

[Linux Basic Commands]

[Linux Basic Commands]

1. Linux Commands

(1) Basic Commands

- Collection

- `cd / ls`
- `mkdir / touch`
- `rm / mv / cp`
- `echo / cat`
- `size`
- `grep` and `|` and `>`, `<`

(2) Useful Setup for Linux Terminal

- Change `hostname`:
- Eliminate `username@hostname:~$` in Terminal :

(3) `sudo ... update / upgrade`

- Fetch the latest version of the package list
- Download and install the updates for each outdated package

2. Vi/Vim

(1) Useful Setup

(2) Useful Hot Keys

- 1) `:sh` in vi command mode, then `[ctrl] + d` in terminal back to vi
- 2) `:! ...` in vi command mode
- 3) `:split -> :e file2.c -> [ctrl] ww` to switch -> `:q` to quit
- 4) `[ctrl]+v` in vi cmd. mode -> `[shift]+i` then do sth -> `[esc]`
- 5) `[ctrl]+v` in vi cmd. mode -> `c` to delete

3. Commands of Compilation with gcc/g++/tcc

(1) `gcc / g++`

- Link `math.h` Library
- Compile `file.c` with `gcc / g++`

(2) `tcc`

- Compile `file.c`

1. Linux Commands

(1) Basic Commands

- Collection

File Commands	System Info
ls - directory listing ls -al - formatted listing with hidden files cd dir - change directory to <i>dir</i> cd - change to home cd - - change back to previous directory pwd - show current directory mkdir dir - create a directory <i>dir</i> rm file - delete <i>file</i> rm -r dir - delete directory <i>dir</i> rm -f file - force remove <i>file</i> rm -rf dir - force remove directory <i>dir</i> * cp file1 file2 - copy <i>file1</i> to <i>file2</i> cp -r dir1 dir2 - copy <i>dir1</i> to <i>dir2</i> ; create <i>dir2</i> if it doesn't exist mv file1 file2 - rename or move <i>file1</i> to <i>file2</i> if <i>file2</i> is an existing directory, moves <i>file1</i> into directory <i>file2</i> ln -s file link - create symbolic link <i>link</i> to <i>file</i> touch file - create or update <i>file</i> cat > file - places standard input into <i>file</i> more file - output the contents of <i>file</i> head file - output the first 10 lines of <i>file</i> tail file - output the last 10 lines of <i>file</i> tail -f file - output the contents of <i>file</i> as it grows, starting with the last 10 lines	date - show the current date and time cal - show this month's calendar uptime - show current uptime w - display who is online whoami - who you are logged in as finger user - display information about <i>user</i> uname -a - show kernel information cat /proc/cpuinfo - cpu information cat /proc/meminfo - memory information man command - show the manual for <i>command</i> df - show disk usage du - show directory space usage free - show memory and swap usage whereis app - show possible locations of <i>app</i> which app - show which <i>app</i> will be run by default type app - show which <i>app</i> or builtin will be run
Process Management	Compression
ps - display your currently active processes top - display all running processes kill pid - kill process id <i>pid</i> killall proc - kill all processes named <i>proc</i> * jobs - lists stopped or background jobs bg - resume a stopped job in the background fg - brings the most recent job to foreground fg n - brings job <i>n</i> to the foreground	tar cf file.tar files - create a tar named <i>file.tar</i> containing <i>files</i> tar xf file.tar - extract the files from <i>file.tar</i> tar czf file.tar.gz files - create a tar with Gzip compression tar xzf file.tar.gz - extract a tar using Gzip tar cjf file.tar.bz2 - create a tar with Bzip2 compression tar xjf file.tar.bz2 - extract a tar using Bzip2 gzip file - compresses <i>file</i> and renames it to <i>file.gz</i> gzip -d file.gz - expands <i>file.gz</i> back to <i>file</i>
File Permissions	Network
chmod octal file - change the permissions of <i>file</i> to <i>octal</i> , which can be found separately for user, group, and world by adding: <ul style="list-style-type: none"> 4 - read (r) 2 - write (w) 1 - execute (x) Examples: chmod 777 - read, write, execute for all chmod 755 - rwx for owner, rx for group and world For more options, see man chmod .	ping host - ping <i>host</i> and output results whois domain - get whois information for <i>domain</i> dig domain - get DNS information for <i>domain</i> dig -x host - reverse lookup <i>host</i> wget file - download <i>file</i> wget -c file - continue a stopped download
SSH	Installation
ssh [-p port] user@host - connect to <i>host</i> as <i>user</i> , optionally on custom port <i>port</i> scp [-P port] user@host:path1 path2 - copy the remote file at <i>path1</i> to local location <i>path2</i> ssh-copy-id user@host - add your key to <i>host</i> for <i>user</i> to enable a keyed or passwordless login	Install from source: ./configure make make install dpkg -i pkg.deb - install a package (Debian) rpm -Uvh pkg.rpm - install a package (RPM)
Searching	Shortcuts
grep pattern files - search for <i>pattern</i> in <i>files</i> grep -r pattern dir - search for <i>pattern</i> in <i>dir</i> command grep pattern - search for <i>pattern</i> in the output of <i>command</i> locate file - find all instances of <i>file</i>	Ctrl+C - halts the current command Ctrl+Z - stops the current command, resume with fg in the foreground or bg in the background Ctrl+D - log out of current session, similar to exit Ctrl+W - erases one word in the current line Ctrl+U - erases the whole line Ctrl+R - type to bring up a recent command !! - repeats the last command exit - log out of current session

* use with extreme caution.



- **cd / ls**

- `~`: Home directory
- `.`: Current directory
- `..`: Parent directory
- `*`: A **wildcard character** used to match any number of characters (including none)

```
# ls
$ ls
$ ls -l
$ ls file -l
$ ls /dir1/dir2
    ./dir1/dir2
    ~/dir1/dir2
$ ls ./dir1 -l

# cd
$ cd dir1
$ cd dir1/dir2
$ cd .. # go back to the parent directory

# *
$ ls *
$ cp * /path/to/destination/ # move all files from the current directory to the
specified destination
$ mv * /path/to/destination/ # move all files from the current directory to the specified
destination
$ find . -name "*.txt" # using the find tool, searches for all files with a ".txt"
extension in the
                        # current directory and its subdirectories
```

- **mkdir / touch**

```
$ mkdir newDir # creat a new folder (directory) called 'newDir'
$ touch newfile # creat a new file called `newfile`
$ touch newfile.c
```

- **rm / mv / cp**

```

# rm
$ rm file
$ rm dir1 -r # '-r': Recursively remove directories and their contents.
$ rm dir1 -rf # '-f': Force the removal without prompting for confirmation
$ rm -r * # '*': A wildcard character that matches any file or directory in the current
directory
        # When combined, rm -r * instructs the system to recursively remove all files
and directories in
        # the current directory.

# mv
$ mv oldname newname

# cp
$ cp oldfile

```

- ehco / cat

```

# echo
$ echo "hello world!"
$ echo "hello world!" > file # write in (overlap at the same time) 'file' with "hello
world"
$ echo [input] | ./file.out # let [input] be the input when executing 'file.out'
$ echo $? # check the return value of program; 0 means the program has been executed
successfully, while
        # non-zero value means that

# cat
$ cat file
$ cat file1 > file2

```

- size

```

$ size file.c
$ size file.out

```

- grep and | and >, <

- [left] | [right]: Let [left] be some input of [right].
- [left] > [right]: Let the output of [left] be the input of [right]
- [left] < [right]: Let [right] be the input of [left]
- e.g.

```

$ gcc -E file.c | grep -v "something" # delete the lines containing "something"
$ cat file.c | grep "something"
$ ./file [arg] | grep "something" > result_file
$ ./file < input.txt > output.txt

```

(2) Useful Setup for Linux Terminal

- Change `hostname`:

```
$ (sudo) vi /etc/hostname
```

- then change it to whatever you want and restart linux(ubuntu).

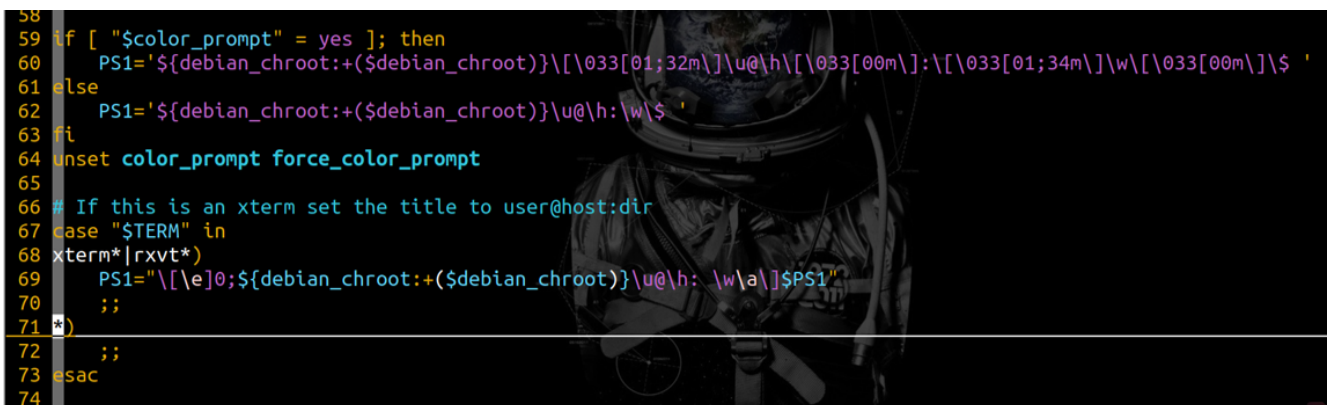
- Eliminate `username@hostname:~$` in Terminal:

- Command to **recover** file `.bashrc`:

```
$ cp /etc/skel/.bashrc ~/
```

- Change file `.bashrc`:

```
$ vi ~/.bashrc +60 # go to line 60
```



```
58
59 if [ "$color_prompt" = yes ]; then
60     PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
61 else
62     PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
63 fi
64 unset color_prompt force_color_prompt
65
66 # If this is an xterm set the title to user@host:dir
67 case "$TERM" in
68 xterm*|rxvt*)
69     PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
70     ;;
71 *)
72     ;;
73 esac
74
```

- Line 60:
 - delete `\u@\h` and `:` between `[00m\]` and `\[\033...`
 - insert a blank `' '`
- Line 62:
 - delete `\u@\h:`
 - insert a blank `' '`
- Line 69:
 - delete `\u@\h:`
 - insert a blank `' '`

(3) `sudo ... update / upgrade`

- Fetch the latest version of the package list

```
$ sudo apt-get update
```

- Download and install the updates for each outdated package

```
$ sudo apt-get upgrade
```

2. Vi / Vim

(1) Useful Setup

- Type the following command at Terminal:

```
$ (sudo) vi /etc/vim/vimrc
```

- Then add the following statements via vim at the front of file `vimrc`.

```
# Define tab as 4 spaces
set tabstop=4
set shiftwidth=4
set expandtab

# Number Index Column
set nu

# Cursor
set cursorline
set cursorcolumn

syntax on
set smarttab
set mouse=a
set autoindent
```

(2) Useful Hot Keys

- 1) `:sh` in vi command mode, then `[ctrl] + d` in terminal back to vi
- 2) `:! ...` in vi command mode
- 3) `:split` -> `:e file2.c` -> `[ctrl] ww` to switch -> `:q` to quit
- 4) `[ctrl]+v` in vi cmd. mode -> `[shift]+i` then do sth -> `[esc]`
- 5) `[ctrl]+v` in vi cmd. mode -> `c` to delete

3. Commands of Compilation with gcc/g++/tcc

(1) gcc / g++

- Link `math.h` Library

```
$ gcc file.c -o file -lm
```

- Compile `file.c` with `gcc / g++`

- Basic Commands:

```
$ gcc file.c
$ gcc file.c -o file
```

- Useful Postfix :

```
-w      # Lowercase 'w': Ignore warnings
-W      # Capitalcase 'w': Show warnings
-Wall   # Show all warnings
-Werror # Take warnings as Errors
```

(2) tcc

- Compile `file.c`

- Compile and run `file.c`

```
$ tcc file.c -run
```

- Compile and run `file.c` without showing warnings

```
$ tcc file.c -run -w
```