

*Linux*环境及开发工具应用实践

-系统操作及管理

zhaofang@email.buptsse.cn



软件学院 赵方

目录

1. 系统登录及文件系统

2. **Linux**常用命令

3. 文件及目录操作

4. **Linux**基本管理命令

登录



redhat

[Session](#) [Language](#) [System](#)



G · N · O · M · E

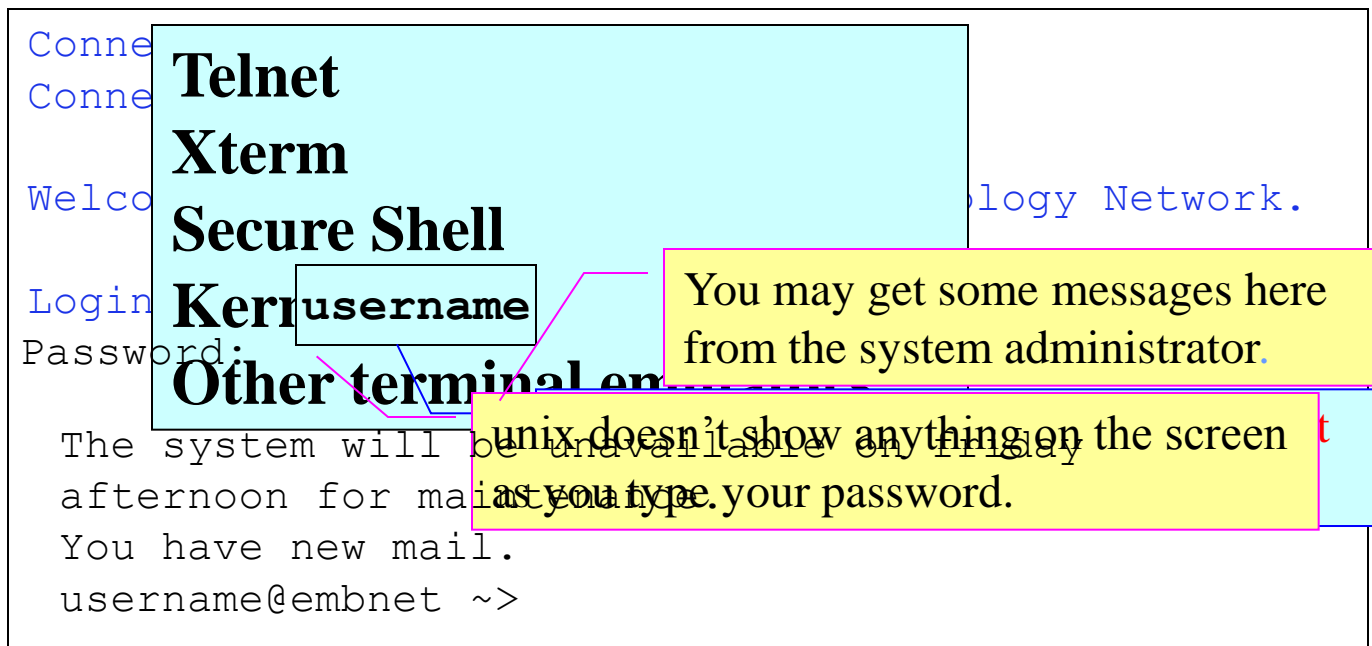
Welcome to liming

Login:

Please enter your login

登录模式

Connect to the unix machine using a suitable program on your local machine.



登出

You can use one of the keys:

- 1)** exit
- 2)** Ctrl+d
- 3)** logout

question: Will you interrupt the system?

Shell命令提示

Several different shells but they behave more or less the same

1. The Prompt.

```
username@embnet ~>
```

The prompt can be your username, the machine you are logged in to, and the present location.

Shell命令行格式

2. Commands

```
username@embnet ~> ls -ald *.txt
```

The shell breaks the command up into individual words

The first word is a command

By default the boundary between words is a space.

To get the shell to treat a phrase that includes spaces as a single word, put it in

Options control how the program runs.

'-a -l -d' is equivalent to '-ald'

them to the list of arguments

如何获取帮助

You can get help on a command by using the command

`'man command'`

If you do not know what a command is called, use the option `'-k'` to get a list of commands that may be relevant

`'man -k word'`

Try using the options `'-f'`, `'-help'`, or `'--help'` if you can't find the man page. This will find all manual pages containing word in the short description of the command.

如何获取帮助



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ man cal

CAL<1>                                System Reference Manual                                CAL<1>

NAME
    cal - displays a calendar

SYNOPSIS
    cal [-m jy13] [month [year]]

DESCRIPTION
    Cal displays a simple calendar.  If arguments are not specified, the current month is displayed.  The options are as follows:

    -1      Display single month output (use if cal was built with -3 as default to get older traditional output)

    -3      Display prev/current/next month output (use if cal was built with traditional -1 as default to get newer improved output)

    -m      Display Monday as the first day of the week.  (The default is Sunday.)

    -j      Display Julian dates (days one-based, numbered from January 1).

:
```

文件名命名规则

Filenames can contain any normal text character including spaces and special characters.

Filenames can

It is best to stick to a-z, A-Z,

If a filename contains a special character or a space you may need to put quotes around the whole path.

It is best to keep them short

as it saves typing.

Special characters in filenames can cause problems with some programs.

特殊宏字符

* ? " "

Any single character.
including none.

& | > <

Cause the process to run in the
background

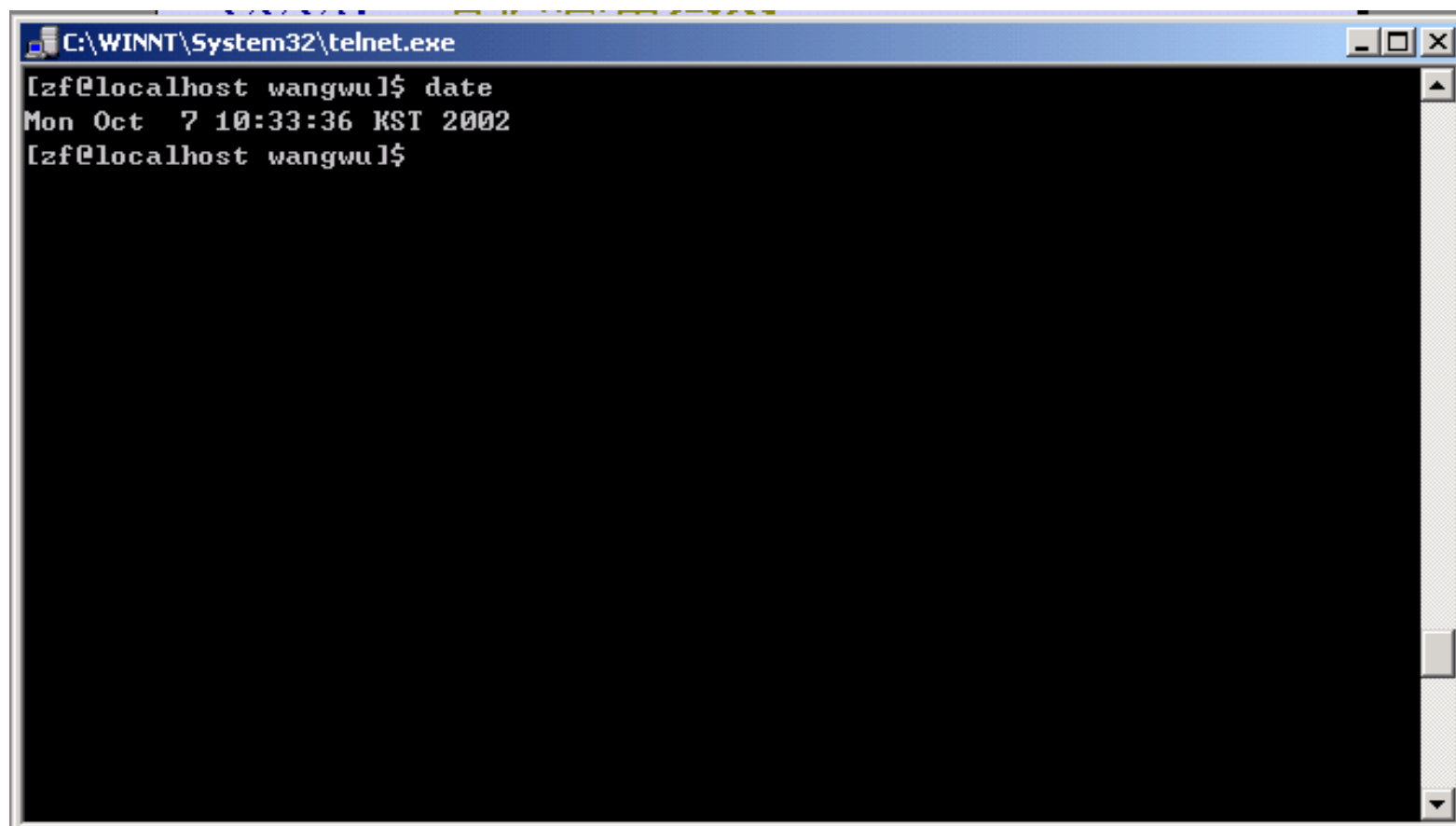
\ \$ \ ;

Backslash.
Change the meaning of the
next character.
value

Some special characters can lose their special meaning if they are inside quotes.

常用命令

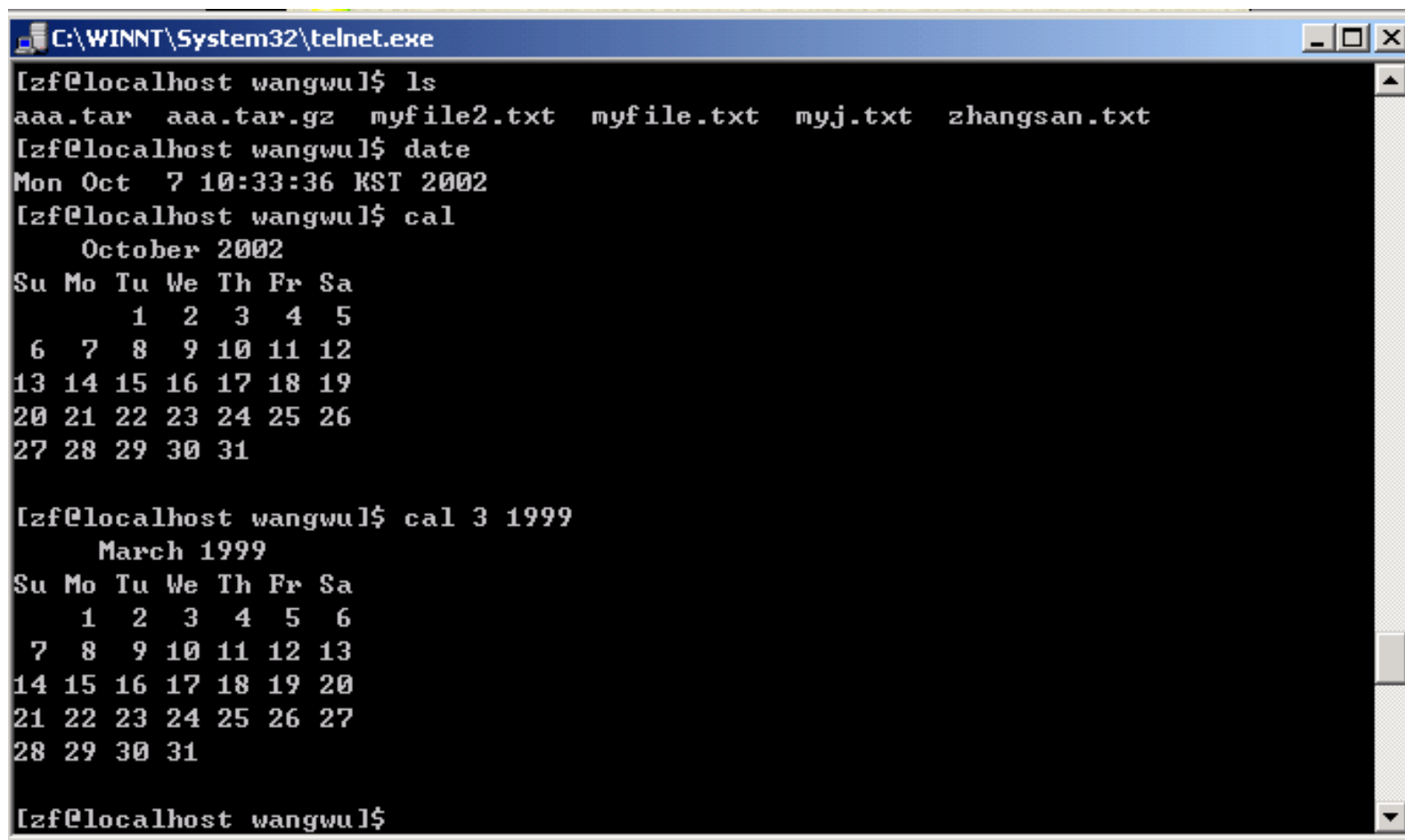
- ❖ **date**命令：告诉用户设置在Linux系统时钟的当前日期和时间。

A screenshot of a Windows command prompt window titled "C:\WINNT\System32\telnet.exe". The window shows a telnet session with the prompt "[zf@localhost wangwu]". The user has entered the command "date", and the output is "Mon Oct 7 10:33:36 KST 2002". The prompt is now "[zf@localhost wangwu]".

```
C:\WINNT\System32\telnet.exe
[zf@localhost wangwu]$ date
Mon Oct 7 10:33:36 KST 2002
[zf@localhost wangwu]$
```

常用命令

❖ cal命令：可以显示一个日历



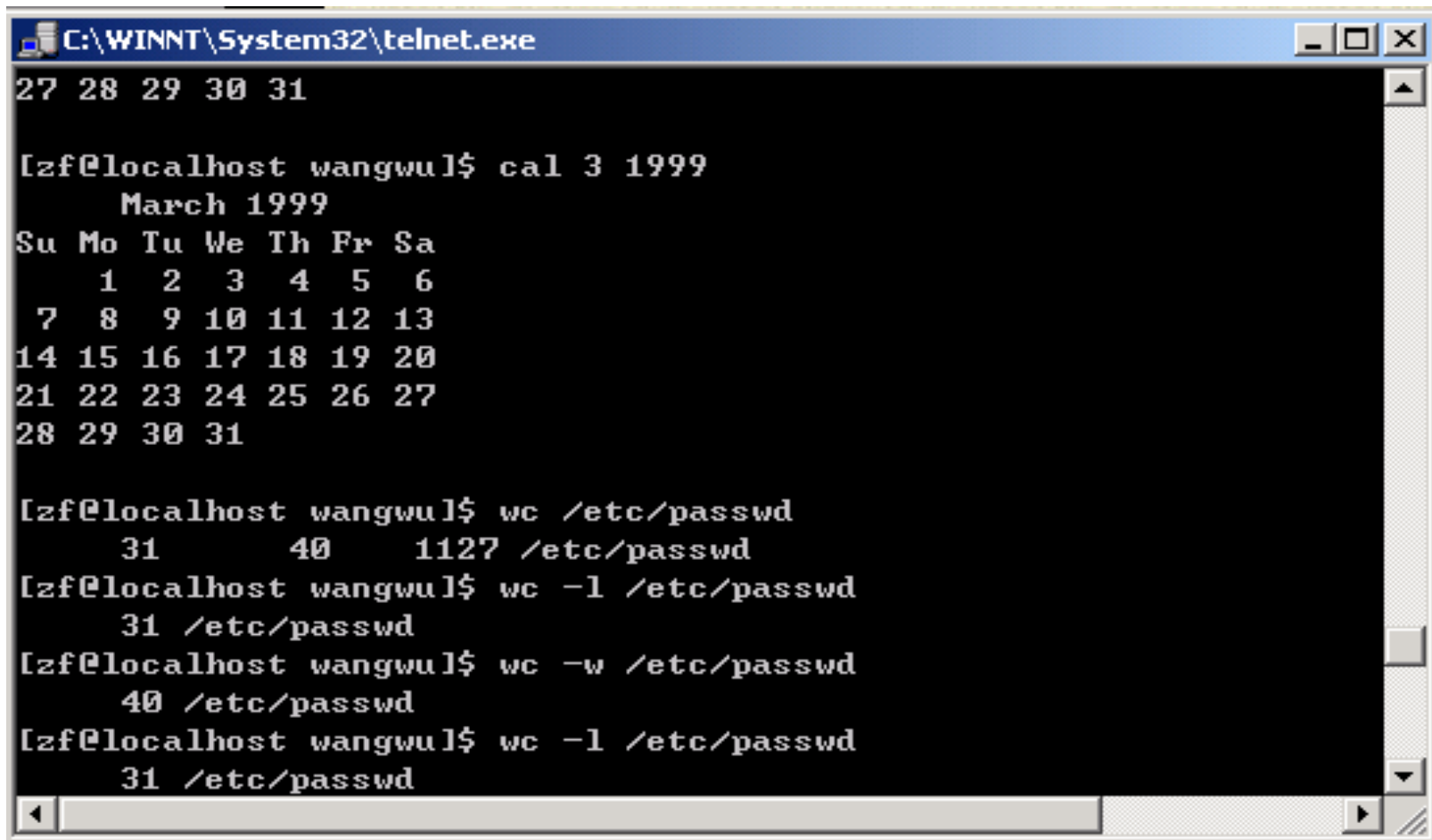
```
C:\WINNT\System32\telnet.exe
[zf@localhost wangwu]$ ls
aaa.tar  aaa.tar.gz  myfile2.txt  myfile.txt  myj.txt  zhangsan.txt
[zf@localhost wangwu]$ date
Mon Oct  7 10:33:36 KST 2002
[zf@localhost wangwu]$ cal
    October 2002
Su Mo Tu We Th Fr Sa
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

[zf@localhost wangwu]$ cal 3 1999
    March 1999
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

[zf@localhost wangwu]$
```

常用命令

❖ WC命令：统计字符数。（-l -w -c）



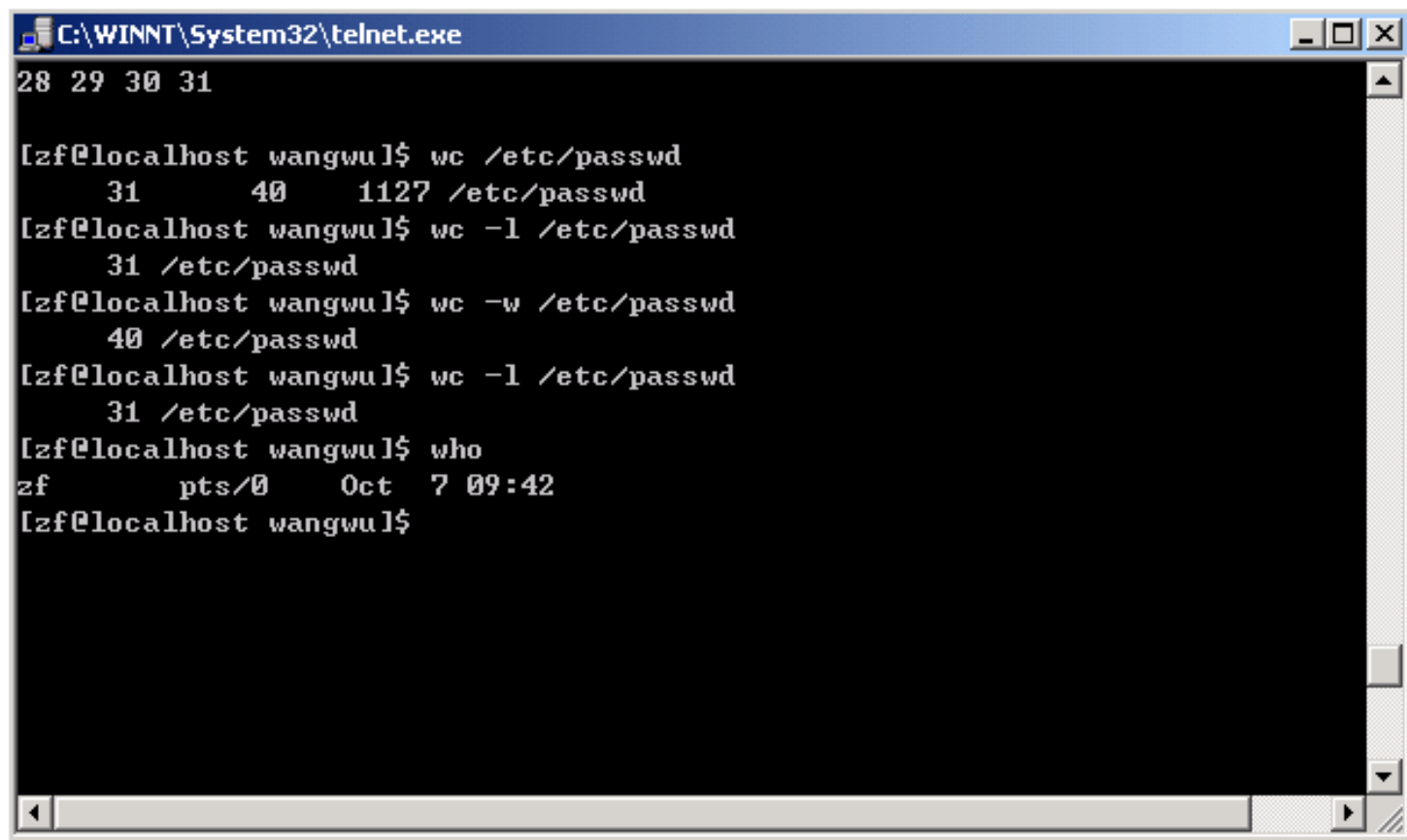
```
C:\WINNT\System32\telnet.exe
27 28 29 30 31

[zf@localhost wangwu]$ cal 3 1999
    March 1999
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

[zf@localhost wangwu]$ wc /etc/passwd
   31      40    1127 /etc/passwd
[zf@localhost wangwu]$ wc -l /etc/passwd
   31 /etc/passwd
[zf@localhost wangwu]$ wc -w /etc/passwd
   40 /etc/passwd
[zf@localhost wangwu]$ wc -l /etc/passwd
   31 /etc/passwd
```

常用命令

- ❖ **who**命令：显示所有登录系统的用户的登录名、登录时间、及使用的终端。

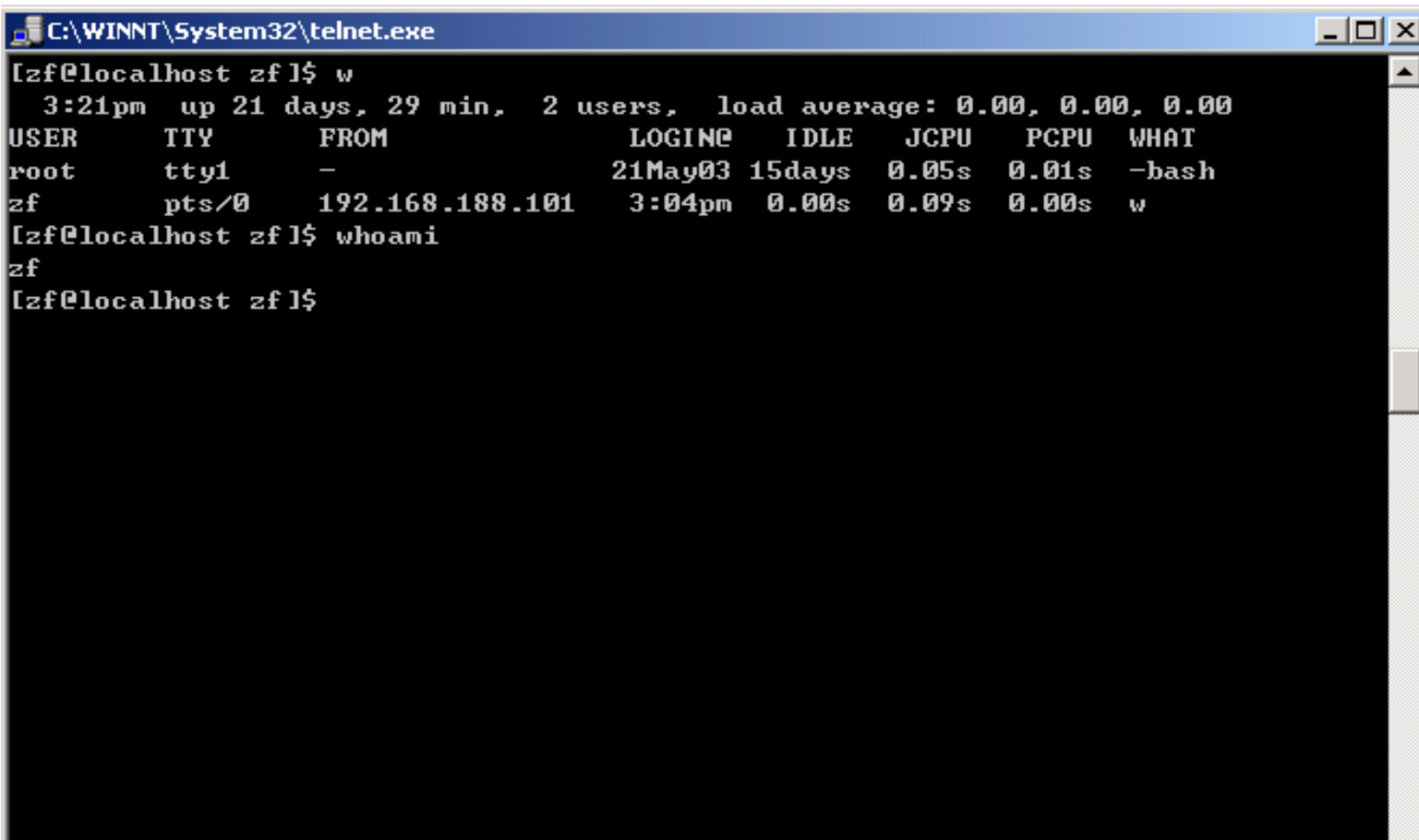


```
C:\WINNT\System32\telnet.exe
28 29 30 31

[zf@localhost wangwu]$ wc /etc/passwd
  31      40    1127 /etc/passwd
[zf@localhost wangwu]$ wc -l /etc/passwd
  31 /etc/passwd
[zf@localhost wangwu]$ wc -w /etc/passwd
  40 /etc/passwd
[zf@localhost wangwu]$ wc -l /etc/passwd
  31 /etc/passwd
[zf@localhost wangwu]$ who
zf      pts/0    Oct  7 09:42
[zf@localhost wangwu]$
```

常用命令

❖ w和whoami命令

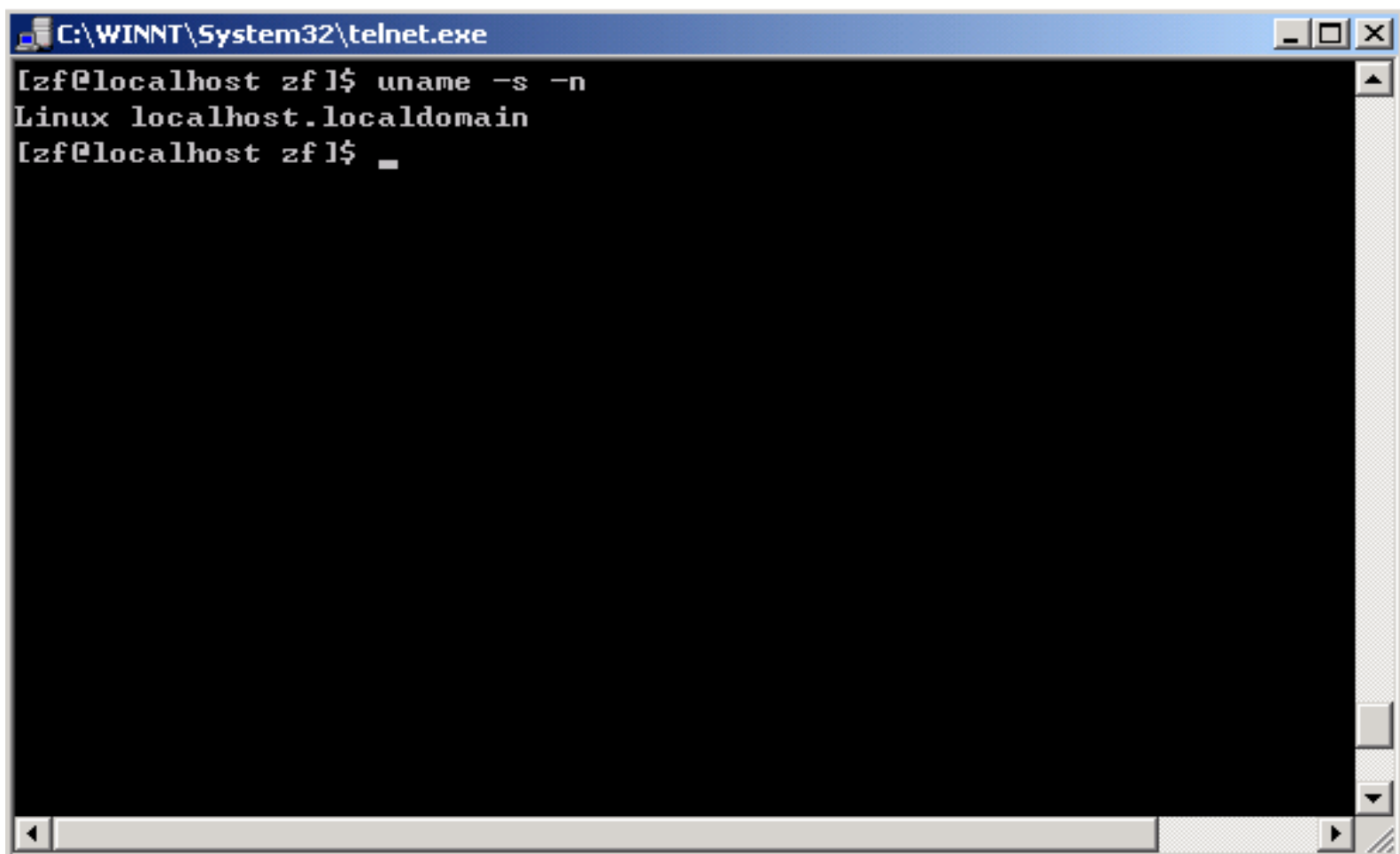


The screenshot shows a Windows command prompt window titled "C:\WINNT\System32\telnet.exe". The user is logged in as "zf" on "localhost". The user enters the command "w", which displays system status and a table of active users. The user then enters "whoami", which returns "zf".

```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ w
  3:21pm  up 21 days, 29 min,  2 users,  load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@  IDLE   JCPU   PCPU   WHAT
root      tty1     -             21May03 15days  0.05s  0.01s  -bash
zf        pts/0    192.168.188.101  3:04pm  0.00s  0.09s  0.00s  w
[zf@localhost zf]$ whoami
zf
[zf@localhost zf]$
```


常用命令

❖ **uname**命令：显示系统名、版本号等

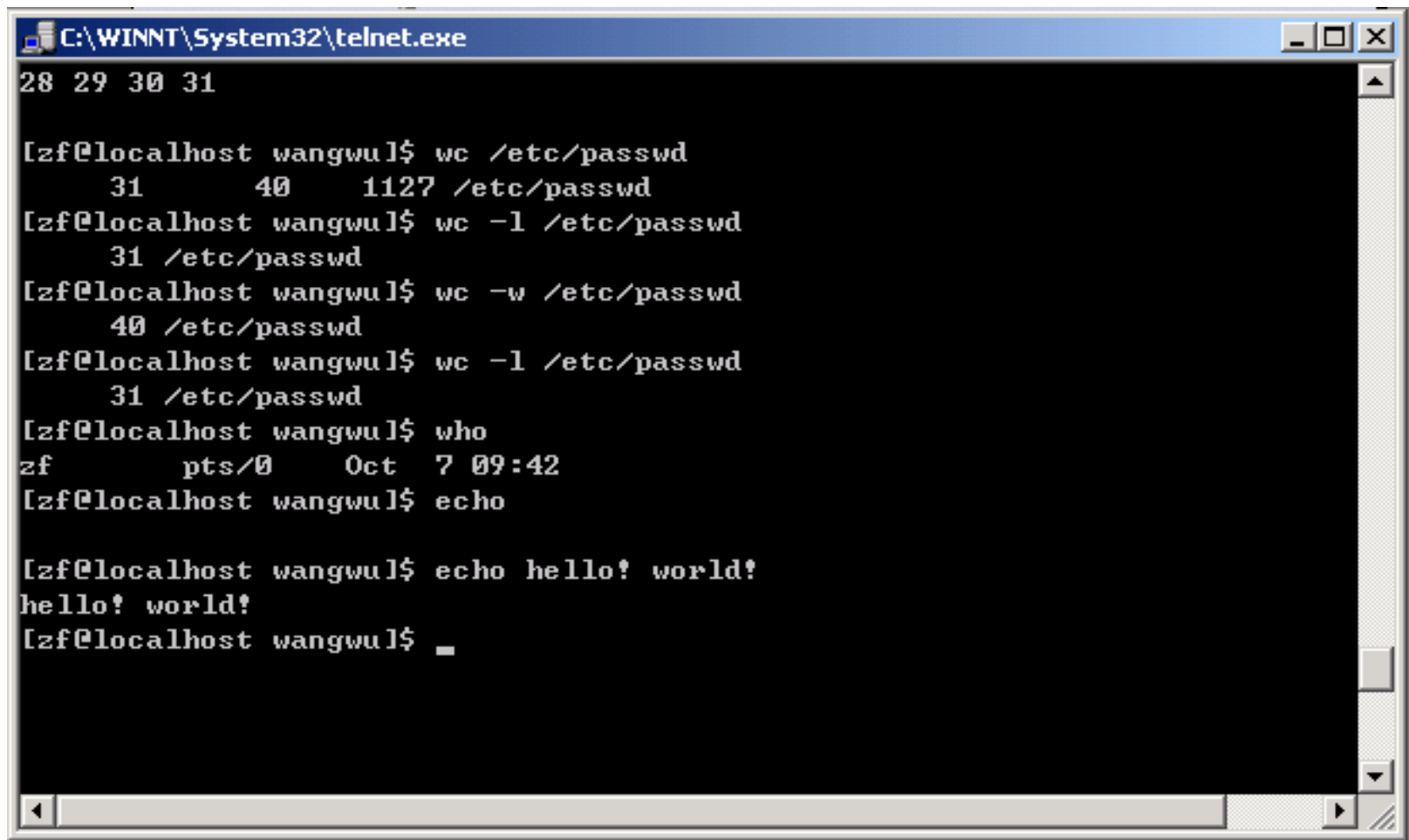


```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ uname -s -n
Linux localhost.localdomain
[zf@localhost zf]$
```

The screenshot shows a Windows telnet window titled "C:\WINNT\System32\telnet.exe". The terminal displays the command "uname -s -n" being executed, which returns "Linux localhost.localdomain". The prompt "[zf@localhost zf]" is visible before and after the command.

常用命令

❖ echo命令：显示输出



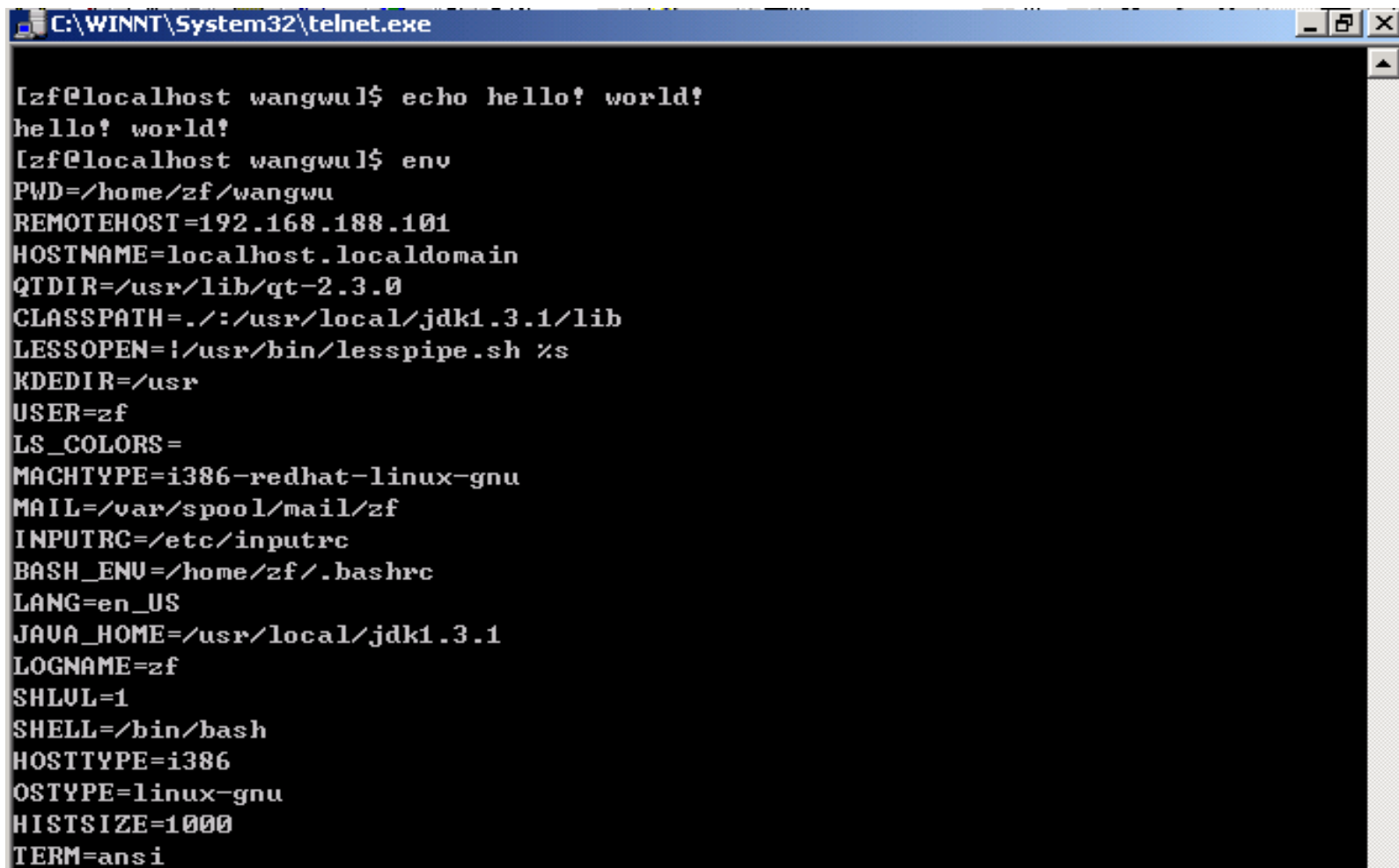
```
C:\WINNT\System32\telnet.exe
28 29 30 31

[zf@localhost wangwul$ wc /etc/passwd
  31      40     1127 /etc/passwd
[zf@localhost wangwul$ wc -l /etc/passwd
  31 /etc/passwd
[zf@localhost wangwul$ wc -w /etc/passwd
  40 /etc/passwd
[zf@localhost wangwul$ wc -c /etc/passwd
  31 /etc/passwd
[zf@localhost wangwul$ who
zf          pts/0    Oct  7 09:42
[zf@localhost wangwul$ echo

[zf@localhost wangwul$ echo hello! world!
hello! world!
[zf@localhost wangwul$ _
```

常用命令

❖ env命令：显示系统环境参数。

A screenshot of a Windows command prompt window titled "C:\WINNT\System32\telnet.exe". The window shows a telnet session with a user named 'zf' at a host named 'localhost'. The user has entered the command 'env', and the output displays various system environment variables. The variables listed include PWD, REMOTEHOST, HOSTNAME, QTDIR, CLASSPATH, LESSOPEN, KDEDIR, USER, LS_COLORS, MACHTYPE, MAIL, INPUTRC, BASH_ENV, LANG, JAVA_HOME, LOGNAME, SHLVL, SHELL, HOSTTYPE, OSTYPE, HISTSIZE, and TERM.

```
C:\WINNT\System32\telnet.exe

[zf@localhost wangwu]$ echo hello! world!
hello! world!
[zf@localhost wangwu]$ env
PWD=/home/zf/wangwu
REMOTEHOST=192.168.188.101
HOSTNAME=localhost.localdomain
QTDIR=/usr/lib/qt-2.3.0
CLASSPATH=./:/usr/local/jdk1.3.1/lib
LESSOPEN=!/usr/bin/lesspipe.sh %s
KDEDIR=/usr
USER=zf
LS_COLORS=
MACHTYPE=i386-redhat-linux-gnu
MAIL=/var/spool/mail/zf
INPUTRC=/etc/inputrc
BASH_ENV=/home/zf/.bashrc
LANG=en_US
JAVA_HOME=/usr/local/jdk1.3.1
LOGNAME=zf
SHLVL=1
SHELL=/bin/bash
HOSTTYPE=i386
OSTYPE=linux-gnu
HISTSIZE=1000
TERM=ansi
```

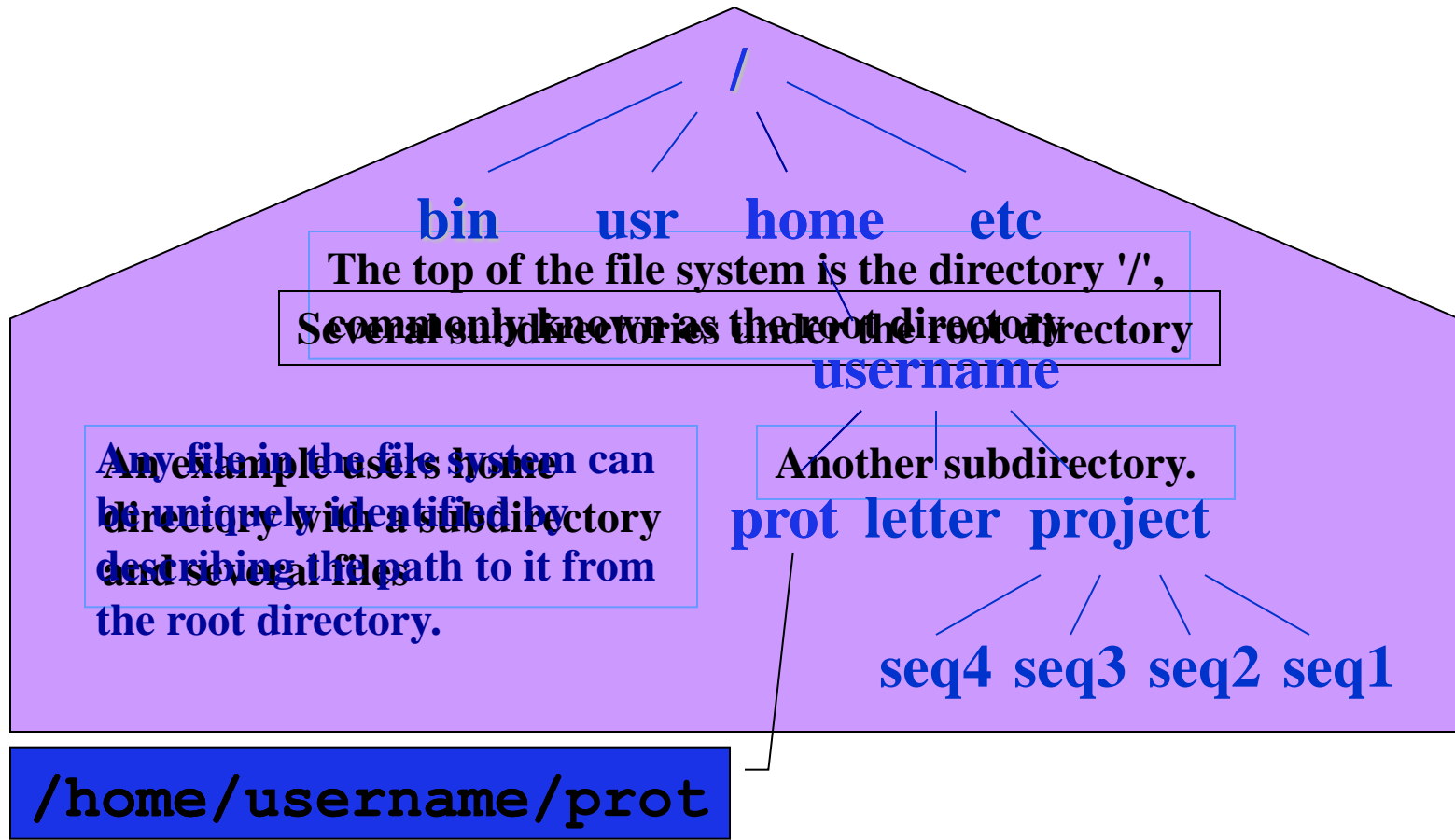
常用命令

❖ 其它常用命令：

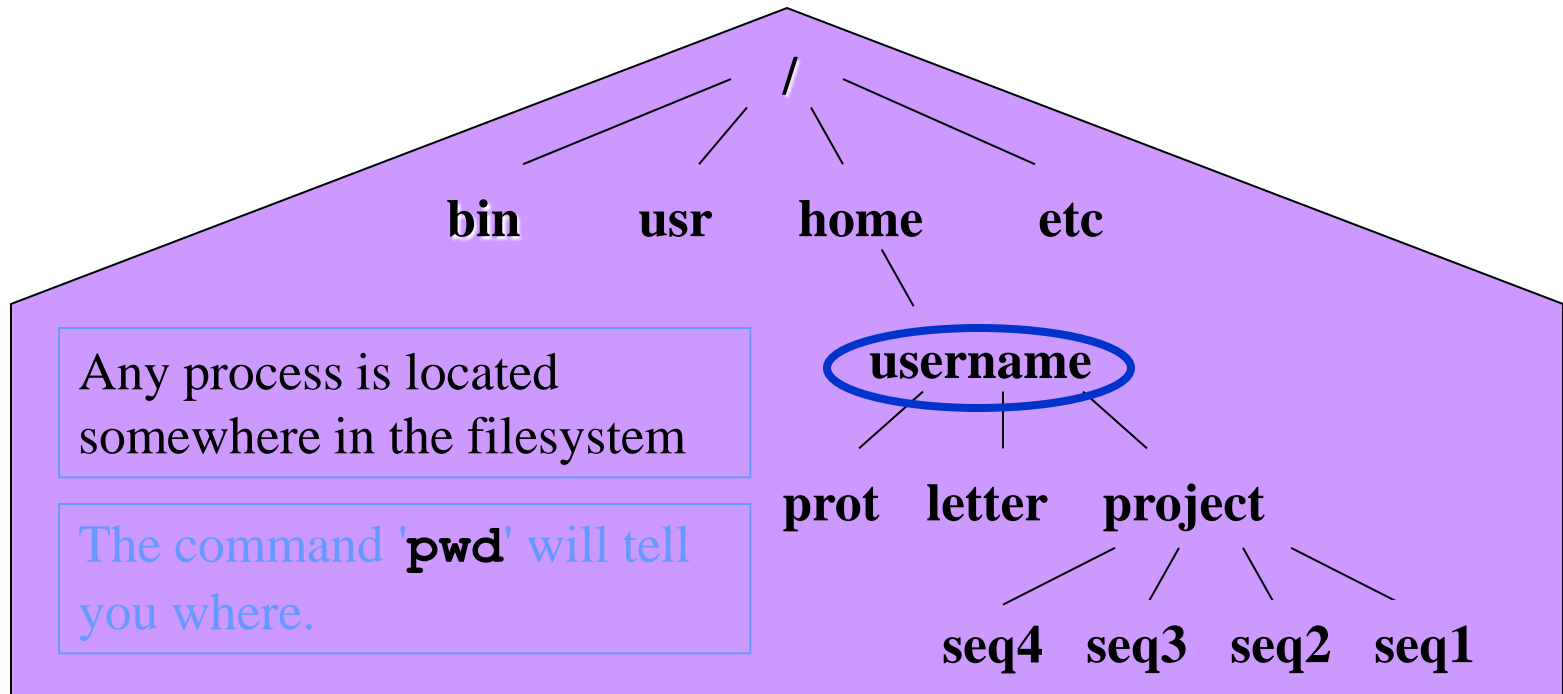
❖ **clear**: 清屏

❖ **banner**: 打印标签，用于打印登录时的欢迎消息和打印分隔。

Linux文件系统结构



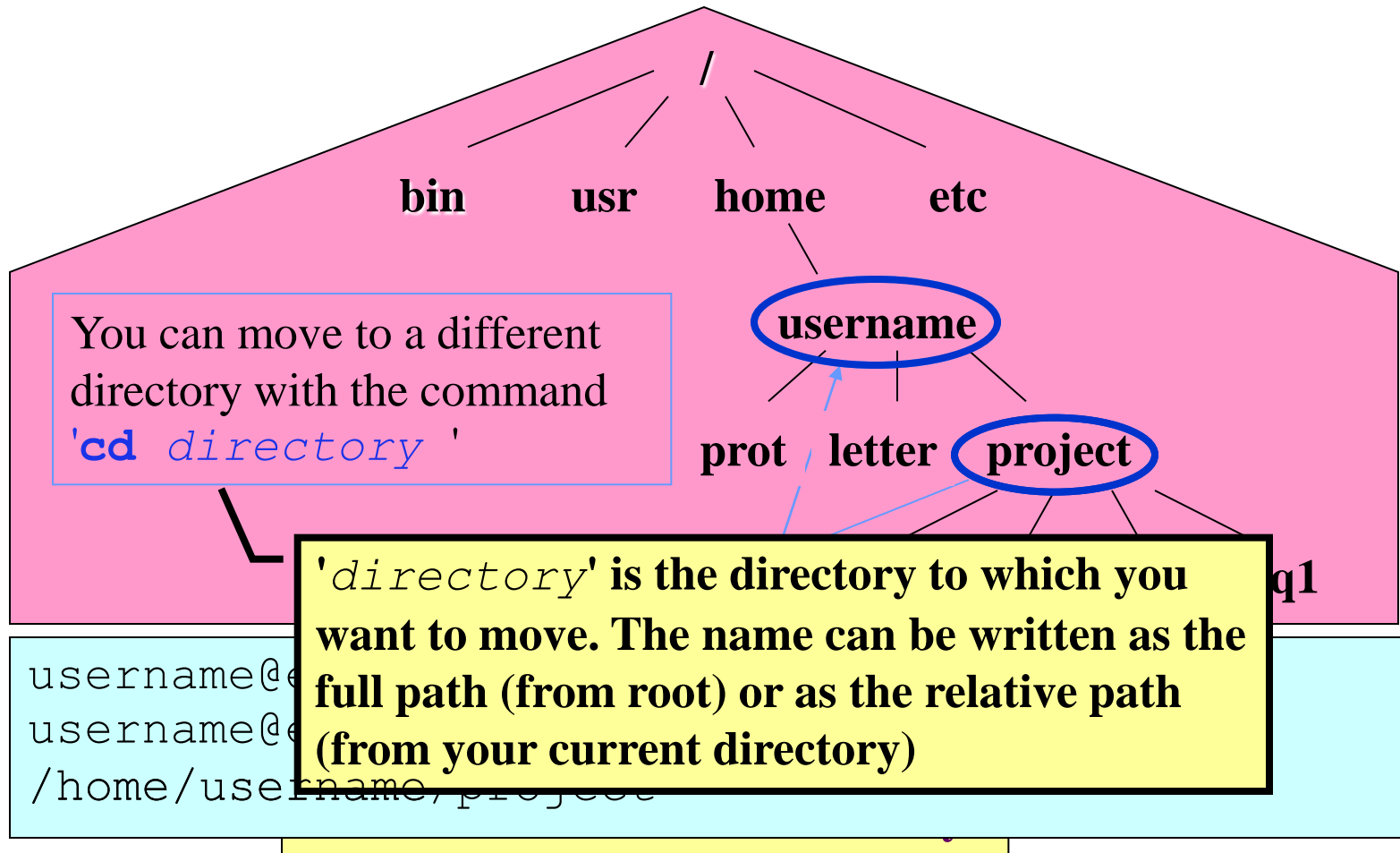
Linux目录结构



```
username@embnet ~> pwd  
/home/username
```

'~' is a unix shortcut meaning 'your home directory'

修改目录操作



Linux文件

You can view the permissions for a file by listing it in long format with the command
`'ls -l filename'`

```
username@embnet ~> ls -l letter  
-rwxr--r-- 1 username users 6048 Aug 17 16:07 letter
```

Type Permissions The user The files group

-- ordinary file
d - directory
l - link (shortcut)

目录操作命令

- ❖ **ls** 列出目录的内容。
- ❖ **-a** 列出全部目录项，包括“.”，和“..”。
- ❖ **-l** (**long**) 强行是每行只显示一个文件名。
- R** 递归显示子目录下的内容。

注： **-a** 通常用于列出所以隐含文件（以“.”开头的文件）
-l 可用于列出文件的存取权限，如：
drw-r-- r--

文件与目录

C:\WINNT\System32\telnet.exe

[zf@localhost zf]\$ ls -l

total 96

-rwxrwxr--	1	zf	zf	179	Nov	22	16:51	break
-rwxrwxr--	1	zf	zf	410	Nov	22	15:58	case
drwxr-xr-x	2	zf	zf	4096	Jul	10	16:59	Desktop
-rwxrwxr--	1	zf	zf	632	Nov	22	16:10	end_loop
-rwxrwxr--	1	zf	zf	180	Nov	22	15:02	expr1
-rwxrwxr--	1	zf	zf	78	Nov	22	16:26	for
-rwxrwxr--	1	zf	zf	115	Nov	22	16:31	for2
-rwxrwxr--	1	zf	zf	326	Nov	22	15:45	if
-rwxrwxr--	1	zf	zf	253	Nov	22	15:21	int_test
-rw-----	1	zf	zf	391	Jul	17	16:19	mbox
-rwxrwxr--	1	zf	zf	230	Nov	22	15:13	mycom
drwxrwxr-x	2	zf	zf	4096	Oct	6	11:38	mydir
-rw-rw-r--	1	zf	zf	263	Aug	9	09:58	mydir.tar.gz
-rwxrwxr--	1	zf	zf	150	Nov	22	10:19	read
-rwxrwxr--	1	zf	zf	136	Nov	22	16:14	shift
-rwxrwxr-x	1	zf	zf	14329	Jul	15	15:38	test1
-rwxrwxr--	1	zf	zf	120	Nov	22	09:48	var1
drwxrwxr-x	2	zf	zf	4096	Oct	7	10:26	wangwu
-rwxrwxr--	1	zf	zf	120	Nov	22	16:07	while
-rwxrwxr--	1	zf	zf	173	Nov	22	16:38	while2
drwxrwxr-x	2	zf	zf	4096	Oct	7	09:52	zhangsan

[zf@localhost zf]\$

文件与目录

❖ the first character indicates the file's type

"-" means it is a plain file

"d" means it is a directory

"p" means it is a pipe

"b" means it is a block special file

"c" means it is a character special file

Linux文件特征

"Everything is a file"

- An ordinary file contains data.
- A directory contains other files.
- A link is a file that is a shortcut to another file.

~~• The data could be any type of document. There are many other types of file.~~

This is also known as a folder on some systems. A

Files can have more than one name, and be in

You don't need to worry about these. You probably won't come across one in normal use of the system.

Linux文件权限

- Every file is protected to a greater or lesser extent.

- Permissions determine **who** can read, write, or execute a given file.

Owner

Group

World

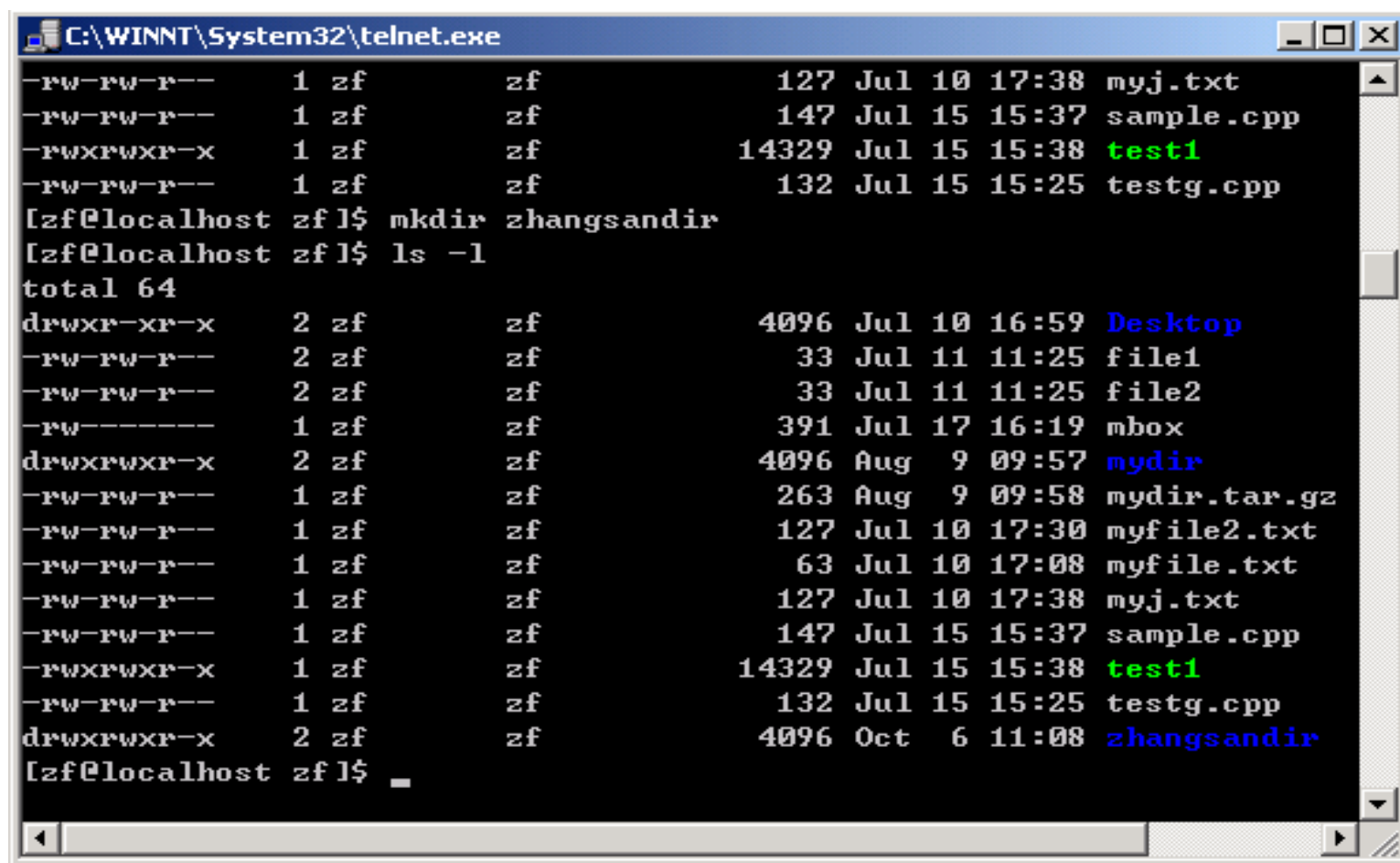
The user who owns the file

Other users in the same
All the other users in the
system.

- Files can have read, write or execute permission for each of the three types of user.

目录操作命令

❖ mkdir 建立目录

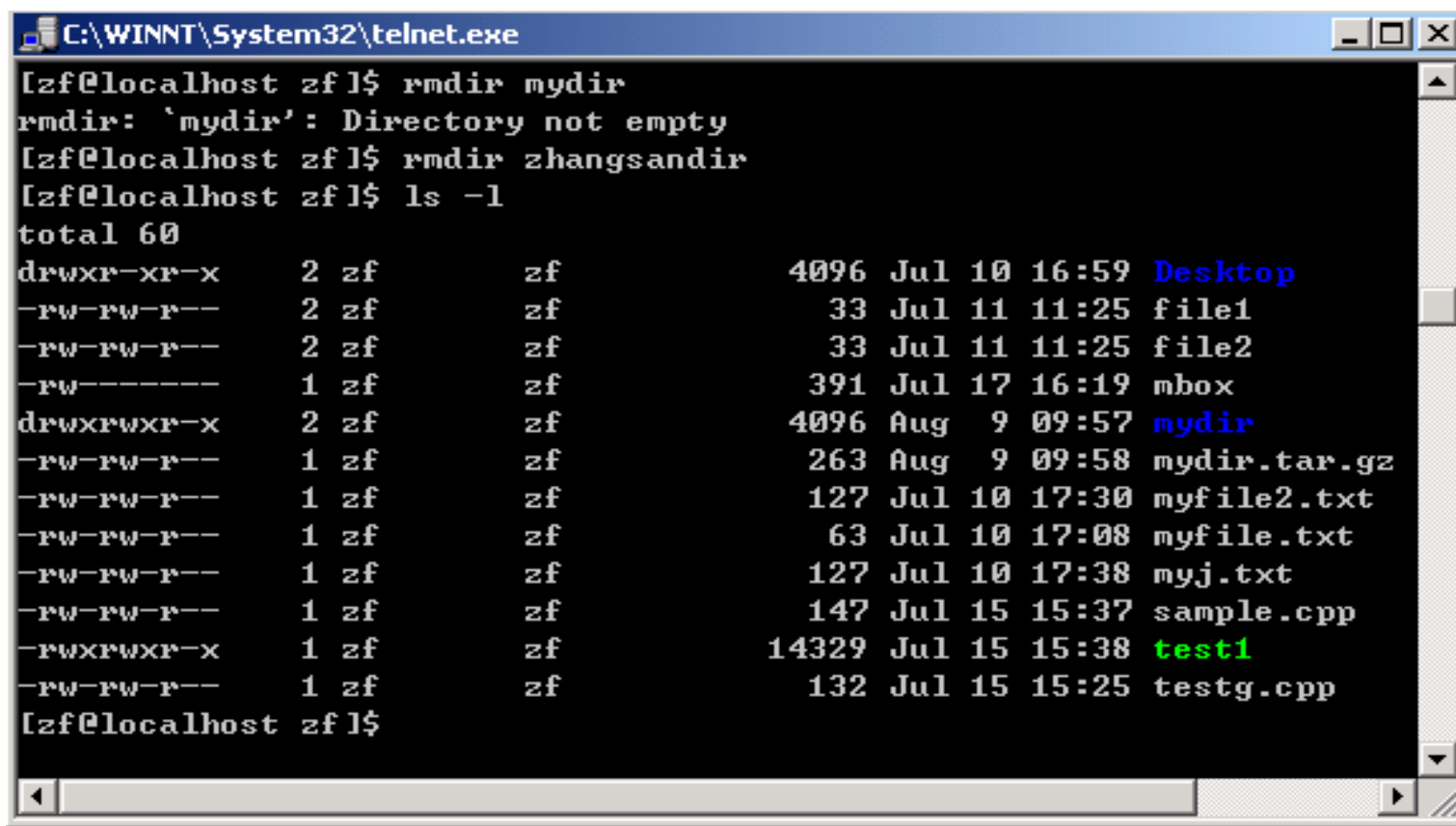


The screenshot shows a telnet window titled "C:\WINNT\System32\telnet.exe". The user is at a prompt [zf@localhost zf] and has just executed the command `mkdir zhangsmdir`. The prompt has changed to [zf@localhost zf] and the user has entered `ls -l`. The output shows a list of files and directories with their permissions, sizes, dates, times, and names. The new directory `zhangsmdir` is listed at the bottom with permissions `drwxrwxr-x`, size 4096, and date Oct 6 11:08.

```
C:\WINNT\System32\telnet.exe
-rw-rw-r-- 1 zf      zf           127 Jul 10 17:38 myj.txt
-rw-rw-r-- 1 zf      zf           147 Jul 15 15:37 sample.cpp
-rwxrwxr-x 1 zf      zf          14329 Jul 15 15:38 test1
-rw-rw-r-- 1 zf      zf           132 Jul 15 15:25 testg.cpp
[zf@localhost zf]$ mkdir zhangsmdir
[zf@localhost zf]$ ls -l
total 64
drwxr-xr-x  2 zf      zf           4096 Jul 10 16:59 Desktop
-rw-rw-r--  2 zf      zf            33 Jul 11 11:25 file1
-rw-rw-r--  2 zf      zf            33 Jul 11 11:25 file2
-rw-----  1 zf      zf           391 Jul 17 16:19 mbox
drwxrwxr-x  2 zf      zf           4096 Aug  9 09:57 mydir
-rw-rw-r--  1 zf      zf           263 Aug  9 09:58 mydir.tar.gz
-rw-rw-r--  1 zf      zf           127 Jul 10 17:30 myfile2.txt
-rw-rw-r--  1 zf      zf            63 Jul 10 17:08 myfile.txt
-rw-rw-r--  1 zf      zf           127 Jul 10 17:38 myj.txt
-rw-rw-r--  1 zf      zf           147 Jul 15 15:37 sample.cpp
-rwxrwxr-x  1 zf      zf          14329 Jul 15 15:38 test1
-rw-rw-r--  1 zf      zf           132 Jul 15 15:25 testg.cpp
drwxrwxr-x  2 zf      zf           4096 Oct  6 11:08 zhangsmdir
[zf@localhost zf]$
```

目录操作命令

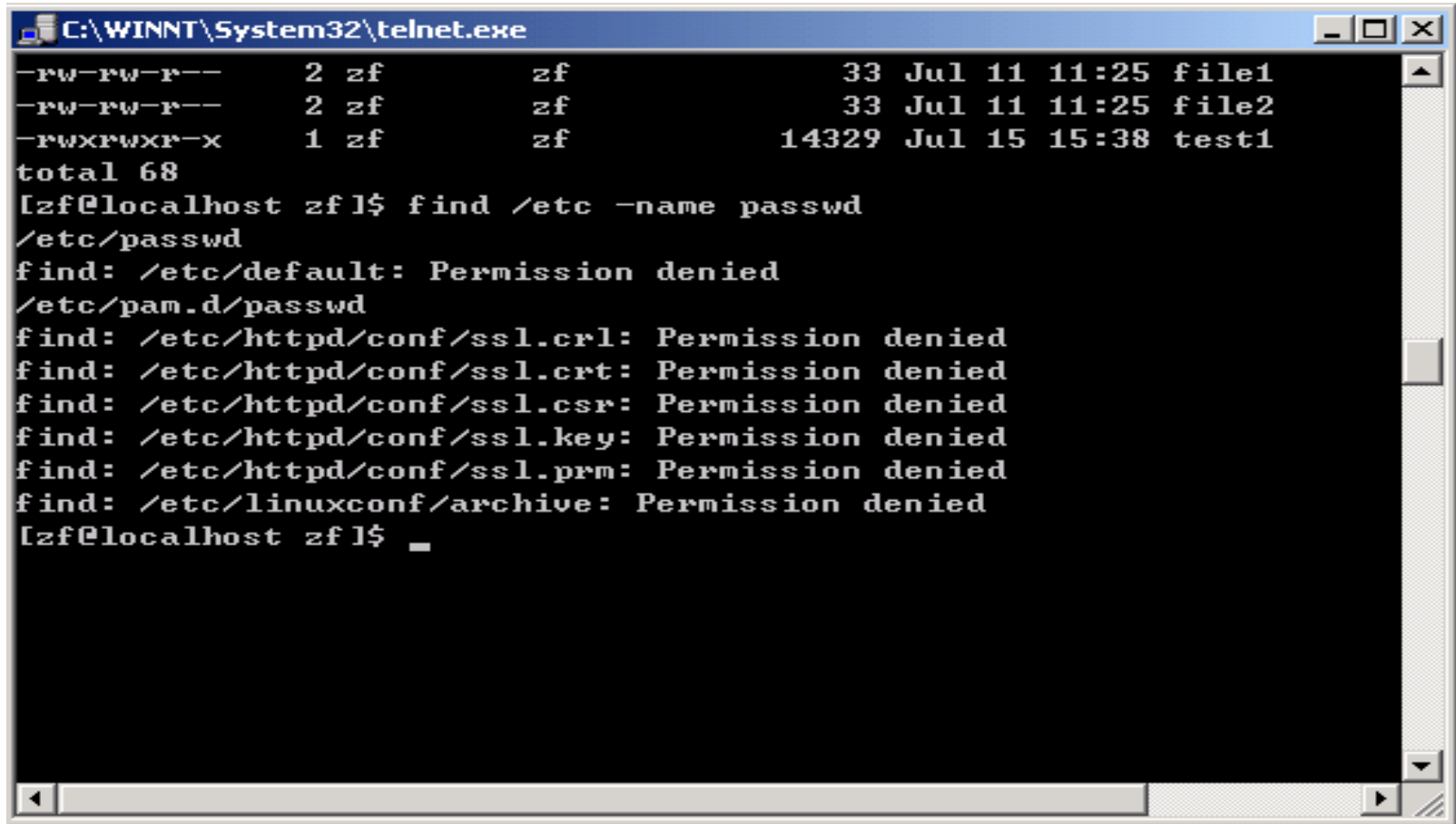
- ❖ **rmdir** 删除目录。从一个目录中删除一个或多个子目录项。注意和**rm**命令的区别。



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ rmdir mydir
rmdir: 'mydir': Directory not empty
[zf@localhost zf]$ rmdir zhangsmdir
[zf@localhost zf]$ ls -l
total 60
drwxr-xr-x  2 zf      zf      4096 Jul 10 16:59 Desktop
-rw-rw-r--  2 zf      zf      33 Jul 11 11:25 file1
-rw-rw-r--  2 zf      zf      33 Jul 11 11:25 file2
-rw-----  1 zf      zf     391 Jul 17 16:19 mbox
drwxrwxr-x  2 zf      zf     4096 Aug  9 09:57 mydir
-rw-rw-r--  1 zf      zf     263 Aug  9 09:58 mydir.tar.gz
-rw-rw-r--  1 zf      zf     127 Jul 10 17:30 myfile2.txt
-rw-rw-r--  1 zf      zf      63 Jul 10 17:08 myfile.txt
-rw-rw-r--  1 zf      zf     127 Jul 10 17:38 myj.txt
-rw-rw-r--  1 zf      zf     147 Jul 15 15:37 sample.cpp
-rwxrwxr-x  1 zf      zf    14329 Jul 15 15:38 test1
-rw-rw-r--  1 zf      zf      132 Jul 15 15:25 testg.cpp
[zf@localhost zf]$
```

目录查找命令

❖ find命令 在指定位置查找文件。



```
C:\WINNT\System32\telnet.exe
-rw-rw-r--      2 zf      zf              33 Jul 11 11:25 file1
-rw-rw-r--      2 zf      zf              33 Jul 11 11:25 file2
-rwxrwxr-x      1 zf      zf            14329 Jul 15 15:38 test1
total 68
[zf@localhost zf]$ find /etc -name passwd
/etc/passwd
find: /etc/default: Permission denied
/etc/pam.d/passwd
find: /etc/httpd/conf/ssl.crl: Permission denied
find: /etc/httpd/conf/ssl.crt: Permission denied
find: /etc/httpd/conf/ssl.csr: Permission denied
find: /etc/httpd/conf/ssl.key: Permission denied
find: /etc/httpd/conf/ssl.prm: Permission denied
find: /etc/linuxconf/archive: Permission denied
[zf@localhost zf]$ _
```


文件操作命令

You can print the contents of one or more files to the screen with the command:

```
'cat file1 file2 ...'
```

You can view the contents of one or more files a page at a time on the screen with the command:

```
'more file1 file2 ...'
```

cat prints the whole file at once, so a file longer than just a few lines will run off the top of your screen.

You can print the first few lines of a file with the command:

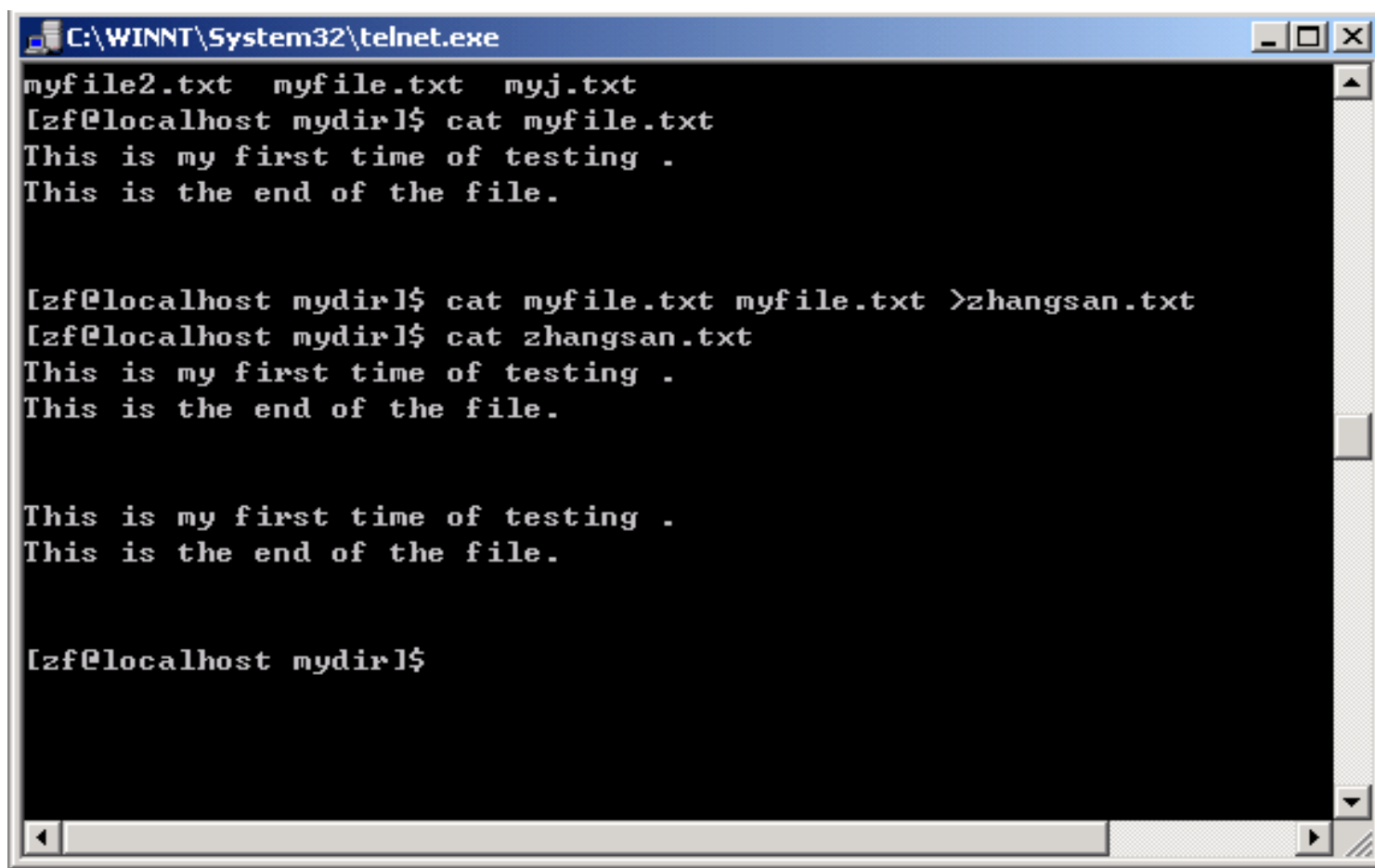
```
'head file1 file2 ...'
```

more will let you search through a file, go backwards and forwards and has many other functions.

The last few lines can be viewed with 'tail'

文件显示命令

❖ **cat命令** 用于连接并显示文件。



```
C:\WINNT\System32\telnet.exe
myfile2.txt  myfile.txt  myj.txt
[lzf@localhost mydir] $ cat myfile.txt
This is my first time of testing .
This is the end of the file.

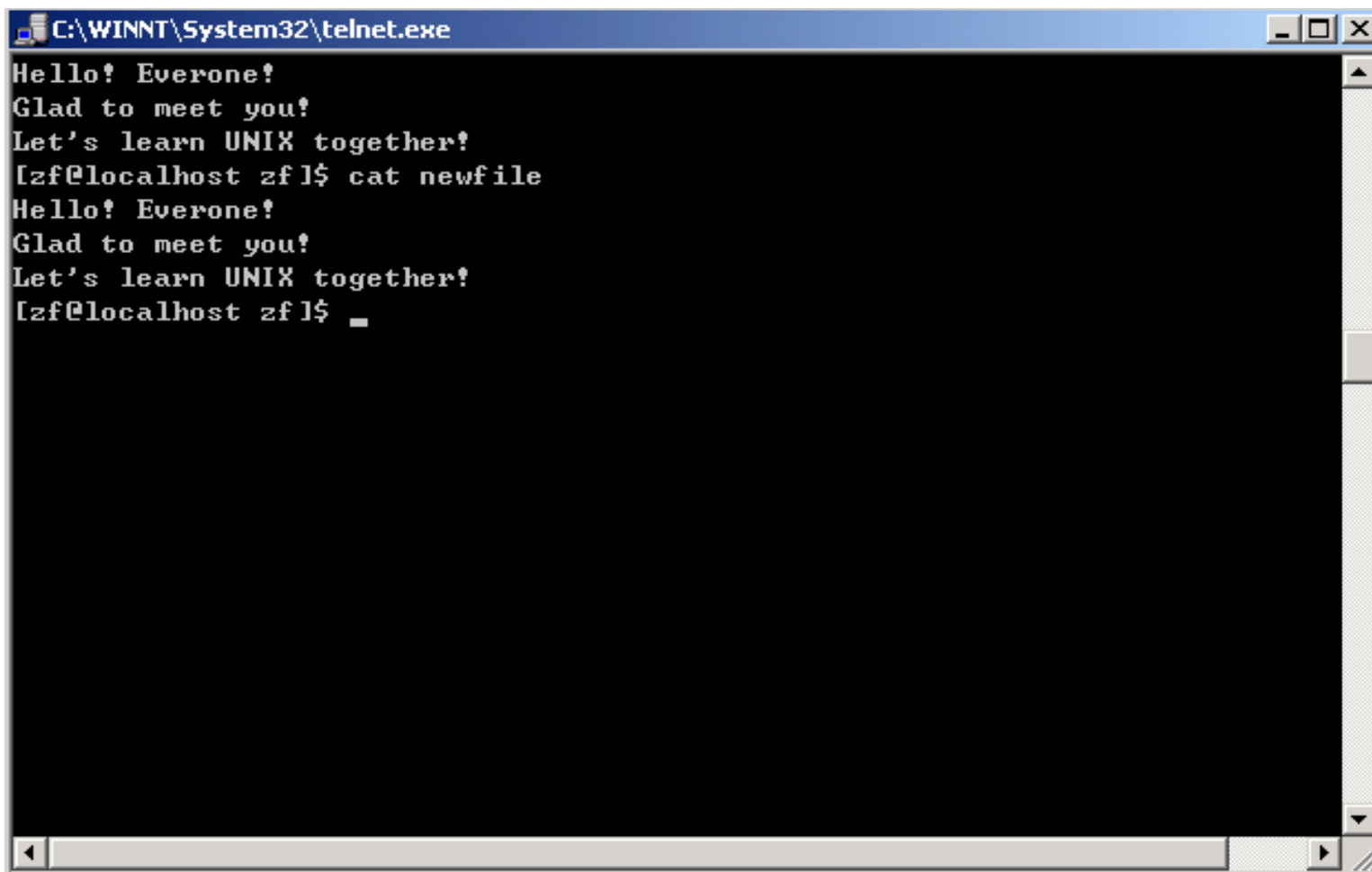
[lzf@localhost mydir] $ cat myfile.txt myfile.txt >zhangsan.txt
[lzf@localhost mydir] $ cat zhangsan.txt
This is my first time of testing .
This is the end of the file.

This is my first time of testing .
This is the end of the file.

[lzf@localhost mydir] $
```

文件显示命令

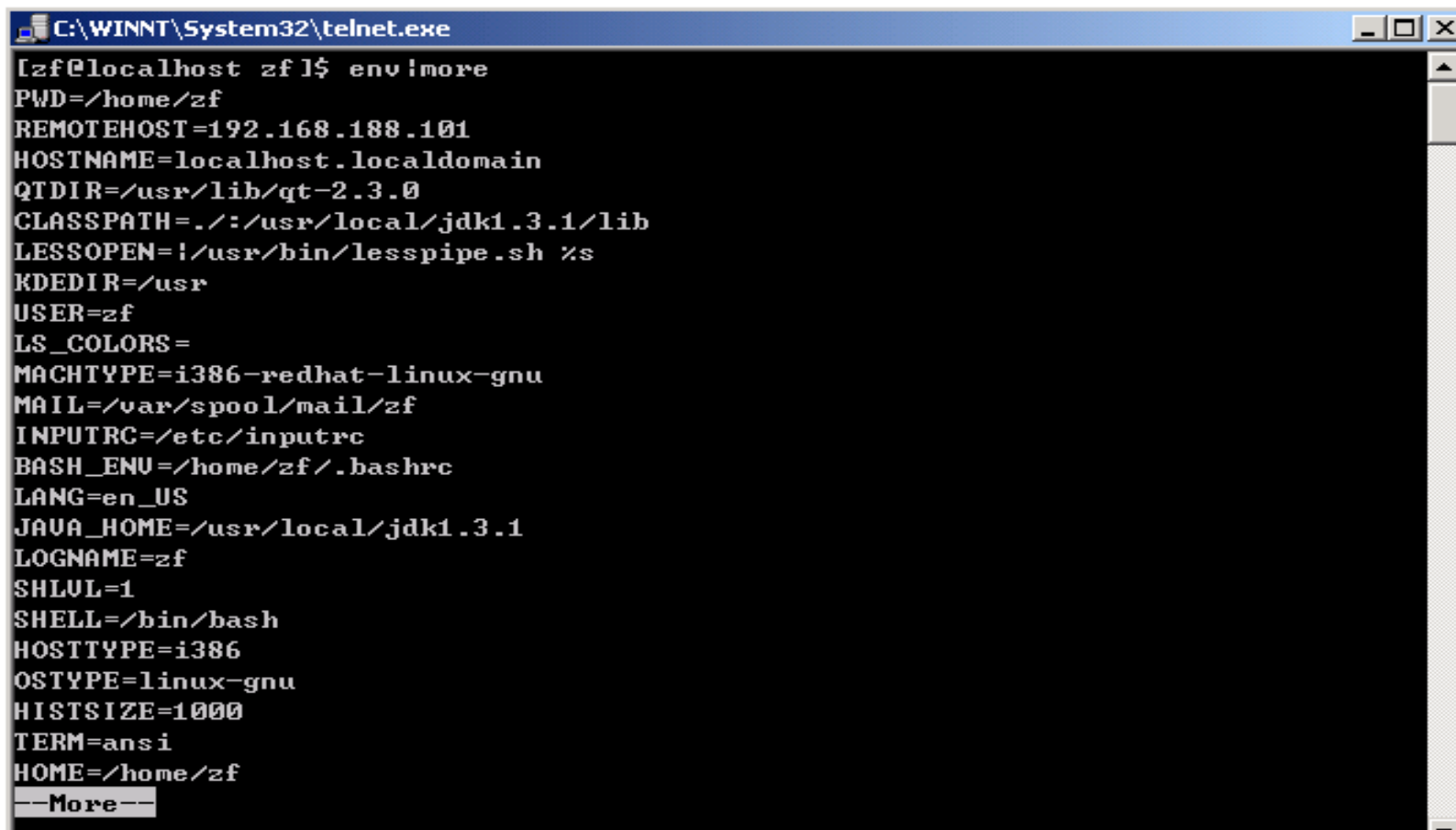
❖ cat命令也可以用于创建文件



```
C:\WINNT\System32\telnet.exe
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
[zf@localhost zf] $ cat newfile
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
[zf@localhost zf] $
```

文件显示命令

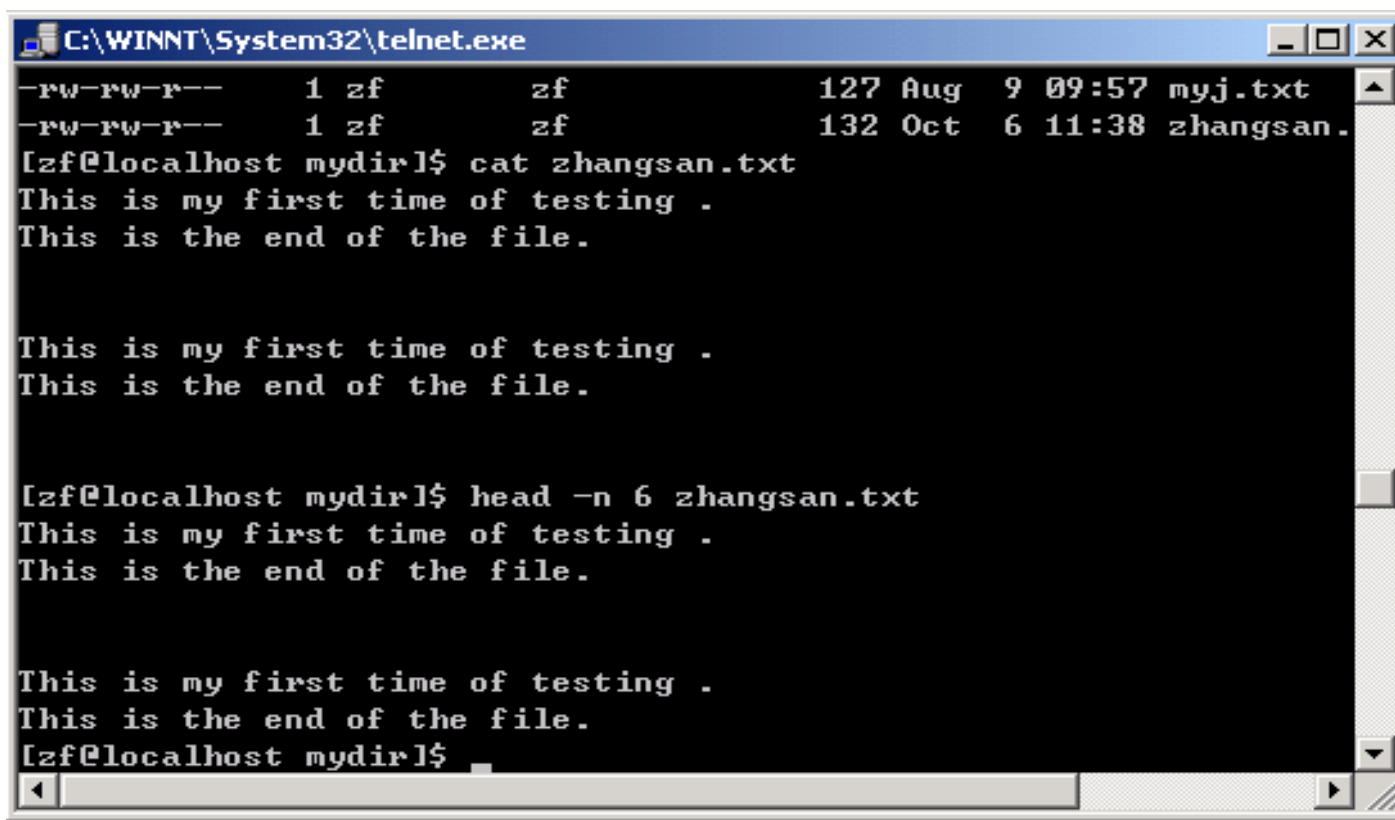
❖ **more命令** 用于逐屏显示文件，多用于显示长文件。

A screenshot of a Windows command prompt window titled "C:\WINNT\System32\telnet.exe". The window shows a telnet session with a user named "zf" at a "localhost". The user has entered the command "env | more", which displays a list of environment variables. The variables include PWD, REMOTEHOST, HOSTNAME, QTDIR, CLASSPATH, LESSOPEN, KDEDIR, USER, LS_COLORS, MACHTYPE, MAIL, INPUTRC, BASH_ENV, LANG, JAVA_HOME, LOGNAME, SHLVL, SHELL, HOSTTYPE, OSTYPE, HISTSIZE, TERM, and HOME. The list is paginated, with "More" visible at the bottom of the screen.

```
C:\WINNT\System32\telnet.exe
[zf@localhost zf] $ env | more
PWD=/home/zf
REMOTEHOST=192.168.188.101
HOSTNAME=localhost.localdomain
QTDIR=/usr/lib/qt-2.3.0
CLASSPATH=./:/usr/local/jdk1.3.1/lib
LESSOPEN=!/usr/bin/lesspipe.sh %s
KDEDIR=/usr
USER=zf
LS_COLORS=
MACHTYPE=i386-redhat-linux-gnu
MAIL=/var/spool/mail/zf
INPUTRC=/etc/inputrc
BASH_ENV=/home/zf/.bashrc
LANG=en_US
JAVA_HOME=/usr/local/jdk1.3.1
LOGNAME=zf
SHLVL=1
SHELL=/bin/bash
HOSTTYPE=i386
OSTYPE=linux-gnu
HISTSIZE=1000
TERM=ansi
HOME=/home/zf
--More--
```

文件显示命令

- ❖ head命令 显示文件的前面几行。
- ❖ head [-n number] [file...]



```
C:\WINNT\System32\telnet.exe
-rw-rw-r--  1 zf      zf          127 Aug  9 09:57 myj.txt
-rw-rw-r--  1 zf      zf          132 Oct  6 11:38 zhangsan.
[zf@localhost mydir]$ cat zhangsan.txt
This is my first time of testing .
This is the end of the file.

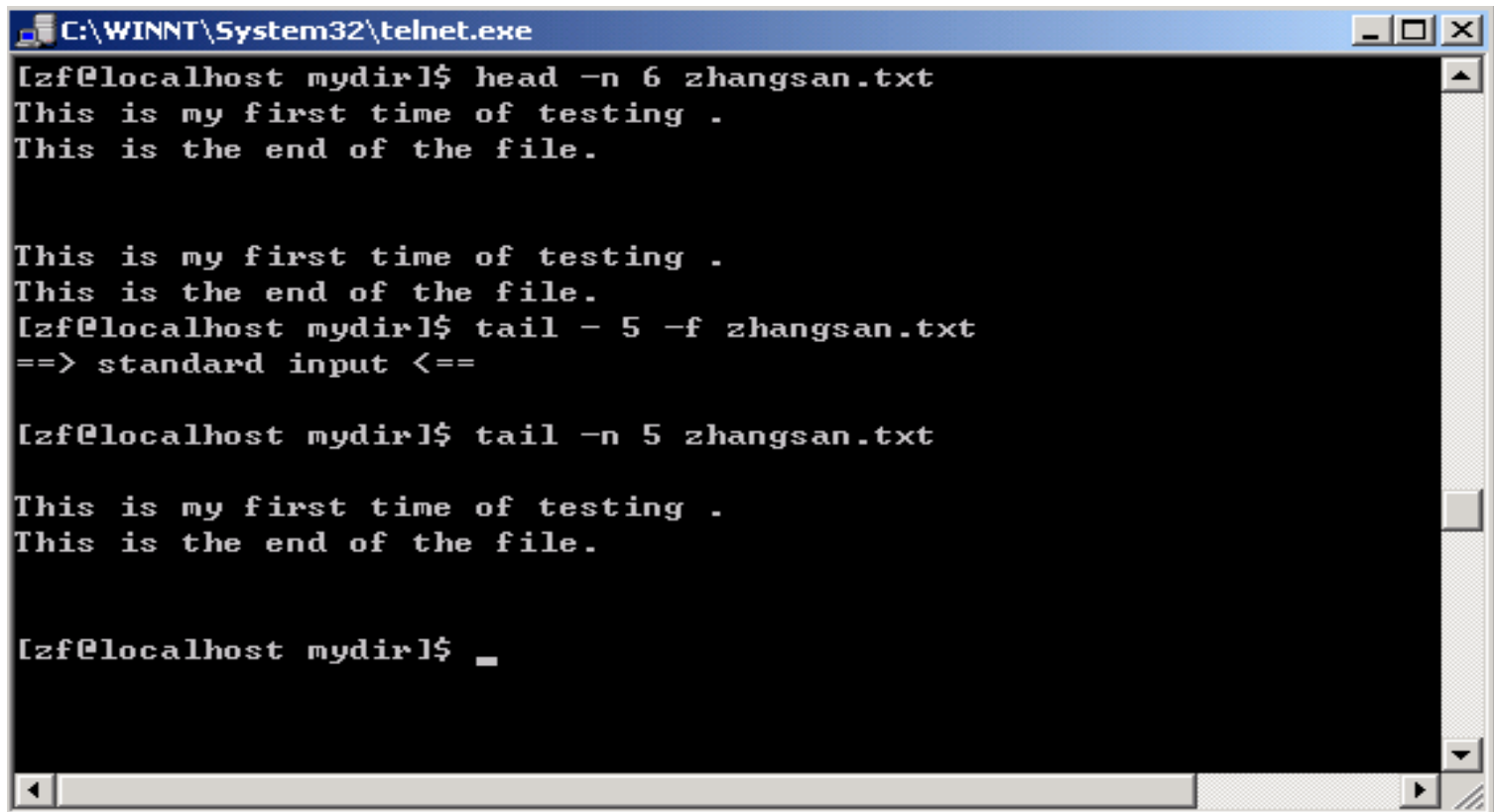
This is my first time of testing .
This is the end of the file.

[zf@localhost mydir]$ head -n 6 zhangsan.txt
This is my first time of testing .
This is the end of the file.

This is my first time of testing .
This is the end of the file.
[zf@localhost mydir]$
```

文件显示命令

- ❖ Tail命令 显示一个文件的最后部分。
- ❖ Tail [+/-n [number]] [-f] [file]



```
C:\WINNT\System32\telnet.exe
[zf@localhost mydir]$ head -n 6 zhangsan.txt
This is my first time of testing .
This is the end of the file.

This is my first time of testing .
This is the end of the file.
[zf@localhost mydir]$ tail - 5 -f zhangsan.txt
==> standard input <==

[zf@localhost mydir]$ tail -n 5 zhangsan.txt

This is my first time of testing .
This is the end of the file.

[zf@localhost mydir]$ _
```

文件操作命令

You can copy a file with the command
'**cp** *oldfilename newfilename*'

If *newfilename* is a directory.

```
username@embnet ~> ls
letter          project
username@embnet ~> cp letter draft
username@embnet ~> ls
draft           letter          project
username@embnet ~>
```

Warning:

If a file called *newfilename* already exists

The command '**mv** *oldfilename newfilename*'
can be used to rename a file

文件的复制命令

- ❖ **cp**命令 用于复制文件。
- ❖ **cp [-fir] source_file target_file**
- ❖ **-f** 强制复制，覆盖已存在的文件
- ❖ **-i** 交互式拷贝
- ❖ **-r** 递归复制文件层次。
- ❖ 例如：
- ❖ **cp -f /usr/zhangsan/* /usr/lisi/.**

文件的复制命令

```
C:\WINNT\System32\telnet.exe

-rw-rw-r-- 1 zf      zf          63 Jul 10 17:08 myfile.txt
-rw-rw-r-- 1 zf      zf          127 Jul 10 17:38 myj.txt
-rw-rw-r-- 1 zf      zf          147 Jul 15 15:37 sample.cpp
-rwxrwxr-x 1 zf      zf        14329 Jul 15 15:38 test1
-rw-rw-r-- 1 zf      zf          132 Jul 15 15:25 testg.cpp
drwxrwxr-x 2 zf      zf         4096 Oct  7 09:43 zhangsan
[zf@localhost zf]$ ls -l lisi
total 0
[zf@localhost zf]$ cp ./zhangsan/* ./lisi/*
cp: copying multiple files, but last argument './lisi/*' is not a direc
Try 'cp --help' for more information.
[zf@localhost zf]$ cp ./zhangsan/* ./lisi/.
[zf@localhost zf]$ ls -l lisi
total 16
-rw-rw-r-- 1 zf      zf          127 Oct  7 09:45 myfile2.txt
-rw-rw-r-- 1 zf      zf           66 Oct  7 09:45 myfile.txt
-rw-rw-r-- 1 zf      zf          127 Oct  7 09:45 myj.txt
-rw-rw-r-- 1 zf      zf          132 Oct  7 09:45 zhangsan.txt
[zf@localhost zf]$
```

文件删除命令

- ❖ **rm**命令 删除文件和目录
- ❖ **rm [-fir] file ...**
- ❖ **-f** 即使用户对文件没有写权限，也强行把它删去。
- ❖ **-i** 交互式删除。
- ❖ **-r** 递归删除指定目录的所有内容以及目录本身。

文件删除命令

```
C:\WINNT\System32\telnet.exe

-rw-rw-r-- 1 zf      zf          127 Oct  7 09:45 myj.txt
-rw-rw-r-- 1 zf      zf          132 Oct  7 09:45 zhangsan.txt
[zf@localhost zf]$ rm -r lisi
[zf@localhost zf]$ ls -l
total 64
drwxr-xr-x  2 zf      zf          4096 Jul 10 16:59 Desktop
-rw-rw-r--  2 zf      zf           33 Jul 11 11:25 file1
-rw-rw-r--  2 zf      zf           33 Jul 11 11:25 file2
-rw-----  1 zf      zf          391 Jul 17 16:19 mbox
drwxrwxr-x  2 zf      zf          4096 Oct  6 11:38 mydir
-rw-rw-r--  1 zf      zf          263 Aug  9 09:58 mydir.tar.gz
-rw-rw-r--  1 zf      zf          127 Jul 10 17:30 myfile2.txt
-rw-rw-r--  1 zf      zf           63 Jul 10 17:08 myfile.txt
-rw-rw-r--  1 zf      zf          127 Jul 10 17:38 myj.txt
-rw-rw-r--  1 zf      zf          147 Jul 15 15:37 sample.cpp
-rwxrwxr-x  1 zf      zf        14329 Jul 15 15:38 test1
-rw-rw-r--  1 zf      zf          132 Jul 15 15:25 testg.cpp
drwxrwxr-x  2 zf      zf          4096 Oct  7 09:43 zhangsan
[zf@localhost zf]$
```

文件移动命令

- ❖ **mv**命令 移动或者重新命名文件和目录。
- ❖ **mv [-fi] source_file target_file**
- ❖ **mv [-fi] source_file ... target_directory**
- ❖ 在第一种格式 中，完成移动或改名操作。这里，目标文件不是目录。
- ❖ 在第二种格式中， **mv**命令把每个源文件移到新目录中。
- ❖ **-f**禁止所有的提示。
- ❖ **-I**要进行交互式操作。

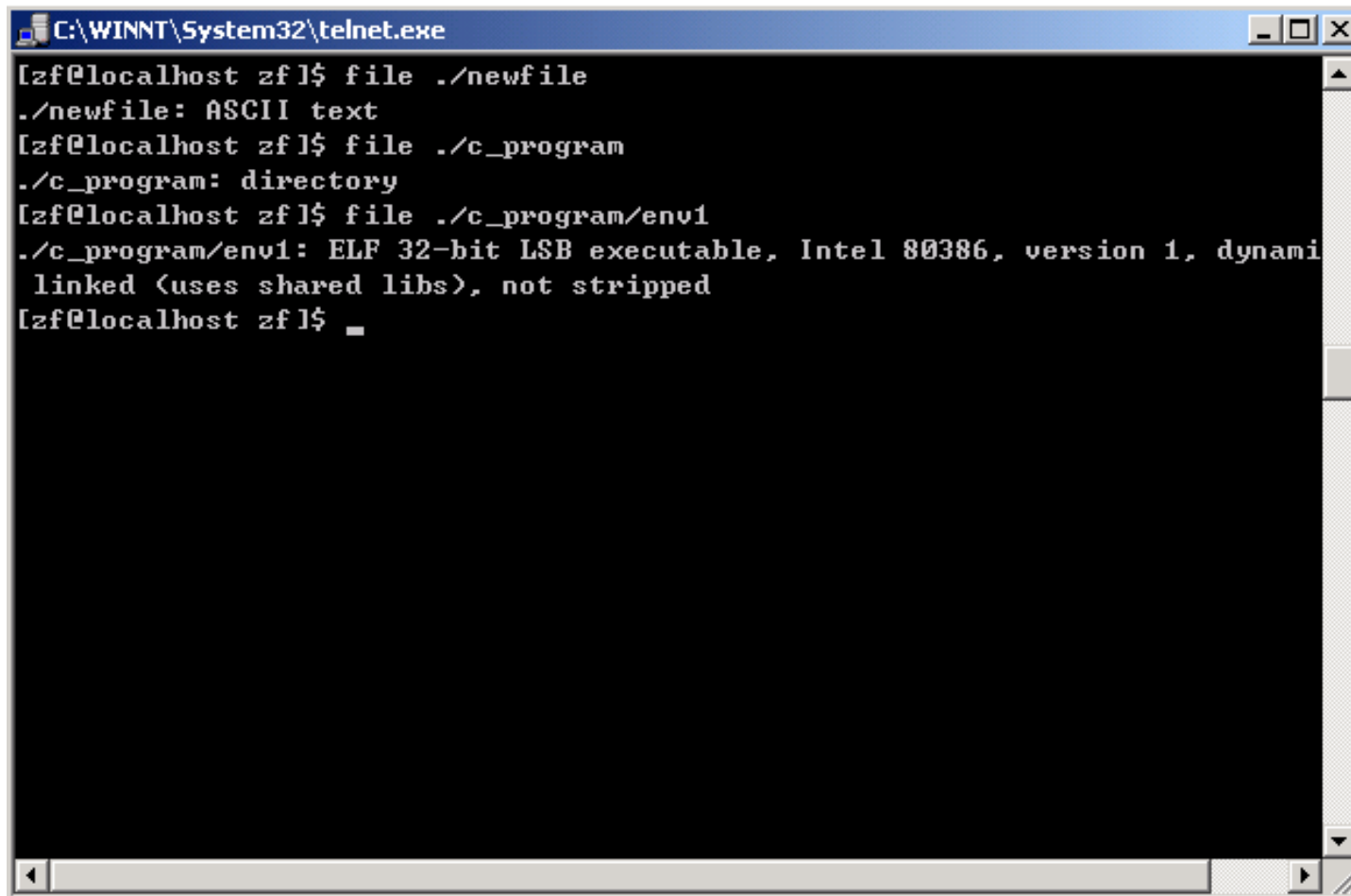
文件移动命令

```
C:\WINNT\System32\telnet.exe

-rw-rw-r--  1 zf      zf           63 Jul 10 17:08 myfile.txt
-rw-rw-r--  1 zf      zf          127 Jul 10 17:38 myj.txt
-rw-rw-r--  1 zf      zf          147 Jul 15 15:37 sample.cpp
-rwxrwxr-x  1 zf      zf        14329 Jul 15 15:38 test1
-rw-rw-r--  1 zf      zf          132 Jul 15 15:25 testg.cpp
drwxrwxr-x  2 zf      zf         4096 Oct  7 09:43 zhangsan
[zf@localhost zf] $ mkdir wangwu
[zf@localhost zf] $ ls -l wangwu
total 0
[zf@localhost zf] $ mv ./zhangsan/* wangwu
[zf@localhost zf] $ ls -l zhangsan
total 0
[zf@localhost zf] $ ls -l wangwu
total 16
-rw-rw-r--  1 zf      zf          127 Oct  7 09:43 myfile2.txt
-rw-rw-r--  1 zf      zf           66 Oct  7 09:43 myfile.txt
-rw-rw-r--  1 zf      zf          127 Oct  7 09:43 myj.txt
-rw-rw-r--  1 zf      zf          132 Oct  7 09:43 zhangsan.txt
[zf@localhost zf] $
```

有关查看文件类型的命令

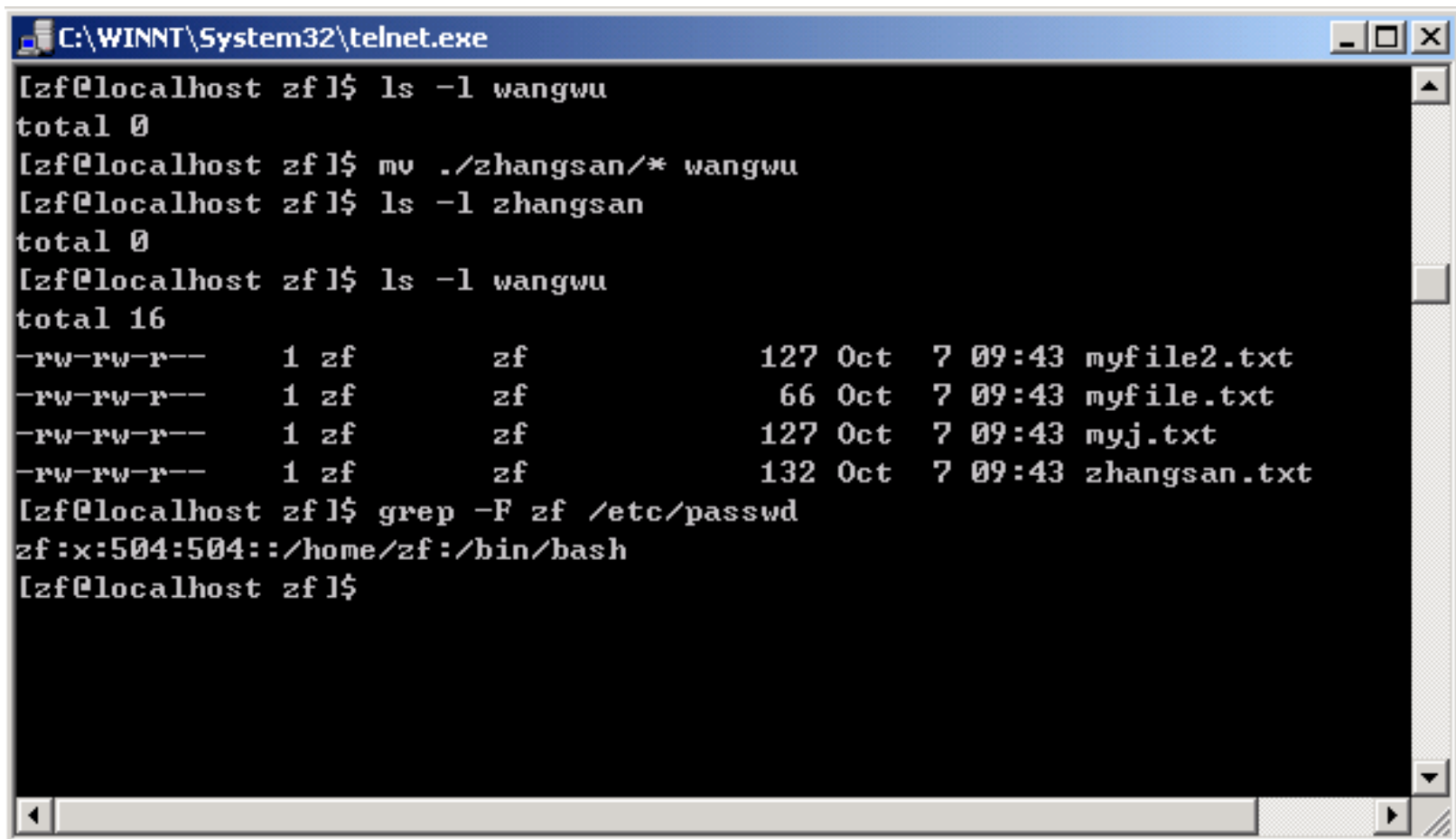
- ❖ Linux中file命令用于判定文件的类型。



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ file ./newfile
./newfile: ASCII text
[zf@localhost zf]$ file ./c_program
./c_program: directory
[zf@localhost zf]$ file ./c_program/env1
./c_program/env1: ELF 32-bit LSB executable, Intel 80386, version 1, dynamically
linked (uses shared libs), not stripped
[zf@localhost zf]$
```

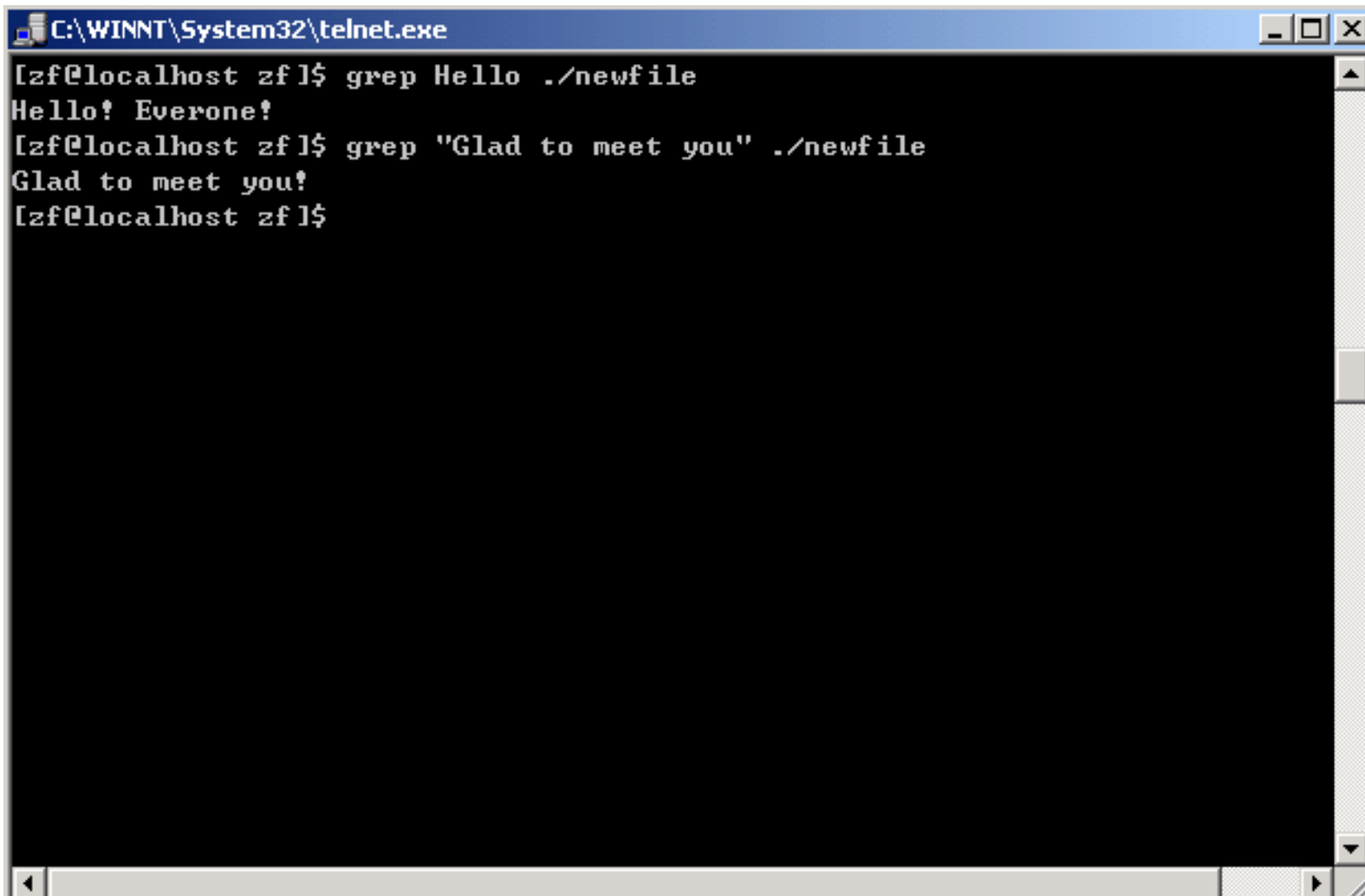
查找文件内容

❖ **grep**命令 按给定模式查找文件。



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ ls -l wangwu
total 0
[zf@localhost zf]$ mv ./zhangsan/* wangwu
[zf@localhost zf]$ ls -l zhangsan
total 0
[zf@localhost zf]$ ls -l wangwu
total 16
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf       66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf     127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf     132 Oct  7 09:43 zhangsan.txt
[zf@localhost zf]$ grep -F zf /etc/passwd
zf:x:504:504::/home/zf:/bin/bash
[zf@localhost zf]$
```

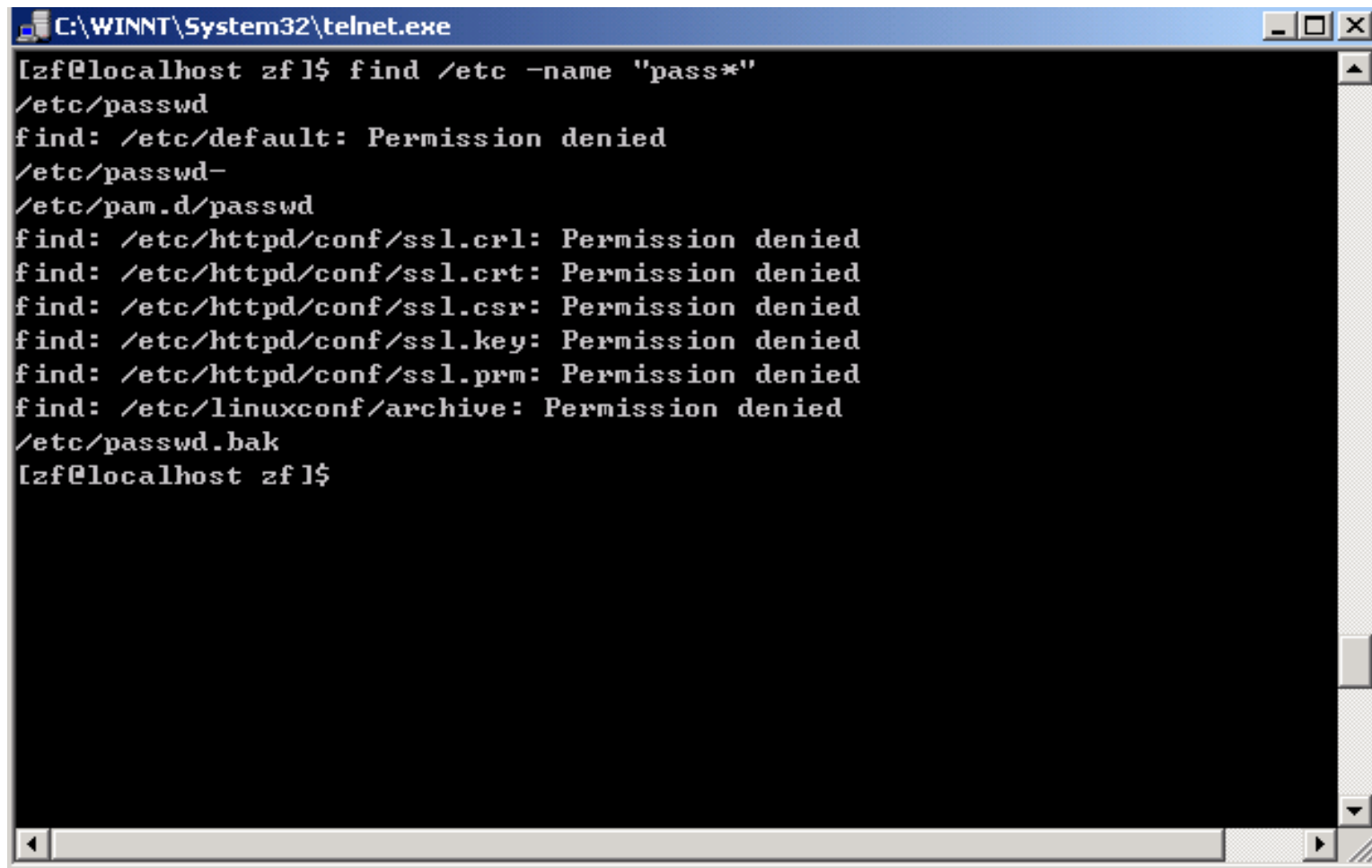
查找文件内容



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ grep Hello ./newfile
Hello! Everone!
[zf@localhost zf]$ grep "Glad to meet you" ./newfile
Glad to meet you!
[zf@localhost zf]$
```


查找文件

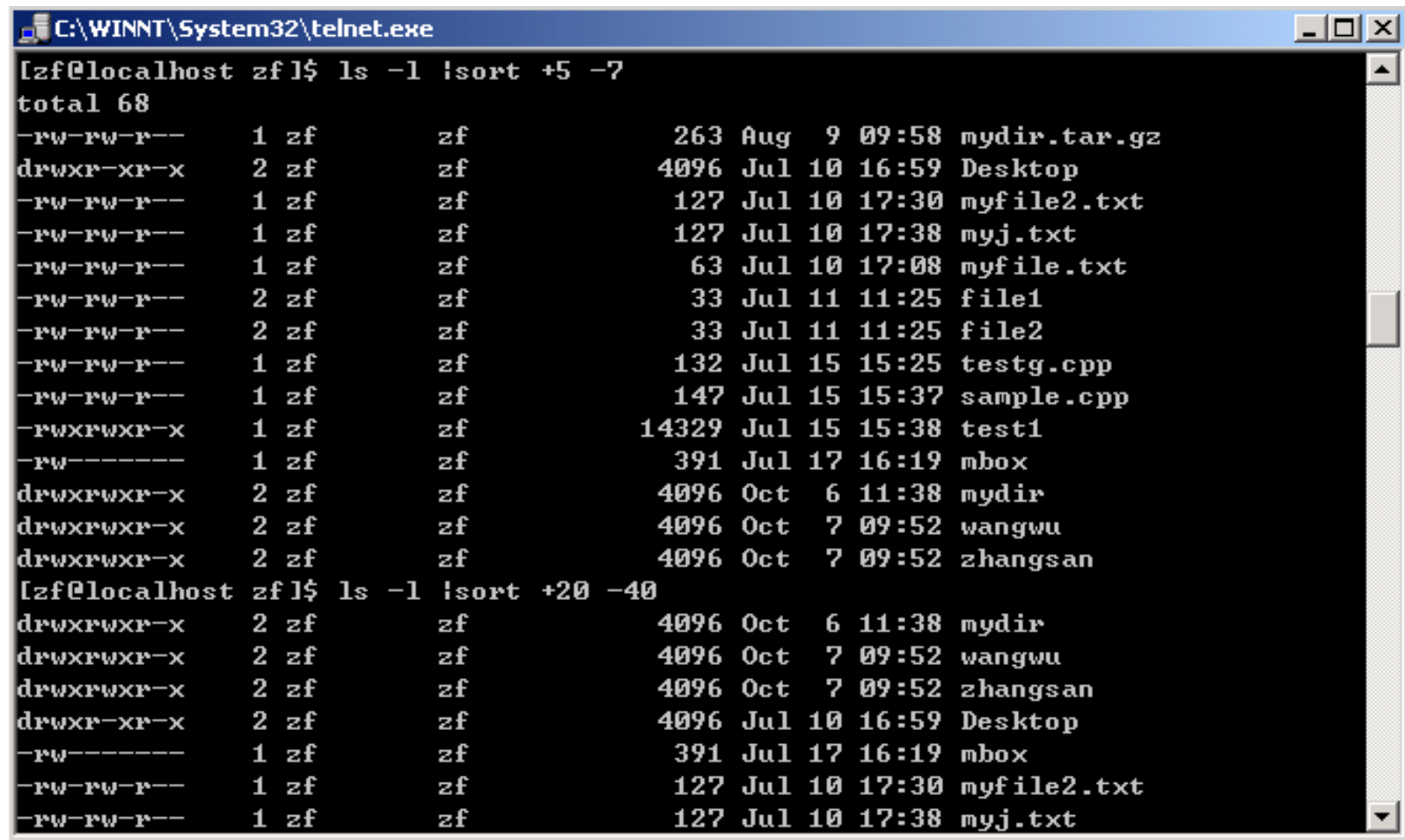
❖ 命令**find**用于在系统中搜索一个指定文件。



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ find /etc -name "pass*"
/etc/passwd
find: /etc/default: Permission denied
/etc/passwd-
/etc/pam.d/passwd
find: /etc/httpd/conf/ssl.crl: Permission denied
find: /etc/httpd/conf/ssl.crt: Permission denied
find: /etc/httpd/conf/ssl.csr: Permission denied
find: /etc/httpd/conf/ssl.key: Permission denied
find: /etc/httpd/conf/ssl.prm: Permission denied
find: /etc/linuxconf/archive: Permission denied
/etc/passwd.bak
[zf@localhost zf]$
```

对文件内容进行排序

❖ **sort命令** 对文件进行排序与合并。



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf] $ ls -l | sort +5 -7
total 68
-rw-rw-r-- 1 zf zf 263 Aug 9 09:58 mydir.tar.gz
drwxr-xr-x 2 zf zf 4096 Jul 10 16:59 Desktop
-rw-rw-r-- 1 zf zf 127 Jul 10 17:30 myfile2.txt
-rw-rw-r-- 1 zf zf 127 Jul 10 17:38 myj.txt
-rw-rw-r-- 1 zf zf 63 Jul 10 17:08 myfile.txt
-rw-rw-r-- 2 zf zf 33 Jul 11 11:25 file1
-rw-rw-r-- 2 zf zf 33 Jul 11 11:25 file2
-rw-rw-r-- 1 zf zf 132 Jul 15 15:25 testg.cpp
-rw-rw-r-- 1 zf zf 147 Jul 15 15:37 sample.cpp
-rwxrwxr-x 1 zf zf 14329 Jul 15 15:38 test1
-rw----- 1 zf zf 391 Jul 17 16:19 mbox
drwxrwxr-x 2 zf zf 4096 Oct 6 11:38 mydir
drwxrwxr-x 2 zf zf 4096 Oct 7 09:52 wangwu
drwxrwxr-x 2 zf zf 4096 Oct 7 09:52 zhangsan
[zf@localhost zf] $ ls -l | sort +20 -40
drwxrwxr-x 2 zf zf 4096 Oct 6 11:38 mydir
drwxrwxr-x 2 zf zf 4096 Oct 7 09:52 wangwu
drwxrwxr-x 2 zf zf 4096 Oct 7 09:52 zhangsan
drwxr-xr-x 2 zf zf 4096 Jul 10 16:59 Desktop
-rw----- 1 zf zf 391 Jul 17 16:19 mbox
-rw-rw-r-- 1 zf zf 127 Jul 10 17:30 myfile2.txt
-rw-rw-r-- 1 zf zf 127 Jul 10 17:38 myj.txt
```

对文件进行链接

- ❖ 当用户需要将不同的名字指定到同一个文件时，可以创建一个虚拟文件的链接指向它所链接的一些其它文件。
- ❖ 硬链接：
 - ❖ `$ln newfile hardlink`
 - ❖ `$`
- ❖ 软链接：
 - ❖ `$ln -s newfile softlink`
 - ❖ `$`

对文件进行链接



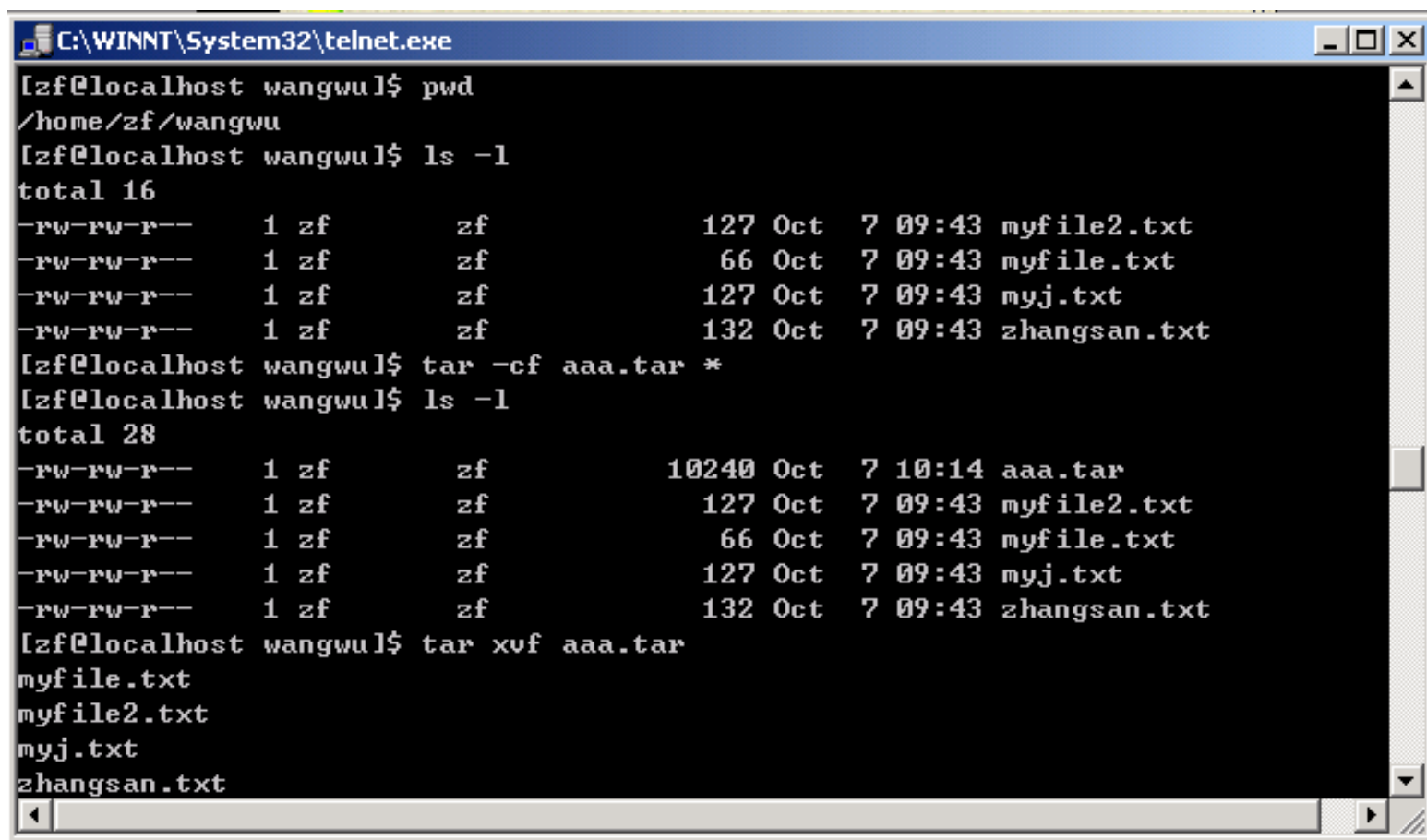
```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ cat testfile
This is a test!
The test make file link!
[zf@localhost zf]$ ls -l testfile
-rw-rw-r-- 1 zf      zf      41 Jun 11 16:17 testfile
[zf@localhost zf]$ ln testfile hard
[zf@localhost zf]$ ls -l testfile
-rw-rw-r-- 2 zf      zf      41 Jun 11 16:17 testfile
[zf@localhost zf]$ ls -l hard
-rw-rw-r-- 2 zf      zf      41 Jun 11 16:17 hard
[zf@localhost zf]$ ln -s testfile soft
[zf@localhost zf]$ ls -l testfile
-rw-rw-r-- 2 zf      zf      41 Jun 11 16:17 testfile
[zf@localhost zf]$ ls -l soft
lrwxrwxrwx 1 zf      zf      8 Jun 11 16:19 soft -> testfile
[zf@localhost zf]$
```

文件的归档和压缩

- ❖ **tar**命令是UNIX传统的备份和恢复方法。它的主要优点在于它可以跨很多厂商的UNIX计算机。
- ❖ `$ tar -cvf /dev/rmt/0m /home/zf /etc`
- ❖ 参数:
 - ❖ **-c** 表示生成备份
 - ❖ **-t** 列出备份内容
 - ❖ **-f** 后面跟一个文件名
 - ❖ **-x** 从一个**tar**备份中恢复数据

文件的归档和压缩

❖ tar命令

A screenshot of a telnet window titled 'C:\WINNT\System32\telnet.exe'. The window shows a terminal session with a user 'zf' at 'localhost' in the directory '/home/zf/wangwu'. The user runs 'pwd' and 'ls -l', showing four files: myfile2.txt, myfile.txt, myj.txt, and zhangsan.txt. Then, the user runs 'tar -cf aaa.tar *' to create an archive. A subsequent 'ls -l' shows the new file 'aaa.tar' added to the directory. Finally, the user runs 'tar xvf aaa.tar' to extract the files, which are listed one by one.

```
C:\WINNT\System32\telnet.exe
[zf@localhost wangwu]$ pwd
/home/zf/wangwu
[zf@localhost wangwu]$ ls -l
total 16
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf       66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf     127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf     132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ tar -cf aaa.tar *
[zf@localhost wangwu]$ ls -l
total 28
-rw-rw-r-- 1 zf      zf    10240 Oct  7 10:14 aaa.tar
-rw-rw-r-- 1 zf      zf     127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf       66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf     127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf     132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ tar xvf aaa.tar
myfile.txt
myfile2.txt
myj.txt
zhangsan.txt
```

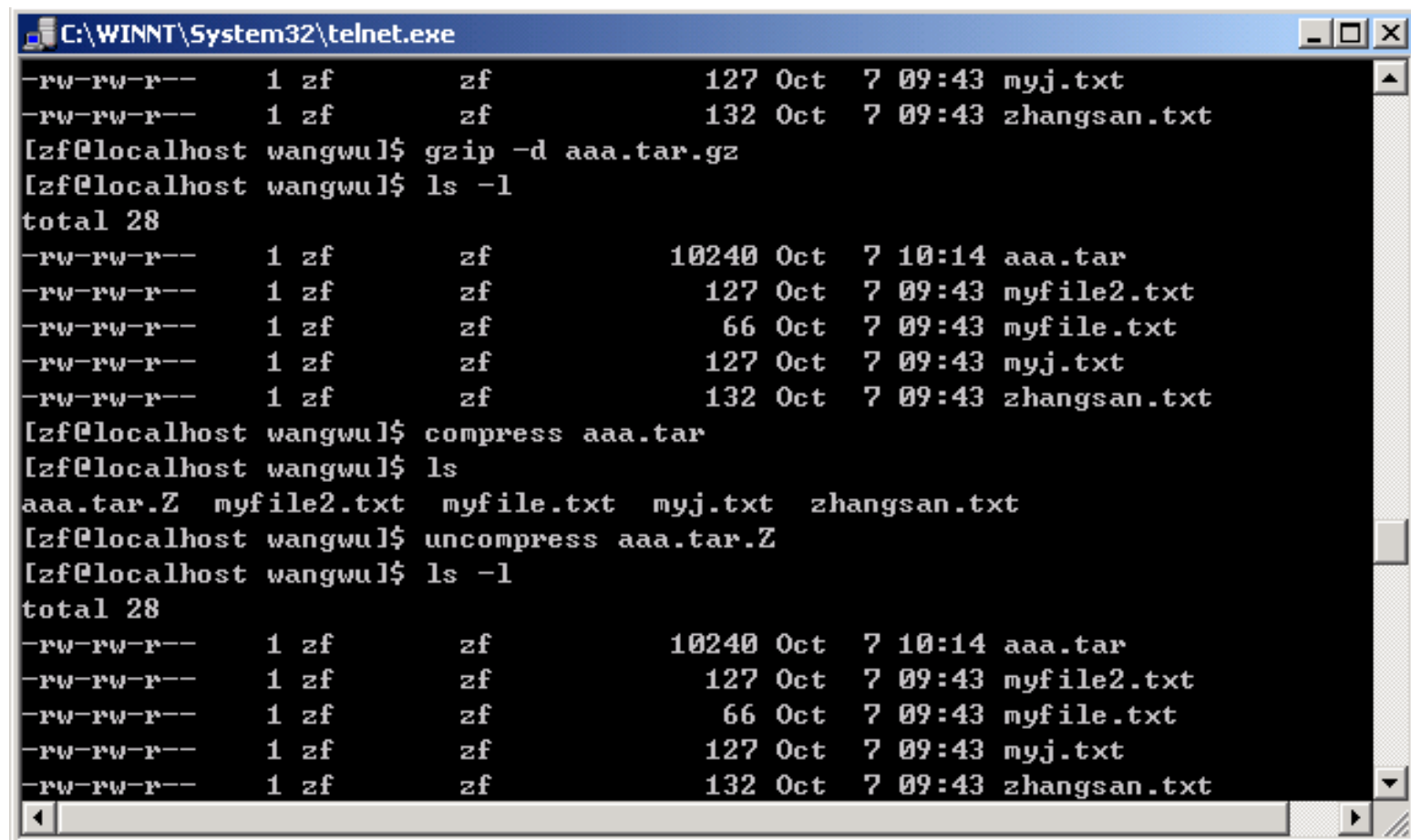
文件的归档和压缩

❖ gzip

```
C:\WINNT\System32\telnet.exe
[zf@localhost wangwu]$ ls -l
total 28
-rw-rw-r-- 1 zf      zf      10240 Oct  7 10:14 aaa.tar
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf       66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf      132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ gzip aaa.tar
[zf@localhost wangwu]$ ls -l
total 20
-rw-rw-r-- 1 zf      zf      265 Oct  7 10:14 aaa.tar.gz
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf       66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf      132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ gzip -d aaa.tar.gz
[zf@localhost wangwu]$ ls -l
total 28
-rw-rw-r-- 1 zf      zf      10240 Oct  7 10:14 aaa.tar
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf       66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf      127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf      132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$
```

文件的归档和压缩

❖ compress



```
C:\WINNT\System32\telnet.exe
-rw-rw-r-- 1 zf      zf          127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf          132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ gzip -d aaa.tar.gz
[zf@localhost wangwu]$ ls -l
total 28
-rw-rw-r-- 1 zf      zf          10240 Oct  7 10:14 aaa.tar
-rw-rw-r-- 1 zf      zf          127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf           66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf          127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf          132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ compress aaa.tar
[zf@localhost wangwu]$ ls
aaa.tar.Z  myfile2.txt  myfile.txt  myj.txt  zhangsan.txt
[zf@localhost wangwu]$ uncompress aaa.tar.Z
[zf@localhost wangwu]$ ls -l
total 28
-rw-rw-r-- 1 zf      zf          10240 Oct  7 10:14 aaa.tar
-rw-rw-r-- 1 zf      zf          127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf           66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf          127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf          132 Oct  7 09:43 zhangsan.txt
```

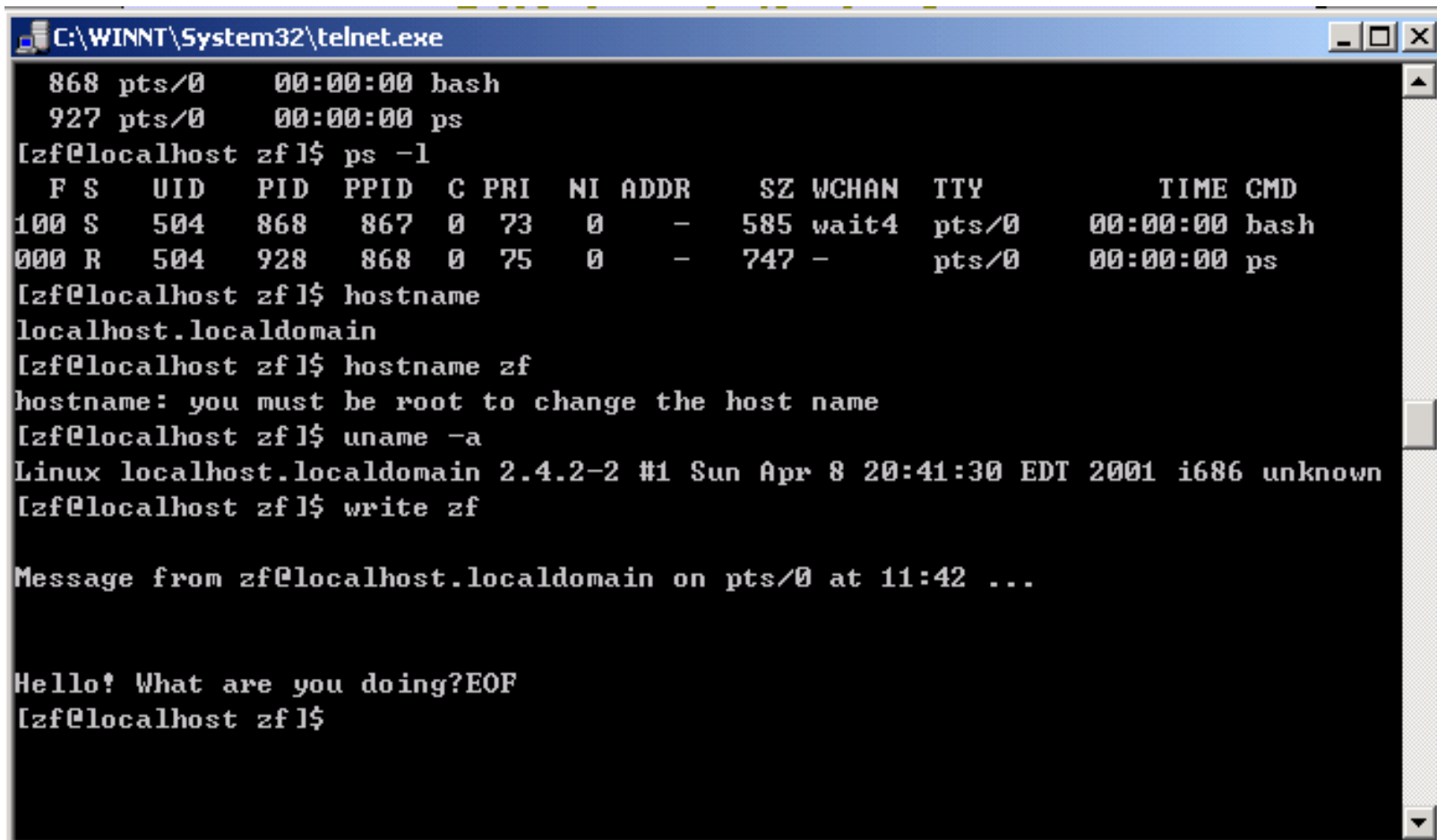

文件的归档和压缩

❖ zip

```
C:\WINNT\System32\telnet.exe
-rw-rw-r-- 1 zf      zf          132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ ls -l
total 28
-rw-rw-r-- 1 zf      zf          10240 Oct  7 10:14 aaa.tar
-rw-rw-r-- 1 zf      zf           127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf           66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf           127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf           132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ zip aaa.tar.gz aaa.tar
  adding: aaa.tar (deflated 98%)
[zf@localhost wangwu]$ ls -l
total 32
-rw-rw-r-- 1 zf      zf          10240 Oct  7 10:14 aaa.tar
-rw-rw-r-- 1 zf      zf           385 Oct  7 10:26 aaa.tar.gz
-rw-rw-r-- 1 zf      zf           127 Oct  7 09:43 myfile2.txt
-rw-rw-r-- 1 zf      zf           66 Oct  7 09:43 myfile.txt
-rw-rw-r-- 1 zf      zf           127 Oct  7 09:43 myj.txt
-rw-rw-r-- 1 zf      zf           132 Oct  7 09:43 zhangsan.txt
[zf@localhost wangwu]$ zip -f aaa.tar.gz
[zf@localhost wangwu]$ ls
aaa.tar  aaa.tar.gz  myfile2.txt  myfile.txt  myj.txt  zhangsan.txt
[zf@localhost wangwu]$
```

通信和网络命令

❖ write



```
C:\WINNT\System32\telnet.exe

868 pts/0      00:00:00 bash
927 pts/0      00:00:00 ps
[zf@localhost zf]$ ps -l
  F S      UID      PID  PPID  C PRI   NI ADDR      SZ WCHAN    TTY          TIME CMD
100 S      504      868    867   0  73    0  -       585 wait4   pts/0      00:00:00 bash
000 R      504      928    868   0  75    0  -       747 -        pts/0      00:00:00 ps
[zf@localhost zf]$ hostname
localhost.localdomain
[zf@localhost zf]$ hostname zf
hostname: you must be root to change the host name
[zf@localhost zf]$ uname -a
Linux localhost.localdomain 2.4.2-2 #1 Sun Apr 8 20:41:30 EDT 2001 i686 unknown
[zf@localhost zf]$ write zf

Message from zf@localhost.localdomain on pts/0 at 11:42 ...

Hello! What are you doing?EOF
[zf@localhost zf]$
```

通信和网络命令

❖ ping

```
C:\WINNT\System32\telnet.exe
Linux localhost.localdomain 2.4.2-2 #1 Sun Apr 8 20:41:30 EDT 2001 i686 unknown
[zf@localhost zf]$ write zf

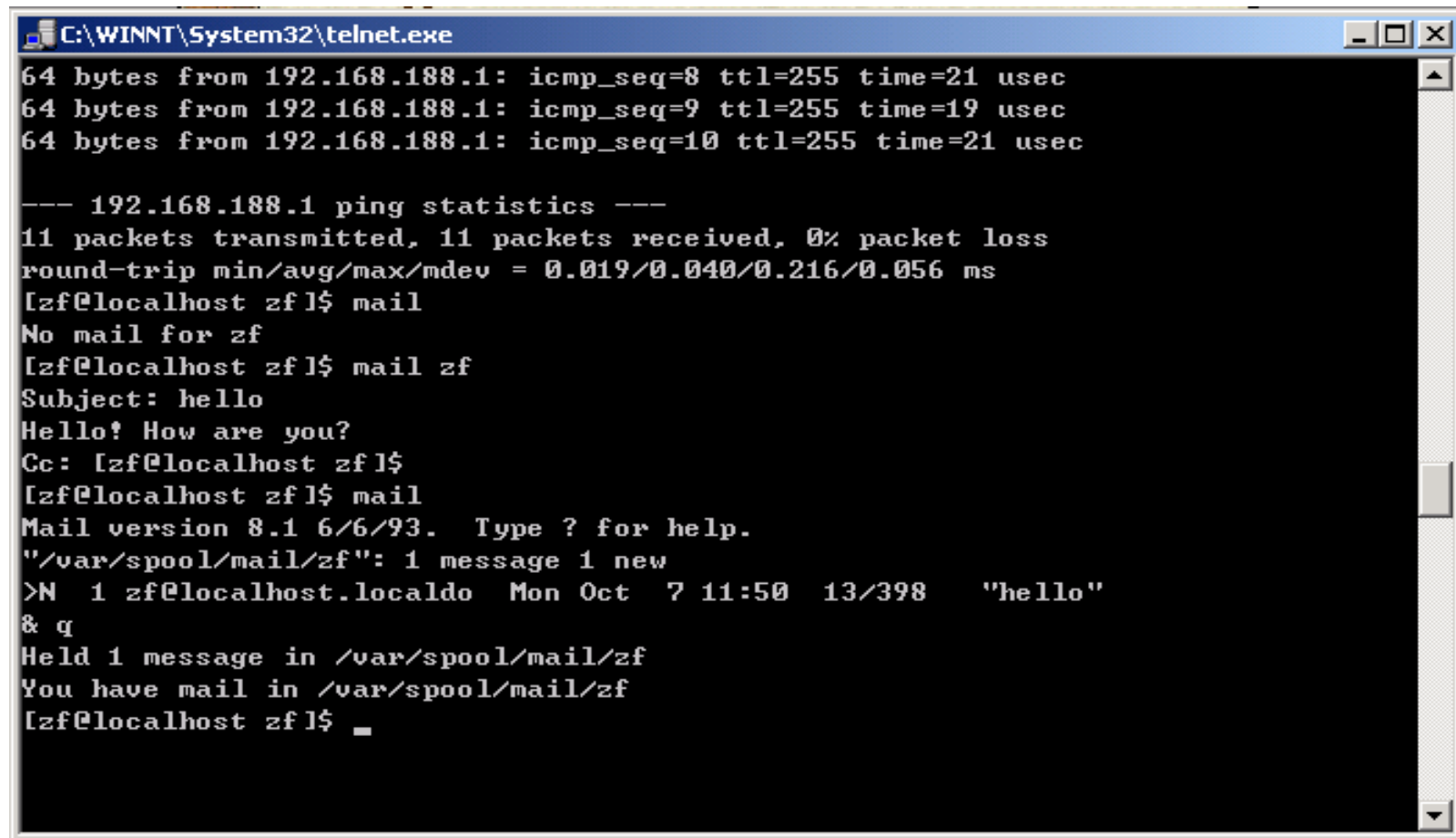
Message from zf@localhost.localdomain on pts/0 at 11:42 ...

Hello! What are you doing?EOF
[zf@localhost zf]$ ping 192.168.188.1
PING 192.168.188.1 (192.168.188.1) from 192.168.188.1 : 56(84) bytes of data.
Warning: time of day goes back, taking countermeasures.
64 bytes from 192.168.188.1: icmp_seq=0 ttl=255 time=216 usec
64 bytes from 192.168.188.1: icmp_seq=1 ttl=255 time=39 usec
64 bytes from 192.168.188.1: icmp_seq=2 ttl=255 time=21 usec
64 bytes from 192.168.188.1: icmp_seq=3 ttl=255 time=23 usec
64 bytes from 192.168.188.1: icmp_seq=4 ttl=255 time=21 usec
64 bytes from 192.168.188.1: icmp_seq=5 ttl=255 time=23 usec
64 bytes from 192.168.188.1: icmp_seq=6 ttl=255 time=22 usec
64 bytes from 192.168.188.1: icmp_seq=7 ttl=255 time=20 usec
64 bytes from 192.168.188.1: icmp_seq=8 ttl=255 time=21 usec
64 bytes from 192.168.188.1: icmp_seq=9 ttl=255 time=19 usec
64 bytes from 192.168.188.1: icmp_seq=10 ttl=255 time=21 usec

--- 192.168.188.1 ping statistics ---
11 packets transmitted, 11 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.019/0.040/0.216/0.056 ms
[zf@localhost zf]$
```

通信和网络命令

❖ mail

A screenshot of a Windows telnet window titled "C:\WINNT\System32\telnet.exe". The window shows a series of commands and their outputs. It starts with three ping commands to 192.168.188.1, followed by a ping statistics summary. Then, the user enters the 'mail' command, which shows 'No mail for zf'. Next, the user enters 'mail zf', which shows the subject 'hello' and a greeting 'Hello! How are you?'. The user then enters 'Cc: [zf@localhost zf]', followed by another 'mail' command. This leads to a mail version message and a list of messages, including one from 'zf@localhost.localdo' with subject 'hello'. Finally, the user enters '& q', and the system responds with 'Held 1 message in /var/spool/mail/zf' and 'You have mail in /var/spool/mail/zf'. The prompt returns to '[zf@localhost zf]\$_'.

```
C:\WINNT\System32\telnet.exe
64 bytes from 192.168.188.1: icmp_seq=8 ttl=255 time=21 usec
64 bytes from 192.168.188.1: icmp_seq=9 ttl=255 time=19 usec
64 bytes from 192.168.188.1: icmp_seq=10 ttl=255 time=21 usec

--- 192.168.188.1 ping statistics ---
11 packets transmitted, 11 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.019/0.040/0.216/0.056 ms
[zf@localhost zf]$ mail
No mail for zf
[zf@localhost zf]$ mail zf
Subject: hello
Hello! How are you?
Cc: [zf@localhost zf]$
[zf@localhost zf]$ mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/zf": 1 message 1 new
>N 1 zf@localhost.localdo  Mon Oct  7 11:50  13/398  "hello"
& q
Held 1 message in /var/spool/mail/zf
You have mail in /var/spool/mail/zf
[zf@localhost zf]$ _
```

输入/输出重定向

- ❖ 标准输入（通常是指用户的键盘）：是一个程序读取输入的地方。
- ❖ 标准输出（通常是指终端显示屏幕）：是命令或程序显示结果的地方；
- ❖ 标准出错输出（正常情况下，标准错误消息也显示在终端屏幕上）：是命令或程序显示输出错误信息的地方；
- ❖ **Linux Shell**可以按照用户的需要把这些数据流重新定向到一个文件、设备或其它命名上。我们把这个过程称为“输入/输出重定向”。

重定向

重定向分为： 输出重定向、输入重定向和错误重定向。

< 实现输入重定向。

<< 把其它内容追加到一个文件中。

>或>> 实现输出重定向，用户可以使用输出重定向把 一个命令的输出重定向到一个文件

1) ls -l /etc>dir

将ls命令生成的/tmp目录下的一个清单存到当前目录中的dir文件，而不在屏幕输出。

2) ls -l /usr>>dir

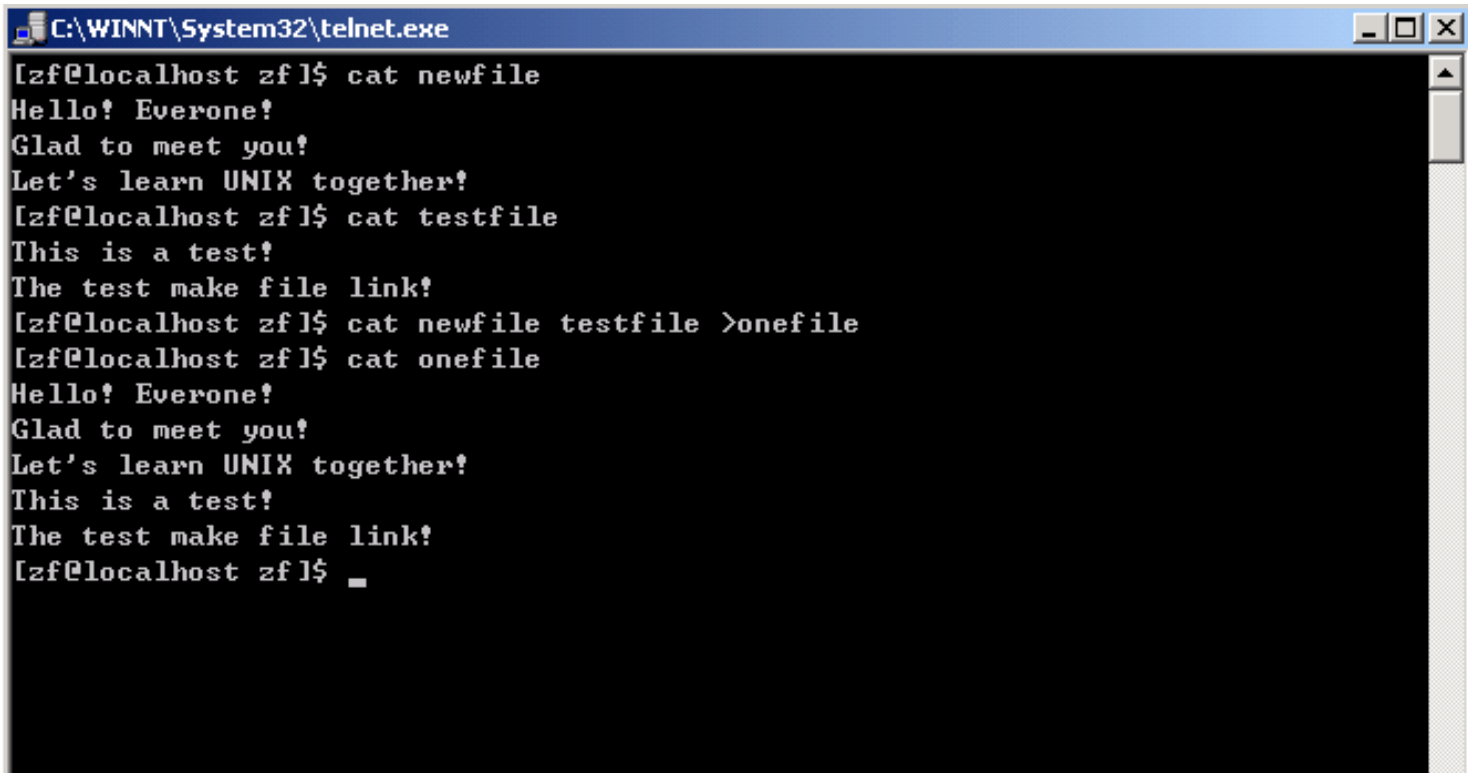
将ls命令生成的/usr目录的一个清单以追加的方式存到当前目录中的dir文件中。

重定向

```
选定 C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ ls -l >dir
[zf@localhost zf]$ more dir
total 100
-rwxrw-r--  1 zf      zf      179 Nov 22 16:51 break
-rwxrw-r--  1 zf      zf      410 Nov 22 15:58 case
-rw-----  1 zf      zf         1 Nov 25 15:25 dead.letter
drwxr-xr-x  2 zf      zf    4096 Jul 10 16:59 Desktop
-rw-rw-r--  1 zf      zf         0 Nov 27 16:35 dir
-rwxrw-r--  1 zf      zf     632 Nov 22 16:10 end_loop
-rwxrw-r--  1 zf      zf     180 Nov 22 15:02 expr1
-rwxrw-r--  1 zf      zf      78 Nov 22 16:26 for
-rwxrw-r--  1 zf      zf     115 Nov 22 16:31 for2
-rwxrw-r--  1 zf      zf     326 Nov 22 15:45 if
-rwxrw-r--  1 zf      zf     253 Nov 22 15:21 int_test
-rw-----  1 zf      zf     391 Jul 17 16:19 mbox
-rwxrw-r--  1 zf      zf     230 Nov 22 15:13 mycom
drwxrwxr-x  2 zf      zf    4096 Oct  6 11:38 mydir
-rw-rw-r--  1 zf      zf     263 Aug  9 09:58 mydir.tar.gz
-rwxrw-r--  1 zf      zf     150 Nov 22 10:19 read
-rwxrw-r--  1 zf      zf     136 Nov 22 16:14 shift
-rwxrwxr-x  1 zf      zf   14329 Jul 15 15:38 test1
-rwxrw-r--  1 zf      zf     120 Nov 22 09:48 var1
drwxrwxr-x  2 zf      zf    4096 Oct  7 10:26 wangwu
-rwxrw-r--  1 zf      zf     120 Nov 22 16:07 while
-rwxrw-r--  1 zf      zf     173 Nov 22 16:38 while2
--More--(95%)
```

重定向

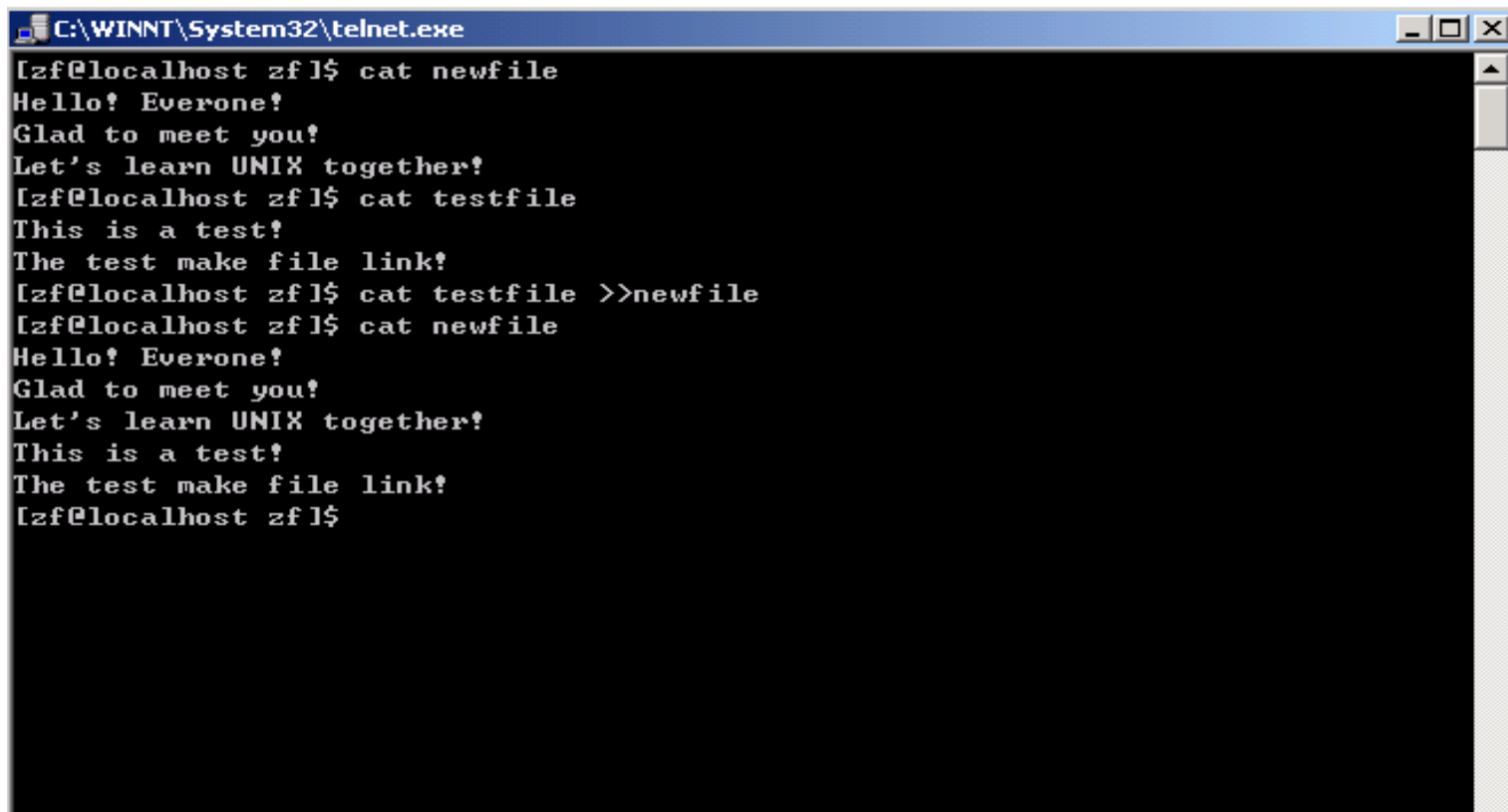
- ❖ 连接两个或多个文件
- ❖ 使用**cat**命令并重定向输出到一个文件可以连接两个或多个文件。

A screenshot of a Windows command prompt window titled "C:\WINNT\System32\telnet.exe". The window shows a telnet session with a user at the "zf@localhost" prompt. The user enters "cat newfile", and the output is "Hello! Everone!", "Glad to meet you!", and "Let's learn UNIX together!". Then, the user enters "cat testfile", and the output is "This is a test!" and "The test make file link!". Finally, the user enters "cat newfile testfile >onefile", followed by "cat onefile", which displays the combined output of both files. The session ends with the user entering a carriage return.

```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ cat newfile
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
[zf@localhost zf]$ cat testfile
This is a test!
The test make file link!
[zf@localhost zf]$ cat newfile testfile >onefile
[zf@localhost zf]$ cat onefile
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
This is a test!
The test make file link!
[zf@localhost zf]$
```


重定向

- ❖ 追加到一个文件
- ❖ 可以使用双重定向输出符号“>>”，保留文件以前的内容。这种情况下，命令输出追加到另一个文件中。

A screenshot of a Windows command prompt window titled "C:\WINNT\System32\telnet.exe". The window shows a telnet session with a user named 'zf' at 'localhost'. The user enters 'cat newfile' and sees the output: 'Hello! Everone!', 'Glad to meet you!', and 'Let's learn UNIX together!'. Then, the user enters 'cat testfile' and sees: 'This is a test!', 'The test make file link!'. Finally, the user enters 'cat testfile >>newfile' and then 'cat newfile', which shows the same output as 'cat testfile', demonstrating that the output of 'cat testfile' has been appended to 'newfile'.

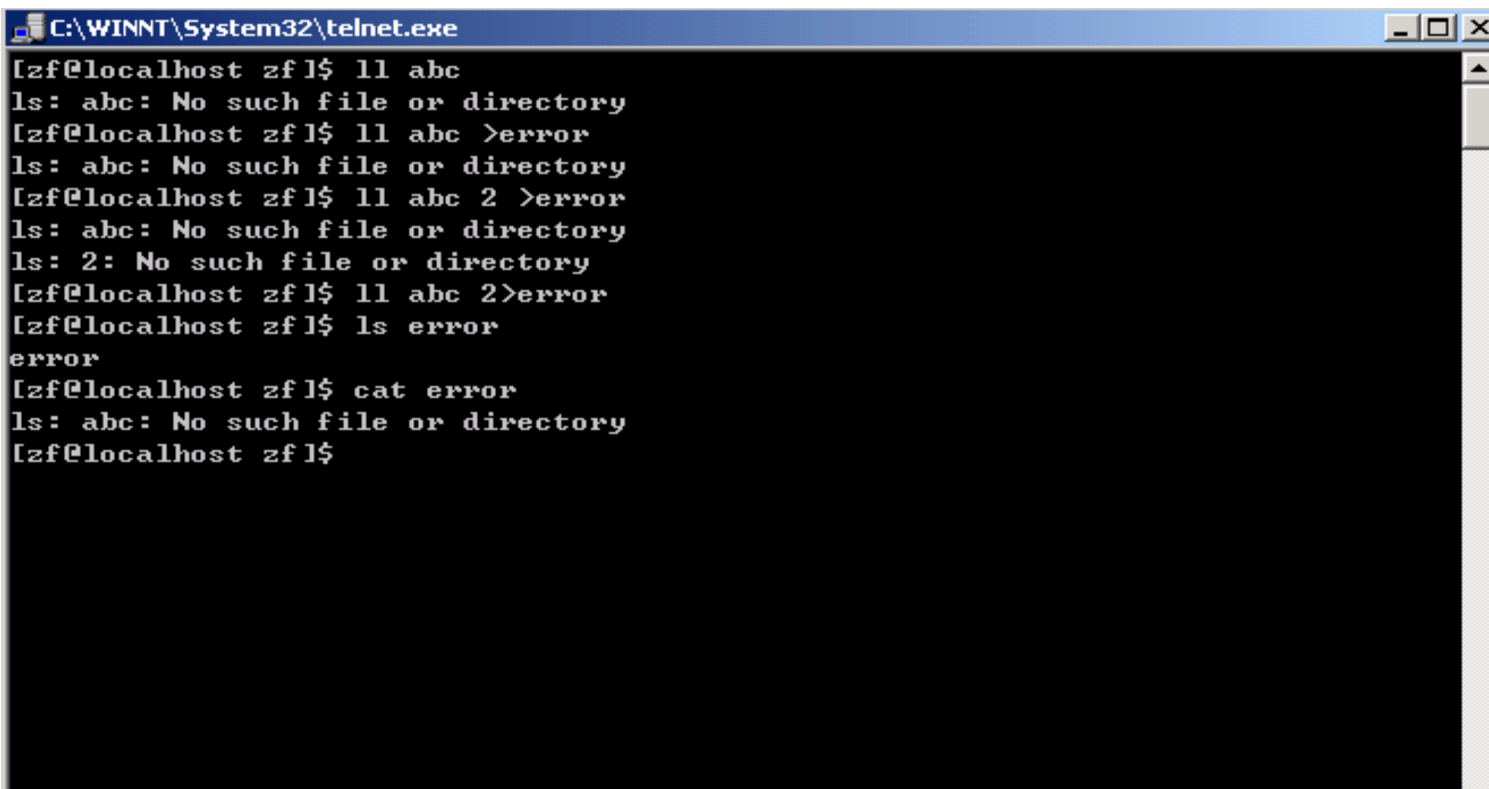
```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ cat newfile
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
[zf@localhost zf]$ cat testfile
This is a test!
The test make file link!
[zf@localhost zf]$ cat testfile >>newfile
[zf@localhost zf]$ cat newfile
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
This is a test!
The test make file link!
[zf@localhost zf]$
```

重定向

- ❖ 重定向标准输出到一个设备
- ❖ 除了重定向一个命令的输出到一个文件，也可以把它重定向到一个设备，因为UNIX系统将设备当做文件。
- ❖ `$echo "Hello! I am petter!" > /dev/tty01`
- ❖ 重定向标准输入
- ❖ 使用“<”重定向输入。
- ❖ 例如：用户已经创建好了一个文件**letter**。如果希望通过电子邮件发送给用户**petter**。可以使用下面方式：
- ❖ `$mail petter < letter`
- ❖ `$`

重定向

- ❖ 标准错误重定向
- ❖ 没有专门的符号用于重定向`stderr`。可以同样使用“<”或“>”符号，但需在它前面补一个数字2。



```
C:\WINNT\System32\telnet.exe
[zf@localhost zf]$ ll abc
ls: abc: No such file or directory
[zf@localhost zf]$ ll abc >error
ls: abc: No such file or directory
[zf@localhost zf]$ ll abc 2 >error
ls: abc: No such file or directory
ls: 2: No such file or directory
[zf@localhost zf]$ ll abc 2>error
[zf@localhost zf]$ ls error
error
[zf@localhost zf]$ cat error
ls: abc: No such file or directory
[zf@localhost zf]$
```

管道

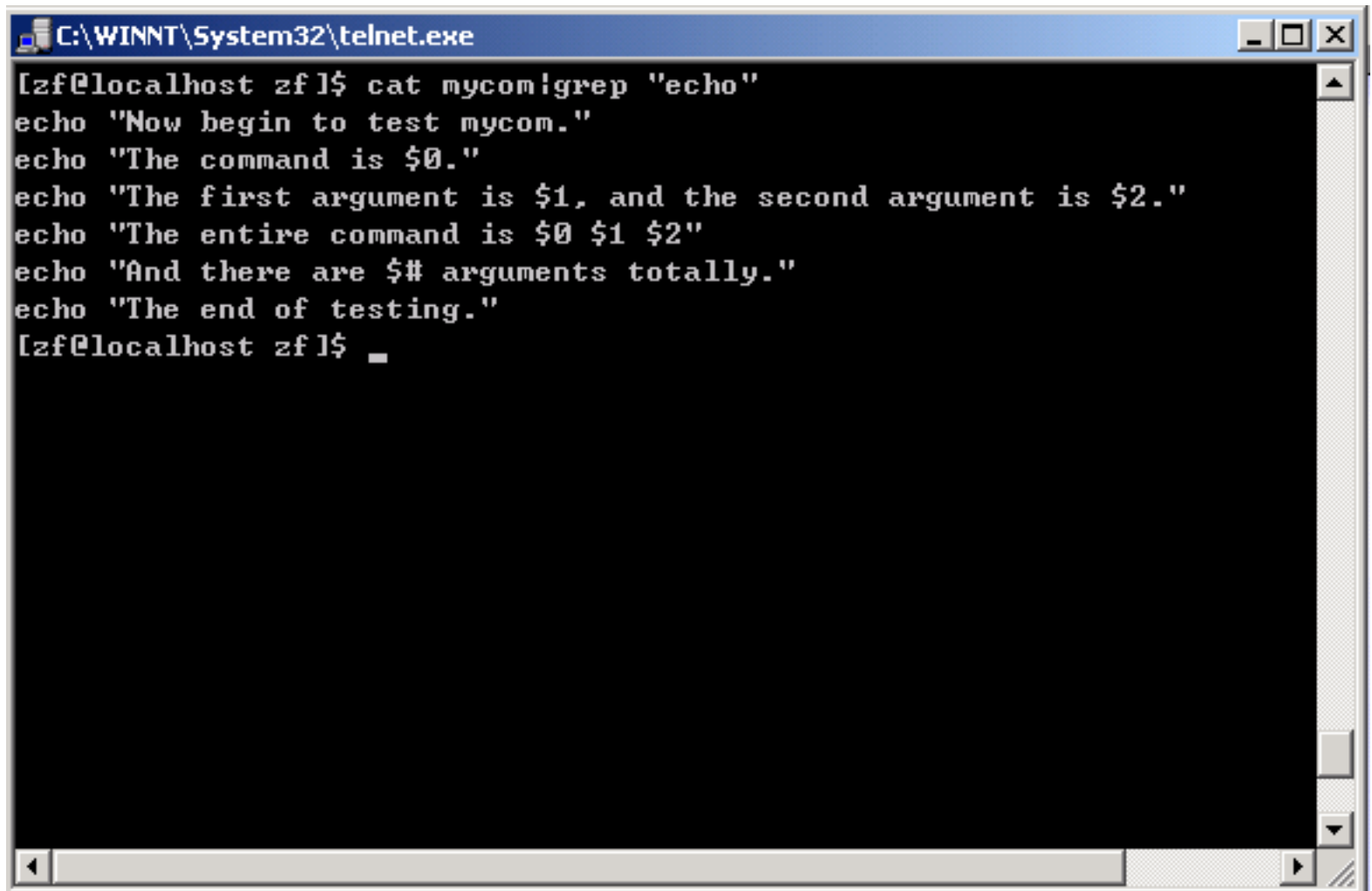
管道即将一条命令执行后产生的结果数据通过标准输出送给后一条命令，作为该命令的输入数据。

管道使用命令“|”将这些命令前后衔接在一起，形成一个管道线，格式为：

命令1|命令2|.....|命令3

管道线中的每一条命令都作为一个单独的进程运行，每一条命令的输出作为下一条命令的输入。命令执行的顺序是从左至右。

管道



A screenshot of a Windows telnet session window. The title bar reads "C:\WINNT\System32\telnet.exe". The window contains a black terminal area with white text. The text shows a user prompt "[zfc@localhost zfc]" followed by a command "cat mycom|grep 'echo'". The output consists of six lines: "echo 'Now begin to test mycom.'", "echo 'The command is \$0.'", "echo 'The first argument is \$1, and the second argument is \$2.'", "echo 'The entire command is \$0 \$1 \$2'", "echo 'And there are \$# arguments totally.'", and "echo 'The end of testing.'". The prompt returns to "[zfc@localhost zfc]" followed by a cursor. The window has standard Windows controls (minimize, maximize, close) in the top right and a scrollbar on the right side.

```
C:\WINNT\System32\telnet.exe

[zfc@localhost zfc]$ cat mycom|grep "echo"
echo "Now begin to test mycom."
echo "The command is $0."
echo "The first argument is $1, and the second argument is $2."
echo "The entire command is $0 $1 $2"
echo "And there are $# arguments totally."
echo "The end of testing."
[zfc@localhost zfc]$
```

管道

- ❖ 管道经常用于过滤、修改或维护一个命令的输出。

- ❖ 1、用管道做过滤器

- ❖ 很多时候不需要一个命令产生的全部输出。
这种情况下，用户可以从有命令产生的输出中过滤想要的信息。

- ❖ 2、用管道进行数据处理



管道

cat *

```
cat: cprogram: Is a directory
ls: abc: No such file or directory
This is a test!
The test make file link!
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
This is a test!
The test make file link!
cat: java: Is a directory
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
This is a test!
The test make file link!
Hello! Everone!
Glad to meet you!
Let's learn UNIX together!
This is a test!
The test make file link!
cat: shell: Is a directory
This is a test!
The test make file link!
cat: test: Is a directory
This is a test!
The test make file link!
[zf@localhost zf]$ _
```

管道

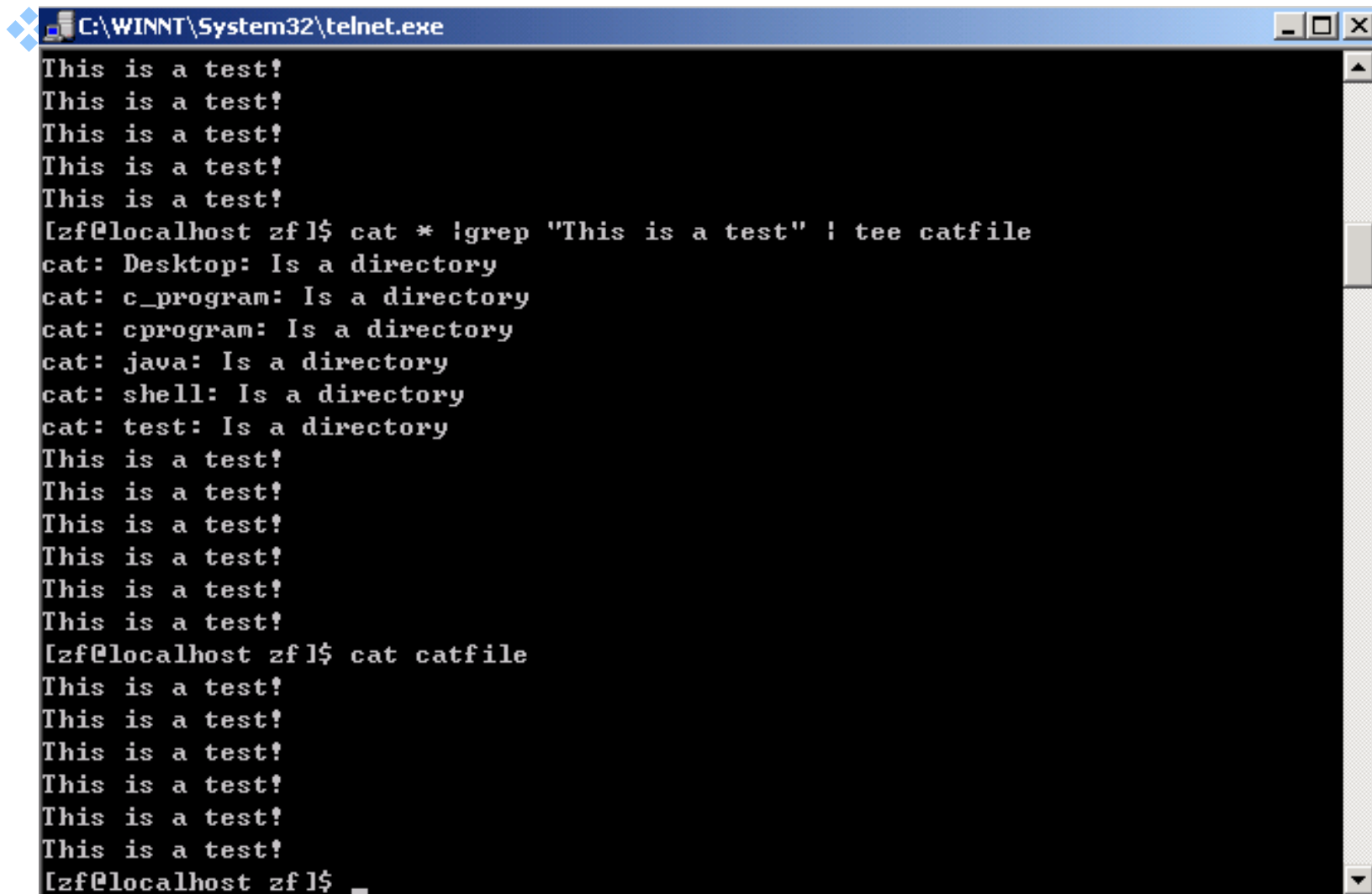


```
The test make file link!  
Hello! Everone!  
Glad to meet you!  
Let's learn UNIX together!  
This is a test!  
The test make file link!  
cat: shell: Is a directory  
This is a test!  
The test make file link!  
cat: test: Is a directory  
This is a test!  
The test make file link!  
[zf@localhost zf]$ cat * |grep "This is a test"  
cat: Desktop: Is a directory  
cat: c_program: Is a directory  
cat: cprogram: Is a directory  
cat: java: Is a directory  
cat: shell: Is a directory  
cat: test: Is a directory  
This is a test!  
This is a test!  
This is a test!  
This is a test!  
This is a test!  
This is a test!  
[zf@localhost zf]$
```


管道

- ❖ **T接头**
- ❖ 是一种特殊类型的管道，与现实生活中的T型管道接头很相似。
- ❖ 用于重定向一个管道数据到多个地方。
- ❖ **tee**命令用于首先一个T接头功能。它从**stdin**得到输入并把同样的内容写入**stdout**和别的文件中。

管道



```
C:\WINNT\System32\telnet.exe
This is a test!
This is a test!
This is a test!
This is a test!
This is a test!
[zf@localhost zf]$ cat * !grep "This is a test" ! tee catfile
cat: Desktop: Is a directory
cat: c_program: Is a directory
cat: cprogram: Is a directory
cat: java: Is a directory
cat: shell: Is a directory
cat: test: Is a directory
This is a test!
This is a test!
This is a test!
This is a test!
This is a test!
This is a test!
[zf@localhost zf]$ cat catfile
This is a test!
This is a test!
This is a test!
This is a test!
This is a test!
This is a test!
[zf@localhost zf]$
```

命令间隔符

一个命令行中可以若干条命令，各个命令之间可以用下表的字符进行间隔。

; 用 “;”间隔的各命令按顺序依次执行

&& 前后命令的执行存在“逻辑与”关系，只有&&前面的命令执行成功后，后面的命令才被执行

| | 前后命令的执行存在“逻辑或”关系，只有“| |”前面的命令执行失败，后面的命令才被执行。

命令间隔符（续一）

当几个命令的间隔符同时出现在同一个命令行上时，其优先级为：

- 1)** `;` 的优先级最低
- 2)** `|` 和 `&&` 具有相同的优先级
- 3)** 相同优先级，按从左到右的结合原则执行命令行
- 4)** 使用 “`()`” 可以组合命令行中的命令，改变执行顺序

命令间隔符（续二）

❖ 4) `date; cat file | wc`

只有`cat`命令的信息通过管道送给`wc`命令

5) `(date; cat file) | wc`

`date`和`cat`命令的信息都通过管道送给`wc`命令

命令间隔符（续三）



```
选定 C:\WINNT\System32\telnet.exe
sqlserver:x:1002:1002::/home/sqlserver:/bin/bash
[zf@localhost zf]$ cat /etc/passwd |grep zf
zf:x:504:504::/home/zf:/bin/bash
[zf@localhost zf]$ date ; pwd
Mon Oct 7 11:28:31 KST 2002
/home/zf
[zf@localhost zf]$ ld
ld: no input files
[zf@localhost zf]$ ls
Desktop  file2  mydir.tar.gz  myj.txt  testg.cpp
dir.txt  mbox   myfile2.txt   sample.cpp  wangwu
file1    mydir  myfile.txt    test1      zhangsan
[zf@localhost zf]$ date; pwd
Mon Oct 7 11:29:15 KST 2002
/home/zf
[zf@localhost zf]$ date;cat myfile.txt |wc
Mon Oct 7 11:29:44 KST 2002
      1      15      63
[zf@localhost zf]$ <date;cat myfile.txt> |wc
      2      21      92
[zf@localhost zf]$
```

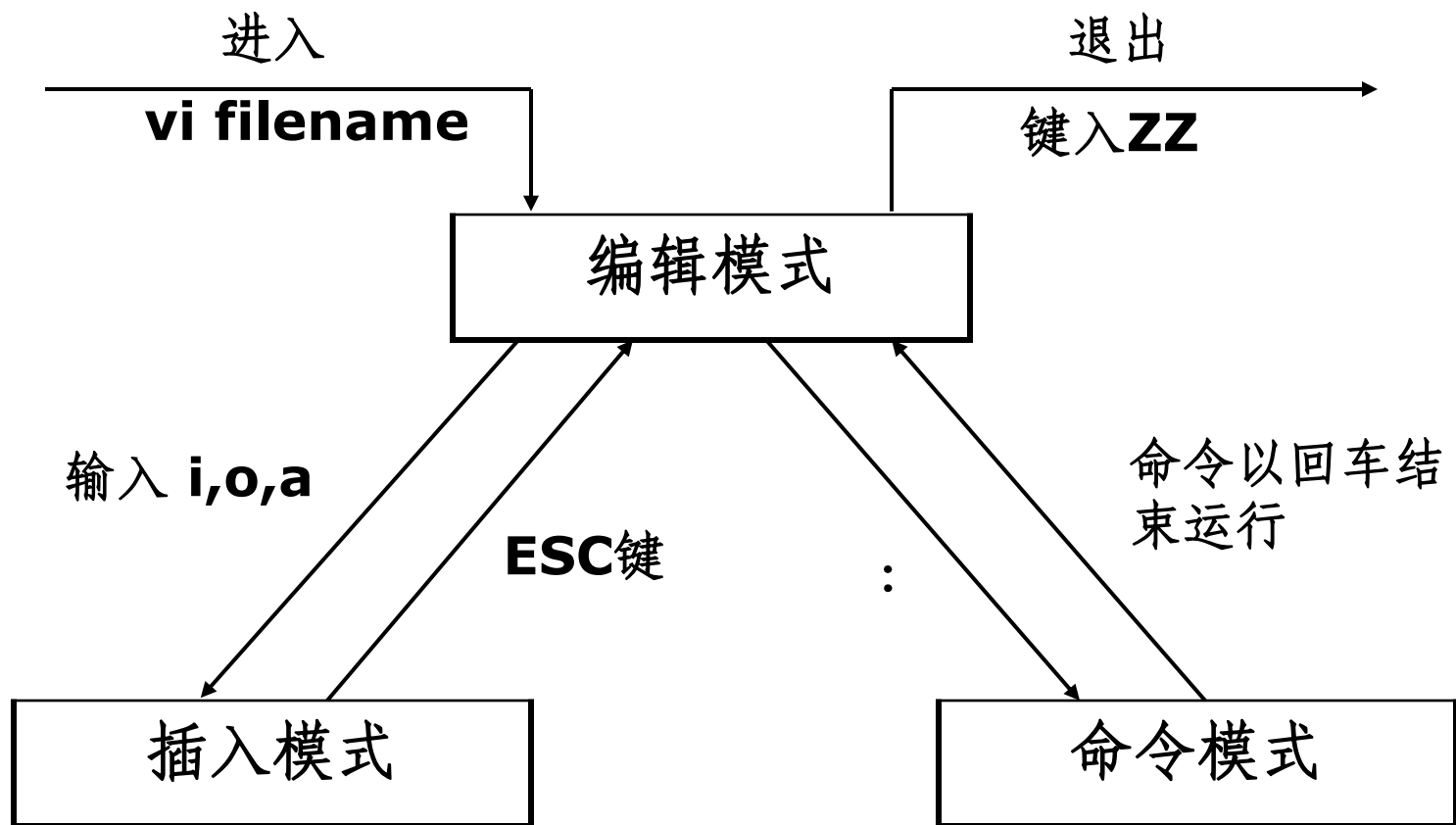
Vi 简介

Vi是Visual interface的简称，它是一个功能强大的全屏幕文本编辑器，是UNIX最重要的应用程序之一，它的作用是建立、编辑、显示ASCII码的文本文件。

Vi没有菜单，只有命令。

注意：Vi命令并不锁住所编辑的文件，因此多个用户可能同时编辑一个文件，那么最后保存的文件版本将被保留。

Vi 的三种运行模式



移动光标



命令	描述
j 或下箭头	向下移动一行
k 或上箭头	向上移动一行
h 或左箭头	左移一个字符
l 或右箭头	右移一个字符
w	右移一个词
W	右移一个以空格分隔的词
b	左移一个词
B	左移一个以空格分隔的词
0	移到行首

移动光标（续一）



命令	描述
\$	到行尾
(到句子的开头
)	到句子的末尾
{	到段落的开头
}	到段落的末尾
Ctrl-F	向前翻页
Ctrl-B	向后翻页
nG	到第n行
G	到最后一行

插入文本



命令	描述
i	在当前字符前插入文本
I	在行首插入文本
a	在当前字符后添加文本
A	在行末添加文本
o	在当前行后面插入一空行
O	在当前行前面插入一空行
R	以改写方式输入文本

删除、拷贝文本



命令	描述
x	删除光标处的字符
X	删除光标前面的字符
dd	删除光标所在的整行
d\$	删除从光标所在处开始到行尾的内容
d0	删除从光标前一个字符开始到行首的内容
db	删除前一个词
dG	删除当前行一直到文件结尾

文本替换命令



命令	描述
r	仍保持在命令模式替换当前字符
s	替换当前字符并进入插入模式
R	直到按下 Esc 键才替换多个字符
cw	修改下一个词的开头
cc	修改整个一行
cG	从文件的最后一行开始修改

查找和替换命令



命令	描述
/text	从当前光标位置开始向前查找text
?text	从当前光标位置开始向后查找text
/	向前重复前一次的查找
?	向后重复前一次的查找
n	在同一个方向重复查找
N	在反方向重复查找

文本查找和替换命令



命令	描述
:s/oldtext /newtext	向前查找 oldtext 并替换成 newtext
m,ns/oldtext /newtext	从 m 行到 n 行向前查找 oldtext 并替换成 newtext
:s/oldtext /newtext/g	在这个文件中查找 oldtext 并替换成 newtext
/<space>text	如果在/和 text 之间输入多个空格，只查找整个词
/^text	只在一行的开始查找
/text\$	只在一行的结尾查找
/ (多于一个单词)	用括弧查找多个单词

剪切、拷贝和粘贴命令



命令	描述
yy	拷贝当前行
nyy	从当前行位置开始拷贝 n 行
p	在当前光标位置后粘贴拷贝的文本
P	在当前光标位置前粘贴拷贝的文本
: m a	移动当前行并剪贴在第 a 行后面
:a, bmc	移动第 a 至 b 行并剪贴在第 c 行后面
:a, btc	拷贝第 a 至 b 行并剪贴在第 c 行后面

对vi选项的设置



命令	描述
: set number	在vi中为每一行设行号
: set all	列出所有有效的选项
:set autoindent	约束下一行使之与当前行有同样的字符数
: set readonly	设置当前文件为只读。不保存修改
:set wrapmargin=n	设置右侧装订线等于n.
: set showmode	显示用户所在的“插入”、“替换一个字符”或“替换”模式

撤消和重做命令

- ❖ 如果用户对文件的修改有误，可以撤消所做的修改。
- ❖ 使用命令“u”命令可以撤消刚才的修改。
- ❖ 也可以使用命令“U”重做对当前行的修改。
- ❖ 为了重做被撤消的内容，可以使用命令“.”。

导入和导出文本

- ❖ 导入：在特定的位置上将一个磁盘文件插入编辑器。
- ❖ 导出：打开多个文件，从一个打开的文件中拷贝文本并把它保存为磁盘上的另一个文件。
- ❖ 导入一个磁盘文件到编辑器：
- ❖ 为了插入一个磁盘文件到已打开文件的位置上，使用命令“**r filename**”。其中，**filename**是一个用完全路径命名的文件名。
- ❖ 从当前文件导出文本：
- ❖ 可以导入任意行使它们作为一个新文件并保存在磁盘中。使用命令“**w**”。例：
- ❖ **5,36 w filename**
- ❖ 表示：将文件的第**5**行至第**36**行所有的文本导出到文件名为**filename**的文件。

保存文件和退出Vi



命令	描述
ZZ	保存并退出
: w filename	写入文件
: w	写入文件
: wq	保存并退出
: x	保存并退出
: wq!	强制写入并退出
: q!	不保存文件，直接退出
: q	退出Vi

Linux的运行级别

- ❖ Linux 系统的运行分成很多不同的级别（**runlevel**）。所谓的运行级别实际上就是对系统进行的一定设置，使得在某个运行级别下系统会启动运行一组特定的进程。Linux系统常用的运行级别有七个。
- ❖ 系统的运行级别是在文件/**etc/inittab**中定义的，在此文件中描述了在每个运行级别下，系统将执行哪些启动脚本、启动哪些进程。系统启动后的第一个进程**init**将根据此文件中的设置来修建其他系统进程。

Linux的运行级别


代号	说明
0	所有进程将被终止，机器将有序的停止，关机时系统处于这个运行级别
1	单用户模式。用于系统维护，只有少数进程运行，同时所有服务也不启动
2	多用户模式。和运行级别3一样，只是网络文件系统（NFS）服务没被启动
3	多用户模式。允许多用户登录系统，是系统默认的启动级别
4	留给用户自定义的运行级别
5	多用户模式，并且在系统启动后运行X-Window，给出一个图形化的登录窗口
6	所有进程被终止，系统重新启动

Linux运行级别的切换

- ❖ 在`inittab`文件中，操作`initdefault`将在系统初始化之后启动预设的运行级别，用户可以通过更改此项设置来改变系统的预设运行级别。
- ❖ 用户也可以在系统运行过程当中来改变系统的运行级别，方法是用`init`命令，后面加上要切换到的运行级别。

编辑/etc/inittab文件切换单用户

```
# inittab      This file describes how the INIT process should set up
#              the system in a certain run-level.
#
# Author:      Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
#              Modified for RHS Linux by Marc Ewing and Donnie Barnes
#
#
# Default runlevel. The runlevels used by RHS are:
#  0 - halt (Do NOT set initdefault to this)
#  1 - Single user mode
#  2 - Multiuser, without NFS (The same as 3, if you do not have networking)
#  3 - Full multiuser mode
#  4 - unused
#  5 - X11
#  6 - reboot (Do NOT set initdefault to this)
#
id:3:initdefault:
```



- ❖ 在安装系统时，若选择文字模式登录系统，则此处的默认值为“3”。若选择直接启动窗口系统模式，则选择默认值为“5”，改成“1”会进入单用户模式。

Linux的关闭和重新启动

- shutdown命令
- halt命令
- reboot命令
- init命令

Shutdown命令选项

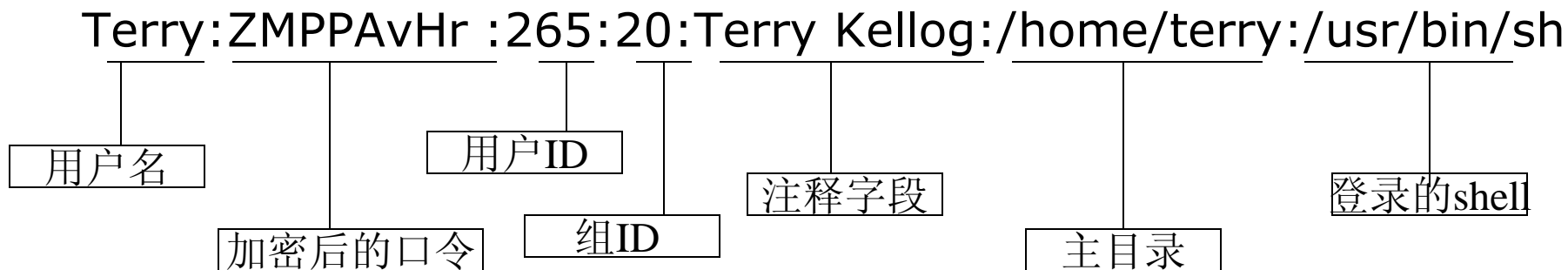
选项	说明
-t sec	等待sec秒后再通知init关闭系统
-k	并不真的关闭系统，只是给每个用户发送警告信息
-r	关闭之后重新启动系统
-h	关闭之后停滞系统
-f	重新启动系统后不用fsck检查磁盘
-F	重新启动系统后强制用fsck检查磁盘
Time	设置关闭系统的时刻，后面加上时间，格式是hh:mm
Warning-message	发给用户的警告信息

用户账号管理

❖ 1) 超级用户的权限

- 增加或删除一个用户的帐户
- 增加或删除用户对计算机资源的使用限制
- 设置系统的日期或时间
- 建立设备文件
- 创建、安装和卸载文件系统
- 修改进程的根目录
- 修改文件的属性
- 关闭系统

/etc/passwd文件



注意：passwd文件的权限应为只读（r—r—r--），所有者必须为root

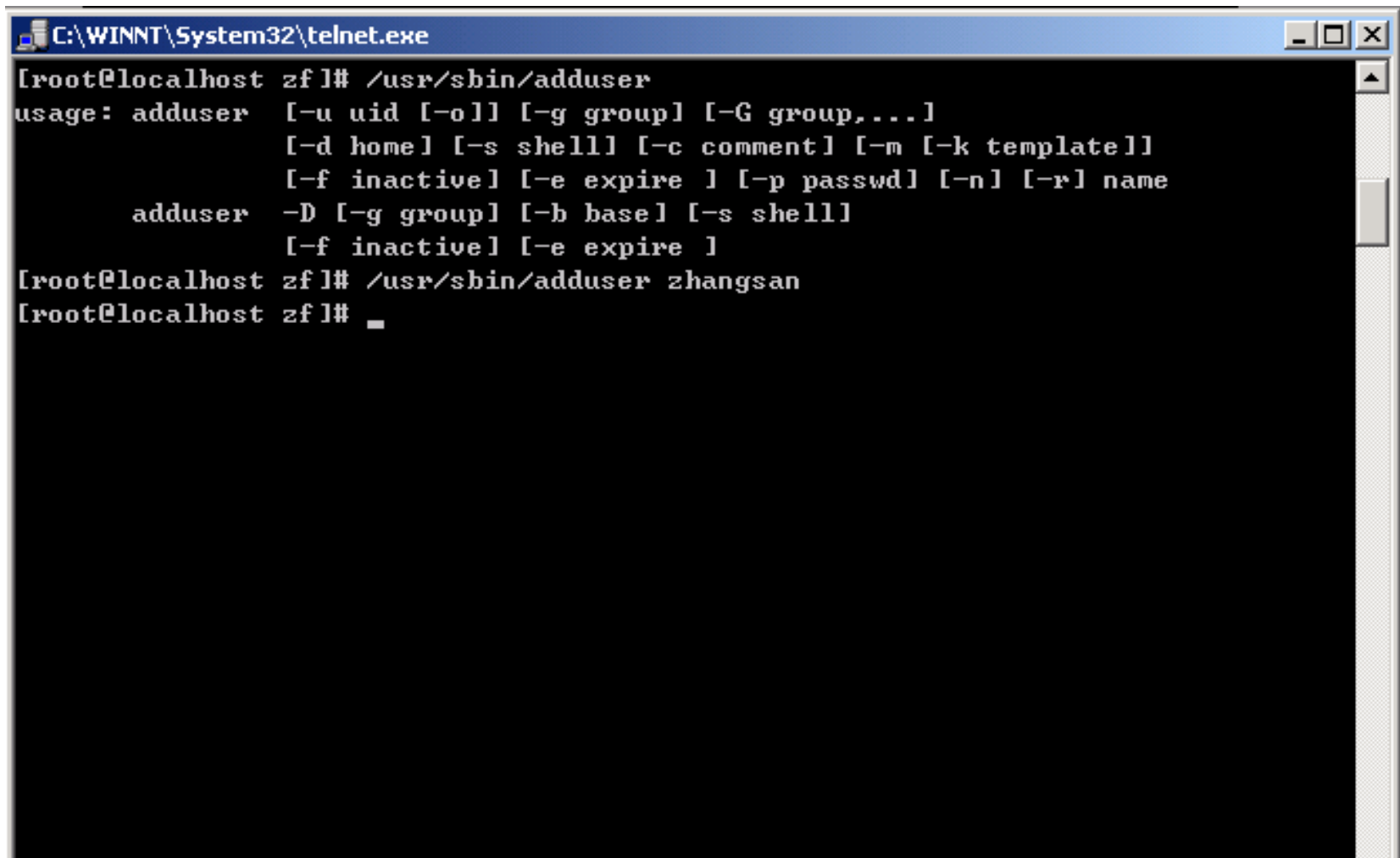
#vipw 命令 将/etc/passwd拷贝到/etc/passwd.tmp，并在/etc/passwd中加锁。

检查/etc/passwd文件用/usr/sbin/pwck。

用户管理

- ❖ 添加用户命令： **useradd**或**adduser**
- ❖ 格式： **useradd** 选项 用户名
- ❖ 说明：通常不使用任何选项，系统自动指定用户组、主目录等，无口令，禁用
- ❖ 步骤：
 - ❖ 1) 定义用户账号的标识信息：用户登录名、**uid**、缺省用户组名
 - ❖ 2) 指定用户账号的原始口令
 - ❖ 3) 指定用户的登录目录
 - ❖ 4) 将以上信息加入/**etc/passwd**中。

用户管理



```
C:\WINNT\System32\telnet.exe
[root@localhost zfl]# /usr/sbin/adduser
usage: adduser  [-u uid [-o]] [-g group] [-G group,...]
               [-d home] [-s shell] [-c comment] [-m [-k template]]
               [-f inactive] [-e expire ] [-p passwd] [-n] [-r] name
      adduser  -D [-g group] [-b base] [-s shell]
               [-f inactive] [-e expire ]
[root@localhost zfl]# /usr/sbin/adduser zhangsan
[root@localhost zfl]#
```

用户管理

❖ 可使用的选项:

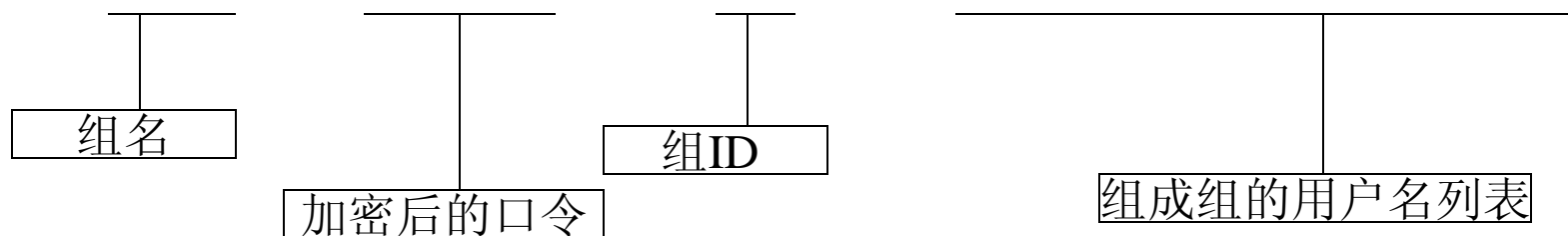
- ❖ **-r** 指定该用户为系统用户
- ❖ **-d 目录** 指定用户的主目录
- ❖ **-e 日期** 指定帐号的过期日期
- ❖ **-g 用户组** 指定主用户组，必须存在
- ❖ **-c 文本** 注释信息，一般指定用户全名

用户管理

- G 用户组,用户组... 指定所属的其他组
- m 若主目录不存在，则创建
- s shell路径 指定用户的登录shell
- u 用户号 指定用户号
- M 不创建主目录

/etc/group文件

sales:salesabc:102:bugs,daffy,elemer,marvin



注意：一个用户可能是多个组的成员，可以用`newgrp`命令改变到其他组。改变到新组并不改变用户在`/etc/passwd`文件中的主组。

检查`/etc/group`文件用`/usr/sbin/grpck`命令。

用户管理

- ❖ 添加用户组命令: **groupadd**
- ❖ 格式: **groupadd** 选项 用户组名
- ❖ 说明: 通常不使用任何选项, 系统自动指定用户组号
- ❖ 或使用命令:
- ❖ **newgrp**

用户管理

❖ 可使用的选项:

-r 指定该用户组为系统用户组，组号
小于500

-g 组号

❖ 相关文件: **passwd shadow group**

用户管理

删除用户命令： **userdel**

格式： **userdel** 选项 用户名

可使用的选项：

-r 把用户主目录中的文件一起删除

删除用户组命令： **groupdel**

格式： **groupdel** 用户组名

用户管理

修改用户帐号信息命令： **usermod**

格式： **usermod** 选项 帐号

功能： 修改帐号、口令、禁用帐号

可使用的选项：

-d 、 -g、 -G、 -p、 -s、 -u与useradd命令的选项意义相同，指定新值

-L 锁定口令，即禁用帐号,不能与p一起使用

-U 帐号解锁

用户管理

修改用户口令命令：**passwd** 选项 用户名

可使用的选项：

- s 报告口令状态
- l 锁定口令，即禁用帐号
- u 口令解锁
- d 使帐号无口令

例如：**passwd fred** 修改用户**fred**的口令

passwd -l fred 禁用帐号

用命令行修改用户密码

- ❖ \$ passwd
- ❖ Changing password for bugs
- ❖ Old password:
- ❖ New password:
- ❖ Re_enter new password:
- ❖ \$
- ❖ #passwd bugs
- ❖ New password:
- ❖ Re_enter new password:
- ❖ #

用户管理

修改用户组信息命令：**groupmod**

格式：**groupmod** 选项 组名

功能：修改用户组的组名和组号

可使用的选项：

-g 组号

把用户组号改为指定的组号

-n 组名

把用户组名改为指定的组名

用户管理

使当前用户成为超级用户：

su

然后输入超级用户口令即可

设置搜索路径：

编辑用户主目录中**.profile**文件的**PATH**

PATH=目录**1**:目录**2**:目录**3**: . . .

Export PATH

用户管理

显示系统信息命令：**uptime**

功能：显示当前时间、系统工作时间、
登录的用户数、作业数

显示用户身份的命令：**w**、**who**、**whoami**

w显示目前注册的用户及用户正运行的命令**who** 显示所有已登录用户

whoami 显示当前用户

用户管理

显示登录用户信息的命令：**finger**

格式：**finger** 选项 用户**1** 用户**2**@主机

选项：

- s** 显示详细信息

- l** 以多行形式显示

例如：

finger 显示所有已登录用户

finger -s fred

进程管理概念

- ✓ 进程是程序的一次动态执行过程，同一程序多次执行，则建立不同的进程
- ✓ **Linux**系统中所有操作都是通过进程来实现
- ✓ 进程划分为**3**类：
 - 核心态进程
 - 用户态进程
 - 后台进程

进程管理概念

系统类进程:

- ✓ 系统核心专用的特殊进程，可以通过键入命令：

ps -rc|grep SYS

看到哪些进程属于系统类进程

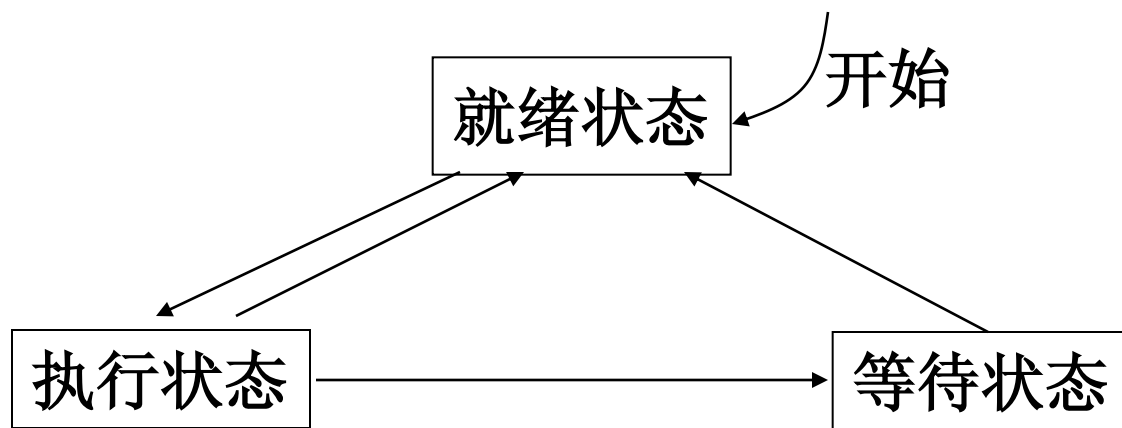
- ✓ 系统类进程采用固定优先级策略进行调度，其优先级由核心确定，且不改变

进程管理概念

进程状态：进程从被创建到终止这一时间内所处的位置

进程有三种状态：就绪、等待、执行

进程之间的转换见下图：

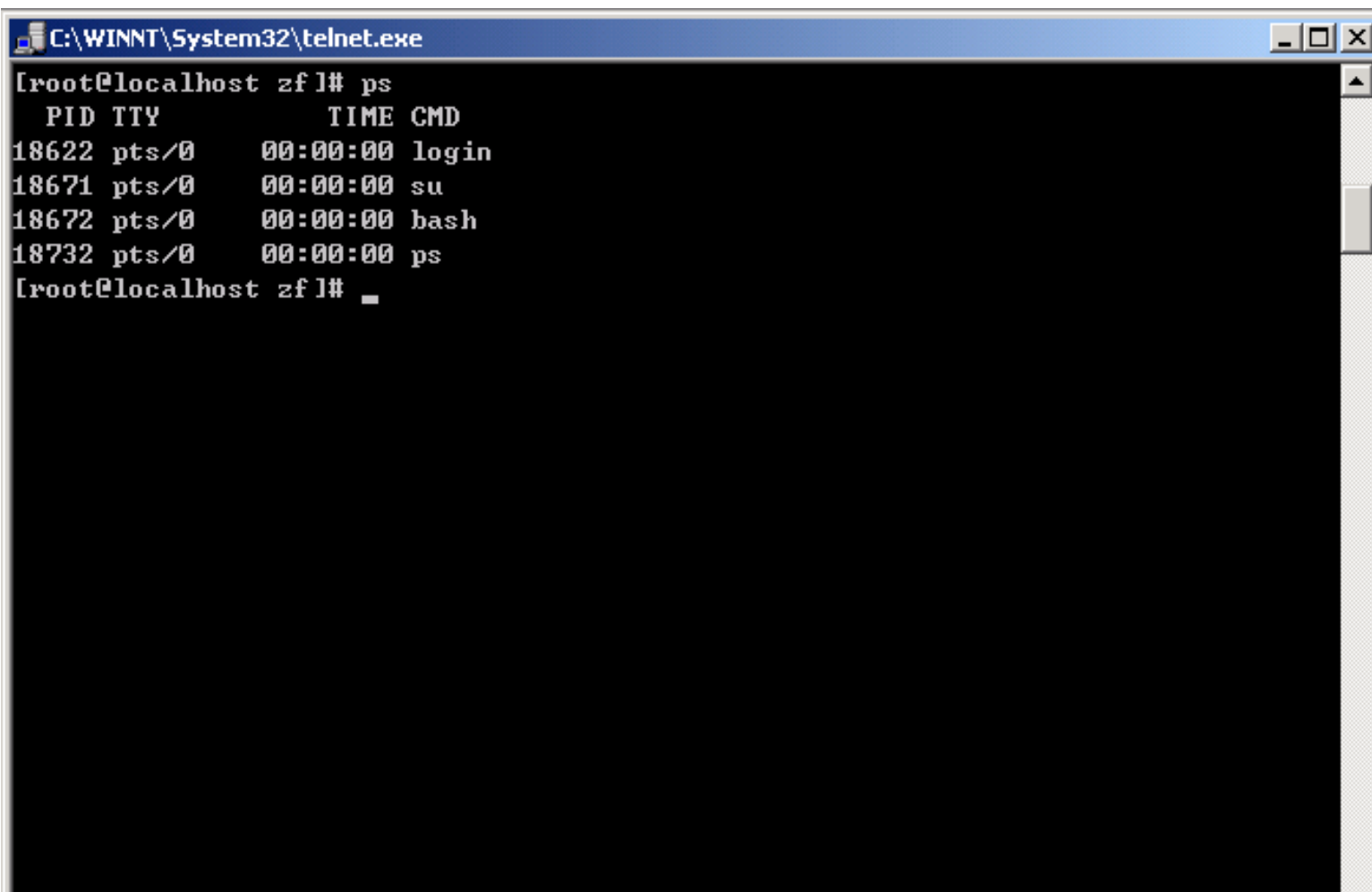


进程的状态转换

进程管理

- ❖ 进程状态查看命令：**ps**
- ❖ 格式：**ps** 选项
- ❖ 用于选择进程的选项：
 - A** 列出所有进程
 - a** 列出在某终端上运行的所有进程
 - r** 仅列出正在运行的进程

进程管理

A screenshot of a Windows telnet session window. The title bar reads 'C:\WINNT\System32\telnet.exe'. The terminal text shows a user at a root prompt on localhost typing 'ps', which returns a list of running processes including login, su, bash, and ps.

```
C:\WINNT\System32\telnet.exe
[root@localhost zfl]# ps
  PID TTY          TIME CMD
 18622 pts/0        00:00:00 login
 18671 pts/0        00:00:00 su
 18672 pts/0        00:00:00 bash
 18732 pts/0        00:00:00 ps
[root@localhost zfl]#
```


进程管理

- C** 命令名 列出执行此命令的进程
- U** 用户 列出此用户运行的进程
- u** 用户 列出此用户运行的进程
- t tty** 列出在此**tty**上运行的进程

用于控制输出格式的选项：

- f** 以完全列表的形式输出
- l** 以长格式输出

进程管理

-H 输出进程层次关系

❖ 进程状态码:

R 正在运行

S 正在睡眠

T 已经停止

< 高优先级进程

N 低优先级进程

进程管理

进程通常在前台执行，可以用**&**命令使进程在后台执行

&命令格式：

命令名**&**

查看后台进程：**jobs**

使进程转到前台执行：**fg** 作业号

使进程在后台继续执行：**bg** 作业号

进程管理

进程的实时监控命令命令：**top**

功能：以交互界面的方式，对进程进行实时持续的监控

格式：**top** 选项

d 秒数 设置查看的频率

p 进程号 仅对指定的进程进行监控

q 没有时间间隔地查看

进程管理

- s** 以安全模式查看，防止交互式命令带来的危险
- i** 不显示空闲的进程
- c** 显示完整的命令行
- n** 次数 以既定频率查看指定的次数后退出

❖ 在**top**的界面中可以使用的交互式命令：

空格键 立即查看进程状态

进程管理

k	结束一个进程
i	取消空闲进程的显示
n	显示指定数量的进程
s	改变查看的频率
Ctrl+l	清除屏幕，并重新显示

进程管理

❖ 结束进程命令：**kill**

kill 进程号

结束进程的执行

kill -s 信号 进程号

给指定进程发出信号

kill -p 进程号

得到指定进程的名字

kill -l

得到所有的信号

kill -l 数字

得到指定数字对应的信号

❖ 结束所有进程命令：**killall**

❖ 使进程转入睡眠：**sleep**

进程管理

进程通常在前台执行，可以用**&**命令使进程在后台执行

&命令格式：

命令名**&**

查看后台进程：**jobs**

使进程转到前台执行：**fg** 作业号

使进程在后台继续执行：**bg** 作业号

进程管理

进程的实时监控命令命令：**top**

功能：以交互界面的方式，对进程进行实时持续的监控

格式：**top** 选项

d 秒数 设置查看的频率

p 进程号 仅对指定的进程进行监控

q 没有时间间隔地查看

进程管理

❖ 结束进程命令：**kill**

kill 进程号

结束进程的执行

kill -s 信号 进程号

给指定进程发出信号

kill -p 进程号

得到指定进程的名字

kill -l

得到所有的信号

kill -l 数字

得到指定数字对应的信号

❖ 结束所有进程命令：**killall**

❖ 使进程转入睡眠：**sleep**

系统状态监控

- ❖ 显示用户标识 id
- ❖ 报告用户名、用户id及用户组id。

作业调度

at命令：使系统在指定的时间执行调度命令

格式：**at** 时间 ;进入**at**命令提示符

at 时间 **-f** 文件

例如：**at 2300**

at 18:20 ; **at**命令提示符下输入命令

at now + 2 hours -f my_jobs

at 6:30 next week -f my_jobs

作业调度

crontab命令：使系统在指定的时间周期性地执行调度命令

格式： **crontab -l** 列出调度任务

crontab -e 编辑调度任务

crontab -r 删除调度任务

Crontab文件的格式：

分 时 日 月 星期 命令

Thank You !

zhaofang@email.buptsse.cn

