

# Command line operation

Be familiar with working in command line interface has much significance for using and managing Linux operation system. This chapter will introduce knowledge of carry out shell operation in Asianux 2.0.

## 1.1 Basic knowledge

The following knowledge about Linux shell and directory is the basic of studying this chapter.

### 1.1.1 File naming

The maximum length of file name in Linux is 256 characters. Usually it composes of letter, number, “.” (point), “\_” (underline) and “-” (subtraction sign). It can not contain “/” in file name, because that “/” means the list separator of root directory or path in Linux directory tree (like “\” in DOS).

Linux system can support asterisk wildcard in file name. Detail as follows:

Asterisk (\*): match zero or several characters;

Interrogation mark (?): match any character;

[ab1 A-F]: amtch any character that listed in muster. In this example, this muster a, b, 1 or any majuscule character between A and F;

### 1.1.2 Path

The passed path when system looks for files is called path name. It can use filename directly when using files under current directory; if you want to use files under other directories, you must point out this file is in which directory.

There have two kinds of path according to different beginning of searching file: absolute path and relative path. The path that starts from root directory is called absolute path, and path that starts from current directory is called relative path, relative path will change with working directory of user.

Same with DOS, there has “.” file which delegates current directory and “..” file which delegates parent directory of current directory in every directory. Relative path generally

starts from “..”.



*In Linux directory tree, the symbol which delegates root directory or list separator is “/”.*

### 1.1.3 File type

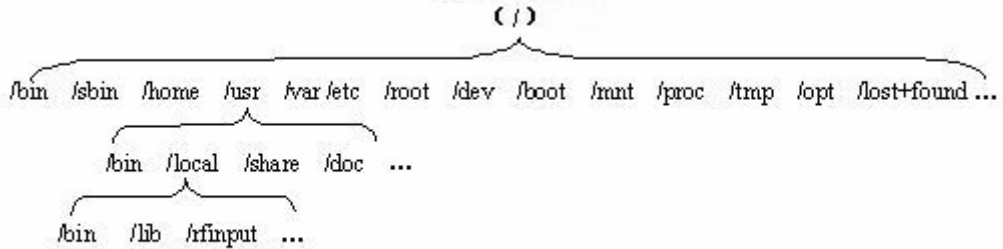
Asianux 2.0 system can support the following file types: common file, directory file, device file and symbol links file.

<b>Common file</b>	Includes text file, data file, executable binary system program etc.
<b>Directory file</b>	Simply called directory, Linux regards directory as a especial file, use it to form hierarchical tree structure of file system. Every directory file at least contains two files, “..” means superior directory, “.” means this directory itself.
<b>Device file</b>	Device file is a kind of particular file, Linux uses them to identify each device driver, and kernel uses them to communicate with hard devices. There are two class of particular device file: character device and block device.
<b>Symbol links</b>	A special file, the data they stores is the path that leads to some file in file system. When using symbol links file, system will visit this stored path automatically.

### 1.1.4 Directory structure

Through the knowledge of system directory structure, we can conveniently know where is the file we want when carrying out file operation and system management.

File system of Asianux 2.0 adopt hierarchical tree structure. Namely: a root directory (usually use “/”) contains several sub directories or files; sub-directory also can contain sub directories or file information, this way it can form an inverted tree. “Root” and “Branch” of tree delegate directory or folder, and “Leaf” delegates file. As it shown in the following figure.



Linux directory tree structure

The following table lists out main system directory and its simple description:

/bin:	Store command file that can be used by common user. Directory /usr/bin also can be used to store user commands.
/sbin:	Generally, it stores commands that used by non-common user (sometime common user may also use). Directory /usr/sbin also contains many system commands.
/etc:	Configuring file of system.
/root:	Main directory of administrator (root or super user).
/usr:	Includes files and directory that directly corresponds with system user, some main application can also be stored under this directory.
/home:	The location of user's main directory, it stores user file (configuring file, document and data of user itself).
/dev:	Device file. Device is expressed by file form in Linux, it can simply operate device according to the mode which operated on file.
/mnt:	File system mount point. It is generally used to install mobile medium, division of other file system (such as DOS), network shared file system or any Installable File System (IFS).
/lib:	Include many shared storeroom file that used by programs in /bin and /sbin. Directory /usr/lib/ includes more storeroom files which are used in user's programs.

- `/boot:` Includes kernel and files that are used when starting up other system.
- `/var:` Includes some files that are changed frequently. For example, illusive spool directory, file log directory, lock file, temporary file etc.
- `/proc:` Memory mapping file system of OS, it is a virtual file system (does not occupy any disk space). When you check them, the information you see is information in memory, these file is helpful to know about inner information of system.
- `/initrd:` Mount directory of inited.img mapping file and load directory of needed device module when starting up computer.
- `/opt:` Store files and programs that can be installed optionally. It is mainly used by the third-party developer to install and uninstall their software packages.
- `/tmp:` Temporary directory of user and program, files in this directory will be cleared up by system automatically.
- `/lost+found` Files that recovered from system repairing process.
- :

### 1.1.5 Brief introduction of Shell

When user works in command line, it does not communicate with OS kernel, command explainer accepts commands, and transmits it to correlative program after analyzing. System will start correlative shell automatically when entering Asianux 2.0 environment, shell is a kind of command line interpreter. It provides interface between user and OS. The default shell if Asianux 2.0 is bash.

The basic format of bash command is as following:

**Command name [option] [parameter1] [parameter2]...**

The content in square brackets is optional to command.

**[option]** : There has some especial definition to command, generally begins with “-”, several options can be connected with “-”, for example, `ls -l -a` is same eith `ls -la`.

**[parameter]** : Provides running information of commands, or used file name during

command running process.



*When inputting user name, password filename and command name, we have to notice majuscule and small letter, because they have different meaning in Linux system.*



*Command, option and parameter must be separated by space. Successive space will be regarded as single space by shell.*

- **Key in commands**

Input correlative commands in shell prompt, press enter button to confirm; shell will read this command and execute. If system can not find command, it will display: “Command not found”, now it needs to examine spell of commands.

Use “;” to separate two command, in this way, we can input several commands in one line. Executing order of commands is same with inputting order.

- **Fill command**

When the input command directory is deep or the file name is long, system will search for matching option in possible command or filename and full fill it automatically as long as you press <TAB>. If there are more than one file match the input character string, press <TAB> twice, system will list all matching filenames.

- **Histry record**

Shell will remember all the commands that have ever been input, you can select former command as long as you press UP and DOWN.

Examples:

clear: Refurbish screen;

date: Display date and time on screen;

echo: Redisplay content in command line to standard output;

cal: Display month and calendar.

## **1.1.6 System help**

Asianux 2.0 has powerful system and network functions, has plenty of applications and large of complex operation commands. In order to help users to operate successfully, system provides all kinds of online help information so that users can inquire at any time.

- **Online manual**

Use online user manual through man command, system can display online help information of any command. It uses command name as parameter, the grammar format of this command is:

**man command**

Usually used Linux system help manual is classified by chapter; it locates under /usr/man.

For example, the following command line will display manual page of cal command:

**\$ man cal**

Use command “**man man**”, it will display usage of man command.

- It can also display reference information of commands by adding **--help** parameter after command that is inquired.
- Use **help command** can list help of many inner commands.
- **whatis** command can lookup brief help information through command name, format of command is: **whatis keyword**.

## **1.2 Directory operating command**

### **1.2.1 Check directory**

The command to check content of directory is ls, it defaulted displays content of current directory, you can give one or several directory names on location of command line parameter, so that you can check these directories. Format of this command is:

**ls [option]...[filename]...**

ls command has several command line options, for example:

- a: List all files, include the file that begins with “.”;
- d: If it is followed by a directory, then use this parameter to output the name of this directory only;

- l: Use long format to display file list, include link quantity, owner, size, last modifying time, right etc.
- t: Sort according to change time but not file name;
- C: Sort filename longitudinally by line;
- F: Add a sign after filename to show file type;
- Cx: Sort filename by raw;
- CF: List filename in directory by line, this command add a character to filename to distinguish types of directory and files;
  - Append a diagonal (/) to filename of directory
  - Append a asterisk (\*) to name of executable file
  - Append a @ to sign link file
  - Do not append any character to name of common file
- CR: Display all files of current directory and all files of sub-directory along directory tree by multi-column, it can also be called recursive list. This command can distinguish directory and executable files, this is, append a character to filename.

## 1.2.2 Change work directory

Enter a directory, or we can say change current work directory by using cd command, the grammar format of this command is:

**cd directory name**

cd command has unique parameter, that is path name that means objective directory (relative path name and absolute path name)

Use “” to move work directory to upper directory: cd ..

In order to return to home directory of user from any place of system, we can use cd command that does not contain any parameter.

## 1.2.3 Establish directory

Use mkdir command to establish one or several directories, so that can organize files effectively. Its grammar format is:

**mkdir [option] directory name [directory name...]**

Same sub-directory should contain similar files. For example, it should set up a sub-directory that contains all database files; another sub-directory contains Excel file; and another one is

used to store correlative file of certain directory.

**-p** option: establish directory and its sub-directory at one time.

**mkdir -p directory name/sub-directory name**

### 1.2.4 Delete directory

If directory is not used any more, or disk space reaches the limit, it needs to delete useless directory from file system.

Use rmdir command to delete one or several sub-directories from directory, its grammar format is as following:

**rmdir [option] directory name [directory name...]**

Sub-directory must be blank directory before deleting. That is to say, all files in this directory must be clean away. If there have other files in this directory, then it can not use rmdir command to delete it.

Current work directory must be above the deleting directory, it can not be the directory itself which is to be deleted, and also can not be the sub-directory of this directory.

**-p** option: delete appointed directory and its sub-directory recursively. That is: if the appointed directory has sub-directory, delete its sub-directory firstly, and then delete this directory.

**-r** option: delete all files in this directory and delete this directory itself recursively. For detailed introduction of deleting file commands, please refer to 3.5 setion.

### 1.2.5 Display current directory

When operating, mostly we can not remember current directory, command pwd can display the location of user in directory tree. For example:

```
# pwd
```

```
# /usr/local/rfinput/bin
```

The information that system gives means that the current directory of user is /usr/local/rfinput/bin

## 1.3 File operating command

### 1.3.1 Display text file

Text file is composed of printable character and control character (such as tabs and Wrap Prompt). Several commands can display text file.

- **cat command**

General grammar of cat command is:

**cat [option] filename [filename...]**

After running this command, the content of appointed file will display on standard output (generally screen). If the content of file is very long, and one screen can not display all the content, it will appears scroll screen, in order to control scroll screen, you can press <Ctrl+S> to stop scroll; press <Ctrl+Q> to recover scroll.

Meanings of option are as follows:

- v: use a especial form to display controlling character, eliminate LFO and TAB
- n: display serial number of output row
- b: display serial number of non-vacant row

- **head command**

Its grammar is:

**head [display number of rows] filename [filename...]**

head command displays several forefront rows of appointed file on screen, amount is decided by “display number of rows”, the default is 10.

- **tail command**

Its grammar is:

**tail [display number of rows] filename [filename...]**

**tail [+n] filename [filename...]**

Display the last several rows of appointed file on screen, amount of rows is decided by

“**display number of rows**”; or start to display from appointed row, to the last of this file.

- **more command**

more command displays content of file, and it displays one screen every time. Its grammar is:

**more [option] filename [filename]**

It can appear hint information at the bottom of the screen to give the percentage of displayed file.

It can use several different methods to answer hint:

——Press <**Space**> to display the content of next screen.

——Press <**Enter**> , it will only display content of the next row.

——Press </>, and input a mode, it can search for the next matching mode in text.

——Press <h>, it will display help screen, there has correlative help information on this screen.

——Press <b>, it will display content of the previous screen.

——Press <q> to quit more command.

### 1.3.2 Establish new file

It can use command and application to establish new file, such as text editor, it is used specially to put useful data to file. However, sometime it may only need file that only has filename, namely blank file.

Linux system provides touch command to establish new blank file. Its grammar is:

**touch filename [filename...]**

The filename that does not exist will be established as blank file. The time log of existed file will be updated to current time (defaulted mode); their data will be hold

### 1.3.3 Copy file

Use cp command to back up files, or make personal backup of other user's file.

It can use cp command to copy a source file to an destination, or copy a serials of file to an objective directory, its grammar is:

**cp source file destination**

**cp source file1 [source file2...] destination**

In first grammar format, source file will be copy to destination.

——if destination is a directory file, then it will copy source file to this directory, and the filename will be hold;

——if destination is not a directory file, then source file will be copied to this destination, and the content of the later file will be destroyed, but filename will be hold.

In the second grammar format, all the source files will be copied to destination ---- this destination must be directory file, the name of all files will be hold.



*cp command copy a file, and the source file will be hold. If copying a file to an existed destination, then, the content of destination will be destroyed!*

### 1.3.4 Move and rename file

mv command is used to move file or rename file. The grammar of this command is:

**mv source file destination**

**mv source file1 [source file2...] destination**

In first usage, there have two different results when source file is moved to destination:

——if destination is the path of certain file, source file will be move to this directory, and the filename will be hold;

——if destination is not directory file, then the name of source file will be changed to this destinationname, and cover the existed file that has same name;

In the second usage, all the source files will be moved to destination, the destination here must be directory file. All the files that are moved to this directory will hold their names.



*If a file is moved to an existed destination, content of the destination will be covered!*

If source file and destination are under same directory, the function of mv is to rename file, for example:

**mv oldname newname**

### 1.3.5 Delete file

Use `rm` command to delete unnecessary files and directories. The grammar of this command is:

```
rm [option] filename1 [filename2...]
```

Before deleting file, you'd better look over content of file to make sure whether you really want to delete it.

**-i** option: this option is especially useful when using extended name character of file to delete multiple files. This option will ask user for confirmation one by one, you must input "y" or "Y", the file can only be deleted after you press <Enter>. If you only press <Enter> or other character, the file will not be deleted.

**-r** option: it can be used to delete directory. When a directory is deleted, all files and sub-directories will be deleted. *Note, it is a very dangerous command option!*

### 1.3.6 File link

Asianux has a function that give a file several names, it is called link. In this way, is you modify one file, it can accomplish modification of all correlative link files.

`ln` command is used to establish link, its grammar is:

```
ln source file destination
```

```
ln source file1 [source file2...] destination
```

In the first grammar format, if destination is path to certain directory file, source file will be linked to this directory, filename will be hold; if destination is not a path to certain directory file, source file will be linked to this destination, and cover the existed file that has same filename.

In the second grammar format, all the source files will be linked to destination ---- this destination must be directory file. Name of all the files will be hold.

File link have two forms, they are hard link and symbol link.

- **Hard link**

In default situation, `ln` command establishes hard link.

The quantity of hard links of a file can be read from the second line of long list format of

directory, the quantity of file link that has no additional link is 1. In command will increase link quantity, rm command will decrease link quantity. A file, unless its link quantity is 0, will not be deleted physically from system.

To hard link, there has following limit: it can not make hard link to directory file; can not make hard link between different file systems.

- **Symbol link**

Symbol link can also be called soft link, it links a path name to a file, in fact, it is just a text file, and it contains path name of another file that it provides link to it. Another file is a file actually contains all data. All commands that read and write file content, when they are used on symbol link, will visit actual file along link direction.

If add s option to ln command, then it will establish symbol link, for example:

```
ln -s source destination
```

Symbol link does not have limit like hard link, it can make symbol link to directory file, and it can also make symbol link between different file systems.

### **1.3.7 Comparison of file content**

- **Compare text file**

diff command is used to compare text files, and display differences between two files. Its general format is:

```
diff file1 file2...
```

If two file are totally same, then it will not display any output. If there has difference, it will display difference of two files by section.

- **Compare data file**

cmp command compares any two files that contain text or data. Its general grammar is:

```
cmp file1 file2
```

Because that binary data can not be displayed on screen, cmp command just simply reports from which byte that it appear difference.

### **1.3.8 Find file**

- **find command**

find command is used to search for file and location of directory. The grammar of this command is:

**find path naem [option]**

Generally used options are:

**-print:** ouput the result of searching. If it did not appint any option, the default is `-print`.

For example, the basal usage of find command is to list all the files and sub-directories of appointed directory:

**# find /usr -print**

**-name:** Search according to filename.

**-size:** Search according to file size.

For example, the following command will search for files that their size are over 100K under directory /usr:

**# find /usr -size 100k**

**-user:** Search according to file owner.

**-type:** Search according to file type. Familiar types are:

b	block special file
c	character speacial file
f	common file
l	symbol link file
d	directory file

- **locate command**

locate is a convenient and fast searching command of files and directories. The grammar of this command is:

**locate filename [option]**

The precondition of using locate command is to establish a slocate database which is used to locate file or directory, and this database should be updated momently, in this way, it can ensure the veracity of locate command's searching results.

The following command is used to establish slocate database from /:

```
# locate -u
```

It can start to search for file after establishing database. For example, in order to search for all files about telnet command, we can use:

```
# locate telnet
```

locate command will examine all files and directories that match telnet in its database and display the result on screen.

The command to update slocate database is updatedb, it needs to use root identity to execute this command.

In general situation, administrator will set to use cron program to update database automatically in evening. cron is a background guardian process, it execute planned task termly.

### **1.3.9 Search for text in file**

grep command is used to find word or phrase which is appointed mode in text file, and display all rows which contains given character string to standard ouput device. The grammar of grep command is:

```
grep [option] searching mode filename [filename...]
```

In default situation, grep will match upper/lower case in searching mode; if you don't want to match upper/lower case, you can use option -i.

For example, the following command will find the appearing times of character string "hello world" in all files under /etc directory and its sub-directories:

```
# grep 'hello world' /etc/*/*
```

Searching mode may be the unique parameter, if it used shell meta-character in mode, it usually uses single quotes (') to include it.

## **1.4 File popedom operation**

In multi-user OS, consider about security, it needs to add every file and directory visiting popedom, strictly regulate popedom of every user. At the same time, user can give appropriate popedom to own files, so that can avoid other people from modifying and visiting.

### **1.4.1 Change file owner**

Linux distribute a owner for every file, it is called file owner, the control of file is decided by file owner or super user (root). The establisher of file or directory owns especial using right of the established file or directory.

Ownership of file can be chagned, chown command is used to change the ownership of certain file or directory. The grammar format of chown command is:

**chown [option] user or group file1 [file2...]**

User can be username or user ID. File is file list that is to change popedom and separated by space, it can use asterisk wildcard to express filename.

If it has changed ownership of file or directory, original file owner will not have popedom of this file or directory any more.

Administrator usually uses chown command, after copy file to another user's directory, let user own right to use this file.

### **1.4.2 Change user group**

In Linux, every file belongs to same user group. When establishing a file or directory, system will give it a user group relationship, all members in user group can use this file or directory.

The sign of file user group relationship is GID. GID of file can only be modified by file owner or super user (root). chgrp command can change GID of file, its grammar format is:

**chgrp [option] group filename**

group is user group ID. filename is file list that is to changed its group and they must be separated by space, it supports asterisk wildcard.

### **1.4.3 File popedom setting**

Every file and directory in Linux system has visiting permission popedom, it is used to make sure who can and how to visit and operate files and directories.

There are three different types of users which are regulated by visiting popedom:

- file owner (owner)
- group user (group)
- other users that can visit system (others)

Visiting popedom regulates three mode of visiting file or directory:

- read (r)
- write (w)
- executable or searchable (x)

When using `ls -l` or `l` command to display the detailed information of file or directory, the most left line is visiting popedom of file. The meaning of every character is:

- **File visiting popedom**

- Read (r): Only allow appointed user to read its content, but deny any modifying operation on it. It needs read popedom if use content of visited file as input command. Such as: `cat`, `more` etc.
- Write w): Allow appointed user to open and modify file. Such as commands `vi`, `cp` etc.
- Execute (x): Appoint user to execute this file as a program.

- **Directory visiting popedom**

- Read (r): It can list files that are stored under this directory. This popedom allows shell to use extended name character of file to list matching filename.
- Write (w): Allow user to delete or add new file from directory, generally only directory owner has write popedom.
- Execute (x): Allow searching in directory, and use `cd` command to change work directory to this directory.

## 1.4.4 Change popedom of file

### 1.4.4.1 Change popedom by symbol mode

chmod is used to change visiting popedom of file or directory. Users can use it to control visiting popedom of file or directory. Only file owner or super user (root) have right to use chmod to change visiting popedom of file or directory.

The grammar of chmod command is:

**chmod key filename**

key is compose of the following options:

**[who] [operating sign] [mode]**

Operating object who can be any one of the following letter or combination of them:

- u user, owner of file or directory.
  
- g group, means users in same group, all users that have same group ID with file owner.
  
- o others, means other users.
  
- a all, means all users. It is default of system.

Operating symbol can be:

- + Add certain popedom
- Cancel certain popedom
- = Endow given popedom and cancel all the other popedoms (if it has)

Popedom that mode express can be discretional combination of the following letters:

- r readable
- w writable
- x executable

- s When executing file, set owner of process or group ID to be owner of this file.
- t save text of program to swapping device
- u own same popedom with file owner
- g have same popedom with users that in same group with file owner
- o have same popedom with other users

These three parts must be input in order. It can use multiple key, but must use comma to separate them.

#### 1.4.4.2 Change popedom by absolute mode

Generally, it can also use chmod command mix with different types of key to set popedom. Now, use number to delegate different popedoms. The key here contains three (or more than three) numbers, every number means popedom to different types of users.

The meanings of these numbers are:

0 means denying this popedom, 1 means executable popedom, 2 means writable popedom, 4 means readable popedom, and then add them. So, the format of number attribute should be three octal numbers from 0 to 7, its order is (u) (g) (o).

Generally, key appears by three octal numbers, the first number delegates user's popedom, the second number means group popedom, and the third means other user's popedom.

For example, if you want to make file owner of myfile and users of same group have read and write popedoms, but pther user only can read, you can use the following command to appoint popedom:

**chmod 664 myfile**

#### 1.4.5 Defualt popedom

In default situationm system will set the popedom of established common to be -rw-r-r--, it means that file owner can read and write this file (rw), but users in same group and other users only can read; also, in default configuration, it sets popedom of every home directory of user to be drwx-----, that is to say, only file owner can read, write and search for this file (rwx), this means user can not read content in directory of other user.

User can modify the default popedom of newly established file, for example, we can use the

following command:

```
umask u=rwx,g=,o=
```

It will give all popedom to file owner when establishing this file, but users in same group and other users do not have any popedom.

## **1.5 Direction and pipeline**

Execute a shell command generally can open three standard files automatically, they are standard input file (stdin), it usually correspond to terminal keyboard; standard output file (stdout) and standard error output file (stderr), these two files all correspond to terminal screen. Process gets data from standard input file, output normal output data to standard output file, and output error information to standard error file.

The following will take cat command for example, the function of cat command is to read data from file which is given from command line, and send these data directly to standard output. For example, use the following command:

```
$ cat config
```

It will display content of config file to screen in turn. But, if there is no parameter in command line of cat, it will read data from standard input, and send it to standard output. For example:

```
$ cat
```

```
Hello world
```

```
Hello world
```

```
$
```

There are some problems to use standard input/output file directly:

- 1、 When inputting data from standard terminal, the input data can only be used once, you have to input these data again if you want to use them next time; and when input on terminal, it is inconvenient to modify if input incorrectly.
- 2、 The information that is output to terminal screen can only be read but can not be modified. User can not do further process to the output content, such as use output of one command as input of another command to do further process.

In order to solve the above problems, Linux system induct another two mechanisms for transmission of input and output, they are input/output redirection and pipeline.

## 1.5.1 Input redirection

Input redirection means that redirect standard input of command (or executable program) to appointed file. That is to say, input can be not only from keyboard but also appointed files.

For example, command `wc` statistic row number, word number and character number of appointed file. If you only input on command line:

```
$ wc
```

`wc` will wait for user's input, this time shell can not work, all the text that input from keyboard will appear on screen, but there has no result, `wc` can write result on screen only till press <Ctrl+D>.

If give a filename as parameter of `wc` command, `wc` will return row number, word number and character number of this file.

Another method to sent content of `/etc/passwd` content to `wc` command is to redirect input of `wc`. The general format of input redirection is:

```
commanf<filename
```

It can use the following command to redirect the input of `wc` command to `/etc/passwd` file:

```
$ wc < /etc/passwd
```

```
20 23 726
```

Most commands appoint filename of the input file in command-line by using parameters, so input redirection is seldom used. Otherwise, when using a command that do not accept filename as input parameter, or the content to be input exist in a file, we can use input redirection to solve these problems.

## 1.5.2 Output redirection

Output redirection means redirecting standard output or standard error output of command (or executable program) to appointed file. In this way, the output of this command will not display on screen, but be written into appointed file.

Output redirection is more usually used than input redirection. For example, if certain command has many outputs, it can not be display on screen totally, then redirect output into a file, and open this file by using text editor, then we can check output information; if you want to store output of a command, you can also use this method. Furthermore, output redirection

can be used to use output of one command as input of another command.

The commonly format of output redirection is:

**command>filename**

For example:

**\$ ls > directory.out**

**\$ cat directory.out**

ch1.doc ch2.doc ch3.doc chimp config mail/ test/

Save output of ls command as a file named directory.out.



*If the file behind “>” has already existed, then this file will be covered.*

In order to avoid that only content of output redirection of current command can be stored in appointed file in output redirection, shell provides a kind of appending method.

Output appending redirection is similar with output redirection, the difference between them is only that the function of output appending redirection appends result of output to the last of the appointed file, and the content of original file will not be destroyed.

If you want to append output result of a command to the last of appointed file, you can use operating prompt “>>” of appending redirection. The format is:

**command>>filename**

For example:

**\$ ls \*.doc>>directory.out**

**\$ cat directory.out**

ch1.doc ch2.doc ch3.doc chimp config mail/ test/

ch1.doc ch2.doc ch3.doc

Same with standard output redirection of program; error output can also be redirected. Use 2> (or appending symbol 2>>) to express redirection of error output device. Such as the following command:

```
$ ls /usr/tmp 2> err.file
```

We can see the normal output result of program on the screen, but send any error information of program to file err.file for examining in future.

We can also use another output redirection operating symbol (&>) to send standard output and error output to same file. For example:

```
$ ls /usr/tmp &> output.file
```

Using redirection to get commands together, it can implement new functions that single system command can not provide. For example, use the following command list, which is to statistic file number under /usr/bin directory.

```
$ ls /usr/bin > /tmp/dir
```

```
$ wc -w < /tmp/dir
```

```
459
```

### **1.5.3 Pipeline**

There are two methods to use output of a program or command as input of another program or command; one is to hang together two commands or programs through a temporary file, such as the above example that /tmp/dir file links ls and wc commands together; another one is the pipeline function that is provided by Linux, this method is more convenient than the former method.

Pipeline can join a series of commands, this means that the output of the first command will be send to the second command through pipeline as the input of it, the output of the second command will be used as the input of the third command, and so on. The content that is displayed on screen is the output of the last command.

Set up a pipeline by using pipeline prompt “|”. Use pipeline to rewrite the above example:

```
$ ls /usr/bin|wc -w
```

```
1789
```

Another example:

```
$ cat sample.txt|grep “High” |wc -l
```

Pipeline sends output of cat command to grep command. grep command will search for word

“High” in input, the output of grep command will be all the lines that contain word “High”, and this output will be sent to wc command. wc command will compute numbers of rows in input. Assume that the content of sample.txt file is as following:

Things to do today:

Low:Go grocery shopping

High:Return movie

High:Clear level 3 in Alien vs. Predator

Then the result of this pipeline is 2.

## 1.6 Process and task controlling command

To be simple, process is an executing process. In Linux system, when executing any command, it will establish one or multiple processes. That is to say, command is implemented through processes.

From process point of view, we can understand multi-task conception of Linux OS better. For administrator, managing system processes is important part of daily work.

- **Use ps to check status of process in system**

We can use ps command to observe status of process, and it will display status of current process. We can make sure which process is running according to the displayed information, and whether certain process is hanged or meets some problems, how long has the process run, resources that are used by process, relative priority of the process, and PID of the process. This information is useful for users and it’s more important for administrator.

The commonly usage of ps command is:

**ps [option] [arguments]...**

ps command has the following main parameters:

- a: display status of all user processes in system
- f: display tree form directory of process and sub-process
- l: use long list form to display process’s information

- r: only display running processes
- u: display process's information by using user format, give username and beginning time
- pids: display information of process that is appointed with ID

If it does not carry any option, ps command will list PID of every process that is correlative with current shell. The result is as following:

```
PID TTY TIME CMD
596 pts/0 00:00:00 bash
627 pts/0 00:00:00 vi
628 pts/0 00:00:00 ps
```

Among them, the meaning of each field is as follows:

- PID: ID number of process
- TTY: Terminal number that starts this course
- TIME: Report total CPU time that process used
- CMD: Process name which is running

In order to gain an integrated information list of processes, you can use ps command that contains the following options:

### **ps -aux**

Except listing out the above fields, it also lists out using rate of CPU (%CPU), using rate of memory (%MEM), dummy image size (SIZE), resident data stream size (RSS), terminal number (TTY), and status (STAT).

- **top command**

top command is used to read information of computer system, these information contains current system data and status of process and so on. After inputting top command, the output of screen is as following:

```

5:23pm up 1:39, 4 users, load average: 0.46, 0.35, 0.33
66 processes: 64 sleeping, 2 running, 0 zombie, 0 stopped
CPU states: 0.5% user, 1.7% system, 0.0% nice, 97.6% idle
Mem: 126432K av, 118336K used, 8096K free, 144K shrd, 3668K buff
Swap: 265032K av, 85644K used, 179388K free, 21628K cached

```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
834	root	17	0	58552	6364	6144	S	0.7	5.0	3:00	X
1470	root	17	0	1156	1112	884	R	0.7	0.8	0:00	top
1015	root	19	0	3756	3284	3072	S	0.5	2.5	0:25	ksnapshot
1	root	8	0	108	64	64	S	0.0	0.0	0:07	init
2	root	9	0	0	0	0	SH	0.0	0.0	0:00	keventd
3	root	9	0	0	0	0	SH	0.0	0.0	0:00	kapm-idled
4	root	19	19	0	0	0	SHN	0.0	0.0	0:00	ksoftirqd_CPU0
5	root	9	0	0	0	0	SH	0.0	0.0	0:01	kswapd
6	root	9	0	0	0	0	SH	0.0	0.0	0:00	kreclaimd
7	root	9	0	0	0	0	SH	0.0	0.0	0:00	bdfiush
8	root	9	0	0	0	0	SH	0.0	0.0	0:00	kupdated
9	root	-1	-20	0	0	0	SW<	0.0	0.0	0:00	mdrecoveryd
65	root	9	0	864	36	36	S	0.0	0.0	0:00	minilogd
73	root	9	0	0	0	0	SH	0.0	0.0	0:00	khubd
555	root	8	0	64	4	4	S	0.0	0.0	0:00	apmd
623	root	9	0	108	4	4	S	0.0	0.0	0:00	automount
679	root	9	0	88	4	4	S	0.0	0.0	0:00	lpd
738	root	9	0	104	60	60	S	0.0	0.0	0:00	xm

Every line in this figure displays detailed information of system, the meaning of the beginning rows are as follows:

- Uptime:** Display current time, passed time from last starting up, quantity of activated users and CPU's average occupied status in past 1, 5 and 15 minutes.
- Process:** Display all the processes in system, and classify processes according to hanging, running, establishing and stopping.
- CPU states:** Statistic current status of CPU that occupied by users and system
- Mem:** Statistic current occupied status of memory
- Swap:** Statistic current occupied status of swap area

It has displayed list of process in top command, it contains: PID, user, priority, nice parameter, needed memory information (SIZE, RSS, SHARE), status (STAT), CPU occupied percentage, occupied memory information, used time of computer and program transference of each process (COMMAND), and so on. About detailed using information of top command, please check on line help.

- Use kill command to stop process

During running, may be at one moment, some processes in system have some problems, they can not run normally, and they can not quit normally too. This time, we can use kill command to stop executing of process, release system resource occupied by these processes. The format of usually used kill command is:

**kill [-s signal] pid**

**kill -l [signal]**

The meanings of the options and parameters of command are as follows:

**pid** gives PID of the process that is needed to be end, it can use command ps to get PID of process. It can list out many process PIDs in command kill.

**-s signal** is a optional parameter, it is used to send signal to process. In default situation, command kill sends signal to process TERM, and this signal notice process to quit. If process does not accept this signal, it can use parameter -9 to stop process forcedly.

**-l**, this parameter requires kill command to list out all signals that it can send to processes.

- **Use at to arrange task**

at command can implement that run tasks you've arranged in appointed time. Common usage of at command is as follows:

**at [option] time [date]**

**at hh:mm:** Use appointed hour (hh) and minute (mm) (24 hour) to arrange tasks;

**at hh:mm month day year:** Use appointed year, month, day, hour (hh) and minute (mm) to arrange tasks;

**at -l:** List out arranged tasks;

**at now +count time\_units:** The time that task runs is arranged in the time that current time adds count time units. Unit of time (time\_units) can be minute, hour, day or week;

**at -d job\_id:** Cancel tasks that have same task id with job\_id;

Commands that are attempred by at command is command list that input after at command-line. Command list of at can be got from standard input (stdin). If standard input

comes from keyboard, you should key in <Ctrl+D> after inputting commands to express end of inputting.

## 1.7 Basic network commands

Asianux 2.0 has powerful network functions; it provides abundant of network applications, and totally supports TCP/IP protocols. Under network environment, it can carry out long-distance login, long-distance command transferring and transmit files and so on.

### 1.7.1 telnet command

telnet command is a long-distance login tool in Linux, it only needs legal registered name and password if you want to use local machine to visit long-distance computers. telnet can also allow user to log in their own computer from long-distance net point, so as to accomplish operations such as checking e-mail, editing files and running programs through network or telephone line. But telnet can only work under character terminal mode, but can not work under graphical user interface.

The basic usage of telnet is:

**telnet [option] IP address/host name**

After key in commands, telnet will start up a long-distance dialogue, the option parameters that can be used by this command mainly are:

- d** Start up debugging function
- a** Register automatically
- n tracefile** Open tracking program, and save data of tracking program into tracefile
- e escape\_char** Set ESC of dialogue to be escape\_char
- l user** Send username to long-distance system in order to register automatically. This parameter automatically contains -a parameter.
- port** Point out port number that connects to long-distance system, if do not appoint, it will connect to default port

After successfully connecting to long-distance computer, telnet will display login information, and notice user to input registered name and password. If it registered successfully, it can

start to work.

After using telnet, it needs to exit login and return to local shell command prompt.

## 1.7.2 ftp command

FTP (File Transfer Protocol) is a simple and effective method to transmit files between TCP/IP network computers. The function of ftp command is to transmit files between local computer and long-distance computer. It allows user to transmit ASCII files and binary files. During ftp dialogue, users can connect to another computer through using ftp client program. User can move up and down on directories, list out directory, copy files from long-distance computer to local computer and transmit files from local computer to long-distance system. Of course, the precondition is that you must have popedoms of operation between local and long-distance systems.

The basic format of ftp command is:

**ftp [option] IPaddress/host name**

It can use help command to get commands list that can be used, and it can also appoint idiographic command name after help command to get explanation of this command.

<b>ls</b>	List out current directories of long-distance computer
<b>cd</b>	Change working directory on long-distance computer
<b>lcd</b>	Change working directory on local computer
<b>ascii</b>	Set file transmitting mode to be ASCII mode
<b>binary</b>	Set file transmitting mode to be binary system mode
<b>close</b>	End current ftp dialogue
<b>hash</b>	Display a # every time after finishing to data in data buffer
<b>get (mget)</b>	Transmit appointed files from long-distance computer to local computer
<b>put (mput)</b>	Transmit appointed files from local computer to long-distance computer
<b>open</b>	Connect to long-distance ftp web site
<b>quit</b>	Cut off connection to long-distance and exit from ftp

- ?            Display local help information
- !            Go to Shell

Along with developing of Internet, websites that provide information resources usually can not set up account for every user who wants to use FTP, thus it appears a kind of anonymous FTP mechanism: users can use anonymous user name, use their own email address as password to visit most shared information resources. But because of some security reasons, resources that can be visited by anonymous FTP are limit, and furthermore, some web sites do not provide this kind of service.

### 1.7.3 ping commad

ping command is used to make sure whether host computers on network can reach and has reached speed. The format of ping command is:

**ping [option] IP address/host name**

ping command sends packages that have fixed size to opposites side, and require the other side to return it. When stopping ping command, it will display some statistic data. Judge whether it returns and returning time through data, users can make sure whether the other side is reachable, whether it's turned on, and delay time of network. If you want to exit, please press <Ctrl+C> to intermit.

### 1.7.4 finger command

Use finger command to inquire information of system user, the format of this command is:

**finger [option] username@host name**

After running finger command, it will display certain user's information such as username, host directory, stagnancy time, logging time, logging shell and so on. When inquire user information on long-distance computer, it needs to add "@host name" after username.