

- ubuntu command
  - system info
  - work place
  - text edit
  - directory
  - Pipeline&Shell
  - if for while
  - user
  - permission
  - secret permission
  - access control list (ACL)
  - Secure Shell (SSH)
  - File Transfer Protocol (FTP)
    - FTP COMMAND

# ubuntu command

---

## system info

---

- `echo [str]` say hello
  - `echo $DISPLAY`
  - `echo hello world`
- `ps` show process
  - `-a` show all process
  - `-u` show user
  - `-x` no terminal process
- `top` linux task manager
- `pidof [PROCESS_file]` show process id
  - `pidof bash`
- `kill [PROCESS_file]` kill process
- `killall [PROCESS_file]` kill all process
- `uname -a` linux version info
- `ifconfig` internet info
- `uptime` upload time
- `free -h` storage info

- `who` user info
- `last` login history info
- `history` command history
- `sosreport` collect system info to diagnose
- `whereis [command]` see command location
  - all command store in `/usr/bin`

## work place

---

- `pwd` present working directory
- `cd` change directory
  - `cd ..` back to super directory
  - `cd ~` back to origin directory
  - `cd -` back to previous directory
- `ls` list directory
  - `-a` all directory
  - `-l` details of directory

## text edit

---

- `cat [file]` read short text
  - `cat [file]|tr [old] [new]` change old char to new char
- `more [file]` read long text
- `head -n [int] [file]` see n-lines text
- `tail [file]` see last n-line text
  - `-n [int]` last n-line
  - `-f` keep fresh
- `wc [file]` word count
  - `-l` number of lines
  - `-w` number of word
  - `-c` number of bytes
- `stat [file]` storage and data info
- `cut -d [char] -f1 [file]` see info by column
- `diff --brief [file1] [file2]` whether two text are same or different

## directory

- `touch [file]` create file
- `mkdir [file]` make directory
  - `-p a/b/c` create recursive directory
- `cp [file1] [file2]` copy files
  - `-i` if file exist, whether cover that
- `mv [file1] [file2]` cut file(if file1 and file2 have same directory ->rename)
- `rm [file]` remove file
  - `-rf` reinforce `br`
  - `-r [dirname]` remove directory
- `dd if=[file] of=[file]` copy file with special limitation
  - `count=[int] bs=[int M]` block number and block size
  - *example:* `dd if=[file] of=xxx.iso file [file]` check file format
- `tar [gzip] [file]` compress/decompress file
  - `-c` create zip file
  - `-x` unzip
  - `-t` see zip file
  - `-z` gzip
  - `-j` bzip2
  - `-v` show zip/unzip process
  - `-f` object name
  - `-p` save limitation and attribute
  - `-P` use absolute path
  - `-C` zip to directory

*example:*

- `tar -cvf [file.tar] [dir]` create .tar file
  - `tar -cvzf [file.tar.gz] [dir]` create .tar.gz file
  - `tar -xvf [file.tar/file.tar.gz] -C [dir]` unzip .tar/.tar.gz file to directory
  - `tar -tvf [file.tar/file.tar.gz]` list .tar/.tar.gz file
  - `tar -xvf [file.tar] [file]` unzip simple file from .tar file
    - `--wildcards 'regex'` unzip specific files
  - `tar -zxvf [file.tar.gz] [file]` unzip simple file from .tar.gz file
    - `--wildcards 'regex'` unzip specific files
  - `tar -rvf [file.tar] [file]` add file into .tar file
- `grep [str] [file]` search text info
    - `-n` line
    - `-v` reverse select
  - `find [dir] [OPTION] -print` find file info
    - `-name [regex]` match name
    - `-perm [int]` permission
    - `-mtime -[int] +[int]` modified time(- in n days,+before n days)
    - `-atime -[int] +[int]` access time
    - `-ctime -[int] +[int]` change time
    - `--type b/d/c/p/l/f` match file type (block device,dialog,char device,pipeline,link,text)
    - `-prune [dir]` omit specific directory
    - `-exec` execute demand

## Pipeline&Shell

---

- `[command]>[file]` command output to file
- `[command]>>[file]` command output add to file
- `[command1] | [command2]` command 1 output as command 2 input
- `bash [file.sh] [$VARIABLE]` execute script
- `at [time]` execute command at time
- `&& || ! = !=` logic

- [ test command ]
  - [ -d [dir] ] is directory?
  - [ -e [file] ] is exist?
  - [ -f [file] ] is normal file?
  - [ -r [file] ] readable?
  - [ -w [file] ] writable?
  - [ -x [file] ] executable?
  - [ -z [str] ] void?
- compare operator
  - [ [int] -eq [int] ] equal
  - [ [int] -ne [int] ] not equal
  - [ [int] -gt [int] ] greater than
  - [ [int] -lt [int] ] less than
  - [ [int] -ge [int] ] greater equal
  - [ [int] -le [int] ] less equal

## if for while

---

- if

```
if [condition]
then [command]
elif [condition]
then [command]
else [command]
fi
```

- for

```
for [$VARIABLE] in [LIST]
do
[command]
done
```

- while

```
while [condition]
do
[command]
done
```

# user

---

- `useradd [name]` create new user
  - `-m` create directory name in home
  - `-d [dir]` home directory
  - `-e [YYYY-MM-DD]` expire time
  - `-u [int]` user id
  - `-g [groupname]` add the user in some group
  - `-s [dir]` terminal select(usually `/bin/bash`)
- `id [name]` see user info
- `cat /etc/passwd` see all user
- `cat /etc/shadow` see user password
- `groupadd [str]` create new group
- `usermod [OPTION] [name]` modify user attribute
  - `-p [password]` change password
  - `-c [str]` add comments
  - `-md [dir]` move the user home to new directory
  - `-e [YYYY-MM-DD]` expire time
  - `-g [groupname]` add the user in some group
  - `-L` lock user
  - `-U` unlock user
  - `-u [int]` change user id
  - `-s [dir]` change terminal
- `passwd [OPTION] [name]` password
  - `-l` lock user
  - `-u` unlock user
  - `--stdin` use standard input change

- **-d** can not set password
- **-e** change password next time
- **-S** show whether user is locked
- **userdel** **[OPTION]** **[name]** delete user
  - **-f** force delete
  - **-r** delete user and home

## permission

---

- **su** **[name]** or **su -** **[name]** change login user
- **sudo** **[command]** superuser execute command
  - trick: move password when using sudo
    1. **whereis** **[command]** **command path**
    2. **visudo**
    3. add **[command]** **ALL=NOPASSWD:[command path]**
- **chmod** **WhoWhatWhich** **[file]** change mode of file
  - Who: **ugo**a(user,group,other,all)
  - What: **+-=**(add,remove,set exact)
  - Which: **rwxs**t(read,write,execute,SGID,SBIT)
  - **-R** recursion for file
- **chown** **[name]:[groupname]** **[file]** change group of file
- **-R** recursion for file

## secret permission

---

- **chattr** **+[OPTION]/-[OPTION]** **[file]** change attribute of file secretly
  - **i** unchangable
  - **a** append only
  - **S** synchronize to disk
  - **s** remove from disk forever
  - **A** not change the access time

- **b** not change save time
- **D** check zip file fail
- **d** when using dump, omit this file
- **c** gzip this file
- **u** delete the file but recoverable
- **X** access zip file directly
- **lsattr [file]** see attribute of file secretly

## access control list (ACL)

---

- **getfacl [file]** check file permission
- **set [OPTION] [file]** set file permission
  - **-m u:[USERNAME]:[rwx]** set user permission
  - **-m g:[GROUPNAME]:[rwx]** set group permission
  - **-x u:[USERNAME]:[rwx]** delete user permission
  - **-x g:[GROUPNAME]:[rwx]** delete group permission
  - **-b** delete all ACL permission
  - **-d u:[USERNAME]:[rwx]** set default user permission
  - **-d g:[GROUPNAME]:[rwx]** set default group permission
  - **-k** delete default group permission
  - **-R** use regression

## Secure Shell (SSH)

---

1. **apt install openssh-server** create SSH server
2. **service ssh status** check whether SSH is open
3. **service ssh start** start SSH server
4. **ssh root@[ip address]** use SSH login

## File Transfer Protocol (FTP)

---

1. **apt install vsftpd** install vsftpd
2. **useradd [ftp name] -md [ftp dir] -s /sbin/nologin** create FTP user and FTP file
3. **chown -R [ftp name]:[groupname] [ftp dir]** change FTP permission



4. `passwd [ftp name]` set FTP password
5. `whereis vsftpd` find **`vsftpd.conf`** to set vsftpd
6. `service vsftpd status` check FTP status
7. `service vsftpd start` FTP service start
8. `ftp [ip address]` login ftp

## FTP COMMAND

- `bye` or `close` or `quit` Terminates an FTP connection.
- `cd` Changes the current working directory on the FTP host server.
- `cwd` Changes the current directory to the specified remote directory.
- `dir` Requests a directory of files uploaded or available for download.
- `get` Downloads a single file.
- `ls` Requests a list of file names uploaded or available for download.
- `mget` Interactively downloads multiple files.
- `mput` Interactively uploads multiple files.
- `open` Starts an FTP connection.
- `pasv` Tells the server to enter passive mode, in which the server waits for the client to establish a connection rather than attempting to connect to a port the client specifies.
- `put` Uploads a single file.
- `pwd` Queries the current working directory.
- `ren` Renames or moves a file.
- `site` Executes a site-specific command.
- `type` Sets the file transfer mode:ASCII Binary