速成教育
SPEED UP EDUCATION

CSCB09

Software Tools and Systems Programming

导师： **VC**

UTSC Week 01 Class | 2024/1/17

Disclaimer

This study handout is provided by Speed Up Education and its affiliated tutors. The study handout seeks to support your study process and should be used as a complement, not substitute to the university course material, lecture notes, problem, sets, past tests and other available resources.

The study handout is distributed for free to the students participating in Speed Up Education classes and is not for sale or other commercial uses whatsoever. We kindly ask you to refrain from copying or selling in part or in whole any information provided in the handout.

Thanks for choosing Speed Up Education. We wish you the best of luck in your exam.

Unix Commands

Special Terms:

directory (folder) a collection of files and sub-directories
stdin (standard in) by default is terminal
stdout (standard out) by default is terminal

Commands:

`pwd`

Output the current working directory path to stdout

`ls`

Output a list of the file and sub directory of the current working directory to stdout

- `-l` Show all Information

`cd [dir path]`

Change your working directory to the given path

- `.` current directory
- `..` paraent directory

`cat`

Read data from stdin and output to stdout

`cat [file name]`

Ouput the content of the given file to stdout

`mkdir [dir name]`

Create a new directory with the given name in the current working directory

`clear`

Clear the terminal

`sort`

Read data from stdin, sort it, and output to stdout

- `-r` Reverse order

`sort [file name]`

Sort the content of the file, output to stdout

`uniq`

Read data from stdin, remove duplicates , and output to stdout

`uniq [file name]`

Read the content of the file, remove duplicates, output to stdout

```
cp [file/dir name] [dir path]
```

Copy a file/dir to a different path

```
mv [file/dir name] [dir path]
```

Rename a file/dir or move it to a differernt path

```
rm [file/dir name]
```

Remove a file/dir

```
diff [file name1] [file name2]
```

Compare files, Show where they differ.

```
wc [file name]
```

Tells you how many lines, words, and characters in the file.

So many more.. Will cover them later in the course ☺

Redirection:

`command that outputs to stdout > filename`

`>` Redirect the data from stdout into the given file.

Create the file if not exist. **Erease** the content of the orginal file.

`cat > a.txt`	Read data from Stdin, and output them into a.txt

`>>` Redirect the data from stdout into the given file.

Create the file if not exist. **Append** the content of the orginal file.

`command that inputs data from stdin < filename`

`<` Redirect the data from stdout into the given file.

Create the file if not exist. Erease the content of the orginal file.

`sort < a.txt`	Sort data from a.txt
`sort < a.txt > b.txt`	Sort data from a.txt and output it into b.txt

`commandA | commandB`

commandA must be a command that outputs data to stdout.

commandB must be a command that inputs data from stdin.

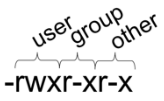
`|` is called **Pipe**. It makes the output data from commandA become the input data for commandB

```
cat a.txt | sort -r | uniq
```

File and Directory Persmission:

Example:

```
drwxr-xr-x
-rw-r--r--
```



first letter: `d` directory, `-` regular file
next 3 letters: user permissions
next 3 letters: group persmissions
last 3 letters: other permissions

permission	**Files**	**Directories**
r	can read the file	can 'ls' the dir
w	can write the file	can modify the dir's contents
x	can execute the file	can 'cd' to the dir

Modify permission for one user type:

Use `chmod (class)(operator)(type) [filename]`

Access class	**Operator.**	**Access Type**
u (user)	+ (add access)	r (read)
g (group)	- (remove access)	w (write)
o (other)	= (set axact access)	x (execute)
a (all)		

`chmod a+r a.txt`	Add Read permission to all users
`chmod g-w a.txt`	Remove Write permission for group users

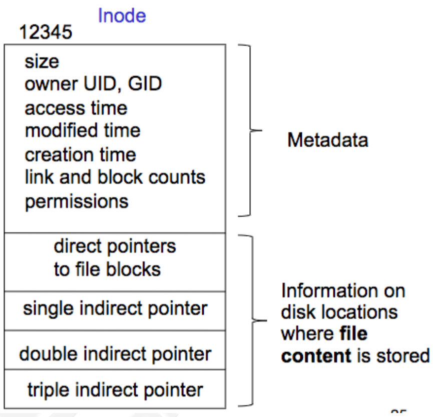
Modify permission using Absolute Form:

Permission	**Number**
Read (r)	4
Write (w)	2
Execute (x)	1

`chmod 751 a.txt`   read, write, execute permissions to yourself (4+2+1=7)
read and execute permissions to users in your group (4+0+1=5)
only execute permissions to others users (0+0+1=1)

File in Unix:

Inode:



- Every file in the Unix system has a **inode** (Index node)
- Contains all file information except the file contents and name.
- Just like a personal ID or a passport (without a name!)
- They contain the following
  - Inode Number
  - File Size
  - Owner information
  - Permissions
  - File Type
  - Number of links
  - ..etc
- Every directory is a file itself!
- `ls -li filename` to see the iNode Number

Soft link (Symbolic Link)

- It is a **pointer** to the original file (Shortcut in Windows)
- Smaller file size
- **Different Inode Number**
- If we delete the original file, the softlinks become useless!
- *Think as a Shortcut of the file.*
- `ln –s targetFile nameOfLink`

Hard link

- Different name of the same file
- Same file size
- **Same Inode Number**
- If the oringal file is deleted, the hard links will still contain the data that were in the orginal file
- *Think as a copy of the file*
- `ln targetFile nameOfLink`