Linux Command Usage

1. apt

apt (Advanced Packaging Tool) is a shell front-end package manager in Debian and Ubuntu.

The apt command provides commands to search, install, upgrade, and remove one, multiple, or all software packages. The commands are concise and easy to remember.

The apt command execution requires superuser (root) privileges.

- ### Grammar Explanation

  ```
  apt [options] [command] [package ...]
  ```

  - **options:** Optional, options include -h (help), -y (answer "yes" to all prompts during installation), -q (do not display installation progress), and so on.
  - **command:** The operation to be performed.
  - **package:** The package name to be installed.

- ### Common Commands

  - Update the apt software source database: **sudo apt update**

  - Update installed software packages: **sudo apt upgrade**

    List upgradeable software packages and their versions: **apt list --upgradeable**

    Upgrade software packages, removing any that need to be updated: **sudo apt full-upgrade**

  - Install a specified software package: **sudo apt install** `<package_name>`

    Install multiple software packages: **sudo apt install** `<package_1> <package_2> <package_3>`

    - Display detailed information about a software package, such as its version, installation size, dependencies, etc: **sudo apt show** `<package_name>`
    - Remove a software package: **sudo apt remove** `<package_name>`
    - Clean up unused dependencies and library files: **sudo apt autoremove**
    - Remove a software package and its configuration files: **sudo apt purge** `<package_name>`
    - Search for a software package: **sudo apt search** `<package_name>`
    - List all installed packages: **apt list --installed**
    - List version information of all installed packages: **apt list --all-versions**

2. dmesg

The `dmesg` command is used to view or control the kernel ring buffer.

The kernel stores the kernel boot logs in the ring buffer. If you didn't have time to view the information during boot-up, you can use `dmesg` to view it.

- ### Grammar Explanation

  ```
  dmesg [options]

  dmesg -C / dmesg --clear
  dmesg -c / dmesg --read-clear [options]
  ```

- ### Option Explanation

  - -c, --read-clear: Display the information and then clear the contents of the ring buffer.
  - -C, --clear: Clear the contents of the ring buffer.

- ### Common Commands

  - Display all kernel log content in the ring buffer

    ```
    dmesg
    ```

  - Save the kernel log to a file

    ```
    dmesg > kernel.log
    ```

  - Clear the cached logs; useful for reducing log content when debugging drivers

    ```
    dmesg -C
    ```

3. dpkg-deb

The **dpkg-deb command** is a software package management tool for Debian Linux. It can perform packaging and unpackaging operations on software packages and provide package information.

Grammar Explanation

```
dpkg-deb [<option> ...] <command>
```

Command Description

The dpkg-deb command not only has options to set, but also requires a command to execute different functions.

- -b: Create a Debian software package.
- -c: Display the file list in the package.
- -e: Extract the control information.
- -f: Print field contents to standard output.
- -x: Extract files from the package to the specified directory.
- -X: Extract files from the package to the specified directory and display the detailed process of extraction.
- -w: Display package information.
- -I: Display detailed package information.
- -R: Extract control information and archive manifest file.

Option Description

- `-v, --verbose`: Enable verbose output.
- `-D, --debug`: Enable debug output.
    - `--showformat=<format>`: Use alternative format for `--show`.
    - `--deb-format=<format>`: Choose archive format. Allowed values are 0.939000, 2.0 (default).
    - `--nocheck`: Disable control file checking (build bad packages).
    - `--root-owner-group`: Force file owner and group to be root.
    - `--[no-]uniform-compression`: Use compression parameters on all members. If specified, uniform compression parameters will be used.
- `-z#`: Set compression level during build.
- `-Z<type>`: Set compression type to be used during build. Allowed types are gzip, xz, zstd, none.
- `-S<strategy>`: Set compression strategy during build. Allowed values are none, extreme (xz), filtered, huffman, rle, fixed (gzip).

Common Commands

- Extract program files

```
dpkg-deb -x hobot-configs_2.2.0-20231030133209_arm64.deb
```

- Extract control files

```
dpkg-deb -e hobot-configs_2.2.0-20231030133209_arm64.deb hobot-configs/DEBIAN
```

- Query the content of the deb package

  ```
  dpkg-deb -c hobot-configs_2.2.0-20231030133209_arm64.deb
  ```

4. dpkg

Install, create, and manage software packages on Debian Linux systems.

The **dpkg command** is a utility tool used on Debian Linux systems to install, create, and manage software packages.

- ## Grammar Explanation

  ```
  dpkg [<option> ...] <command>
  ```

- ## Command Description

  The dpkg command has options for setting and requires a command to execute different functionalities.

  - -i: Install a software package.
  - -r: Remove a software package.
  - -P: Remove a software package and its configuration files.
  - -L: List files belonging to a specified software package.
  - -l: List software package status concisely.
  - -S: Search for software packages containing specified files.
  - --unpack: Unpack a software package.
  - -c: Show the list of files inside a software package.
  - --configure: Configure a software package.

- ## Option Description

  - --admindir= directory: Use directory instead of /var/lib/dpkg.
  - --root= directory: Install to a different root directory.
  - --instdir=directory: Change the installation directory while maintaining the management directory.
  - --path-exclude= expression: Do not install paths that match the shell expression.
  - --path-include= expression: Include an additional pattern after exclusion patterns.
  - -O|--selected-only: Ignore software packages that are not selected for installation or upgrade.
  - -E|--skip-same-version: Ignore software packages with the same version as the installed package.
  - -G|--refuse-downgrade: Ignore software packages with versions earlier than the installed package.
  - -B|--auto-deconfigure: Install even if it affects other software packages.
  - --[no-]triggers: Skip or force the handling of triggers that follow.
  - --verify-format=format: Check the output format ('rpm' is supported).
  - --no-debsig: Do not attempt to verify the signature of the software package.
  - -D|--debug=octal: Enable debugging (see -Dhelp or --debug=help for more information).
  - --status-logger=command: Send status updates to the standard input of command.
  - --log=filename: Log status updates and operation information to filename.
  - --ignore-depends=package,...: Ignore all dependencies related to package.

  - --force-...: Ignore encountered problems (see --force-help for more details).
  - --no-force-...|--refuse-...: Abort execution when encountering problems.
  - --abort-after `<n>` Abort after encountering `<n>` errors.

- ## Common Commands

  - Install package

    ```
    dpkg -i package.deb
    ```

  - Remove package

    ```
    dpkg -r package
    ```

  - Remove package (including configuration files)

    ```
    dpkg -P package
    ```

  - List the files associated with the package

    ```
    dpkg -L package
    ```

  - Show the version of the package

    ```
    dpkg -l package
    ```

  - Unpack the contents of a deb package

    ```
    dpkg --unpack package.deb
    ```

  - Search for the package that owns the specified file or keyword

    ```
    dpkg -S keyword
    ```

  - List all currently installed packages

    ```
    dpkg -l
    ```

  - List the contents of a deb package

    ```
    dpkg -c package.deb
    ```

  - Configure the package

    ```
    dpkg --configure package
    ```

5. find

The `find` command is used to search for files and directories in a specified directory.

It can use different options to filter and limit the search results. Any string before the parameters is considered as the directory name to search for. If no parameters are set, the `find` command will search for subdirectories and files in the current directory and display all the found subdirectories and files.

- ## Grammar Explanation

  ```
  find [-H] [-L] [-P] [-Olevel] [-D debugopts] [path...] [expression]
  ```

- ## Option Description

  path is the directory path to search, which can be a directory or file name. Multiple paths can be specified, separated by spaces. If no path is specified, it defaults to the current directory.

  expression is an optional parameter that specifies the search conditions, such as file names, file types, file sizes, etc.

  There are dozens of options that can be used in the expression. Here are the most commonly used ones:

  - `-name pattern`: Search by file name, supports wildcard characters `*` and `?`.
  - `-iname pattern`: Similar to `-name`, but ignores case differences.
  - `-type type`: Search by file type, can be `f` (regular file), `d` (directory), `l` (symbolic link), etc.
  - `-size [+-]size[cwbkMG]`: Search by file size, supports using `+` or `-` to specify larger or smaller sizes. Units can be `c` (bytes), `w` (words), `b` (blocks), `k` (KB), `M` (MB), or `G` (GB).
  - `-mtime days`: Search by modification time, supports using `+` or `-` to specify days before or after a certain time. `days` is an integer representing the number of days.
  - `-user username`: Search by file owner.
  - `-group groupname`: Search by file group.

  The time parameters used in the `find` command are as follows:

  - `-amin n`: Search for files accessed within the last `n` minutes.
  - `-atime n`: Search for files accessed within the last `n*24` hours.
  - `-cmin n`: Search for files with status changes (e.g., permissions) within the last `n` minutes.
  - `-ctime n`: Search for files with status changes (e.g., permissions) within the last `n*24` hours.
  - `-mmin n`: Search for files modified within the last `n` minutes.
  - `-mtime n`: Search for files modified within the last `n*24` hours.

  In these parameters, `n` can be a positive number, negative number, or zero. A positive number represents files modified or accessed within the specified time period, a negative number represents files modified or accessed before the specified time period, and zero represents files modified or accessed at the current time.

  For example, `-mtime 0` searches for files modified today, and `-mtime -7` searches for files modified more than a week ago.

  Explanation of the time parameter `n`:

  - `+n`: Search for files or directories older than `n` days.
  - `-n`: Search for files or directories with attribute changes within the last `n` days.
  - `n`: Search for files or directories with attribute changes on a specific day `n` days ago.

Common Commands

List all files and folders in the current directory and subdirectories:

```
find .
```

Find a file named file.txt in the current directory:

```
find . -name file.txt
```

List all files with the **.c** file extension in the current directory and its subdirectories:

```
find . -name "*.c"
```

Same as above, but ignore case:

```
find . -iname "*.c"
```

List all files in the current directory and its subdirectories:

```
find . -type f
```

Find files larger than 1MB in /home directory:

```
find . -size +1M
```

Search for files smaller than 10KB:

```
find . -type f -size -10k
```

Search for files that are exactly 10KB:

```
find . -type f -size 10k
```

- File size units:
    - **b** —— block (512 bytes)
    - **c** —— byte
    - **w** —— word (2 bytes)
    - **k** —— kilobyte
    - **M** —— megabyte
    - **G** —— gigabyte

Find files modified in /var/log directory 7 days ago

```
find /var/log -mtime +7
```

List all files that were last updated 20 days ago in the current directory and its subdirectories, exactly 20 days ago

```
find . -ctime  20
```

List all files that were last updated 20 days ago or earlier in the current directory and its subdirectories

```
find . -ctime  +20
```

List all files that were last updated within the last 20 days in the current directory and its subdirectories

```
find . -ctime  20
```

Find regular files in /var/log directory whose modification time is more than 7 days ago, and prompt before deleting them

```
find /var/log -type f -mtime +7 -ok rm {} \;
```

Find files in the current directory whose owner has read and write permissions, and the group and other users have read permissions

```
find . -type f -perm 644 -exec ls -l {} \;
```

List all ordinary files in the system with a length of 0, and display their complete paths.

```
find / -type f -size 0 -exec ls -l {} \;
```

Search for all files ending with .txt and .pdf in the current directory and subdirectories.

```
find . \( -name "*.txt" -o -name "*.pdf" \)
or
find . -name "*.txt" -o -name "*.pdf"
```

Match file paths or filenames.

```
find /usr/ -path "*local*"
```

Match file paths based on regular expressions.

```
find . -regex ".*\(\.txt\|\.pdf\)$"
```

Same as above, but ignore case.

```
find . -iregex ".*\(\.txt\|\.pdf\)$"
```

Negate the parameter, find files in /home that do not end with .txt.

```
find /home ! -name "*.txt"
```

Search based on file type.

```
find . -type type_parameter
```

- List of type_parameters:
    - **f** for regular files
    - **l** for symbolic links
    - **d** for directories
    - **c** for character devices
    - **b** block device
    - **s** socket
    - **p** FIFO

Based on directory depth search, with a maximum depth limit of 3

```
find . -maxdepth 3 -type f
```

Search for all files that are at least 2 subdirectories deep from the current directory

```
find . -mindepth 2 -type f
```

Delete all `.log` files in the current directory

```
find . -type f -name "*.log" -delete
```

Search for files with permission 777 in the current directory

```
find . -type f -perm 777
```

Find `.conf` files in the current directory whose permissions are not 644

```
find . -type f -name "*.conf" ! -perm 644
```

Find all files owned by the user `sunrise` in the current directory

```
find . -type f -user sunrise
```

Find all files owned by the group `sunrise` in the current directory

```
find . -type f -group sunrise
```

Find all files with owner `root` in the current directory and change the ownership to user `sunrise`

```
find . -type f -user root -exec chown sunrise {} \;
```

In the above example, is used in combination with the **-exec** option to match all files and will be replaced with the respective file name.

Find all the `.txt` files under the `home` directory and delete them:

```
find $HOME/. -name "*.txt" -ok rm {} \;
```

In the above example, **-ok** is similar to **-exec** but prompts for confirmation before executing the operation.

Find all `.txt` files in the current directory and concatenate them into a file named `all.txt`:

```
find . -type f -name "*.txt" -exec cat {} \; > /all.txt
```

Search for all `.txt` files in the current directory or its subdirectories, but skip the subdirectory `sk`:

```
find . -path "./sk" -prune -o -name "*.txt" -print
```

> ⚠ `./sk` should not be written as `./sk/` or it will have no effect.

Ignore two directories:

```
find . \( -path ./sk -o -path ./st \) -prune -o -name "*.txt" -print
```

> ⚠ If using relative paths, `./` must be added.

To list all files with zero length:

```
find . -empty
```

Count the number of lines in code files:

```
find . -name "*.c" | xargs cat | grep -v ^$ | wc -l # Code line count,
excluding blank lines.
```

6. grep command

Powerful text search tool.

grep (global search regular expression (RE) and print out the line) is a powerful text search tool that can search text using regular expressions and print out the matching lines. It is used to filter/search for specific characters. It can be used in combination with various commands, making it highly flexible.

Similar commands include egrep, fgrep, rgrep.

- ### Grammar Explanation

  ```
    grep [OPTION]... PATTERNS [FILE]...
    grep [OPTION...] PATTERNS [FILE...]
    grep [OPTION...] -e PATTERNS ... [FILE...]
    grep [OPTION...] -f PATTERN_FILE ... [FILE...]
  ```

  - **PATTERNS** - Represents the string or regular expression to search for.
  - **FILE** - Represents the file name to search in. Multiple files can be searched at once. If the `FILE` parameter is omitted, it defaults to reading data from standard input.

- ## Option Explanation

  Commonly used options:

  - `-i`: Ignore case for matching.
  - `-v`: Print only non-matching lines.
  - `-n`: Display line numbers of matching lines.
  - `-r`: Recursively search files in subdirectories.
  - `-l`: Print only the names of matching files.
  - `-c`: Print only the count of matching lines.

  Additional parameter explanations:

  - **-a or --text**: Do not ignore binary data.
  - **-A**`<num\>` **or --after-context=**`<num\>`: In addition to displaying the column that matches the pattern, display the lines after that line.
  - **-b or --byte-offset**: Before displaying the line that matches the pattern, indicate the number of the first character in that line.
  - **-B**`<num\>` **or --before-context=**`<num\>`: In addition to displaying the column that matches the pattern, display the lines before that line.
  - **-c or --count**: Count the number of matching lines.
  - **-C**`<num\>` **or --context=**`<num\>` **or -**`<num\>`: In addition to displaying the column that matches the pattern, display the lines before and after that line.
  - **-d** `<action\>` **or --directories=**`<action\>`: When searching directories instead of files, this parameter must be used, otherwise the grep command will report an error and stop.
  - **-e**`<pattern\>` **or --regexp=**`<pattern\>`: Specify a string as the pattern for searching file contents.
  - **-E or --extended-regexp**: Use extended regular expressions as the pattern.
  - **-f**`<pattern_file\>` **or --file=**`<pattern_file\>`: Specify a pattern file that contains one or more pattern styles for grep to search for in file contents, with one pattern style per line.
  - **-F or --fixed-regexp**: Treat the pattern as a list of fixed strings.
  - **-G or --basic-regexp**: Treat the pattern as a plain representation.
  - **-h or --no-filename**: Do not display the file name before the line that matches the pattern.
  - **-H or --with-filename**: Displays the file name before the line that matches the pattern.
  - **-i or --ignore-case**: Ignores case differences in the pattern.
  - **-l or --file-with-matches**: Lists the file names that contain matches of the specified pattern.
  - **-L or --files-without-match**: Lists the file names that do not contain any matches of the specified pattern.
  - **-n or --line-number**: Displays the line numbers before the lines that match the pattern.
  - **-o or --only-matching**: Only displays the matching part of the pattern.
  - **-q or --quiet or --silent**: Does not display any information.
  - **-r or --recursive**: The same effect as specifying "-d recurse".
  - **-s or --no-messages**: Does not display error messages.
  - **-v or --invert-match**: Displays all lines that do not contain the matching text.
  - **-V or --version**: Displays version information.
  - **-w or --word-regexp**: Only displays whole words that match the pattern.
  - **-x --line-regexp**: Only displays whole lines that match the pattern.
  - **-y**: The same effect as specifying "-i".

  Regular expressions:

```
^    # Anchors the match to the beginning of a line. Example: '^grep'
matches any line that starts with 'grep'.
$    # Anchors the match to the end of a line. Example: 'grep$' matches any
line that ends with 'grep'.
.    # Matches any single character except a newline character. Example:
'gr.p' matches 'gr' followed by any character, then 'p'.
*    # Matches zero or more of the preceding character. Example: '*grep'
matches any line that has zero or more spaces followed by 'grep'.
.*   # Matches any sequence of characters.
[]   # Matches a single character within the specified range. Example:
'[Gg]rep' matches 'Grep' or 'grep'.
[^]  # Matches a single character not within the specified range. Example:
'[^A-Z]rep' matches any line that does not start with a letter between A and
Z, followed by 'rep'.
\(..\)  # Marks a matching pattern. Example: '\(love\)', 'love' is marked as
pattern 1.
\<      # Anchors the match to the beginning of a word. Example: '\<grep'
matches any line with a word that starts with 'grep'.
\>      # Anchors the match to the end of a word. Example: 'grep\>' matches
any line with a word that ends with 'grep'.
x\{m\}  # Repeats the character 'x' exactly 'm' times. Example: 'O\{5\}'
matches any line with exactly 5 'o's.
x\{m,\}   # Repeats the character 'x' at least 'm' times. Example: 'o\{5,\}'
matches any line with at least 5 'o's.
x\{m,n\}  # Repeats the character 'x' at least 'm' times but no more than
'n' times. Example: 'o\{5,10\}' matches any line with 5 to 10 'o's.
\w    # Matches word and digit characters, equivalent to [A-Za-z0-9].
Example: 'G\w*p' matches a word that starts with 'G', followed by zero or
more word or digit characters, and ends with 'p'.
\W    # Matches a non-word character. Example: '\W' matches any non-word
character.
\b    # Matches a word boundary. Example: '\bgrep\b' only matches the word
'grep'.
```

- ## Common commands

Searches for a word in a file and returns the lines that contain the pattern **"match_pattern"**.

```
grep match_pattern file_name
grep "match_pattern" file_name
```

Searches for a pattern in multiple files.

```
grep "match_pattern" file_1 file_2 file_3 ...
```

Recursively search for lines matching the regular expression `pattern` in all files within the directory `dir`, and print the filenames and line numbers of the matching lines:

```
grep -r -n pattern dir/
```

Search for the string `world` in standard input and only print the number of matching lines:

```
echo "hello world" | grep -c world
```

In the current directory, search for files with the word "file" in their suffix that contain the string "test", and print the lines with that string. You can use the following command for this:

```
grep test *file
```

Search for files that meet the criteria recursively. For example, search for files in the specified directory `/etc/` and its subdirectories (if any) that contain the string `update`, and print the lines with that string. The command to use is:

```
grep -r update /etc
```

Perform a reverse search, using the `-v` parameter to print the contents of lines that do not match the criteria. Search for files with `conf` in their filenames that do not contain `test` on the lines. The command to use is:

```
grep -v test *conf*
```

Use the `--color=auto` option to mark the matching text with color:

```
grep "match_pattern" file_name --color=auto
```

Use the `-E` option to use regular expressions:

```
grep -E "[1-9]+"
# or
egrep "[1-9]+"
```

Use the `-P` option to use Perl-compatible regular expressions:

```
grep -P "(\d{3}\-){2}\d{4}" file_name
```

Only output the matching part in the file with the **-o** option.

```
echo this is a test line. | grep -o -E "[a-z]+\."
line.

echo this is a test line. | egrep -o "[a-z]+\."
line.
```

Count the number of lines in a file or text that contain the matching string with the **-c** option.

```
grep -c "text" file_name
```

Search for records in the command line history that have entered the `git` command.

```
history | grep git
```

Output the number of lines that contain the matching string with the **-n** option.

```
grep "text" -n file_name
# Or
cat file_name | grep "text" -n
# For multiple files
grep "text" -n file_1 file_2
```

Print the character or byte offset where the pattern match is located.

```
echo gun is not unix | grep -b -o "not"
# The character offset of the string in a line is calculated from the first
character of that line, with the starting value as 0. The **-b -o** options
are generally used together.
```

Search multiple files and find which files contain the matching text.

```
grep -l "text" file1 file2 file3...
```

7. ifconfig

ifconfig is a command used to configure and manage network interfaces. It allows users to view and modify the configuration of network interfaces, including IP address, subnet mask, MAC address, MTU, broadcast address, point-to-point address, etc.

- ### Grammar Explanation

```
ifconfig [-a] [-v] [-s] <interface> [[<AF>] <address>]
  [add <address>[/<prefixlen>]]
  [del <address>[/<prefixlen>]]
  [[-]broadcast [<address>]]  [[-]pointopoint [<address>]]
  [netmask <address>]  [dstaddr <address>]  [tunnel <address>]
  [outfill <NN>] [keepalive <NN>]
  [hw <HW> <address>]  [mtu <NN>]
  [[-]trailers]  [[-]arp]  [[-]allmulti]
  [multicast]  [[-]promisc]
  [mem_start <NN>]  [io_addr <NN>]  [irq <NN>]  [media <type>]
  [txqueuelen <NN>]
  [[-]dynamic]
  [up|down] ...
```

- ### Option Explanation

  - `ifconfig`: displays all configured and active network interfaces along with their status.
  - `ifconfig -a`: displays all network interfaces, including those that are not active.
  - `ifconfig <interface>`: displays the configuration of the specified network interface.
  - `ifconfig <interface> up`: activates the specified network interface.
  - `ifconfig <interface> down`: deactivates the specified network interface.
  - `ifconfig <interface> add <address>`: adds an IP address to the specified network interface.
  - `ifconfig <interface> del <address>`: removes an IP address from the specified network interface.

  - `ifconfig <interface> netmask <address>`: sets the subnet mask for the specified network interface.
  - `ifconfig <interface> broadcast <address>`: sets the broadcast address.
  - `ifconfig <interface> pointopoint <address>`: sets the point-to-point address.
  - `ifconfig <interface> hw <HW> <address>`: sets the MAC address.
  - `ifconfig <interface> mtu <NN>`: sets the MTU (Maximum Transmission Unit).
  - `ifconfig <interface> arp`: enables ARP (Address Resolution Protocol).
  - `ifconfig <interface> promisc`: enables promiscuous mode, which allows receiving all packets passing through the network interface.
  - `ifconfig <interface> multicast`: enables multicast mode.
  - `ifconfig <interface> dynamic`: enables dynamic configuration.
  - `ifconfig -s`: displays network interface information in a concise format.
  - `ifconfig -v`: displays detailed information.

- ## Common commands

  Common commands

  ```
  ifconfig   # Network interface in active state
  ifconfig -a  # All configured network interfaces, regardless of whether they are active
  ifconfig eth0  # Display network card information for eth0
  ```

  Start and stop specified network card

  ```
  ifconfig eth0 up
  ifconfig eth0 down
  ```

  Configure IP address

  ```
  ifconfig eth0 192.168.1.10
  ifconfig eth0 192.168.1.10 netmask 255.255.255.0
  ifconfig eth0 192.168.1.10 netmask 255.255.255.0 broadcast 192.168.1.255
  ```

8. ip

The **ip** command is similar to the `ifconfig` command, but more powerful. Its main function is to display or configure network devices.

The **ip** command is an enhanced version of the network configuration tool in Linux, which replaces the ifconfig command.

- ## Grammar Explanation

  ```
  ip [ OPTIONS ] OBJECT { COMMAND | help }
  ip [ -force ] -batch filename
  ```

  - **OBJECT**:

```
OBJECT := { link | address | addrlabel | route | rule | neigh | ntable |
       tunnel | tuntap | maddress | mroute | mrule | monitor | xfrm |
       netns | l2tp | macsec | tcp_metrics | token }

-V: Display version information of the command;
-s: Output more detailed information;
-f: Force the usage of the specified protocol family;
-4: Specify that the network layer protocol used is IPv4;
-6: Specify that the network layer protocol used is IPv6;
-0: Output each record in a single line, even if the content is long;
-r: When displaying hosts, use domain names instead of IP addresses.
```

- **OPTIONS**:

```
OPTIONS := { -V[ersion] | -s[tatistics] | -d[etails] | -r[esolve] |
        -h[uman-readable] | -iec |
        -f[amily] { inet | inet6 | ipx | dnet | bridge | link } |
        -4 | -6 | -I | -D | -B | -0 |
        -l[oops] { maximum-addr-flush-attempts } |
        -o[neline] | -t[imestamp] | -ts[hort] | -b[atch] [filename] |
        -rc[vbuf] [size] | -n[etns] name | -a[ll] }

Network object: Specify the network object to manage;
Specific operation: Perform specific operation on the specified network object;
help: Display help information on the supported operation commands of the network object.
```

Common commands

```
ip link show                     # Show network interface information
ip link set eth0 up             # Enable network card
ip link set eth0 down            # Disable network card
ip link set eth0 promisc on      # Enable promiscuous mode for network card
ip link set eth0 promisc off     # Disable promiscuous mode for network card
ip link set eth0 txqueuelen 1200 # Set the queue length for the network card
ip link set eth0 mtu 1400        # Set the maximum transmission unit for the network card

ip addr show     # Show IP information for the network card
ip addr add 192.168.0.1/24 dev eth0 # Assign IP address 192.168.0.1 to eth0 network card
ip addr del 192.168.0.1/24 dev eth0 # Delete IP address of eth0 network card

ip route show # Show system routes
ip route add default via 192.168.1.254   # Set the default route for the system
ip route list                 # View route information
ip route add 192.168.1.0/24  via  192.168.0.254 dev eth0 # Set the gateway for 192.168.4.0 network segment to 192.168.0.254, using eth0 interface
ip route add default via  192.168.0.254  dev eth0        # Set the default gateway to 192.168.0.254
ip route del 192.168.1.0/24   # Delete the gateway for 192.168.4.0 network segment
ip route del default          # Delete the default route
ip route delete 192.168.1.0/24 dev eth0 # Delete route
```

Get all network interfaces of the host

```
ip link | grep -E '^[0-9]' | awk -F: '{print $2}'
```

9. mount

mount is a command used to mount file systems.

- ### Grammar Explanation

```
mount [-l|-h|-V]
mount -a [-fFnrsvw] [-t fstype] [-O optlist]
mount [-fnrsvw] [-o options] device|dir
mount [-fnrsvw] [-t fstype] [-o options] device dir
```

- ### Option Explanation

```
-V: Display the version of the program
-h: Display the help message
-v: Display verbose information, usually used for debugging with -f
-a: Mount all file systems defined in /etc/fstab
-F: This command is usually used together with -a, it creates a process for
each mount action. It can speed up mounting of a large number of NFS file
systems.
-f: Usually used for debugging purposes. It makes mount simulate the entire
mounting process without actually performing the mount action. It is often
used together with -v.
-n: By default, mount writes a record in /etc/mtab after mounting. This
option cancels that action when there is no writable file system available
in the system.
-s-r: Equivalent to -o ro
-w: Equivalent to -o rw
-L: Mounts a partition with a specific label.
-U: Unmounts the file system with the specified partition number. -L and -U
only make sense when there is a /proc/partitions file available.
-t: Specifies the file system type. Usually not necessary, as mount
automatically selects the correct type.
-o async: Enables asynchronous mode, where all file read and write
operations are performed asynchronously.
-o sync: Executes in sync mode.
-o atime, -o noatime: When atime is enabled, the system updates the "last
accessed time" of a file each time it is read. This option can be turned off
to reduce the number of writes when using flash file systems.
-o auto, -o noauto: Enables/disables auto-mounting mode.
-o defaults: Uses default options rw, suid, dev, exec, auto, nouser, and
async.
-o dev, -o nodev -o exec, -o noexec: Allows/executes execution of files.
-o suid, -o nosuid: Allows execution of files with root privileges.
-o user, -o nouser: Allows users to perform mount/umount actions.
```

```
-o remount: Remounts a file system that is already mounted with different
options. For example, if the system is originally mounted as read-only, it
can be remounted in read-write mode.
-o ro: Mounts in read-only mode.
-o rw: Mounts in read-write mode.
-o loop= : Uses loop mode to mount a file as a disk partition.
```

- ### Common commands

  Mount /dev/hda1 under /mnt:

  ```
  mount /dev/hda1 /mnt
  ```

  Mount /dev/hda1 under /mnt in read-only mode.

  ```
  mount -o ro /dev/hda1 /mnt
  ```

  Remount the root directory "/" in read-write mode.

  ```
  mount -o remount,rw /
  ```

  Mount nfs network file system.

  ```
  mount -t nfs -o nolock 192.168.1.20:/home/ /tmp/nfs
  ```

10. netstat

The **netstat command** is used to print the status of the network system in Linux, allowing you to know the network situation of the entire system.

- ### Grammar Explanation

  ```
  netstat [-vWeenNcCF] [<Af>] -r
  netstat {-V|--version|-h|--help}
  netstat [-vWnNcaeol] [<Socket> ...]
  netstat { [-vWeenNac] -i | [-cnNe] -M | -s [-6tuw] }
  ```

- ### Option Explanation

  - `-A`: List the relevant addresses in the network type connection.
  - `-r, --route`: Display the routing table and list the routing information of the system.
  - `-i, --interfaces`: Display network interface information, including interface names, IP addresses, and other relevant information.
  - `-g, --groups`: Display multicast group member information, including which network members are in the multicast group.
  - `-s, --statistics`: Display network statistics information, similar to SNMP (Simple Network Management Protocol), providing detailed statistics on network activity.
  - `-M, --masquerade`: Display masquerade connection information, usually used in Network Address Translation (NAT) networks.

  - `-v, --verbose`: Display detailed information, providing more information to help diagnose network problems.
  - `-W, --wide`: Do not truncate the IP address to display complete IP address information.
  - `-n, --numeric`: Do not resolve hostnames or port names to display IP addresses, port numbers, and user information in numeric format.
  - `--numeric-hosts`: Do not resolve hostnames.
  - `--numeric-ports`: Do not resolve port names.
  - `--numeric-users`: Do not resolve user names.
  - `-N, --symbolic`: Resolve hardware names and display symbolic names for hardware devices.
  - `-e, --extend`: Display additional information. Use this option twice to get the maximum detailed information.
  - `-p, --programs`: Display the PID (Process Identifier) and program names to show process information related to sockets.
  - `-o, --timers`: Display timer information, including the timer status of sockets.
  - `-c, --continuous`: Make `netstat` continuously print the selected information every second for continuous monitoring.
  - `-l, --listening`: Only display listening server sockets.
  - `-a, --all`: Display all sockets, including connected and unconnected ones.
  - `-F, --fib`: Display Forwarding Information Base (FIB).
  - `-C, --cache`: Display the routing cache instead of the Forwarding Information Base.
  - `-Z, --context`: Display SELinux security context for displaying SELinux security information of sockets.
  - `-v, --verbose`: Enable verbose output to provide more information to the user about ongoing operations. Particularly useful when dealing with unconfigured address families, providing some useful information.
  - `-o, --timers`: Include information related to network timers.
  - `-p, --program`: Display the PID and name of the program that each socket belongs to.
  - `-l, --listening`: Only display listening sockets. By default, these are omitted.
  - `-a, --all`: Display both listening and non-listening sockets. When using the `--interfaces` option, display disabled interfaces.
  - `-F`: Print routing information from the FIB (default).
  - `-C`: Print routing information from the cache.

Common commands

Display detailed network conditions

```
netstat -a       #List all ports
netstat -at      #List all TCP ports
netstat -au      #List all UDP ports
```

Display current registered UDP connections

```
netstat -nu
```

Display usage of UDP port numbers

```
netstat -apu
```

Display network card list

```
netstat -i
```

Display multicast group relationships

```
netstat -g
```

Display network statistics

```
netstat -s    #Display statistics of all ports
netstat -st   #Display statistics of TCP ports
netstat -su   #Display statistics of UDP ports
```

Display listening sockets

```
netstat -l      #Only display listening ports
netstat -lt     #Only list all listening TCP ports
netstat -lu     #Only list all listening UDP ports
netstat -lx     #Only list all listening UNIX ports
```

Display PID and process name in netstat output

```
netstat -pt
```

`netstat -p` can be used with other options to add "PID/Process Name" to netstat output.

Continuously output netstat information

```
netstat -c   #output network information every second
```

Display kernel routing information

```
netstat -r
```

Use `netstat -rn` to display in numerical format, without querying host names.

Find ports that programs are running on

Not all processes can be found, those without permission will not be displayed, view all information with root permission.

```
netstat -ap | grep ssh
```

Find processes running on a specific port

```
netstat -an | grep ':80'
```

Find process ID by port

```
netstat -anp|grep 8081 | grep LISTEN|awk '{printf $7}'|cut -d/ -f1
```

11. nohup

nohup (short for no hang up) is used to run a command in the background without being affected by terminal closures.

By default (when not redirected), it outputs a file named nohup.out to the current directory. If the nohup.out file in the current directory is not writable, the output will be redirected to the `$HOME/nohup.out` file. If no file can be created or opened for appending, the specified command in the "command" parameter will not be executable. If the standard error is a terminal, all output from the specified command that is written to the standard error will be redirected to the same file descriptor as the standard output.

- ## Grammar Explanation

  ```
  nohup COMMAND [ARG]... [  & ]
  nohup OPTION
  ```

  COMMAND: The command to be executed.

  ARG: Any additional parameters that can be used to specify an output file.

  &: Allows the command to be executed in the background, even after the terminal is closed.

- ## Option Explanation

  - `--help`: Display help information.
  - `--version`: Display version information.

- ## Common commands

  The following command executes the runoob.sh script in the background under the root directory:

  ```
  nohup /root/runoob.sh &
  ```

  To stop the execution, you need to use the following command to find the PID of the running script using nohup, and then use the kill command to delete it:

  ```
  ps -aux | grep "runoob.sh"
  ```

  The following command executes the runoob.sh script in the background under the root directory and redirects the input to the runoob.log file:

  ```
  nohup /root/runoob.sh > runoob.log 2>&1 &
  ```

  Explanation of `2>&1`:

  Redirect standard error 2 to standard output &1, and then redirect standard output &1 to the runoob.log file.

  - 0 - stdin (standard input)
  - 1 - stdout (standard output)
  - 2 - stderr (standard error output)

12. ps

The **ps command** is used to display the current system process status. Ps is the most basic and powerful process viewing command. By using this command, you can determine which processes are running, their running status, whether processes have ended, if any processes are in a zombie state, and which processes are consuming excessive resources.

Grammar Explanation

```
ps [options]
```

Option Explanation

Basic options:

- `-A, -e`: Display all processes.
- `-a`: Display all processes with a terminal (tty), excluding session leaders.
- `a`: Display all processes with a terminal (tty), including processes of other users.
- `-d`: Display all processes except session leaders.
- `-N, --deselect`: Deselect (invert) the processes.
- `r`: Display only running processes.
- `T`: Display all processes associated with the current terminal.
- `x`: Display processes without a control terminal.

Selection by list:

- `-C <command>`: Select processes by command name.
- `-G, --Group <GID>`: Select processes by real group ID or group name.
- `-g, --group <group>`: Select processes by session or effective group name.
- `-p, p, --pid <PID>`: Select processes by process ID.
- `--ppid <PID>`: Select processes by parent process ID.
- `-q, q, --quick-pid <PID>`: Quick mode, select processes by process ID.
- `-s, --sid <session>`: Select processes by session ID.
- `-t, t, --tty <tty>`: Select processes by terminal.
- `-u, U, --user <UID>`: Select processes by effective user ID or username.
- `-U, --User <UID>`: Select processes by real user ID or username.

Output formats:

- `-F`: Display additional detailed information.
- `-f`: Full format, including the command line.
- `f, --forest`: Display the process tree in ASCII art.
- `-H`: Display process hierarchy.
- `-j`: Job format.
- `j`: BSD job control format.
- `-l`: Long format.
- `l`: BSD long format.
- `-M, Z`: Add security data (for SELinux).
- `-O <format>`: Use default column preloading.
- `-O <format>`: Preload columns in BSD style.
- `-o, -o, --format <format>`: User-defined format.
- `s`: Signal format.
- `u`: User-oriented format.

  - `v`: Virtual memory format.
  - `X`: Register format.
  - `-y`: Do not display flags, display RSS and address (with -l).
  - `--context`: Display security context (for SELinux).
  - `--headers`: Repeat header lines on each page.
  - `--no-headers`: Do not print headers at all.
  - `--cols, --columns, --width <num>`: Set screen width.
  - `--rows, --lines <num>`: Set screen height.

Show threads:

  - `H`: Show threads as if they were processes.
  - `-L`: May include LWP and NLWP columns.
  - `-m, -m`: Display threads after processes.
  - `-T`: May include SPID column.

Miscellaneous options:

  - `-c`: Display scheduling class with -l option.
  - `c`: Display the real command name.
  - `e`: Display the environment after the command.
  - `k, --sort`: Specify sort order, e.g.: [+|-]key[,[+|-]key[,...]].
  - `L`: Display format specifiers.
  - `n`: Display numeric user ID and wchan.
  - `S, --cumulative`: Include some terminated child process data.
  - `-y`: Do not display flags, display RSS (only with -l).
  - `-V, -V, --version`: Display version information and exit.
  - `-w, -w`: Unlimited output width.

Help options:

  - `--help <simple|list|output|threads|misc|all>`: Display help and exit. Different help modes can be selected.

- ## Common commands

List the PID and related information belonging to the currently logged-in user:

```
sunrise@ubuntu:~$ ps -l
F S   UID     PID    PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
4 S  1000    4295    4294  8  80   0 -  3304 do_wai pts/0    00:00:00 bash
0 R  1000    4304    4295  0  80   0 -  3504 -      pts/0    00:00:00 ps
```

  - `F`: represents the flag of this program, where 4 indicates that the user is a super user.
  - S: represents the STAT (state) of this program, and the meanings of different STATs will be explained below
    - `S`: Sleeping - the process is running.
    - `R`: Running - the process is running or ready to run.
    - `D`: Uninterruptible Sleep - the process is uninterruptible.
    - `T`: Stopped - the process has stopped.
    - `Z`: Zombie - the process is a zombie process.
    - `t`: Traced or stopped - the process is being traced or has stopped.
    - `P`: Parked - the process is being traced or has stopped, but it is waiting.
  - `UID`: UID of the process, indicating the user who runs the process.

- `PID`: Process ID, a unique identifier assigned to each process by the operating system.
- `PPID`: Parent Process ID, indicating the ID of the parent process that launched the current process.
- `C`: CPU usage percentage, indicating the percentage of CPU time occupied by the process.
- `PRI`: Process priority.
- `NI`: Nice value of the process, usually used to adjust the process priority.
- `ADDR`: Address space of the process. This is a kernel function that indicates the part of the program in memory. If the program is running, it is usually "-".
- `SZ`: Virtual memory size of the process in pages.
- `WCHAN`: The event or lock that the process is currently waiting for. If it is "-", it means the process is running.
- `TTY`: The terminal associated with the process (if any).
- `TIME`: CPU time the process has been running.
- `CMD`: Command line of the process.

List all the currently running programs in memory:

```
sunrise@ubuntu:~$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root            1  0.4  0.4 167592 10076 ?        Ss   10:16   0:02
/sbin/init
root            2  0.0  0.0      0     0 ?        S    10:16   0:00
[kthreadd]
root            4  0.0  0.0      0     0 ?        I<   10:16   0:00
[kworker/0:0H]
root            5  0.0  0.0      0     0 ?        I    10:16   0:00
[kworker/u8:0]
```

- `USER`: Represents the user to which the process belongs, i.e., the user who runs the process.
- `PID`: represents the process ID, a unique identifier used to identify each process.
- `%CPU`: represents the CPU usage of the process, i.e., the percentage of CPU time the process uses out of the total CPU time.
- `%MEM`: represents the memory usage of the process, i.e., the percentage of physical memory the process uses out of the total physical memory.
- `VSZ`: represents the virtual memory size of the process (Virtual Set Size), i.e., the size of the virtual memory the process can access, usually measured in kilobytes (KB).
- `RSS`: represents the physical memory size of the process (Resident Set Size), i.e., the current amount of physical memory the process occupies, usually measured in kilobytes (KB).
- `TTY`: represents the terminal associated with the process. If the process is not associated with a terminal, it displays "?".
- STAT : represents the state of the process, usually including some of the following states:
    - `R`: Running

        - `S`: Sleeping
        - `D`: Uninterruptible Sleep
        - `Z`: Zombie
        - `T`: Stopped
        - Other states may also exist, and their specific meanings may vary depending on the operating system.
    - `START`: represents the start time of the process, usually in the format of hour:minute.
    - `TIME`: represents the CPU time the process has used, usually in the format of hour:minute:second.
    - `COMMAND`: represents the command line of the process, i.e., the command and its arguments that the process is executing.

List similar program tree displays

```
sunrise@ubuntu:~$ ps -axjf
   PPID     PID    PGID     SID TTY        TPGID STAT   UID   TIME COMMAND
      1    2973    2973    2973 ?             -1 Ss       0   0:00 sshd:
/usr/sbin/sshd -D [listener] 0 of 10-100 startups
   2973    4067    4067    4067 ?             -1 Ss       0   0:00  \_ sshd:
root@pts/0
   4067    4239    4239    4239 pts/0       4364 Ss       0   0:00  |   \_ -
bash
   4239    4294    4294    4239 pts/0       4364 S        0   0:00  |
\_ su sunrise
   4294    4295    4295    4239 pts/0       4364 S     1000   0:00  |
    \_ bash
   4295    4364    4364    4239 pts/0       4364 R+    1000   0:00  |
        \_ ps -axjf
   2973    4069    4069    4069 ?             -1 Ss       0   0:00  \_ sshd:
root@notty
   4069    4242    4242    4242 ?             -1 Ss       0   0:00      \_
/usr/lib/openssh/sftp-server
```

Other commands

```
ps axo pid,comm,pcpu # View PID, name, and CPU usage of processes
ps aux | sort -rnk 4 # Sort processes by memory usage
ps aux | sort -nk 3  # Sort processes by CPU usage
ps -A # Show all process information
ps -u root # Show information of a specific user
ps -efL # View thread count
ps -e -o "%C : %p :%z : %a"|sort -k5 -nr # View processes and sort by memory
usage
ps -ef # Show all process information, including command lines
ps -ef | grep ssh # Common usage of ps and grep, find specific processes
ps -C nginx # Search processes by name or command
ps aux --sort=-pcpu,+pmem # Sort by CPU or memory usage, -descending,
+ascending
ps -f --forest -C nginx # Display process hierarchy in tree style
ps -o pid,uname,comm -C nginx # Display child processes of a parent process
ps -e -o pid,uname=USERNAME,pcpu=CPU_USAGE,pmem,comm # Redefine labels
ps -e -o pid,comm,etime # Show duration of process running
ps -aux | grep named # View detailed information of named process
ps -o command -p 91730 | sed -n 2p # Get service name by process ID
```

13. route

The **route** command is used to display and set the network routing table in the Linux kernel. The routes set by the route command are mainly static routes. To achieve communication between two different subnets, a router connecting both networks or a gateway located in both networks is needed.

Setting routes in Linux systems is usually done to solve the following problems:

If a Linux system is in a LAN (Local Area Network) with a gateway that allows machines to access the Internet, the IP address of this machine needs to be set as the default route for the Linux machine. It should be noted that adding routes directly through the command line using the route command will not be permanently saved. The route will become invalid when the network card or the machine is restarted. To ensure that the route is permanently set, the route command can be added to the `/etc/rc.local` file.

- ## Grammar Explanation

  ```
  route [-nNvee] [-FC] [<AF>]           List kernel routing tables
  route [-v] [-FC] {add|del|flush} ...  Modify routing table for AF.
  ```

  - `-A`: Set the address type.
  - `-v, --verbose`: Display detailed information.
  - `-n, --numeric`: Do not perform DNS reverse lookup and display IP addresses in numeric form.
  - `-e, --extend`: Display the routing table in netstat format.
  - `-F, --fib`: Display the forwarding information base (default).
  - `-C, --cache`: Display the route cache instead of the forwarding information base.
  - `-net`: Show the routing table for a network.
  - `-host`: Show the routing table for a host.

- ## Option Explanation

  - `add`: Used to add a specified routing record, routing the specified destination network or host to the specified network interface.
  - `del`: Used to delete a specified routing record.
  - `target`: Specifies the destination network or host.
  - `gw`: Used to set the default gateway.
  - `mss`: Set the maximum segment size (MSS) for TCP.
  - `window`: Specify the TCP window size for TCP connections through the routing table.
  - `dev`: Specify the network interface represented by the routing record.

- ## Common commands

  Display the current routing table:

  ```
  root@ubuntu:~# route
  Kernel IP routing table
  Destination     Gateway         Genmask         Flags Metric Ref    Use
  Iface
  default         192.168.0.1     0.0.0.0         UG    600    0        0
  wlan0
  default         192.168.1.1     0.0.0.0         UG    700    0        0 eth0
  192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0
  wlan0
  ```

```
192.168.1.0     0.0.0.0         255.255.255.0   U     700    0        0 eth0

root@ubuntu:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    600    0        0
wlan0
0.0.0.0         192.168.1.1     0.0.0.0         UG    700    0        0 eth0
192.168.0.0     0.0.0.0         255.255.255.0   U     600    0        0
wlan0
192.168.1.0     0.0.0.0         255.255.255.0   U     700    0        0 eth0
```

The Flags column indicates the status of the network node, and the Flags symbols are explained as follows:

- `U`: Up, indicating that the route is currently in an active state.
- `H`: Host, indicating that the gateway is a host.
- `G`: Gateway, indicating that the gateway is a router.
- `R`: Reinstate Route, indicating that the route has been initialized again using dynamic routing.
- shell `D`: Dynamically, indicating that the route has been dynamically written.
- `M`: Modified, indicating that the route has been dynamically modified by the routing daemon or router.
- `!`: Indicates that the route is currently in an inactive state.

Adding a gateway / setting a gateway

```
route add -net 192.168.2.0 netmask 255.255.255.0 dev eth0
```

Blocking a route

```
route add -net 192.168.2.0 netmask 255.255.255.0 reject
```

Deleting a route record

```
route del -net 192.168.2.0 netmask 255.255.255.0
route del -net 192.168.2.0 netmask 255.255.255.0 reject
```

Deleting and adding a default gateway

```
route del default gw 192.168.2.1
route add default gw 192.168.2.1
```

14. rsync

Rsync is a fast and powerful file copying tool. It can copy files locally or from another host using any remote shell, and it can also copy files with a remote rsync daemon. It provides numerous options that allow control over various aspects of its behavior and enables flexible specification of the files to be copied. Rsync is known for its incremental transfer algorithm, which reduces the amount of data sent over the network by only sending the differences between the source and existing files in the target. Rsync is widely used for backup and mirroring operations, as well as an improved alternative to the standard copy command for everyday use.

Rsync finds files to be transferred using the default "quick check" algorithm, which looks for files that have changed in size or modification time. When the quick check indicates that the data of a file does not need to be updated, any changes to other preserved attributes on the destination file (according to the requested options) are applied directly.

Grammar Explanation

```
Usage: rsync [OPTION]... SRC [SRC]... DEST
  or   rsync [OPTION]... SRC [SRC]... [USER@]HOST:DEST
  or   rsync [OPTION]... SRC [SRC]... [USER@]HOST::DEST
  or   rsync [OPTION]... SRC [SRC]... rsync://[USER@]HOST[:PORT]/DEST
  or   rsync [OPTION]... [USER@]HOST:SRC [DEST]
  or   rsync [OPTION]... [USER@]HOST::SRC [DEST]
  or   rsync [OPTION]... rsync://[USER@]HOST[:PORT]/SRC [DEST]

The ':' usages connect via remote shell, while '::' & 'rsync://' usages connect
to an rsync daemon, and require SRC or DEST to start with a module name.
```

Simplified syntax:

```
rsync [OPTION...] SRC... [DEST]
```

Option Explanation

- -v, --verbose: Increase verbosity
  - --info=FLAGS: Specify info messages to output
  - --debug=FLAGS: Specify debug messages to output
  - --msgs2stderr: Special handling for debug output
- -q, --quiet: Suppress non-error messages
  - --no-motd: Suppress MOTD in daemon mode (see man page notes)
- -c, --checksum: Skip based on checksum, not modification time and size
- -a, --archive: Archive mode; equivalent to -rlptgoD (excluding -H, -A, -X)
  - --no-OPTION: Turn off implied OPTION (e.g., --no-D)
- -r, --recursive: Recurse into directories
- -R, --relative: Use relative path names
  - --no-implied-dirs: Do not send implied directories with --relative
- -b, --backup: Create backups (see --suffix and --backup-dir)
  - --backup-dir=DIR: Place backup files into a hierarchy based on DIR
  - --suffix=SUFFIX: Set the suffix for backup files (default is ~ if no --backup-dir)
- -u, --update: Skip files that are newer on the receiving end
  - --inplace: Update destination files in-place (see man page)
  - --append: Append data to the shorter file
  - --append-verify: Similar to --append, but verify the checksum of the appended file with the existing data
- -d, --dirs: Transfer directories without recursion
- -l, --links: Copy symbolic links as symbolic links

- -L, --copy-links: Convert symbolic links to the specified file/directory
  - --copy-unsafe-links: Convert "unsafe" symbolic links only
  - --safe-links: Ignore symbolic links pointing outside the source tree
  - --munge-links: Obfuscate symbolic links to make them more secure (but unusable)
- -k, --copy-dirlinks: Convert symbolic links to directories specified in the target directory
- -K, --keep-dirlinks: Treat symbolic link directories on the receiving end as directories
- -H, --hard-links: Preserve hard links
- -p, --perms: Preserve permissions
- -E, --executability: Preserve executability of files
  - --chmod=CHMOD: Affect the permissions of files and/or directories
- -A, --acls: Preserve ACLs (implies --perms)
- -X, --xattrs: Preserve extended attributes
- -o, --owner: Preserve ownership (superuser only)
- -g, --group: Preserve group ownership
  - --devices: Preserve device files (superuser only)
  - --copy-devices: Copy device contents as regular files
  - --specials: Preserve special files
- -D: Equivalent to --devices --specials
- -t, --times: Preserve modification times
- -O, --omit-dir-times: Omit directories from --times
- -J, --omit-link-times: Omit symbolic links from --times
  - --super: Receiver attempts super-user activities
  - --fake-super: Store/recover privileged attributes using xattrs
- -S, --sparse: Convert continuous empty blocks to sparse blocks
  - --preallocate: Preallocate the destination file before writing to it
- -n, --dry-run: Perform a trial run without making any changes
- -W, --whole-file: Transfer whole file (no incremental transfer algorithm used)
  - --checksum-choice=STR: Choose the checksum algorithm
- -x, --one-file-system: Do not cross filesystem boundaries
- -B, --block-size=SIZE: Force a fixed checksum block size
- -e, --rsh=COMMAND: Specify the remote shell to use
  - --rsync-path=PROGRAM: Specify the rsync to run on the remote machine
  - --existing: Skip creating new files on the receiving end
  - --ignore-existing: Skip updating files that already exist on the receiving end
  - --remove-source-files: Sender deletes synchronized files (non-directories)
  - --del: Alias for --delete-during
  - --delete: Remove files in the destination directory that are not present in the source directory
  - --delete-before: Receiver deletes files before transfer
  - --delete-during: Receiver deletes files during transfer
  - --delete-delay: Deletes files after finding the delete operation
  - --delete-after: Receiver deletes files after transfer
  - --delete-excluded: Also remove excluded files from the destination directory
  - --ignore-missing-args: Ignore missing source arguments without error
  - --delete-missing-args: Remove missing source arguments from destination

    - --ignore-errors: Continue deleting even if there are I/O errors
    - --force: Forcefully delete directories even if not empty- --max-delete=NUM: Delete at most NUM files
    - --max-size=SIZE: Do not transfer files larger than SIZE
    - --min-size=SIZE: Do not transfer files smaller than SIZE
    - --partial: Keep partially transferred files
    - --partial-dir=DIR: Put partially transferred files in DIR
    - --delay-updates: Put all updated files in a specified location at the end of the transfer
  - -m, --prune-empty-dirs: Exclude empty directories from the file list
    - --numeric-ids: Don't map uid/gid values through usernames/groups
    - --usermap=STRING: Customize username mapping
    - --groupmap=STRING: Customize group name mapping
    - --chown=USER:GROUP: Simple username/group name mapping
    - --timeout=SECONDS: Set I/O timeout in seconds
    - --contimeout=SECONDS: Set timeout for daemon connection in seconds
  - -I, --ignore-times: Do not skip files that match in size and modification time
  - -M, --remote-option=OPTION: Only send OPTION to remote side
    - --size-only: Skip files that match in size only
  - @, --modify-window=NUM: Set the accuracy of modification time comparison
  - -T, --temp-dir=DIR: Create temporary files in directory DIR
  - -y, --fuzzy: Find similar files as a basis if no target file exists
    - --compare-dest=DIR: Also compare target files relative to DIR
    - --copy-dest=DIR: Include copies of unchanged files
    - --link-dest=DIR: Hard link unchanged files to DIR instead of copying
  - -z, --compress: Compress file data during transfer
    - --compress-level=NUM: Set compression level explicitly
    - --skip-compress=LIST: Skip compression for files with suffixes in LIST
  - -C, --cvs-exclude: Auto-ignore files similar to CVS
  - -f, --filter=RULE: Add a file filter rule
  - -F: Same as --filter='dir-merge /.rsync-filter'
    - --exclude=PATTERN: Exclude files that match PATTERN
    - --exclude-from=FILE: Read exclude patterns from FILE
    - --include=PATTERN: Do not exclude files that match PATTERN
    - --include-from=FILE: Read include patterns from FILE
    - --files-from=FILE: Read a list of source file names from FILE
  - -0, --from0: All *-from/filter files are delimited by 0
  - -s, --protect-args: Do not split on spaces, only split on wildcard special characters
    - --trust-sender: Trust the file list sent by the remote sender
    - --address=ADDRESS: Bind outgoing socket to ADDRESS of daemon
    - --port=PORT: Specify an alternate PORT number for double-colon
    - --sockopts=OPTIONS: Specify custom TCP options
    - --blocking-io: Use blocking I/O for remote shell operations
    - --stats: Provide some file transfer statistics
  - -8, --8-bit-output: Preserve high-bit characters in output
  - -h, --human-readable: Output numbers in a human-readable format
    - --progress: Show progress during transfer

  - -P: Same as --partial --progress
  - -i, --itemize-changes: Output a summary of all updated changes
    - --out-format=FORMAT: Output updates using the specified FORMAT
    - --log-file=FILE: Log operations to the specified FILE
    - --log-file-format=FMT: Log updates using the specified FMT
    - --password-file=FILE: Read daemon access password from FILE- --list-only: List files without copying them
    - --bwlimit=RATE: Limit socket I/O bandwidth
    - --stop-at=y-m-dTh:m : Stop rsync at year-month-dayThour:minute
    - --time-limit=MINS: Stop rsync after MINS minutes
    - --outbuf=N|L|B: Set output buffering to none, line, or block
    - --write-batch=FILE: Write batch update to FILE
    - --only-write-batch=FILE: Similar to --write-batch, but does not update the target
    - --read-batch=FILE: Read batch update from FILE
    - --protocol=NUM: Force the use of an old protocol version
    - --iconv=CONVERT_SPEC: Request character set conversion for file names
    - --checksum-seed=NUM: Set the block/file checksum seed (advanced option)
    - --noatime: Do not change atime when opening source files
  - -4, --ipv4: Use IPv4 preference
  - -6, --ipv6: Use IPv6 preference
    - --version: Display the version number
  - (-h) --help: Display help information (only when used alone with -h)

- ## Common commands

  - Copy local files: Copy files from /app directory to /userdata directory

    ```
    rsync -avSH /app/ /userdata/
    ```

  - Copy local machine content to remote machine

    ```
    rsync -av /app 192.168.1.12:/app
    ```

  - Copy remote machine content to local machine

    ```
    rsync -av 192.168.1.12:/app /app
    ```

  - Copy files from remote rsync server (running rsync in daemon mode) to local machine

    ```
    rsync -av root@192.168.1.12::www /userdata
    ```

  - Copy local machine files to remote rsync server (running rsync in daemon mode). Activate this mode when the DST path information contains the "::" separator.

    ```
    rsync -av /userdata root@192.168.1.12::www
    ```

  - Display file list of remote machine. This is similar to rsync transfer, but omitting the local machine information in the command.Please translate the Chinese parts in the following content into English, while keeping the original format and content:

```
rsync -v rsync://192.168.1.12/app
```

- Specify the password storage file, no need to enter the password, and directly execute rsync transfer.

```
rsync -rvzP --password-file=/etc/rsync.password
rsync@$192.168.1.12::app/ /app
```

15. scp

The Linux scp command is used to copy files and directories between Linux systems.

scp is short for secure copy. It is a secure remote file copy command based on ssh login in the Linux system.

scp is encrypted, while rcp is not encrypted. scp is an enhanced version of rcp.

- ### Grammar Explanation

```
scp [-346BCpqrTv] [-c cipher] [-F ssh_config] [-i identity_file]
            [-J destination] [-l limit] [-o ssh_option] [-P port]
            [-S program] source ... target
```

Simplified form:

```
scp [option] file_source file_target
```

  - **file_source**: Specifies the source file to be copied.
  - **file_target**: The target file. The format is `user@host:filename` (filename is the name of the target file).

- ### Option Explanation

  - -3: Transfers files between two remote hosts through the local host. If this option is not used, data will be transferred directly between the two remote hosts. Note that this option disables the progress display during transfer.
  - -4: Forces scp to use IPv4 addressing only.
  - -6: Forces scp to use IPv6 addressing only.
  - -B: Batch mode. Prevents asking for a password or passphrase.
  - -C: Enables compression. (Passes the -C flag to ssh, which opens the compression feature)
  - -p: Preserves the modification time, access time, and access permissions of the original file.
  - -q: Quiet mode. Disables the progress meter and warning and diagnostic messages from ssh(1).
  - -r: Recursively copies the entire directory. Note that scp will follow symbolic links encountered in directory traversal.
  - -T: Disables strict filename checking. By default, when copying files from a remote host to a local directory, scp checks that the received filenames match the requested filenames on the command line to prevent unexpected or unnecessary files from being sent by the remote side. These checks may cause desired files to be rejected due to

    different ways of filename wildcard interpretation by different operating systems and shells. This option disables these checks but requires complete trust in the server not to send unexpected filenames.
    - -v: Verbose mode. Causes scp and ssh(1) to print debugging messages about their progress. This is helpful for debugging connection, authentication, and configuration problems.
    - -c cipher: Specifies the password used for encryption during data transmission. This option is passed directly to ssh(1).
    - -F ssh_config: Specifies an alternative per-user configuration file to be used instead of the default ssh(1) configuration file. This option is passed directly to ssh(1).
    - -i identity_file: Specifies the identity (private key) file to be used for public key authentication. This option is passed directly to ssh(1).
    - -l limit: Limits the bandwidth used, in Kbit/s.
    - -o ssh_option: Passes options in the format used in ssh_config(5) to ssh. This is useful for specifying options that do not have separate scp command-line flags.
    - -P port: Specifies the port to connect to on the remote host. Note that this option uses uppercase 'P' because lowercase '-p' is already reserved for preserving the modification time and mode of files.
    - -S program: Specifies the program to use for the encrypted connection. The program must understand ssh(1) options.

- ## Common commands

  ### Copying from local to remote

  Command format:

  ```
  scp local_file remote_username@remote_ip:remote_folder
  # or
  scp local_file remote_username@remote_ip:remote_file
  # or
  scp local_file remote_ip:remote_folder
  # or
  scp local_file remote_ip:remote_file
  ```

    - The first and second ones specify the username. After executing the command, you will need to enter the password. The first one specifies only the remote directory and keeps the file name the same, while the second one specifies the file name.
    - The third and fourth ones do not specify the username. After executing the command, you will need to enter the username and password. The third one specifies only the remote directory and keeps the file name the same, while the fourth one specifies the file name.

  Example:

  ```
  scp /home/sunrise/test.c root@192.168.1.10:/userdata
  scp /home/sunrise/test.c root@192.168.1.10:/userdata/test_01.c
  scp /home/sunrise/test.c 192.168.1.10:/userdata
  scp /home/sunrise/test.c 192.168.1.10:/userdata/test_01.c
  ```

  Copying directory command format:

```
scp -r local_folder remote_username@remote_ip:remote_folder
# or
scp -r local_folder remote_ip:remote_folder
```

  - The first one specifies the username. After executing the command, you will need to enter the password.
  - The second one does not specify the username. After executing the command, you will need to enter the username and password.

Example:

```
scp -r /home/sunrise/app/ root@192.168.1.10:/userdata/app/
scp -r /home/sunrise/app/ 192.168.1.10:/userdata/app/
```

The above commands will copy the local `app` directory to the remote `/userdata/app/` directory.

Copying from remote to local

The scp command for copying from remote to local is similar to the above command. Just swap the order of the last two parameters in the command for copying from local to remote.Copy files from a remote machine to a local directory

```
scp sunrise@192.168.1.10:/userdata/log.log /home/sunrise/
```

Download the file `log.log` from the directory `/userdata/` on the machine with IP address 192.168.1.10 to the local directory `/home/sunrise/`.

16. ssh

The **ssh command** is a client connection tool in the openssh suite, which can be used to securely remote login to a server using the ssh encryption protocol.

- ### Grammar Explanation

```
ssh [-46AaCfGgKkMNnqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address:]port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@]host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]
```

  - **destination**: Specifies the remote ssh server to connect to.
  - **command**: Specifies the command to execute on the remote ssh server.

- ### Option Explanation

  - `-4`: Force use of IPv4 addresses.
  - `-6`: Force use of IPv6 addresses.
  - `-A`: Enable authentication agent connection forwarding.
  - `-a`: Disable authentication agent connection forwarding.

  - `-B`: Bind to `bind_interface` address before attempting to connect to the target host. Useful on systems with multiple addresses.
  - `-b`: Use local specified address as the source IP address for corresponding connection.
  - `-C`: Request compression of all data.
  - `-F`: Specify the configuration file for SSH commands.
  - `-f`: Run SSH command in the background.
  - `-g`: Allow remote hosts to connect to local forwarded ports.
  - `-i`: Specify an identity (private key) file.
  - `-l`: Specify the login username to connect to the remote server.
  - `-N`: Do not execute a remote command.
  - `-o`: Specify configuration options.
  - `-p`: Specify the port on the remote server.
  - `-q`: Quiet mode.
  - `-X`: Enable X11 forwarding.
  - `-x`: Disable X11 forwarding.
  - `-y`: Enable trusted X11 forwarding.

- ## Common commands

```
# ssh username@remote_server_address
ssh sunrise@192.168.1.10
# Specify port
ssh -p 2211 sunrise@192.168.1.10

# SSH family
ssh -p 22 user@ip  # Default username is the current username, default port is 22
ssh-keygen # Generate ssh public and private keys for the current user
ssh-keygen -f keyfile -i -m key_format -e -m key_format # key_format: RFC4716/SSH2(default) PKCS8 PEM
ssh-copy-id user@ip:port # Copy the public key of the current user to the ~/.ssh/authorized_keys file on the server that needs SSH, enabling passwordless login
```

Connect to remote server

```
ssh username@remote_host
```

Connect to remote server and specify port

```
ssh -p port username@remote_host
```

Connect to remote server using a key file

```
ssh -i path/to/private_key username@remote_host
```

Execute a remote command locally

```
ssh username@remote_host "command"
```

Forward a local port to a remote server

```
ssh -L local_port:remote_host:remote_port username@remote_host
```

Forward a remote port to the local machine

```
ssh -R remote_port:local_host:local_port username@remote_host
```

17. tar

The **tar command** can be used to create archives for files and directories in Linux. With tar, you can create archives (backup files) for specific files, modify files within an archive, or add new files to an archive. Tar was originally used to create archives on tapes, but now users can create archives on any device. Using the tar command, you can pack a large number of files and directories into a single file, which is very useful for backing up files or combining multiple files into one file for network transmission.

First, let's clarify two concepts: packing and compressing. Packing refers to turning a large number of files or directories into a single file, while compressing means using compression algorithms to turn a large file into a small file.

Why do we need to distinguish between these two concepts? This is because many compression programs in Linux can only compress one file at a time. So when you want to compress a large number of files, you need to first pack these files into one package (using the tar command), and then compress the package using a compression program (gzip or bzip2 command).

- ### Grammar Explanation

```
tar [OPTION...] [FILE]...
```

- ### Option Explanation

```
-A, --catenate, --concatenate   append tar files to an archive
-c, --create                 create a new archive
-d, --diff, --compare        find differences between archive and file system
    --delete                 delete from the archive (not on tape!)
-r, --append                 append files to the end of the archive
-t, --list                   list the contents of an archive
    --test-label             test the volume label and exit
-u, --update                 only append files that are newer than the copy
-x, --extract, --get         extract files from an archive

Operation modifiers:

      --check-device         check device numbers when creating incremental
archives (default)
  -g, --listed-incremental=FILE   handle new-style GNU format incremental
backups
  -G, --incremental          handle old-style GNU format incremental backups
      --ignore-failed-read   do not exit with nonzero on unreadable files
      --level=NUMBER         output level for listing incremental archives
created with --incremental
  -n, --seek                 archive can be seeked
```

```
      --no-check-device      do not check device numbers when creating
incremental archives
      --no-seek              archive cannot be seeked
      --occurrence[=NUMBER]  process only the NUMBERth occurrence of each
file in the archive;
                             this option is only valid when used in
combination with one of the
                             following subcommands: --delete, --diff, --
extract, or --list;
                             and regardless of whether the file list is
given in command
                             line form or specified via -T option; NUMBER
defaults to 1
      --sparse-version=MAJOR[.MINOR]
                             set version of the sparse format to be used
(implied by --sparse)
-S, --sparse               Efficiently handle sparse files

Rewrite control:

-k, --keep-old-files       Don't replace existing files when extracting,
                            treat them as errors
    --keep-directory-symlink   Preserve existing symlinks to directories
when
                            extracting
    --keep-newer-files
                            Do not replace existing files that are newer
than the copy in the archive
    --no-overwrite-dir     Preserve metadata of existing directories
    --overwrite            Overwrite existing files when extracting
    --overwrite-dir        Overwrite metadata of existing directories when
extracting (default)

    --recursive-unlink     Remove directory hierarchy before extracting
directories
    --remove-files         Delete files after adding them to the archive
    --skip-old-files       Don't replace existing files when extracting,
                            silently skip over them
-U, --unlink-first         Delete files that would be overwritten before
extracting
-W, --verify               Attempt to verify the archive after writing

Output streams:

    --ignore-command-error Ignore the exit code of subprocesses
    --no-ignore-command-error
                            Treat non-zero exit code of subprocesses as
error
-O, --to-stdout            Extract files to standard output
    --to-command=COMMAND  Pipe extracted files to another program

File attributes:

    --atime-preserve[=METHOD]
                            Preserve access time on output files, either by
restoring timestamps after reading (default METHOD='replace') or not setting
them at all on the first (METHOD='system') setting
    --delay-directory-restore
```

```
                            Set modification time and directory permissions
only after extraction is complete
    --group=NAME          Force NAME to be the group owner of the files
being added
    --mode=CHANGES         Force the (symbolic) mode of the files being
added to be CHANGES
    --mtime=DATE-OR-FILE   Set the mtime of the added files from DATE-OR-
FILE
-m, --touch                Do not extract modification time of files
    --no-delay-directory-restore
                            Undo the effect of --delay-directory-restore
option
    --no-same-owner
                            Extract files as yourself (default for regular
users)
    --no-same-permissions
                            Use the user's umask when extracting permissions
from the archive (default for regular users)--numeric-owner Always use
numeric IDs for user/group names
    --owner=NAME Force NAME to be the owner of the added files

-p, --preserve-permissions, --same-permissions Preserve file permissions
during extraction (default for super user only)
    --preserve Same as -p and -s
    --same-owner Try to preserve owner relationship during extraction
(default for super user)

-s, --preserve-order, --same-order member arguments are listed in the same
order as the files in the archive

Handling of extended file attributes:

    --acls Enable POSIX ACLs support
    --no-acls Disable POSIX ACLs support
    --no-selinux Disable SELinux context support
    --no-xattrs Disable extended attributes support
    --selinux Enable SELinux context support
    --xattrs Enable extended attributes support
    --xattrs-exclude=MASK Specify the exclude pattern for xattr keys
    --xattrs-include=MASK Specify the include pattern for xattr keys

Device selection and switching:

-f, --file=ARCHIVE Use archive file or device ARCHIVE
    --force-local Treat the archive file as a local archive even if it has a
copy

-F, --info-script=NAME, --new-volume-script=NAME Run script at end of each
tape volume (implied -M)

-L, --tape-length=NUMBER Change tape after writing NUMBER x 1024 bytes

-M, --multi-volume Create/list/extract multiple-volume archive files
    --rmt-command=COMMAND Use the specified rmt COMMAND instead of rmt
    --rsh-command=COMMAND Use the specified remote COMMAND instead of rsh
    --volno-file=FILE Use/update volume number in FILE

Device blocking:
```

```
-b, --blocking-factor=BLOCKS Number of records per BLOCKS × 512 bytes
-B, --read-full-records Reblock as reading (only for 4.2BSD pipes)
-i, --ignore-zeros Ignore zero-byte blocks in the archive (end of file)
    --record-size=NUMBER Number of bytes per record NUMBER × 512

Selecting archive format:

-H, --format=FORMAT Create archive in the specified FORMAT

FORMAT can be one of the following:

    gnu                      GNU tar 1.13.x format
    oldgnu                   GNU format as per tar <= 1.12
    pax                      POSIX 1003.1-2001 (pax) format
    posix                    equivalent to pax
    ustar                    POSIX 1003.1-1988 (ustar) format
    v7                       old V7 tar format

    --old-archive, --portability
                                equivalent to --format=v7
    --pax-option=keyword[[:]=value][,keyword[[:]=value]]...
                                control pax keywords
    --posix                equivalent to --format=posix

-V, --label=TEXT           create archive with volume name TEXT
                            use TEXT as a pattern for list/extract

Compression options:

-a, --auto-compress        use archive suffix to determine the compression
program
-I, --use-compress-program=PROG
                             filter through program PROG (must accept -d
option)
-j, --bzip2                filter the archive through bzip2
-J, --xz                   filter the archive through xz
    --lzip                 filter the archive through lzip
    --lzma                 filter the archive through lzma
    --lzop
    --no-auto-compress     do not use archive suffix to determine the
compression program
-z, --gzip, --gunzip, --ungzip   filter the archive through gzip
-Z, --compress, --uncompress   filter the archive through compress

Local file selection:

    --add-file=FILE        add specified FILE to archive (useful if name
begins with -)
    --backup[=CONTROL]     backup before removal, choose version CONTROL
-C, --directory=DIR        change to directory DIR
    --exclude=PATTERN      exclude files that match PATTERN
    --exclude-backups      exclude backups and lock files
    --exclude-caches       exclude contents of directories containing
CACHEDIR.TAG, except the tag file itself
    --exclude-caches-all   exclude directories containing CACHEDIR.TAG
    --exclude-caches-under exclude all contents of directories containing
CACHEDIR.TAG
```

```
    --exclude-tag=FILE     exclude contents of directories containing FILE,
except the file itself
    --exclude-tag-all=FILE exclude directories containing FILE
    --exclude-tag-under=FILE   exclude all contents of directories
containing FILE
    --exclude-vcs          exclude version control system directories

-h, --dereference           follow symlinks; archive and output files they
point to
    --hard-dereference      Track hard links; archive and output files they
point to

-K, --starting-file=MEMBER-NAME
                            Begin at member MEMBER-NAME when reading the
                            archive
    --newer-mtime=DATE     Compare data and time only when data has changed
    --no-null              Disable the effect of previous --null option
    --no-recursion         Avoid automatic descent in directories
    --no-unquote           Do not quote file names read with -T as ending
quotes
    --null                 Read null-terminated names from -T, disable -C
-N, --newer=DATE-OR-FILE, --after-date=DATE-OR-FILE
                            Only save files that are newer than DATE-OR-FILE
    --one-file-system      Create archive while staying on local file system
-P, --absolute-names       Do not strip leading '/' from file names
    --recursion            Directory recursion (default)
    --suffix=STRING        Backup before removal, overwrite common suffix
('')
                            unless overridden by SIMPLE_BACKUP_SUFFIX
-T, --files-from=FILE      Extract or create files using names from FILE
    --unquote              Use file names read with -T as ending quotes
(default)
-X, --exclude-from=FILE    Exclude patterns listed in FILE

 File name transformations:

    --strip-components=NUMBER   Remove NUMBER leading components from file
names when extracting
    --transform=EXPRESSION, --xform=EXPRESSION
                            Use sed expression instead of EXPRESSION
                            for transforming file names

 File name matching options (affect both exclude and include patterns):

    --anchored             Pattern matches file name at the start
    --ignore-case          Ignore case when matching
    --no-anchored          Pattern matches anything after '/'
                            (default for exclusion)
    --no-ignore-case       Case-sensitive matching (default)
    --no-wildcards         Match strings literally
    --no-wildcards-match-slash   Wildcards do not match '/'
    --wildcards            Use wildcards (default)
    --wildcards-match-slash   Wildcards match '/'

 Informative output:

    --checkpoint[=NUMBER]  Show progress information every NUMBER records
                            (default: 10)
    --checkpoint-action=ACTION   Execute ACTION at each checkpoint--full-
time                    print file time to its full resolution
    --index-file=FILE             send detailed output to FILE
-l, --check-links              print information as long as not all links
are output
    --no-quote-chars=STRING        disable character quoting from STRING
    --quote-chars=STRING           additional quoting characters from STRING
    --quoting-style=STYLE          set the name quoting style; valid STYLE
values are listed below
-R, --block-number             display the number of blocks in each message
of the archive
    --show-defaults                display tar default options
    --show-omitted-dirs            list directories that don't match the
search criteria when listing or extracting
    --show-transformed-names, --show-stored-names     display transformed
file or archive names
    --totals[=SIGNAL]              print total byte count after processing
the archive; when this SIGNAL is triggered with an argument of '-', print
total byte count; allowed signals are: SIGHUP, SIGQUIT, SIGINT, SIGUSR1, and
SIGUSR2; signals without the 'SIG' prefix are also accepted
    --utc                          print file modification time in UTC
format
-v, --verbose                  list files being processed in detail
    --warning=KEYWORD              warning control:
-w, --interactive, --confirmation      require confirmation for each
operation

Compatibility options:

-o                         equivalent to --old-archive when creating an
archive; equivalent to --no-same-owner when extracting an archive

Other options:

-?, --help                 display this help list
    --restrict                 disable certain potentially dangerous options
    --usage                    display a brief usage message
    --version                  print program version

Long options and their corresponding short options have the same required or
optional arguments.

Unless a backup suffix is set with --suffix or SIMPLE_BACKUP_SUFFIX, the
backup suffix is "~".
Backup can be set to a version control using --backup or VERSION_CONTROL,
the possible values are:

none, off        never make backups
t, numbered      make numbered backups
nil, existing    make numbered backups
if numbered backups already exist, otherwise make simple backups
never, simple    always make simple backups

--quoting-style option has the following valid parameters:

  literal
  shell
  shell-always
  c
  c-maybe
  escape
  locale
  clocale

The default value for this tar command is:
--format=gnu -f- -b20 --quoting-style=escape --rmt-command=/etc/rmt
--rsh-command=/usr/bin/ssh
```

Common commands

Compress file without packaging:

```
touch a.c
tar -czvf test.tar.gz a.c   //Compress file a.c to test.tar.gz
```

- z: with gzip attribute
- j: with bz2 attribute
- Z: with compress attribute
- v: display all processes
- O: extract files to standard output

List the content of compressed file:

```
tar -tzvf test.tar.gz
-rw-r--r-- root/root     0 2010-05-24 16:51:59 a.c
```

Extract files:

```
tar -xzvf test.tar.gz
a.c
```

18. top

The **top command** allows you to dynamically view the overall operation of the system in real time. It is a practical tool that combines multiple information monitoring system performance and operational information. It provides a dynamic and interactive real-time view that displays overall performance information of the system and relevant information about running processes.

Grammar Explanation

```
top -hv | -bcEHiOSs1 -d secs -n max -u|U user -p pid(s) -o field -w [cols]
```

Option Explanation

- `-b`: Run in batch mode and directly output the results to a file.
- `-c`: Display the complete command line without truncation.
- `-d <seconds>`: Screen refresh interval time.
- `-I`: Ignore idle processes.
- `-s`: Secure mode.

    - `-S`: Accumulatively display the CPU usage time of processes.
    - `-i`: Do not display idle or useless processes.
    - `-u <username>`: Specify the username.
    - `-p <pid>`: Specify the process.
    - `-n <number>`: Number of times to loop display.
    - `-H`: Display resource occupation of all threads.

- ## top Interactive Commands

  Some interactive commands that can be used during the execution of the top command. These commands are single-letter, and if the -s option is used in the command line, some commands may be blocked.

    - `h`: Display the help screen, providing a brief summary of commands.
    - `k`: Terminate a process.
    - `i`: Ignore idle and zombie processes (toggle command).
    - `q`: Quit the program.
    - `r`: Reschedule the priority of a process.
    - `S`: Switch to cumulative mode.
    - `s`: Change the delay time between two refreshes, in seconds (if decimals are used, in milliseconds). Entering a value of 0 will continuously refresh the system. The default value is 5 seconds.
    - `f` or `F`: Add or remove items from the current display.
    - `o` or `O`: Change the order of displayed items.
    - `l`: Switch to display average load and boot time information.
    - `m`: Switch to display memory information.
    - `t`: Switch to display process and CPU status information.
    - `c`: Switch to display command names and complete command lines.
    - `M`: Sort by resident memory size.
    - `P`: Sort by CPU usage percentage.
    - `T`: Sort by time/accumulative time.
    - `w`: Write current settings to the ~/.toprc file.

- ## Display Information

  ```
  top - 14:55:57 up  1:03,  2 users,  load average: 0.62, 0.55, 0.45
  Tasks: 158 total,   1 running, 157 sleeping,   0 stopped,   0 zombie
  %Cpu(s):  3.7 us,  9.3 sy,  0.0 ni, 86.0 id,  0.0 wa,  0.0 hi,  0.9 si,  0.0
  st
  MiB Mem :   1982.2 total,    778.4 free,    645.6 used,    558.1 buff/cache
  MiB Swap:      0.0 total,      0.0 free,      0.0 used.   1307.2 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+
  COMMAND
     4496 root      20   0   14660   3092   2628 R  23.8   0.2   0:00.13 top
     3032 root      20   0       0      0      0 S  14.3   0.0   0:26.43
  RTW_CMD_THREAD
        1 root      20   0  167580  10200   7216 S   0.0   0.5   0:03.09
  systemd
        2 root      20   0       0      0      0 S   0.0   0.0   0:00.00
  kthreadd
        4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00
  kworker/0:0H
        6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00
  mm_percpu_wq
  ```

```
      7 root      20   0       0      0      0 S   0.0   0.0   0:00.13
ksoftirqd/0
```

System Information:

  - uptime: The system's uptime and average load.
  - tasks: The number of running processes and sleeping processes.
  - CPU: The overall CPU usage and usage of each core.
  - Memory: The overall memory usage, free memory, and memory used for buffering and caching.

Process Information:

  - PID: The process identifier.
  - USER: The username of the running process.
  - PR (Priority): The priority of the process.
  - NI (Nice Value): The priority adjustment value of the process.
  - VIRT (Virtual Memory): The size of the process's virtual memory.
  - RES (Resident Memory): The amount of physical memory actually used by the process.
  - SHR (Shared Memory): The amount of memory shared by the process.
  - %CPU: The percentage of CPU usage by the process.
  - %MEM: The percentage of memory usage by the process.
  - TIME+: The cumulative CPU time of the process.

Features and Interactions:

  - Key Commands: When running top, certain key commands can be used for operations, such as pressing "k" to terminate a process and pressing "h" to display help information.
  - Sorting: Processes can be sorted by CPU usage, memory usage, process ID, etc.
  - Refresh Rate: The refresh rate of top can be set to dynamically view system information.

- ## Common Commands

Display Process Information

```
top
```

Show the complete command

```
top -c
```

Display program information in batch mode

```
top -b
```

Display program information in cumulative mode

```
top -S
```

Set the number of information updates

```
top -n 2   # means stop updating display after 2 updates
```

Set the information update interval

```
top -d 3  # means update every 3 seconds
```

Display specific process information

```
top -p 139  # display process information for process number 139, including CPU and memory usage
```

Exit after 10 updates

```
top -n 10
```

Users will not be able to use interactive commands to control processes

```
top -s
```

19. zip

The **zip command** can be used to decompress files or perform packaging operations on files. zip is a widely used compression program, and files compressed with it will produce a compressed file with the extension `.zip`.

- ### Grammar Explanation

  ```
  zip [-options] [-b path] [-t mmddyyyy] [-n suffixes] [zipfile list] [-xi list]
  ```

  - **zipfile list**: Specifies the zip file to be created.
  - File list: Specifies the list of files to be compressed.

- ### Option Explanation

  - `-0`: Only store data without compression.
  - `-1`: Faster compression speed.
  - `-9`: Better compression quality.
  - `-'compression efficiency'`: Compression efficiency is a value between 1-9.
  - `-@`: Reads file names from standard input.
  - `-A`: Adjusts executable self-extracting files.
  - `-b path`: Specifies the temporary directory for storing files.
  - `-c`: Adds a comment to each compressed file.
  - `-d`: Deletes entries in the compressed file.
  - `-D`: Does not add directory entries.
  - `-e`: Encrypts.
  - `-f`: This option is similar to specifying "-u", but not only updates existing files. If some files do not exist in the compressed file, this option will add them to the compressed file.
  - `-F`: Attempts to repair damaged compressed files (-FF attempts a stricter repair).
  - `-h2`: Displays more help.
  - `-i`: Only compresses files that meet the conditions.
  - `-j`: Only saves file names and their contents without storing any directory names.

  - `-J`: Does not record the compression file prefix (used for self-extracting files).
  - `-l`: Converts LF to CR LF (-ll converts CR LF to LF).
  - `-m`: Moves to the compressed file (deletes the operating system file).
  - `-n`: Does not compress files with these suffixes.
  - `-o`: Makes the compressed file have the same date as the newest entry.
  - `-q`: Runs in quiet mode.
  - `-r`: Recursively into directories.
  - `-s`: Changes the delay time between two refreshes, in seconds (if there are decimals, in milliseconds). Entering a value of 0 means the system will refresh continuously, the default value is 5 seconds.
  - `-T`: Tests the integrity of the compressed file.
  - `-u`: Updates (only includes changed or new files).
  - `-v`: Detailed operation, prints version information.
  - `-X`: Excludes additional file attributes.
  - `-y`: Directly saves symbolic links instead of the files they point to. This parameter is only effective on systems like UNIX.
  - `-z`: Adds a comment to the compressed file.

- ## Common Commands

  Pack all files and folders under the `/app` directory into a `app.zip` file in the current directory:

  ```
  zip -q -r app.zip /app
  ```

  If we are in the `/app` directory, we can execute the following command:

  ```
  zip -q -r app.zip *
  ```

  Delete the file `a.c` from the compressed file `cp.zip`:

  ```
  zip -dv cp.zip a.c
  ```