

# Introduction to Linux, II

Introduction to Programming

EE231002

Oct. 5, 2020

# --help

- `--help` explains usage of the command
  - Example, `cp --help`

```
michang — ssh ee231002@140.114.24.31 — 80x24
[ee231002@ws38 ~]$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
   or: cp [OPTION]... SOURCE... DIRECTORY
   or: cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
-a, --archive                same as -dpr
   --backup[=CONTROL]       make a backup of each existing destination file
-b                           like --backup but does not accept an argument
   --copy-contents           copy contents of special files when recursive
-d                           same as --no-dereference --preserve=link
-f, --force                  if an existing destination file cannot be
                              opened, remove it and try again
-i, --interactive            prompt before overwrite
-H                           follow command-line symbolic links
-l, --link                   link files instead of copying
-L, --dereference            always follow symbolic links
-P, --no-dereference         never follow symbolic links
-p                           same as --preserve=mode,ownership,timestamps
   --preserve[=ATTR_LIST]   preserve the specified attributes (default:
                              mode,ownership,timestamps), if possible
                              additional attributes: links, all
-c                           same as --preserve=context
```

# Wild Cards

- `*` is a wild card that match any character strings
  - Example
  - `rm *`
    - Remove all files in the current directory
  - `cp ~ee2310/lab01/*.`
    - Copy all files in ~ee2310/lab01 directory to the current directory
  - `ls *.c`
    - List all `.c` files in the current directory

- `ls -al`: list all files in long format
  - `-a`: list all files including hidden files (files start with `.` character)
  - `-l`: long format
    - File mode, number of links
    - Owner of the file, group of the owner
    - Size of the file in number of bytes
    - Last modification date
    - Name of the file

```
michang — ssh ee231002@140.114.24.31 — 62x11
[ee231002@ws38 lab01]$ ls -l
total 536
-rwxr-xr-x 1 ee231002 course 6996 Sep 12 19:36 a.out
-rw-r--r-- 1 ee231002 course 379 Sep 12 19:39 lab01.c
-rw-r--r-- 1 ee231002 course 31979 Sep 7 14:53 lab01.pdf
-rw-r--r-- 1 ee231002 course 200523 Sep 7 14:53 linux1.pdf
-rw-r--r-- 1 ee231002 course 367 Sep 7 19:26 test1.c
-rw-r--r-- 1 ee231002 course 283034 Sep 7 14:53 vim.pdf

[file mode] [owner] [group][size][last mod tim][ name]
[link]
```

# File Modes

- File mode consists of 10 characters
  - The first character is the entry type
    - `-`: regular file
    - `d`: directory
    - `l`: symbolic link
  - The next 9 characters are divided into 3 fields to represent owner permissions, group permissions and world permissions.
    - `r`: readable; `-`: not readable
    - `w`: writable; `-`: not writable
    - `x`: executable or accessible (directory); `-`: not executable

```
michang — ssh ee231002@140.114.24.31 — 62x11
[ee231002@ws38 lab01]$ ls -l
total 536
-rwxr-xr-x 1 ee231002 course 6996 Sep 12 19:36 a.out
-rw-r--r-- 1 ee231002 course 379 Sep 12 19:39 lab01.c
-rw-r--r-- 1 ee231002 course 31979 Sep 7 14:53 lab01.pdf
-rw-r--r-- 1 ee231002 course 200523 Sep 7 14:53 linux1.pdf
-rw-r--r-- 1 ee231002 course 367 Sep 7 19:26 test1.c
-rw-r--r-- 1 ee231002 course 283034 Sep 7 14:53 vim.pdf

[file mode] [owner] [group][size][last mod tim][ name]
[link]
```

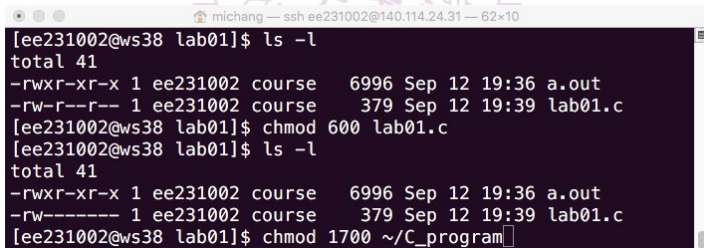
# File Modes

```
michang — ssh ee231002@140.114.24.31 — 62x11
[ee231002@ws38 lab01]$ ls -l
total 536
-rwxr-xr-x 1 ee231002 course 6996 Sep 12 19:36 a.out
-rw-r--r-- 1 ee231002 course 379 Sep 12 19:39 lab01.c
-rw-r--r-- 1 ee231002 course 31979 Sep 7 14:53 lab01.pdf
-rw-r--r-- 1 ee231002 course 200523 Sep 7 14:53 linux1.pdf
-rw-r--r-- 1 ee231002 course 367 Sep 7 19:26 test1.c
-rw-r--r-- 1 ee231002 course 283034 Sep 7 14:53 vim.pdf

[file mode]  [owner]  [group][size][last mod tim][ name]
[link]
```

- The file **a.out**
  - Owner can read, write and execute
  - Group member can read and execute (but not write)
  - The rest of the world can read and execute (but not write)
- The file **lab01.c**
  - Owner can read or write (but not execute)
  - Group member can read (but not write or execute)
  - The rest of the world can read (but not write or execute)

- File mode can be changed using `chmod` (change mode) command
- In the example below, after changing mode
  - `lab01.c` is only owner read/write accessible

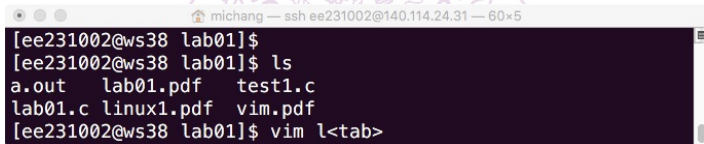


```
michang — ssh ee231002@140.114.24.31 — 62x10
[ee231002@ws38 lab01]$ ls -l
total 41
-rwxr-xr-x 1 ee231002 course 6996 Sep 12 19:36 a.out
-rw-r--r-- 1 ee231002 course 379 Sep 12 19:39 lab01.c
[ee231002@ws38 lab01]$ chmod 600 lab01.c
[ee231002@ws38 lab01]$ ls -l
total 41
-rwxr-xr-x 1 ee231002 course 6996 Sep 12 19:36 a.out
-rw----- 1 ee231002 course 379 Sep 12 19:39 lab01.c
[ee231002@ws38 lab01]$ chmod 1700 ~/C_program
```

- Please issue the command as the last line above to protect your `C_program` directory

# Some Useful linux Commands

- `clear`: clear window
- `↑`: re-enter the previous `linux` command
  - Can key in more than once
- `<tab>`: complete file name if possible
  - In the example below, the last command will be completed as  
`$ vim lab01.c`



A terminal window titled "michang — ssh ee231002@140.114.24.31 — 60x5". The prompt is "[ee231002@ws38 lab01]\$". The user enters "ls", and the terminal displays the output: "a.out lab01.pdf test1.c", "lab01.c linux1.pdf vim.pdf". Then, the user enters "vim l", and the terminal shows "vim l" followed by a tab completion arrow. The final command shown is "vim lab01.c".

```
[ee231002@ws38 lab01]$  
[ee231002@ws38 lab01]$ ls  
a.out  lab01.pdf  test1.c  
lab01.c linux1.pdf vim.pdf  
[ee231002@ws38 lab01]$ vim l<tab>
```