to get familiar with a powerful editor / IDE. It'll probably save you time and trouble in the long run.

Some random opinions / guides:

- <http://lifehacker.com/five-best-text-editors-1564907215>
- <http://www.techradar.com/news/the-best-free-text-editor-2017>
- <https://www.codementor.io/mattgoldspink/best-text-editor-atom-sublime-vim-visual-studio-code-du10872i7>
- <http://blog.liveedu.tv/10-best-text-editors-programming-2016/>

- `man`

Example: `man ls`, `man cp`, `man rm`

Other commands

- `man`

Example: `man ls`, `man cp`, `man rm`

Find out more about these on your own.

- `alias` (giving your own, easy-to-remember names to commands)
- `wc` (counting characters, words, lines)
- `sort`
- `head`, `tail` (first few / last few lines)
- `cmp`, `diff` (comparing two files)
- `ps`, `top`, `kill` (checking what programs are running)
- `find` (finding files or directories)
- `grep` (searching for patterns)
- `awk`, `sed` (programming)

http://cli.learncodethehardway.org/bash_cheat_sheet.pdf

<https://ubuntudanmark.dk/filer/fwunixref.pdf>

<http://www.ucs.cam.ac.uk/docs/leaflets/u5>

<http://mally.stanford.edu/~sr/compuGng/basic-unix.html>

<http://www.math.utah.edu/lab/unix/unix-commands.html>

- 1 Course overview
- 2 Technical details
- 3 UNIX-like systems
 - File system hierarchy
 - Commands
- 4 Computing Lab conventions

File / directory naming conventions

At the beginning of **any** submitted program file (assignment / exam), please write:

```
/*-----  
Name:  
Roll number:  
Date:  
Program description:  
Acknowledgements:  
-----*/
```