

Linux 窗口

窗口列表

Linux 窗口

- [Linux 窗口](#)
- [Linux 窗口](#)
- [Linux 窗口](#)
- [Cron 窗口](#)
- [SystemD](#)
- [Linux 窗口](#)
- [xargs 窗口](#)
- [Arch Linux](#)
- [Fedora](#)

Linux 00



1 /etc/os-release

/etc/os-release

```
$ cat /etc/os-release
```

SystemD

RedHat /etc/redhat-release

2 /etc/issue

/etc/issue

```
$ cat /etc/issue
```

3 hostnamectl

hostnamectl

```
$ hostnamectl
```

4 uname

uname Linux

```
#
```

```
$ uname -a
```

```
#
```

```
$ uname -o
```

```
#
```

```
$ uname -r
```

5 lsb-release

lsb-release

lsb-release

```
$ lsb-release
```

6 /proc/version

/proc/version

```
$ cat /proc/version
```

Linux □□□□□

□□□□□□□□□□

□□□□

```
$ useradd -m [username]
$ passwd [username]
```

□□□□□□□□□□

```
$ usermod -aG wheel,audio,video,storage [username]
```

□□□□□□□□ sudo □□□□□□□□ visudo□□□ sudo □□□□□ /etc/sudoers □□□□
wheel □□□□□□□□

```
$ visudo
```

□□□□

/etc/group □□□□□□□□□□

groups □□□□□□□□□□

```
$ groups [username]
```

□□□□ username □□□□□□□□□□□□□□

id □□□□□□□□□□□□□□□□□□□□□□

```
$ id [username]
```

□□□□□□□□□□□□□□

```
$ cat /etc/group
```

groupadd 

```
$ sudo groupadd [groupName]
```

`gpas swd`

```
$ sudo gpasswd -a [userName] [groupName]
```

usermod

```
$ sudo usermod -aG [groupName] [username]
```

- a

gpasswd

```
$ sudo gpasswd -d [userName] [groupName]
```

groupmod

```
$ sudo groupmod -n [new_group] [old_group]
```

groupdel

```
$ sudo groupdel [groupName]
```

Linux 目录结构

Linux 目录结构

Linux 根目录 /

/root

/home

/home

ruanyf

/home/ruanyf

~ \$HOME

/bin sbin

/bin

ls cp mv mkdir

/sbin

fdisk

/usr

/usr/bin

/usr/lib

/etc

/opt

MS Edge

/opt/microsoft

/lib

/boot

RAM

/media

USB

/media

/mnt

/media

/proc

/run

ID

/srv

Web FTP

FTP

Cron



Cron

cron crond

cron

```
# Ubuntu / Debian
$ sudo systemctl status cron
# or
$ ps aux | grep cron

# rockyOS / RHEL
$ sudo systemctl status crond
# or
$ ps aux | grep crond
```

crontab

crond crontab

crontab crontab crontab

crontab /etc/crontab /etc/cron.d /etc/cron.daily /etc/cron.hourly /etc/cron.weekly /etc/cron.monthly

crontab cron

```
# 普通ユーザ cron 編集
$ crontab -l

# root ユーザ cron 編集
$ sudo crontab -l

# 特定のユーザ cron 編集
$ sudo crontab -u [USERNAME] -l
```

cron 設定

crontab 編集 cron 実行

cron 設定

minute hour day_of_month month day_of_week command_to_execute

設定

- minute 0-59 0 59 0
- hour 0-23 24 14 2
- day_of_month 1-31 1 31
- month 1-12 1 12
- day_of_week 0-6 0 6
- command_to_execute cron

crontab 設定

- () " " "" " "
- (.) day_of_week "1,3,5" "
- (-) "9-17" " 9 5 "
- (/) "*/10" " 10 "

crontab 設定

- @reboot 1

- @year @annually 1 1 12:00 "0 0 1 1 *"
- @monthly 1 12:00 "0 0 1 * *"
- @weekly 12:00 "0 0 * * 0"
- @daily @midnight 12:00 cronjob crontab "0 * * * *"
- @hourly "0 * * * *"

crontab "#"

- * * * * * cron
- 0 * * * * cron
- 0 0 * * * cron
- 0 2 * * * 2 cron
- 0 0 15 * * 15 cron
- 0 0 0 12 * cron
- 0 0 * * 6 3 cron
- 0 15 * * 1-5 3 cron
- */5 * * * * 5 cron
- 0 8-16 * * * 8 4 cron
- 0 4 * * 2,4 4 cron
- @reboot crond cron

L Last " " " "

```
0 4 L * * command - 4
0 4 * * 5L command - 4
```

w weekday " "

```
0 4 15W * * command - 15 4
```

" "

```
0 4 * * 5#2 - 4
```

cron Shell

crontab [] [] [] [] [] []

cron

PATH

```
PATH=/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin
```

[illegible]

PATH SHELL cron Shell

```
SHELL=/bin/sh
```

☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ cron ☐ ☐ ☐ ☐ ☐ Bash ☐

```
result="HELLO WORLD"
```


cron

--	--	--	--

--	--	--	--

[illegible]

```
# 00000000 cron 00
```

```
$ crontab -e
```

```
# 000000 cron 00
```

```
$ sudo crontab -e
```

```
# 00000000 cron 00
```

```
$ sudo crontab -u [USERNAME] -e
```

crontab [] [] [] [] [] []

[illegible]

/etc/cronjob

```
/etc/cron.d/
```

/etc/cronjob

crontab

```
* * * * * user command
```

```
#####
```

```
/etc/cron.*
```

```
#####
```

```
#####
```

- /etc/cron.hourly/
- /etc/cron.daily/
- /etc/cron.weekly/
- /etc/cron.monthly/

```
## cron #####
```

```
crontab #####
```

```
backup.sh ### 3:00
```

```
#####
```

```
0 3 * * * /home/linuxiac/backup.sh
```

```
#####
```

```
cron #####
```

```
crontab #####
```

```
cron
```

```
##
```

```
crontab -e #####
```

```
cron ##
```

```
### cron #####
```

```
#####
```

```
cron #####
```

```
crontab #####
```

```
-r
```

```
# ##### cron #####
```

```
$ crontab -r
```

```
#####
```

```
crontab #####
```

```
cron
```

```
#####
```

```
#####
```

```
-i
```

```
# ##### cron #####
```

```
$ crontab -i -r
```

```
#####
```

```
cron ##
```

```
# ##### cron ##
```

```
$ sudo crontab -r
```

```
# ##### cron ##
```

```
$ sudo crontab -u USERNAME -r
```



```
/etc/crontab
```

/var/log/myservice

[illegible]

--	--	--	--

[illegible]

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

cron

- cron

--	--	--	--	--

cron ☐☐☐☐☐

```
stdout [ ]
```

```
/var/log/syslog
```

RHEL/RHEL □

```
/var/log/cron
```

```
grep [ ] [ ] [ ] [ ] [ ] [ ]
```

- [How to Use Cron on Linux: Tips, Tricks, and Examples](#)
- [Cron Jobs on Linux - Comprehensive Guide with Examples](#)

SystemD



SystemD is Linux's init system. It manages system services using systemctl. You can start, stop, and enable services. For example, to start the SSH service, you would use systemctl start sshd. To enable it to start at boot, use systemctl enable sshd. To check the status of the service, use systemctl status sshd. To view the service's configuration file, use systemctl cat sshd.service. To view the service's logs, use journalctl -u sshd.service. To view the service's dependencies, use systemctl cat sshd.service | grep Requires. To view the service's description, use systemctl cat sshd.service | grep Description. To view the service's documentation, use systemctl cat sshd.service | grep Documentation. To view the service's after dependencies, use systemctl cat sshd.service | grep After. To view the service's condition path exists dependencies, use systemctl cat sshd.service | grep ConditionPathExists. To view the service's environment file, use systemctl cat sshd.service | grep EnvironmentFile. To view the service's exec start pre command, use systemctl cat sshd.service | grep ExecStartPre. To view the service's exec start command, use systemctl cat sshd.service | grep ExecStart. To view the service's exec reload command, use systemctl cat sshd.service | grep ExecReload. To view the service's kill mode, use systemctl cat sshd.service | grep KillMode. To view the service's restart policy, use systemctl cat sshd.service | grep Restart. To view the service's restart prevent exit status, use systemctl cat sshd.service | grep RestartPreventExitStatus. To view the service's type, use systemctl cat sshd.service | grep Type. To view the service's runtime directory, use systemctl cat sshd.service | grep RuntimeDirectory. To view the service's runtime directory mode, use systemctl cat sshd.service | grep RuntimeDirectoryMode.

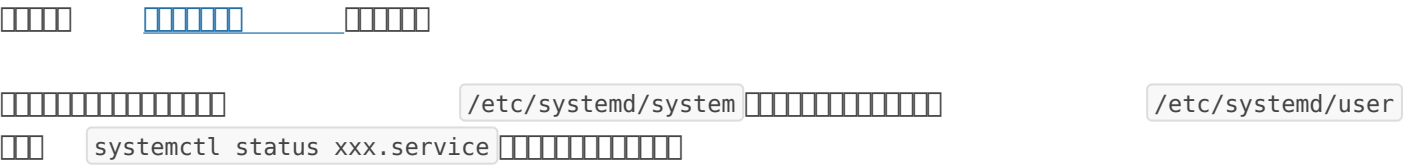


ssh.service

```
[Unit]
Description=OpenBSD Secure Shell server
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify
RuntimeDirectory=sshd
RuntimeDirectoryMode=0755
```

```
[Install]
WantedBy=multi-user.target
Alias=sshd.service
```



Linux 四四四



四四四四四四

fdisk 四四四四四

```
$ sudo fdisk -l
```

四四四四四

/dev/sdc 四

四四四四四四四

/mnt/sdd 四

```
$ sudo mkdir /mnt/sdd
```

四四四

mount 四四四四四四四四

```
$ sudo mount /dev/sdc /mnt/sdc
```

四四四四四

NTFS 四四四四

mount 四四

-t 四四四四四四四

```
# 四四
```

```
$ sudo mount -t ntfs /dev/sdc /mnt/sdc
```

```
# 四四
```

```
$ sudo mount -t ntfs-3g /dev/sdc /mnt/sdc
```

四四四

df 四四四四四四四

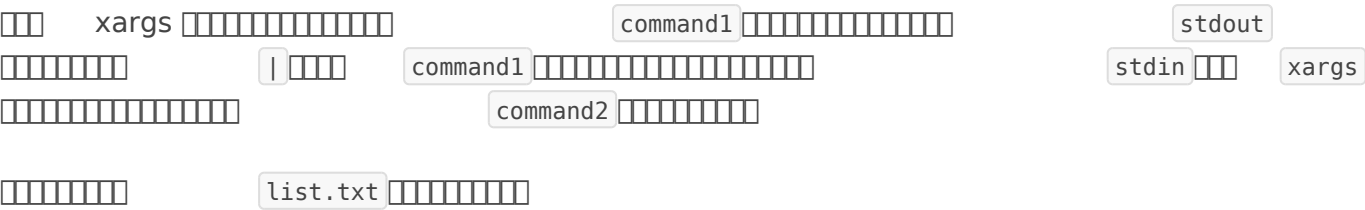
```
$ df -h
```

xargs



xargs [input] stdin[output] | [output]

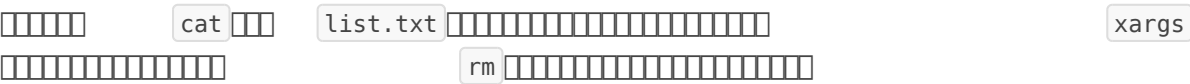
```
$ {{command1}} | xargs {{command2}}
```



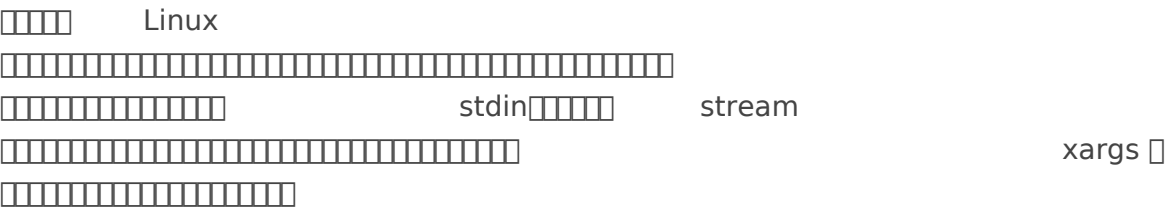
```
a.txt
b.txt
c.txt
```

[output]

```
# cat a.txt b.txt c.txt
$ cat list.txt | xargs rm
```



```
$ rm a.txt b.txt c.txt
```



xargs [input] echo [output] echo [output]

```
$ echo abc
```



				xargs					Linux				

xargs |||||

xargs ||||| xargs echo |

```
$ xargs
# 
$ xargs echo
```

| xargs |||||
Ctrl+d ||||| echo |||||

```
$ xargs
hello (Ctrl + d)
hello
```

||||| xargs ||||| hello ||||| Ctrl+d ||||| echo hello |

|||||

```
$ xargs find -name
"*.txt"
./foo.txt
./hello.txt
```

||||| xargs find -name ||||| "*.txt"
||||| TXT ||||| Ctrl+d ||||| find -name
*.txt |

-d |||||

||||| xargs |||||

```
$ echo "one two three" | xargs mkdir
```

mkdir one two three xargs one two three mkdir

-d

```
$ echo -e "a\tb\tc" | xargs -d "\t" echo
a b c
```

\t a\tb\tc echo -e

-p -t

xargs

-p

```
$ echo 'one two three' | xargs -p touch
touch one two three ?...
```

-t

```
$ echo 'one two three' | xargs -t rm
rm one two three
```

-0 find

xargs

find -print0 null xargs -0

```
$ find /path -type f -print0 | xargs -0 rm
```

/path null

xargs find rm " " " "

```
$ find . -name "*.txt" | xargs grep "abc"
```

XXXXXXXXXX

TXT XXXXXXXXXXXXXXXXXXXX

abc XX

-L XX

XXXXXXXXXX

-L XXXXXXXXXXXXXXXXXXXX

```
$ xargs find -name
"*.txt"
"*.md"
find: paths must precede expression: `*.md'
```

XXXXXXXXXX

"*.txt" XX *.md XXXXXXXXXXXX

find XXXXXXXX

XX -L XXXXXXXXXXXXXXXXXXXX

```
$ xargs -L 1 find -name
"*.txt"
./foo.txt
./hello.txt
"*.md"
./README.md
```

XXXXXXXXXX

-L 1 XXXXXXXXXXXXXXXXXXXX

find -name XX

XXXXXXXXXX

```
$ echo -e "a\nb\nc" | xargs -L 1 echo
a
b
c
```

XXXXXXXXXX

echo XXXXX

echo XXXXXXXXXXXX

-n XX

-L XXXXXXXXXXXXXXXXXXXX

```
$ xargs find -name  
"*.txt" "*.md"  
find: paths must precede expression: `*.md'
```

XXXXXXXXXXXXXXXXXXXX

-n XXXXXXXXXXXXXXX

```
$ xargs -n 1 find -name
```

XXXXXXXXXX

-n 1 XXXXXXXXXXXXXXX

find -name XX

XXXXXXX

```
$ echo {0..9} | xargs -n 2 echo  
0 1  
2 3  
4 5  
6 7  
8 9
```

XXXXXXXXXXXXXXXXXXXX

echo XXXXX

10XXXXXXXXXX

echo XXXXXXX

-l XX

XX xargs XXXXXXXXXXXXXXX

-l XX

-l XXXXXXXXXXXXXXX

```
$ cat foo.txt  
one  
two  
three  
  
$ cat foo.txt | xargs -l file sh -c 'echo file; mkdir file'  
one  
two  
three  
  
$ ls
```

one two three

```
foo.txt
-I file
mkdir file
file
file
```

```
echo
mkdir
echo file;
```

--max-procs

```
xargs
```

```
--max-procs
procs 0
```

```
--max-procs 2
```

```
--max-
```

```
$ docker ps -q | xargs -n 1 --max-procs 0 docker kill
```

```
Docker
```



- [xargs is the inverse function of echo](#)
- [Linux and Unix xargs command tutorial with examples](#), George Ornbo
- [8 Practical Examples of Linux Xargs Command for Beginners](#), Himanshu Arora

Arch Linux



Arch

Arch Manjaro EndeavourOS Garuda Linux

Pacman

Pacman Arch

-Sy

```
$ sudo pacman -Sy
```

-Syu

```
$ sudo pacman -Syu
```

-Ss

```
$ sudo pacman -Ss vlc
```

-Si

```
$ pacman -Si nginx
```

-S

```
$ sudo pacman -S vlc
```

-S

```
$ sudo pacman -S pac1 pac2
```

`-S`

`-Qi`

```
$ pacman -Qi package
```

`--ignore`

```
$ sudo pacman -Syu --ignore=vlc
```

`-Sw`

```
$ pacman -Sw vlc
```

`-Q`

```
$ pacman -Q
```

`-Qs`

```
$ pacman -Qs vlc
```

`-Ql`

```
$ pacman -Ql vlc
```

`-Qo`

```
$ pacman -Qo /usr/bin/vlc
```

`-Rs`

```
$ sudo pacman -Rs vlc
```

`-Rns`

```
$ sudo pacman -Rns vlc
```

```
$ sudo pacman -Rns $(pacman -Qdtq)
```

-U Pacman /var/cache/pacman/pkg

```
$ cd /var/cache/pacman/pkg/  
$ sudo pacman -U vlc-3.0.11-2-x86_64.pkg.tar.zst
```

-Sc /var/cache/pacman/pkg

```
$ sudo pacman -Sc
```

-Scc

```
$ sudo pacman -Scc
```

AUR

AUR Arch User Repository Arch

yay AUR

```
$ git clone https://aur.archlinux.org/yay-bin.git  
$ cd yay-bin  
$ makepkg -si
```

```
$ yay --version
```

-S

```
$ yay -S packageName
```

-Sua AUR

```
$ yay -Sua
```

-Syu

```
$ sudo yay -Syu
```

-Ss AUR

```
$ yay -Ss packageName
```

□□□□ AUR □□□ -Ss □□□□

```
$ yay packageName
```

-R

```
$ yay -R packagename
```

-Rns

-Ps AUR ~/.cache/yay/

\$ yay -Ps

-Sc [] [] [] [] [] [] [] AUR [] []

```
$ yay -Sc
```

[illegible]

```
$ yay -Scc
```

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

□□□□ Reflector □□□□□□□□□□□□□□□□

Python

```
/etc/pacman.d/mirrorlist
```

```
$ sudo pacman -S reflector rsync
```

[illegible]

```
$ sudo cp /etc/pacman.d/mirrorlist /etc/pacman.d/mirrorlist.backup
```

Reflector 10

```
/etc/pacman.d/mirrorlist
```

```
$ sudo reflector --latest 10 --sort rate --save /etc/pacman.d/mirrorlist
```

- `--latest`
- `--sort`
- `--save`

□□□□□ 10 □□□□□□□□□□□□□□□□ □□□□□□□□ □□□□□□□□

```
$ sudo reflector --latest 10 --sort rate --fastest 5 --save /etc/pacman.d/mirrorlist
```

```
--list-countries
```

```
$ reflector --list-countries
```

[illegible]

```
$ sudo reflector --country "US" --latest 10 --sort rate --save /etc/pacman.d/mirrorlist
```

```
$ sudo reflector --country "US,CA" --latest 10 --sort rate --save /etc/pacman.d/mirrorlist
# []
$ sudo reflector --country "United States,Canada" --latest 10 --sort rate --save
/etc/pacman.d/mirrorlist
```

```
--protocol [ ] https http ftp [ ]
```

```
$ sudo reflector --protocol https --latest 10 --sort rate --save /etc/pacman.d/mirrorlist
```

Reflector

SystemD

```
$ sudo systemctl enable --now reflector.timer
```

```
$ sudo systemctl list-timers
```

■■■■■ Reflector ■■■■■■■■

```
$ sudo systemctl enable --now reflector.service
```



- [Finding the Most Up-to-Date and Fastest Arch Linux Mirrors](#)

Fedora

包管理 dnf

Fedora 包管理 rpm 包管理 PackageKit 图形化
DNF 是 GNOME Software 的默认包管理器

DNF 是 Fedora 的默认包管理器

```
# 搜索包
$ sudo dnf search 包名

# 安装包
$ sudo dnf install 包名

# 卸载包
$ sudo dnf remove 包名
```

更新系统

```
$ sudo dnf upgrade --refresh -y
```

dnf 常用命令

- autoremove- 自动卸载无依赖的包
- check-update- 检查更新
- downgrade- 降级
- info- 显示包的详细信息
- reinstall- 重新安装
- upgrade- 升级
- exclude- 排除

dnf 安装语言包 Fedora 默认语言包

安装语言包

```
$ sudo dnf install langpacks-zh_CN
```