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 — 80×24

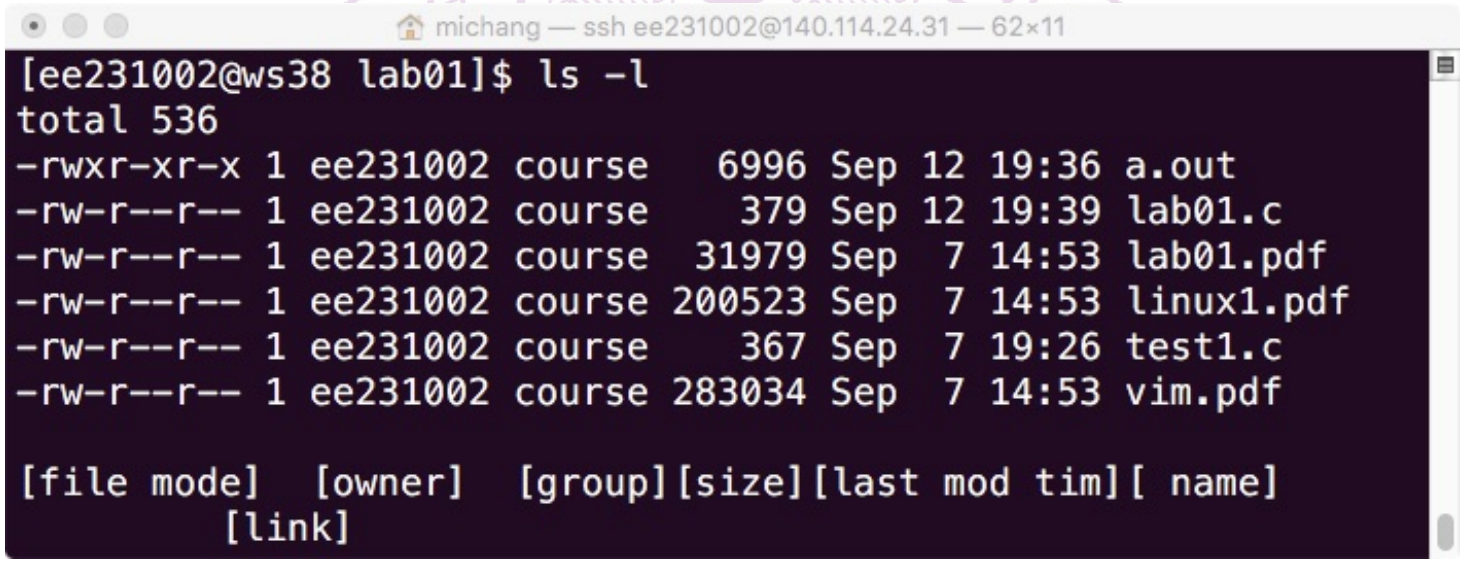
```
[ee231002@ws38 ~]$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

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

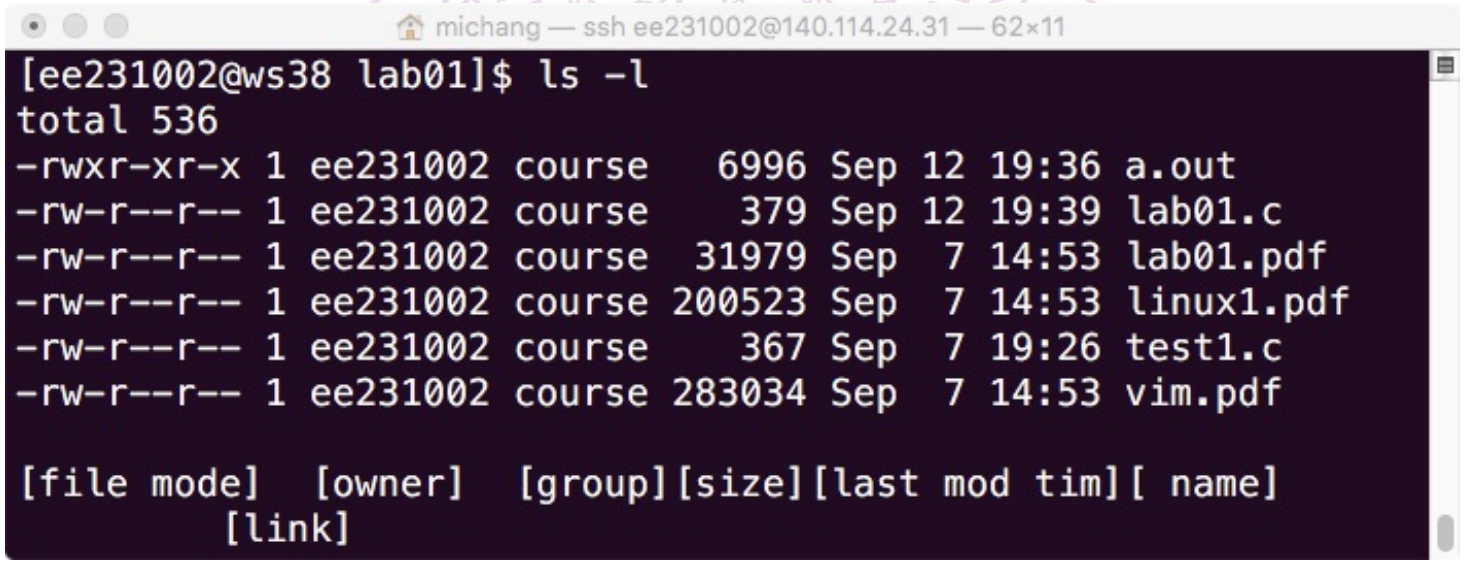
ls

- `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

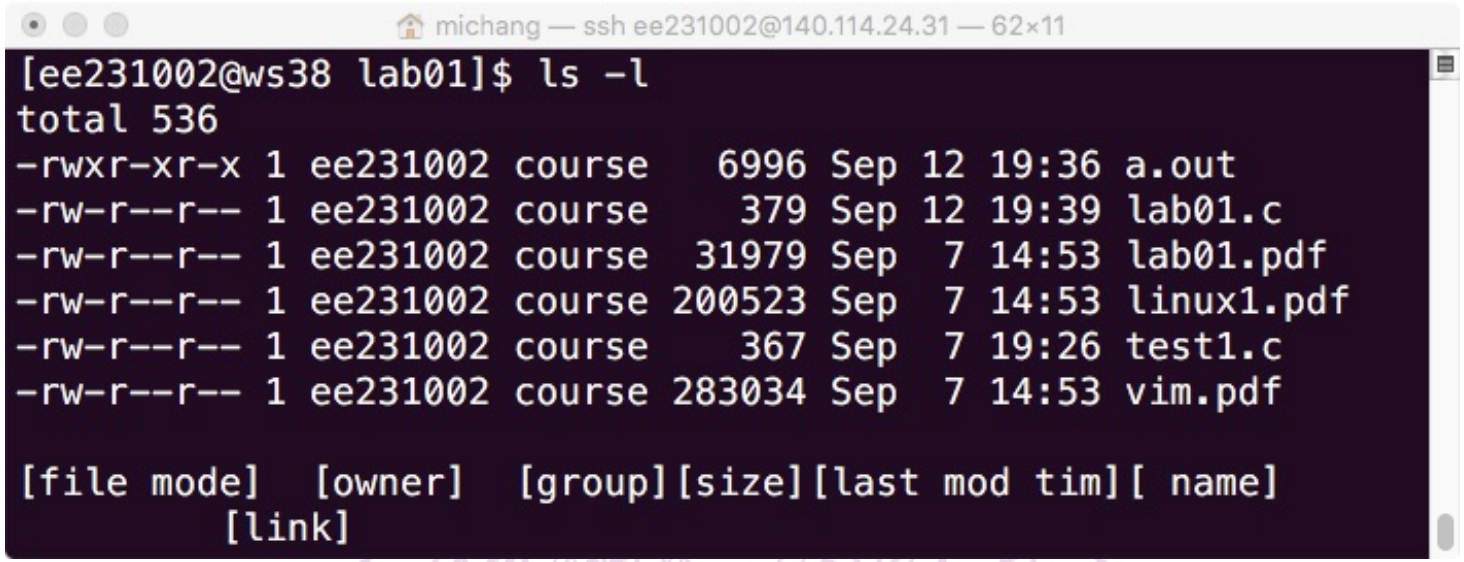


File Modes

- 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



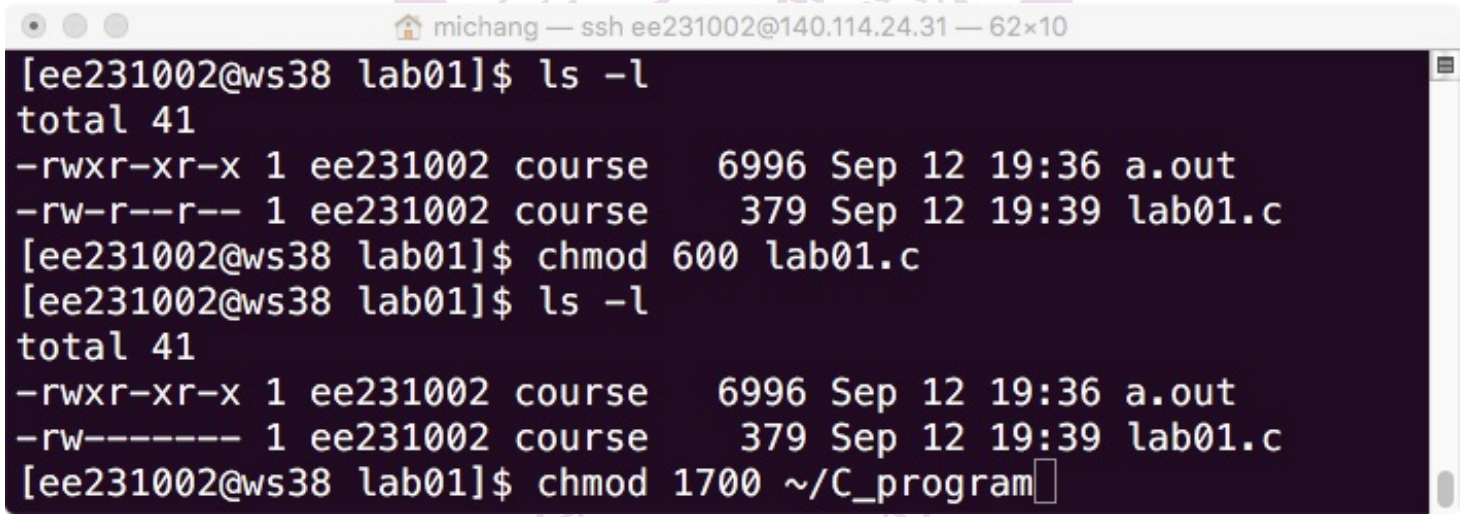
File Modes



- 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)

chmod

- 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



- 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`