

Linux commands

Andy Kim

Shortcuts

Bash command line shortcuts

- ctrl+a e: move to the front / end of a line
- ctrl+u k: cut to the cursor / from the cursor in a line
- ctrl+b f: move back / forward a char
- alt+b f: move back / forward a word
- ctrl+w: delete a word in the left
- ctrl+d h: delete / backspace a char
- ctrl+y: paste
- ctrl+_ : undo

Terminal shortcuts

- ctrl+alt+t: open a terminal
- ctrl+z: stop / suspend
- ctrl+c: kill
- ctrl+d: close (at the beginning of the prompt)
- ctrl+l: clear
- ctrl+r: reverse history search
- ctrl+p n: command history search
- ctrl+s q: stop output to screen / resume
- shift+page-up page-down: scrolling

Linux commands

Basic

- history, !, !#, !<command>: history search/execute
- pwd: present working directory
- which <command>: shows absolute path of the command; useful when there are multiple versions
- ls -asl: a hidden, s size, l long-format
- ls -F: * executable / directory @ symbolic link
- ls -R: recursive to all subdirectories
- clear, cat [-n], touch (creates an empty file), more <file> (page view), head/tail [-n] <file>
- wc [-lwc] <file>: l line, w word, c char count
- cp [-r] dir1 dir2: recursive copy including subdirectories
- mv [-i], rm [-ir]: i interactive, r recursive (to remove all subdirectories)
- chmod [-R] ugoa+-=rwx <file>
- file <file>: shows attributes of the file

Shall commands

- Bash shell user env setting: `~/ .bash_profile(login)`, `~/ .bashrc(login, subshell)`
- Setting environment variable:
 - `$ var1=value1 # define a shell variable`
 - `$ export var1 # declares local variable to global variables`
 - `$ echo $var1 # shows the value of the shell variable`
- Useful `~/ .bashrc` inclusion
 - `export PATH=$PATH:.`
 - `# History search code`
 - `bind ‘“\e[A”: history-search-backward’`
 - `bind ‘“\e[B”: history-search-forward’`
- `. ~/ .bashrc`: executes `~/ .bashrc` (source command) i
- `.` (source command) lets the bash shell script to run in the current terminal
- Command substitution: ‘commands here substituted with results’ (quotations are backtick)
- `>`, `>>`, `|`, `*?[*]`: redirection, append, pipe, and wildcards
- `command&`: background processing
- `jobs [%job_number]`
- `fg %job_number`
- `kill %job_number` or `kill process_number`
- `wait process_number`
- `nohup <command> &`: enables to run the command after logout
- `ps -ef`: shows all info regarding processes
- `ps -ef | grep -w <word>`: whole word matching – the same as `pgrep`
- `find <directory> -name <filename> -print` or `-ls`
- Other file handling commands: `grep`, `sort`, `cmp`, `diff`, `cat` (to merge; e.g., `cat f1 f2 > f3`)
- Disk related commands: `df`, `du [-s:sum]` (refer to man)
- Archive (without compress): `tar` (refer to man)
- Compress: `gzip` (refer to man)
- `wget “<filename>”`: downloads a file from the internet
- `strings <filename>`: `grep` strings from a file (even from a binary)
- defining a function in `.bashrc`

```
c() {  
    if [ "$1" == "" ]  
    then  
        cd ~  
    else  
        cd "$1"  
    fi  
    ls -CF  
}
```
- `$mkdir -p <path to a dir> && cd $_ # executes only if first is successful`
- `$ln -s path/to/file path/to/link`: makes a symbolic link

Other topics

- Bash shell script (start with `#!/bin/bash` ; make executable)
- C-programming
- Process handling
- System admin