



# Centos

```
# cat /etc/redhat-release
CentOS Linux release 7.9.2009 (Core)
```



```
# uname -a
Linux oxoolcommunity.ossii.com.tw 3.10.0-1160.66.1.el7.x86_64 #1 SMP Wed May 18 16:02:34 UTC 2022
x86_64 x86_64 x86_64 GNU/Linux
```

██:██████linux██████

██ CentOS ~~3.10.0-1160.66.1.el7.x86\_64~~██████ yum update ██████

# Debian



```
cat /etc/debian_version
```

# Ubuntu

```
lsb_release -a
```

```
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.5 LTS
Release:        22.04
Codename:       jammy
```

```
cat /etc/issue
```

Ubuntu 22.04.5 LTS \n \l

```
#!/bin/sh ([[ ! -x /bin/bash ]] && shell)
$? [0] [1] [2]
$1 [ ] $2 [ ]
$0 [shell script]
$@ [ ] $1, $2,... [ ] $@ [ ] "$1" "$2" "$3"...
$* [ ] $* [ ] "$1 $2 $3..."
```

`[]""[[]$var[[]][[]]"[]$var[[]][[]]`

```
#!/bin/bash

read -p "Please enter your age:" Y_AGE

read -p "Please enter your friend's age:" F_AGE
sum1=$Y_AGE + $F_AGE

sum2=`expr $Y_AGE + $F_AGE`

sum3=$((Y_AGE + F_AGE))
echo "Y_AGE + F_AGE = sum1"

echo "Y_AGE + F_AGE = sum2"

#Bash Shell
echo "sum1 = $sum1"

echo "sum2 = $sum2"

echo "sum3 = $sum3"
```

```
#ubuntu adduser {}  
sudo adduser {{ }}
```

```

ron@DESKTOP-MB641C8:~$ sudo adduser ron1
Adding user `ron1' ...
Adding new group `ron1' (1001) ...
Adding new user `ron1' (1001) with group `ron1' ...
The home directory `/home/ron1' already exists.  Not copying from `/etc/skel'.
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for ron1
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]

```

~~EX~~ `sudo adduser {username} sudo`

~~User~~ `add [ ] ( [ ] )`

```

-d [ ] home [ ]
-p [ ]
-e [ ] YYYY-MM-DD
-g [ ]
-G [ ]
-M [ ]
-m [ ]/etc/skel/[ ]

```

`[ ] ssh [ ] sftp`

Match `[ ] UsePAM [ ]`

```

Match user user [ ]
ChrootDirectory /srv/ftp # [ ]
AllowTcpForwarding yes # [ ] ssh tcp forwarding
ForceCommand "[ ]"

```

`[ ] ip`

Match User ansible,backup,support Address \*,!176.x.x.x [ ]

`[ ] hostname`

Match user [ ] Host "hostname"

`[ ] ip`

Match address IP

port

Match Localport

ex

Match User user2, LocalPort 2201DenyUsers user2

port

ListenAddress hostname|address

ListenAddress hostname:port

ListenAddress IPv4\_address:port

ListenAddress [hostname|address]:port

AllowUsers AllowGroups denyUsers denyGroups  
ssh ip

```
AllowUsers <user1> <user2> <user3>
```

```
AllowGroups <group>
```

```
DenyUsers *
```

```
DenyGroups no-sshDeny
```

```
Users username@192.168.1.*
```

/etc/hosts.allow /etc/hosts.deny IP

hosts.allow hosts.allow tcp\_wrapper tcpd

sshd

```
ldd `which sshd` | grep libwrap
```

```
libwrap.so.0 => /lib/x86_64-linux-gnu/libwrap.so.0 (0x00007fedac9e6000)
```

:<>:<ip>

EX:

```
sshd : 192.168.1.100
```

```
vsftpd : 192.168.1.*
```

```
vsftpd : ALL
```

```
sshd : 192.168.1.1,192.168.8.*
```

[]:

1. hosts.allow []hosts.deny[]
2. any[]ip[]local[]ip[]EXCEPT[]

## shell []

```
n=1
m=2
echo $n+$m
[]
1+2 []shell[]
```

```
n=1
m=2
echo $((n+m))
[]
3
```

## [][]

```
[][]
echo $PATH #[]
env #[]
set #[]shell[]
export #[]shell[]
a=asd #[]shell[]
export a=asd #[]shell[]
unset a #[]
readonly a #[]
```

## [][][]

```
[]/etc/profile #[]
[]$HOME/.profile #[]
[]/etc/bach.bachrc #[]bash shell
[]$HOME/.bashrc #[]bash shell
```

**[][][]source [] . [][]source /etc/profile [] ./etc/profile**

□□□□

2>/dev/null □□□□□□

□□

□□

```
apt install clamav clamdscan clamav-daemon clamav-freshclam
clamscan -r --quiet -l (□□log□□□) --exclude-dir (□□□□□□) □□□□□□
clamdscan - < (□□□□□□□□)
cat □□□□□□□□| clamdscan -
```

**clamdtop** □□**htop**□□

□□□

□□0□□□□1□□□□2□□□□

**clamscan** □ **clamdscan**□□

**clamscan**□□□□□□

**clamdscan**□□□□□□□□□**clamdtop**□□□□□

□□□□

□□□□"&"

□□□

1|2 (□□□□□□□□□□□□)

1||2 (1□□□□□2)

1&&2 □1□□□□2□

1;2 □□□□□□□□□□

□□□□□□

```
sudo mount -t cifs -o username=<win_share_user>,password=<win_share_password>
//WIN_SHARE_IP/<share_name> /mnt/win_share
```

□□fstab□□

```
□□□□□ □□□□□ cifs defaults,uid=□□□□□□□,gid=□□□□□□□,username=□□□□,password=□□□□,iocharset=utf8 0 0
```

**EX:1.fstab**□□□□□□□□□□**smb**□□□□□□**vers={1.0:2.0:3.0}**

**2.□mount**□□□□□□□□□□**040**

imgimg -o loop

```
sudo mount -O loop {img} {}
```

## mountpoint

```
mountpoint <>
```

-q:

## mail

```
mail -s "This is the subject" somebody@example.com <<< 'This is the message' 
```

```
mail -s "Hello World" user@yourmaildomain.com < /home/user/mailcontent.txt 
```

```
echo "This is the message body" | mail -s "This is the subject" mail@example.com
```

```
mail -s "Hello World" user1@example.com,user2@example.com 
```

-B (BCC)

-C (CC)

-aFrom:Harry\<harry@gmail.com\>

## taskset

( )

```
apt install fcitx5 fcitx5-chewing
```

( )

im-config

fcitx5-mozc

## tasksetxfce4ubuntu xfce fail to start session

gdm3

```
sudo apt install gdm3
```

```
sudo systemctl set-default graphical
```

or

```
sudo systemctl set-default graphical.target
```

```
dpkg-reconfigure tzdata
```

```
dpkg-reconfigure tzdata
```

## iptables

```
sudo ufw allow proto tcp from 192.168.56.0/24 to 192.168.56.203 port 22
```

## port

```
lsof -i -P -n | grep LISTEN  
netstat -tulpn | grep LISTEN  
sudo nmap -sT -O localhost
```

```
du -h --max-depth=1
```

```
du -h --max-depth=1
```

```
du -h --max-depth=1
```

```
df -h | trace df
```

```
trace df
```

```
umount -f
```

```
umount -f {  
killall -9 df
```

```
lshw
```

```
lshw  
dmidecode -t 17 (dmidecode --type memory dmidecode --type 17)
```

```
dmidecode -t 17 (dmidecode --type memory dmidecode --type 17)
```



```
free
free -m
cat /proc/meminfo
```

```
lsmod | grep hardinfo
```

```
sudo apt install hardinfo
```

## netplan

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3: #
      addresses: [10.0.2.15/24] # IP
      gateway4: 10.0.2.1 # IPV4 Getway ip
      nameservers:
        addresses: [8.8.8.8,8.8.4.4] # DNS server ip
      dhcp4: no # dhcp IP
# Host only enp0s8
    enp0s8:
      addresses: [192.168.56.101/24] # IP
      routes:
        - to: 192.168.56.1/24
          via: 192.168.56.1
          metric: 100
      #gateway4: 192.168.56.1 # IPV4 Getway ip
      #nameservers:
      #addresses: [8.8.8.8,8.8.4.4] # DNS server ip
      dhcp4: no # dhcp IP
      dhcp6: no # dhcp IP
version: 2
```

```
sudo netplan try sudo netplan apply
```

```
nmcli network manager
```

```
network:
  version: 2
```

ethernets:

NM-f52160ba-1cb2-4d49-955e-84a6f51adb8d: #

renderer: NetworkManager

match:

name: "enp0s8"

macaddress: "FF:FF:FF:E3:AD:CA" MAC

addresses:

- "192.168.100.51/24"

ipv6-address-generation: "stable-privacy"

wakeonlan: true

networkmanager:

uuid: "f52160ba-1cb2-4d49-955e-84a6f51adb8d"

name: "Ethernet connection 2"

passthrough:

ipv6.method: "ignore"

proxy.\_: ""

# device not managed

1

```
sudo vi /etc/NetworkMnager/NetworkManager.conf
```

```
[main]
```

```
plugins=ifupdown,keyfile,ofono
```

```
dns=dnsmasq
```

```
[ifupdown]
```

```
managed=false <==true
```

```
sudo systemctl restart NetworkManager
```

```
nmcli d ---
```

```
sudo nmcli dev set (device name) managed yes ----device
```

```
sudo mv /usr/lib/NetworkManager/conf.d/10-globally-managed-devices.conf /usr/lib/NetworkManager/conf.d/10-globally-managed-devices.conf_orig
sudo touch /usr/lib/NetworkManager/conf.d/10-globally-managed-devices.conf
sudo systemctl restart NetworkManager
nmcli d ---[ ]
```

## [ ]wifi radio

```
nmcli r wifi on
```

[ ]wifi[ ]

```
nmcli d wifi list
```

[ ]wifi

```
nmcli d wifi connect {SSID} password {password}
```

[ ]wifi

```
nmcli c add type wifi con-name <name> ifname wlan0 ssid <ssid>
nmcli c modify <name> wifi-sec.key-mgmt wpa-psk wifi-sec.psk <password>

#establish wifi ssid
nmcli c up <name>
```

[ ]

```
curl -I http://yourdomain/test.jpg
```

## [ ](wget)

```
wget -O [ ] [ ]
wget -c [ ]
wget -i [ ] [ ]
wget -b log[ ] [ ]
wget --limit-rate=[ ] [ ]
wget --http-user=my_user --http-password=my_password http://www.example.com/my_file[http[ ]
wget --ftp-user=my_user --ftp-password=my_password ftp://ftp.example.com/my_file(ftp[ ])
```

```
wget --tries=50 50
wget --user-agent="Mozilla" 
wget --spider 
wget --mirror -p --convert-links -P  1
wget -r http://www.example.com/()
wget --secure-protocol=TLSv1_2 {url site} --no-check-certificate  TLS1.2
```

**:-nd()-np () -A()**

**EX:wget -r -nd -np -A pdf,png http://www.example.com/path1/path2/**

## Curl

```
curl -o {} {} #
curl -O -C {} #
curl -O --limit-rate {} {} #--max-filesize
```

```
curl -L {} #
curl -v {} #
curl -X {GET|POST|DELETE|PUT} {}
curl -H 'key:value' {} #header
EX:curl -H 'Accept-Language: en-US' -H 'Secret-Message: xyzzy' http://www.example.com/test

curl -X POST -d '{}' |
EX:curl -d '{"user":"zhangsan", "password":"123456"}' -H 'Content-Type:application/json'
http://www.example.com/login

curl -F 'name1=1;name2=2' {} #
```

???? -F ????? HTTP ???? Content-Type: multipart/form-data ?????MIME???  
application/octet-stream

**????? MIME ??????MIME??? "image/png"**

```
curl -F 'file=@photo.png;type=image/png' https://google.com/profile
???????????????????? "photo.png"???????????? "me.png"
curl -F 'file=@photo.png;filename=me.png' https://google.com/profile
```

```
curl -A 'user-agent' { } #Agent
curl -b '{ }' #cookie
curl -u 'user[:password]' { } #
```

```
curl {url} --verbose --tlsv1.2 --tls-max 1.2
```

## Proxy

wget ~/.wgetrc

```
use_proxy=yes
http_proxy=http://proxy.yoyodyne.com:18023/
```

```
wget -e use_proxy=yes -e http_proxy=http://proxy.yoyodyne.com:18023/ http://www.example.com/
```

```
use_proxy=on
http_proxy=http://username:password@proxy.server.address:port/
https_proxy=http://username:password@proxy.server.address:port/
ftp_proxy=http://username:password@proxy.server.address:port/
```

## Flush BOTH DNS caches

You can do it altogether, just copy and paste on a terminal:

```
sudo systemd-resolve --flush-caches
sudo nscd -i hosts
```

## letsencrypt

letsencrypt

```
certbot certonly --webroot --webroot-path=/var/www/html -d www.domain.com
```

## rdp linux

client xrdp

```
sudo apt install xrdp
```

## firewall 3389

```
/etc/xrdp/startwm.sh Xsession
```

```
unset DBUS_SESSION_BUS_ADDRESS
```

```
unset XDG_RUNTIME_DIR
```

```
root
```

```
02:52:37.658 Main Warn could not connect to display :11.0
```

```
02:52:37.658 Main Info Could not load the Qt platform plugin "xcb" in "" even though it was found.
```

```
sudo cp ~/.Xauthority /root
```

## rdp

```
Remmina->Preferences->RDP->Use client keyboard mapping
```

## xrdp firefox

```
firefox snap bug
```

1. `firefox` apt (
2. `firefox`
- 3.

## disk GUI

```
sudo apt install gnome-disk-utility
```

```
badblocks -wsv ( )
```

```
-w -n -s -v -p -o
```

```
mkfs.fat -l ( ) --
```

**-c**

fsck.fat

fsck

0

1

2

4

8

16

32

128 shared library

**-A -R -t -f -a**

**resize**

1.

2.

1. **ext4**

```
growpart /dev/sda 2
```

```
#2 partition 2
```

```
resize2fs /dev/sda2
```

**growpart cloud-guest-utils resize e2fsck**

```
find ( ) -name ( ) ( -iname )
```

```
find ./ -name \*.c -exec grep -wnHA5 main {} \;
```

**-type d: f: l:**

**-perm** **ex:find . -type f -perm 0666 -perm !**

**-exec** **ex:find . -type f -name " " -exec rm -f {} \;** **find / -type d -perm 777 -print {} \;**

**-empty**

**-user-group** ( )

**-nouser-group** ( )

**-path**

**-maxdepth**

**-mindepth**

**-lnks**

**-newer file** file

[illegible]



Y (1970...)

```
%b      512( )
%c      C`ctime'
%k      k%A
%d      0
%f      ( )
%F      -fstype
%g      D
%G      D
%h      ( )
%H
%i      i(16 )
%k      1kB( )
%l      ( )
%m      (8 )
%n
%p
%P
%s
%t      C`ctime'
%Tk      k%A
%u      D
%U      D
`%'`%'( )
-mtime(-mmin)  ex:find /home -type f -mtime 7 ( atime aminctime ci
mtime +7 -mtime -14714
-size  ex:find /home -type f -size 50M(+-- ) ex:find /home -type f -size +50M -200M
find " " -name "*.php" -exec grep -H " " {} \;
find . -type f -mtime -3 | grep -v "( )" | grep -v " " 3
! expr false ture -not
-a(-and)
expr1 -o(-or) expr2 12
```

grep

ls ( )|grep

-i

-n

-V

-r

-e PATTERN, -regexp=PATTERN

-E, -extended-regex

```
--include 00000
-A(B;C) 00000;00000 ex:grep -A 2 tt test.txt 00000200
00000000
```

```
df -h | grep -vE '^Filesystem|tmpfs' | awk '{ print $1 " " $5 }' 0000Filesystem0tmpfs000000001,500
```

```
grep -n $query $file | awk -F:' ' '{print $1}' 0000000000
```

'PATTERN' or "PATTERN"

```
grep -e 'root' /etc/passwd
root:x:0:0:root:/root:/bin/bash
( PATTERN1 | PATTERN 2 )
grep -e '(root|www)' /etc/passwd
[Char] , [^Char]
grep -e b[^e]n /etc/group
```

```
[:alnum:] # 0000000,000 [A-Z,a-z,0-9]
[:alpha:] # 0000,000 [A-Z,a-z]
[:lower:] # 0000,000 [A-Z]
[:upper:] # 0000,000 [a-z]
[:digit:] # 0000,000 [0-9]
[:xdigit:] # 000000 0-9,A(10),B(11),C(12),D(13),E(14),F(15),000 [0-9a-fA-F]
[:space:] # 000000 ,00 000, Tab, CR (00)
[:graph:] # 00000 (0000 Tab 00) 00000.
[:print:] # 0000 ,0000.
[:cntrl:] # 0000.
[:punct:] # 0000.
^ , $
grep -e "^ben" /etc/passwd root@ubuntu:~# grep -e "bash$" /etc/passwd
```

\{\}

```
x\{m\} 000000 x 00 m . '0\{5\}' 0000000 0 000000 5 0.
x\{m,\} 000000 x 00 m . '0\{5\}' 0000000 0 0000000 5 0(00000).
x\{m,n\} 000000 x 00 m . '0\{1,5\}' 0000000 0 000000 1(00) 5(00)0.
[a-z]\{m,n\} 000000 a-z 00 m . 'a-z\{1,5\}' 0000000 0 000000 1(00) 5(00)0.
```

```
00000000 00 #ifconfig 00000 IP , Bcast 0 Mask 0000.
```

```
ifconfig ens33 | grep -e "[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}\.[0-9]\{1,3\}"  
inet addr:172.16.15.130 Bcast:172.16.15.255 Mask:255.255.255.0
```

the package lists or status file could not be parsed or opened

```
sudo rm /var/lib/apt/lists/* -vf  
sudo apt-get clean  
sudo apt-get update
```

kazam

```
sudo apt install kazam  
sudo apt install openshot ( )
```

cron (CRON) info (No MTA installed, discarding output)

```
cron cron MTA EX:postfix  
> /dev/null 2>&1 > 2>&1
```

cron


```
EX:0 * * * * root cmd.sh root cmd.sh  
"*"  
", "2 ex 0,30 or 0,15,30,45  
"- " ex 0-15 or 22-30  
"/n" n  
0 ,6  
@reboot  
@yearly @annually  
@monthly  
@weekly  
@daily @midnight  
@hourly
```

last

```
last ---  
last (name) ---  
  
-n ---  
-F  
-ad ---  
-s : XXXX-XX-XX XX:XX:XX or -Xdays  
-t
```

# mutt

```
apt install mutt
```

```
vim ~/.muttrc 
```

```
set from = "XXXX@yourdomain.com"
set realname = "XXX"
set smtp_url = "smtps://XXXX@yourdomain.com@mail.yourdomain.com:465/"
set smtp_pass = "PASSWORD"
set header_cache = "~/.mutt/cache/headers"
set message_cachedir = "~/.mutt/cache/bodies"
set certificate_file = "~/.mutt/certificates"
```

## ssmtp

```
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=SENDMAIL

# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=mail Domain

# Where will the mail seem to come from?
#rewriteDomain=

# The full hostname
hostname=HOSTNAME

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
FromLineOverride=YES

AuthUser=USER ACCOUNT
```

```
AuthPass=PASSWORD
```

```
#UseSTARTTLS=YES
```

```
UseTLS=YES
```

```
[[
```

```
echo "body" | mutt -s "title" ( ) -b ( ) -c ( ) -a ( )
```

```
mutt -s "title" ( ) -b ( ) -c ( ) < .txt -a ( )
```

## ssh( )

( )mutt( )mail( )ssmtps( )

( )/etc/bash.bashrc( )~/.bashrc

```
echo 'ALERT - Root Shell Access (PVE) on:' `date` `who` | mail -s "Alert: Root Access from `who | cut -d'(' -f2 | cut -d')' -f1`" ( )
```

```
[[[
```

md5sum( )sha256sum

md5sum ( )

md5sum -c ( )

```
[[[[[[[[
```

( )Bash( )\$'` ( )\

## date( )

```
date --date="now" ( )date( )
```

```
date --date="yesterday" ( )date --date="1 days ago")
```

```
date --date="3 days ago" ( )
```

```
date --date="tomorrow" ( )date --date="1 days" ( )
```

```
date --date="3 days" ( )
```

```
date +%Y %_m-%_d, %H:%_M" --date="3 days" ( )" ( )" + " " ( )
```

```
[[[
```

( )\$(date +%Y\_%m\_%d" --date="now")( )

"\_" ( )0( )

"-" ( )0( )

"^" ( )

%a locale's abbreviated weekday name (e.g., Sun)

%A locale's full weekday name (e.g., Sunday)  
 %b locale's abbreviated month name (e.g., Jan)  
 %B locale's full month name (e.g., January)  
 %c locale's date and time (e.g., Thu Mar 3 23:05:25 2005)  
 %C century; like %Y, except omit last two digits (e.g., 21)  
 %d day of month (e.g, 01)  
 %D date; same as %m/%d/%y  
 %e day of month, space padded; same as %\_d  
 %F full date; same as %Y-%m-%d  
 %g last two digits of year of ISO week number (see %G)  
 %G year of ISO week number (see %V); normally useful only with %V  
 %h same as %b  
 %H hour (00..23)  
 %I hour (01..12)  
 %j day of year (001..366)  
 %k hour ( 0..23)  
 %l hour ( 1..12)  
 %m month (01..12)  
 %M minute (00..59)  
 %n a newline  
 %N nanoseconds (000000000..999999999)  
 %p locale's equivalent of either AM or PM; blank if not known  
 %P like %p, but lower case  
 %r locale's 12-hour clock time (e.g., 11:11:04 PM)  
 %R 24-hour hour and minute; same as %H:%M  
 %s seconds since 1970-01-01 00:00:00 UTC  
 %S second (00..60)  
 %t a tab  
 %T time; same as %H:%M:%S  
 %u day of week (1..7); 1 is Monday  
 %U week number of year, with Sunday as first day of week (00..53)  
 %V ISO week number, with Monday as first day of week (01..53)  
 %w day of week (0..6); 0 is Sunday  
 %W week number of year, with Monday as first day of week (00..53)  
 %x locale's date representation (e.g., 12/31/99)  
 %X locale's time representation (e.g., 23:13:48)  
 %y last two digits of year (00..99)  
 %Y year  
 %z +hhmm numeric timezone (e.g., -0400)  
 %:z +hh:mm numeric timezone (e.g., -04:00)  
 %::z +hh:mm:ss numeric time zone (e.g., -04:00:00)  
 %:::z numeric time zone with : to necessary precision (e.g., -04, +05:30)

date -r [ ] [ ]

ex date -r test.txt +%F date -r test.txt +%Y-%m-%d %H:%M:%S'

read [ ]

```
read -s -n1 -p "[ ] ... "  
read -p "[ ]" [ ] [ ]p[ ]  
read -t N [ ] [ ]-t[ ]
```

[ ]

```
cp ([ ]) [ ]  
[ ]  
-r [ ]  
-p [ ]  
-a [ ]
```

```
rsync ([ ]) [ ]  
[ ]  
-a [ ]  
-v [ ]  
-h [ ]  
--bwlimit [ ]1024[ ]1MB
```

echo [ ] tee [ ]

```
echo "Hello World!" | sudo tee -a abc.txt backup.txt
```

[ ]

```
printf "The path of shell interpreter is %s\n" $SHELL >> abc.txt
```

sed [ ]

```
[ ]sed [-nefri] 'command' [ ]  
-n [ ]  
-e [ ]  
-f [ ]sed[ ]  
-r [ ]  
-i [ ]  
command[ ]  
a [ ]sed '5a [ ]' [ ] [ ]6[ ]  
c [ ] sed '2c [ ]' [ ] [ ]2[ ]  
d [ ] sed '3d' [ ] [ ]  
i [ ]  
p [ ]  
s [ ] sed 's/[ ]/[ ]/g' [ ]
```

```
[ ]  
$[ ]  
[ ] 3,5d [ ]3-5[ ]
```

sed -i 's/abc/\ /g' filename.txt

sed 's/^./\ /'

timeout 10 tail -f /var/log/httpd/access.log

## rsync

rsync ( ) / /

-v

-r

-a -rlptgoD

-z

-h

-i

-l

-p

-t

-g

-o

-D

-q -quiet

-c -checksum checksum mod-time size

-R -relative

-u -update

-d -dirs

-l -links

-L -copy-links

-copy-unsafe-links

-safe-links

-k -copy-dirlinks

-K -keep-dirlinks

-H -hard-links

-A -acls ACL -perms

-t -times

--bwlimit

--delete

--exclude

--include

--min-size

--max-size

--remove-source-files

--existing



```

--whole-file [FILE]
--progress [FILE]
-del -delete-during[FILE]
-delete-before [FILE]
-delete-during [FILE]
-delete-delay [FILE]
-delete-after [FILE]
-delete-excluded [FILE]
-ignore-errors [FILE] / O[FILE]
-force [FILE]
-max-delete = NUM[FILE]
-partial [FILE]
-partial-dir = DIR[FILE]
-delay-updates [FILE]
-m -prune-empty-dirs[FILE]
-numeric-ids [FILE]uid / gid[FILE]
-timeout = SECONDS[FILE] / O[FILE]s[FILE]
-contimeout = SECONDS[FILE]
-l -ignore-times[FILE]mtime[FILE]size[FILE]
-size-only [FILE]
-modify-window = NUM[FILE]
-T -temp-dir = DIR[FILE]
-y -fuzzy[FILE]
-compare-dest = DIR[FILE]
-copy-dest = DIR[FILE]-compare-dest[FILE]
-link-dest = DIR[FILE]-compare-dest[FILE]
-compress-level = NUM[FILE]
-skip-compress = LIST[FILE]
-C -cvs-exclude[FILE]CSV[FILE]
-f -filter = RULE[FILE]
-F[FILE]-filter ='dir-merge /.rsync-filter'[FILE]-filter =' - .rsync-filter'
-exclude = PATTERN[FILE]PATTERN
-exclude-from = FILE[FILE]
-include = PATTERN[FILE]
-include-from = FILE[FILE]
-files-from = FILE[FILE]
-O[FILE]-fromO all
-from / filter[FILE]O[FILE]
-s -protect-args[FILE]; [FILE]
-address = ADDRESS[FILE]
-port = PORT[FILE]
-sockopts = OPTIONS[FILE]TCP[FILE]
-blocking-io[FILE]shell[FILE] / O.
-stats[FILE]
-8[FILE]-8-bit-output[FILE]
-h[FILE]-human-readable[FILE]

```

-progress[ ]  
 -P[ ]partial-progress[ ]  
 -i -itemize-changes[ ]  
 -out-format = FORMAT[ ]  
 -log-file = FILE[ ]  
 -log-file-format = FMT[ ]  
 -password-file = FILE[ ]  
 -list-only[ ]  
 -write-batch = FILE[ ]  
 -only-write-batch = FILE[ ]write-batch[ ]  
 -read-batch = FILE[ ]  
 -protocol = NUM[ ]  
 -iconv = CONVERT\_SPEC[ ]  
 -4[ ]-ipv4[ ]Pv4  
 -6[ ]-ipv6[ ]Pv6  
 -version[ ]  
 [ ]-h[ ]-help[ ]-h[ ]-help[ ]  
 [ ]  
 [ ]rsync[ ] rsync -avh /home/ron user@ipaddr:[ ]  
 [ ]rsync -avzh -e "ssh -p [ ]" [ ]/[ ] [ ]/[ ]  
 [ ]rsync server[ ]" "[ ]EX:rsync -avh /home/ron user@ipaddr::[ ]

## SCP[ ]

```
scp -r [ ] username@hostname:[ ]
```

## ssh [ ]

ssh-keygen [ ]

[ ]

-t [ ] EX:rsa

-b [ ] [ ]3072

[ ]

[ ]1:

```
ssh-copy-id -i key_path username@ipaddr
```

[ ]2:

[ ]server[ ]~/.ssh/authorized\_keys [ ]

[ ]/etc/ssh/sshd\_config

PasswordAuthentication no

## sshd\_config

```
port 22 # ssh
listenaddress 0.0.0.0 # ip
hostkey /etc/ssh/ssh_host_key #
authorizedkeysfile #
serverkeybits 1024 #
loggingracetime 600 #
keyregenerationinterval 3600 #
compression no # delay
MaxAuthTries 3 #
PermitRootLogin no # root ssh
Printlastlogin yes #
clientaliveinterval 900 #
clientalivecountmax 0 #
allowusers username #
denyusers username #
allowgroups username #
denygroups username #
permitemptypassword no #
ignorerhosts yes # Rhost
ignoreuserknownhosts yes # known_hosts
hostbasedauthentication no #
x11forwarding no # gui ssh
strictmodes yes #
pubkeyauthentication yes #
GSSAPIauthentication no # GSSAPI
kerberosauthentication no # kerberos
Ciphers aes128-ctr,aes192-ctr,aes256-ctr #
MACs hmac-sha2-256,hmac-sha2-512 #
Banner #
passwordauthentication yes #
```

**authorized\_keys 600 ~/.ssh 700**

## ssh

sshpass

sshpass -p 'password' ssh username@ipaddr 'cmd'

export SSHPASS='password' # /etc/profile \$HOME/.profile

ssh -e ssh username@ipaddr 'cmd'

rename 's/ / /'

EX:rename 's/DSC\_/4inlibra-Ron-[REDACTED]-/' \*.JPG

## nfs[REDACTED]

nfs server[REDACTED]

[REDACTED]

```
sudo apt install nfs-kernel-server
```

[REDACTED]

```
sudo netstat -tulnp | grep rpcbind #[REDACTED]111udp[tcp]
udo netstat -tuln | grep :2049 #[REDACTED]2049[tcp][udp]
```

```
vi /etc/exports #[REDACTED]client[REDACTED]
```

[REDACTED] [REDACTED]NFS[REDACTED]([REDACTED])

ex:/home/magiclen/shared-files 192.168.56.104(rw,sync,no\_subtree\_check)

192.168.100.0/24(ro,sync,no\_subtree\_check) \*.4inlibra.com(ro,sync,no\_subtree\_check)

[REDACTED]

- rw[REDACTED]
- ro[REDACTED]
- root\_squash[NFS[REDACTED]root[REDACTED](UID[REDACTED])root[REDACTED](GID=0)[REDACTED]NFS[REDACTED]UID[REDACTED]GID[REDACTED]65534[REDACTED]
- no\_root\_squash[NFS[REDACTED]root[REDACTED]root[REDACTED]NFS[REDACTED]
- all\_squash[FS[REDACTED]root[REDACTED]root[REDACTED]NFS[REDACTED]UID[REDACTED]GID[REDACTED]65534[REDACTED]
- no\_all\_squash[NFS[REDACTED]root[REDACTED]root[REDACTED]NFS[REDACTED]
- no\_subtree\_check[REDACTED](subtree\_check)[REDACTED]subtree\_check[REDACTED]
- nohide[REDACTED]exported

tree[REDACTED]([REDACTED]/exports/home[REDACTED])[REDACTED]([REDACTED]/exports)[REDACTED]([REDACTED]/exports/home)[REDACTED]

- crossmnt[REDACTED]exported tree[REDACTED]nohide[REDACTED]
- fsid[REDACTED]NFS[REDACTED]ID[REDACTED]ID[REDACTED]0root[REDACTED]ID[REDACTED]fsid=0[REDACTED]fsid=root[REDACTED]

[REDACTED]

```
sudo exportfs -r
```

[REDACTED]-a [REDACTED] -r [REDACTED]-ua[REDACTED]

[REDACTED]

```
sudo exportfs
```

[REDACTED]

```
showmount
```

###

- --exports[ ]
- --directories[ ]
- --all[ ]

nfs client[ ]

###

```
sudo apt install nfs-common
```

###

```
sudo mount -t nfs 192.168.56.103:/home/ron/shared-files /mnt/nfs
```

```
192.168.72.136:/mnt/sharedfolder /mnt/client_sharedfolder nfs defaults 0 0 #fstab
```

- rw[ ]
- ro[ ]
- rsize[ ]NFS[ ]NFS[ ]4096[ ]
- wsize[ ]NFS[ ]NFS[ ]4096[ ]
- udp[ ]UDP[ ]TCP[ ]NFS[ ]3[ ]UDP[ ]
- tcp[ ]UDP[ ]TCP[ ]NFS[ ]3[ ]TCP[ ]
- nfsvers[ ]NFS[ ]
- retry[ ]10000[ ]
- timeo[ ]0.1[ ]0.7[ ]
- hard[ ]RPC[ ]
- soft[ ]RPC[ ]
- fg[ ]
- bg[ ]
- intr[ ]
- nointr[ ]

NFS[ ]:[ ] nfs \_netdev 0 0

###linux[ ]ntfs[ ]

```
apt install ntfs-3g
```

###swap

```
swap -s # free -h
```

###swap

```
sudo mkswap /dev/sda # [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] UUID[ ] [ ] [ ] [ ] -U [ ] [ ] [ ] [ ]  
sudo swapon /dev/sda # [ ] [ ] [ ] [ ] swap[ ] SSD[ ] TRIM[ ] [ ] -d --discard[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]
```

```
fstab swap discard defaults
```

□□□□swap

```
sudo fallocafe -l ( ) /
```

```
sudo dd if=/dev/zero of=() bs=1M count=512 #bs*count
```

```
sudo chmod 600 /[ ]/[ ]/[ ]/[ ]/[ ]/[ ] # [ ][ ][ ]
```

```
sudo mkswap /[ ]/[ ]/[ ]/[ ]/[ ]/[ ]#[ ][ ]swap
```

```
sudo swapon /( [ ][ ][ ] ) #[ ][ ]swap
```

```

fstab swap

```

```
swapoff /[ ]/[ ]/[ ]/[ ]/[ ]/[ ]swap
```

swap

[illegible]

```
vi /etc/sysctl.conf
```

```
vm.swappiness = 10 # 0[ ]100[ ]60[ ]
```

```
dpkg --get-selections | grep mongodb #  
sudo apt-mark hold mongodb mongodb-dev #  
sudo apt-mark auto mongodb-dev #  
sudo apt-mark unhold mongodb-dev #  
sudo apt-mark manual mongodb-dev #  
sudo apt-mark showhold #  
sudo apt-mark showmanual #  
sudo apt-mark showauto | more #
```

deb

```
sudo apt install --fix-missing
```

■■■■■■■

## Ubuntu Kernel■■■■

<http://kernel.ubuntu.com/~kernel-ppa/mainline/> #kernel■■■■

#■■ -lowlatency ■■■■■ kernel■■■ lowlatency ■■■■ -generic■ -rt ■ -preempt ■■■■■■■■ -generic ■■■

```
grep "menuentry 'Ubuntu' /boot/grub/grub.cfg #■■■■■kernel■■
```

```
sudo nano /etc/default/grub #■■■■GRUB_DEFAULT=0 ■■■■
```

```
sudo update-grub #■■■■
```

## umount■■device is busy■■■■■

```
fuser -cuk (■■■■) #■■■■■■■■■■
```

■■■

-c

Same as -m option, used for POSIX compatibility.

-u, --user

Append the user name of the process owner to each PID.

-k, --kill

Kill processes accessing the file. Unless changed with -SIGNAL, SIGKILL is sent. An fuser process never kills itself, but may kill other fuser processes. The effective user ID of the process executing fuser is set to its real user ID before attempting to kill.kill.

## diff■■■■

```
diff -bur ■■1 ■■2
```

■■■b■■■■■■■

u■ unified ■■■■■■■■

r■■■■■

q■■■■■■■

## case ... esac

```
#!/bin/bash
```

```
m='MacDonald'case $m in
```

```
MacDonald*) echo "Ronald McDonald"
```

```
;;
```

```
KFC*) echo "Harland David Sanders"
```

```
;;
```

```

TKK*)    echo "[]"

;;

*)    echo "You are really health!"
esac

```

## for

```

#!/bin/bash
for loop in 1 2 3; do
    echo "number: $loop"
done

```

```

#!/bin/bash
echo -n "[]:"
read F
for ((i=1 ; i<=F ; i++))dotouch $i.js
echo $idone
echo "[] $F []"

```

## while

```

#!/bin/bash
echo -n "[]:"
read FINDEX=1
# []le[]
while [ $INDEX -le $F ]
do
    # []
    touch $INDEX.js
    echo -n "$INDEX"

    # INDEX []1
    (( INDEX++ ))
done
echo ""echo "[] $F []"

```

```

#!/bin/bash
echo "[] Ctrl + C []..."LENGTH=0
while :
do
    echo -ne "\r["

```



```
sleep 0.2
while [ $LENGTH -le 10 ]
do
    sleep 0.1
    echo -n ">"
    (( LENGTH++ ))
done
LENGTH=0
echo -en "\r          "
```

--	--	--	--	--	--	--

```
#!/bin/bash
echo -n "[XXXXXXXXXX]"
read FNINDEX=1
while read line
do
    echo "[XXXXX]: $line"
    (( INDEX++ ))
done <$FN
```

```
#!/bin/bash  
echo -n "■■■■■:"  
  
read F  
  
counter=0  
  
until [ $counter = $F ]; do  
    ((counter++)) #■■■■■  
    echo $counter  
    touch $counter.js  
  
done  
  
echo "■■■ $F ■■■"
```

dd[ ][ ][ ][ ]

```
sudo dd if=/dev/sda of=/dev/sdb
```

```
conv[0][0]sync[0]/O sync[0]noerror[0][0]  
[0]*.img[0].iso[0][0][0][0][0][0]
```

# e2fsck

```
e2fsck -a -y /dev/sda1
```

- a: partition
- b: superblock
- B size: size
- c: partition
- C file: file
- d: e2fsck debug
- f: e2fsck -f
- F: buffer cache
- l list: list
- d : e2fsck debug
- f :
- n: (read-only)
- p: -a
- v:
- y:

```
sudo apt install gthumb
```