
实验一 LINUX 及其使用环境

(一) LINUX 常用命令简介

一、实验目的

目的：

- 1、了解LINUX的Shell命令及使用格式。
- 2、熟悉LINUX的常用基本命令。
- 3、学会如何得到帮助信息。

要求：

1. 通过WINDOWS操作系统中的远程登录程序telnet.exe 登录linux。

2. 使用man命令

使用man命令来获得每个Linux命令的帮助手册

用man ls, man passwd, man pwd命令得到ls、passwd、pwd三个命令的帮助手册。

3. 用who 命令显示当前正在你的Linux系统中使用的用户名字：

- 1) 有多少用户正在使用你的Linux系统？给出显示的结果

- 2) 哪个用户登录的时间最长？给出该用户登录的时间和日期。

4. 使用下面的命令显示有关你计算机系统信息：uname（显示操作系统的名称），uname -n（显示系统域名），uname -p（显示系统的CPU名称）

- 1) 你的操作系统名字是什么？

- 2) 你计算机系统的域名是什么？

- 3) 你计算机系统的CPU名字是什么？

5. 使用whoami命令找到用户名。然后使用who -a命令来看看你的用户名和同一系统其他用户的列表。

6. 使用passwd命令修改你的登录密码。

7. 用命令date显示当前的时间，给出显示的结果。

8. 用pwd显示你的主目录(home directory)名字，给出pwd显示的结果。

9. 使用uptime 命令判断系统已启动运行的时间和当前系统中有多少登录用户，给出显示的结果。

10. 退出系统

二、实验环境及相关情况（包含使用软件、实验设备、主要仪器及材料等）

Linux 发行版

三、实验内容

（一）Linux 的登录与退出

1、登录

在DOS环境下用MS提供的telnet程序（也可使用WINDOWS 自带的telnet图形界面程序或多功能的S-Term终端程序），可使PC作为终端（terminal）登录（login）Linux服务器（UNIX Server）。

（1）执行格式：

telnet hostname(主机名)

或 telnet 主机的IP地址

例： telnet 192.168.0.254

（2）步骤

login: (输入username)

password: (输入密码)

2、退出

在UNIX系统提示符\$下，输入logout或exit。

例：\$ logout

（二）UNIX命令格式

命令 [选项] [处理对象]

例：ls -la mydir

注意：（1）命令一般是小写字串。注意大小写有别；

（2）目录之间的分隔为(/)区别于DOS中的(\)

（3）命令的选项通常以减号(-)间隔，命令和选项之间需要加空格

（4）同一行可有数个命令，命令间应以分号隔开

（5）命令后加上(&)可使该命令后台（background）执行

（三）常用命令

1、目录操作

和DOS相似，UNIX采用树型目录管理结构，由根目录 (/) 开始一层层将子目录建下去，各子目录以 / 隔开。用户login后，工作目录的位置称为 home directory，默认是已用户名为名的文件夹，例如用户mytest 的工作目录是在 /home/mytest文件夹。

UNIX的通配符有三种：'*' 和 '?' 用法与DOS相同， '-' 代表区间内的任一字符，如test[0-5]即代表test0，test1，……，test5的集合。

（1）显示目录文件 ls

执行格式： ls [-atFlR] [name] (name可为文件或目录名称)

例： ls 显示出当前目录下的文件

ls -a 显示出包含隐藏文件的所有文件

ls -t 按照文件最后修改时间显示文件
ls -F 显示出当前目录下的文件及其类型
ls -l 显示目录下所有文件的许可权、拥有者、文件大小、修改时间及名称

ls -lg 同上
ls -R 显示出该目录及其子目录下的文件

(2) 建新目录 mkdir

执行格式: mkdir directory-name

例: mkdir lilan (新建一名为lilan的目录)

(3) 删除目录 rmdir

执行格式: rmdir directory-name 或 rm directory-name

例: rmdir lilan 删除目录 lilan, 但它必须是空目录, 否则无法删除

 rm -r lilan 删除目录 dir1 及其下所有文件及子目录

 rm -rf lilan 不管是否空目录, 统统删除, 而且不给出提示,使用时要小心

(4) 改变工作目录位置 cd

执行格式: cd [name]

例: cd 改变目录位置至用户 login 时的 working directory

 cd lilan 改变目录位置, 至lilan目录

 cd .. 改变目录位置, 至当前目录的上层目录

 cd ../user 改变目录位置, 至上一级目录下的user目录

 cd /dir-name1/dir-name2 改变目录位置, 至绝对路径 (Full path)

 cd - 回到进入当前目录前的上一个目录

(5) 显示当前所在目录 pwd

执行格式: pwd

(6) 显示环境变量

 echo \$HOME 显示父目录

 echo \$PATH 显示可执行文件搜索路径

 env 显示所有环境变量(可能很多,最好用"env | more","env |

grep PATH"等)

2、文件操作

(1) 查看文件(可以是二进制的)内容 cat

执行格式: cat filename 或 more filename 或 cat filename|more

例: cat file1 以连续显示方式, 查看文件 file1 的内容

 more file1

或 `cat file1|more` 以分页方式查看文件的内容，按回车显示下一行，按空格显示下一屏

(2) 删除文件 `rm`

执行格式: `rm filename`

例: `rm file?`
`rm f*`

(3) 复制文件 `cp`

执行格式: `cp [-r] source destination`

例: `cp file1 file2` 将 `file1` 复制成 `file2`
`cp file1 dir1` 将 `file1` 复制到目录 `dir1`
`cp /tmp/file1` 将 `file1` 复制到当前目录
`cp /tmp/file1 file2` 将 `file1` 复制到当前目录名为 `file2`
`cp -r dir1 dir2` (recursive copy)复制整个目录。

(4) 移动或更改文件、目录名称 `mv`

执行格式: `mv source destination`

例: `mv file1 file2` 将文件 `file1`，更名为 `file2`
`mv file1 dir1` 将文件 `file1`，移到目录 `dir1` 下
`mv dir1 dir2`

(5) 比较文件(可以是二进制的)或目录的内容 `diff`

执行格式: `diff [-r] name1 name2` (name1、name2 同为文件或目录)

例: `diff file1 file2` 比较 `file1` 与 `file2` 的不同处
`diff -r dir1 dir2` 比较 `dir1` 与 `dir2` 的不同处

(6) 文件中字符串的查找 `grep`

执行格式: `grep string file`

例: `grep abc file1` //查找并列出串 `abc` 所在的整行文字

(7) 建立文件或目录的链接 `ln`

例: `ln source target1` 建立 `source` 文件(已存在)的硬链接,命名为 `target1`

`ln -s source target2` 建立 `source` 文件的符号链接,命名为 `target2`

3、系统询问与权限口令

(1) 查看系统中的使用者

执行格式: `who`

(2) 查看 `username`

执行格式: `who am I` 查看自己的 `username`

(3) 改变自己的 `username` 的帐号与口令 `su`

执行格式: `su username`

例:	su	username	输入帐号
		password	输入密码

(4) 文件属性的设置 chmod

改变文件或目录的读、写、执行的允许权

执行格式: chmod [-R] mode name

其中: [-R]为递归处理,将指定目录下所有文件及子目录一并处理

mode 为 3-8 位数字, 是文件/目录读、写、执行允许权的缩写(r:read,数字代号为"4", w:write,数字代号为"2", x:execute,数字代号为"1")

mode:	rwX	rwX	rwX
	user	group	other
缩写:	(u)	(g)	(o)

例: chmod 755 dir1 将目录dir1设定成任何人皆有读取及执行的权利, 但只有拥有者可作写修改。其中7=4+2+1,5=4+1

chmod 700 file1 将 file1 设为拥有者可以读、写和执行

chmod o+x file2 将 file2, 增加其它使用者可执行的权利

chmod g+x file3 将 file3, 增加组使用者可执行的权利

chmod o-r file4 将 file4, 除去其它使用者可读取的权利

(5) 改变文件或目录所有权 chown

执行格式: chown [-R] username name

例: chown user file1 将文件 file1 改为 user 所有

chown .fox file1 将文件file1改为fox组所有

chown user.fox file1 将文件file1改为fox组的user所有

chown -R user dir1 将目录 dir1 及其下所有文件和子目录, 改为 user 所有

(6) 检查用户所在组名称 groups

执行格式: groups

(7) 改变文件或目录的组拥有权 chgrp

执行格式: chgrp [-R] groupname name

例: chgrp vlsi file1 将文件 file1 改为 vlsi 组所有

chgrp -R image dir1 将目录 dir1 及其下所有文件和子目录, 改为 image 群组

(8) 改变文件或目录的最后修改时间 touch

执行格式: touch name

4、进程操作

(1) 查看系统目前的进程 ps

执行格式: ps [-aux]

例: `ps` 或 `ps -x` 查看系统中属于自己的 process
`ps -au` 查看系统中所有使用者的 process
`ps -aux` 查看系统中包含系统内部及所有使用者的 process
`ps -aux|grep apache` 找出系统中运行的所有名称中带有"apache"串
的 process

(2) 查看正在 background 中执行的 process

执行格式: `jobs`

(3) 结束或终止进程 `kill`

执行格式: `kill [-9] PID` (PID 为利用 `ps` 命令所查出的 process ID)

例: `kill 456`

或 `kill -9 456` 终止 process ID 为 456 的 process

(4) 后台 (background) 执行 process command 的命令

执行格式: `command &` (在命令后加上 `&`)

例: `gcc file1 &` 在后台编译 file1.c

注意: 按下 `^Z`, 暂停正在执行的 process。键入 `"bg"`, 将所暂停的 process 置入 background 中继续执行。

例: `gcc file1 &`

`^Z`

`stopped`

`bg`

(5) 显示系统中程序的执行状态

例: `top -q` 不断地更新、显示系统程序的执行状态

第一行显示的项目依次为当前时间、系统启动时间、当前系统登录用户数目、平均负载。

第二行为进程情况, 依次为进程总数、休眠进程数、运行进程数、僵死进程数、终止进程数。

第三行为 CPU 状态, 依次为用户占用、系统占用、优先进程占用、闲置进程占用。

第四行为内存状态, 依次为平均可用内存、已用内存、空闲内存、共享内存、缓存使用内存。

第五行为交换状态, 依次为平均可用交换容量、已用容量、闲置容量、高速缓存容量。

PID 每个进程的 ID。

PPID 每个进程的父进程 ID。

UID 每个进程所有者的 UID。

USER 每个进程所有者的用户名。

PRI 每个进程的优先级别。

NI 该进程的优先级值。

SIZE 该进程的代码大小加上数据大小再加上堆栈空间大小的总数。单位是 KB。

TSIZE 该进程的代码大小。

DSIZE 数据和堆栈的大小。

TRS 文本驻留大小。

D 被标记为“不干净”的页项目。

LIB 使用的库页的大小。对于 ELF 进程没有作用。

RSS 该进程占用的物理内存的总数量，单位是 KB。

SHARE 该进程使用共享内存的数量。

STAT 该进程的状态。其中 S 代表休眠状态；D 代表不可中断的休眠状态；R 代表运行状态；Z 代表僵死状态；T 代表停止或跟踪状态。

TIME 该进程自启动以来所占用的总 CPU 时间。如果进入的是累计模式，那么该时间还包括这个进程子进程所占用的时间。且标题会变成 CTIME。

%CPU 该进程自最近一次刷新以来所占用 CPU 时间和总时间的百分比。

%MEM 该进程占用的物理内存占总内存的百分比。

COMMAND 该进程的命令名称，如果一行显示不下，则会进行截取。内存中的进程会有一个完整的命令行。

注意：按"ctrl+c"停止查看

(6) 以树状图显示执行的程序 pstree

例： pstree -h 列出进程树并高亮标出当前执行的程序

(7) 监视虚拟内存 vmstat

vmstat 对系统的虚拟内存、进程、CPU 活动进行监视，同时它也对磁盘和 forks 和 vforks 操作的个数进行汇总。

不足是：vmstat 不能对某个进程进行深入分析，它仅是对系统的整体情况进行分析。

例如：[angel@home /angel]# vmstat

procs			memory				swap		io		system		cpu		
r	b	w	swap	free	buff	cache	si	so	bi	bo	in	cs	us	sy	id
0	0	0	7180	1852	56092	48400	0	0	6	5	24	8	0	0	18

其中：

Procs

r: 等待运行的进程数 b: 处在非中断睡眠状态的进程数 w: 被交换出去的可运行的进程数。

Memory

swap: 虚拟内存使用情况, 单位: KB free: 空闲的内存, 单位 KB

buff: 被用来做为缓存的内存数, 单位: KB

Swap

si: 从磁盘交换到内存的交换页数量, 单位: KB/秒 so: 从内存交换到磁盘的交换页数量, 单位: KB/秒

IO

bi: 发送到块设备的块数, 单位: 块/秒 bo: 从块设备接收到的块数, 单位: 块/秒

System

in: 每秒的中断数, 包括时钟中断 cs: 每秒的环境(上下文)切换次数
CPU 按 CPU 的总使用百分比来显示

us: CPU 使用时间 sy: CPU 系统使用时间 id: 闲置时间

(8) 分析共享内存、信号量和消息队列 ipcs(相关命令 ipcrm: 用于给有权限的用户清除这些量, 注意不要乱清除, 除非该量确实失效了)

例如: [angel@home /angel]# ipcs

----- Shared Memory Segments -----

key	shmid	owner	perms	bytes	nattch	status
0x00280267	0	root	644	1048576	1	
0x61715f01	1	root	666	32000	33	
0x00000000	2	nobody	600	92164	11	dest

----- Semaphore Arrays -----

key	semid	owner	perms	nsems	status
0x00280269	0	root	666	14	
0x61715f02	257	root	777	1	

----- Message Queues -----

key	msqid	owner	perms	used-bytes	messages
-----	-------	-------	-------	------------	----------

(9) 监视用户空间程序发出的全部系统调用 strace

strace 还能显示调用的参数, 以及用符号方式表示的返回值。

strace 从内核中接收信息, 所以一个程序无论是否按调试方式编译(gcc -g)或是否被去掉了调试信息, 都可以被跟踪。

执行格式: strace [-tTeo] executable-program-name

-t : 用来显示调用发生的时间

-T : 显示调用花费的时间

-e : 限定被跟踪的调用类型

-o : 将输出重定向到一个文件中

类似命令: `ltrace [-fiS] executable-program-name`

5、其它常用命令

(1) 命令在线帮助 `man`

执行格式: `man command`

例: `man ls` 查询 `ls` 这个指令的用法

(2) 设定命令记录表长度 `history`

执行格式一: `set history=n`

例: `set history=40`

功能: 设定命令记录表长度为 40 (可记载执行过的前面 40 个命令)

执行格式二: `history` 查看命令记录表的内容

(3) 显示说明 `info`

执行格式: `info command-name`

例: `info gcc`

功能: 查看 `gcc` 的说明,按上下箭头选定菜单,回车进入,"u"键返回上级菜单.

`info` 不加参数则进入最上一级菜单.

(二) LINUX 下 C 语言编写、编译与调试

一、实验目的

- 1、练习用 `vi` 编辑器编辑文本文件;
- 2、复习 C 语言程序基本知识
- 3、练习并掌握 UNIX 提供的 `vi` 编辑器来编译 C 程序

二、实验要求

1、用 `vi` 编写一个简单的、显示"Hello,World!"的 C 程序,用 `gcc` 编译并观察编译后的结果

2、运行生成的可执行文件。

3、`vi`编辑器的使用:

a. 在shell提示符下,输入`vi firscrip`并按<Enter>键。`vi`的界面将出现在显示屏上;

b. 输入`a`, 输入`ls -la`, 并按<Enter>键;

c. 输入`who`, 并按<Enter>键;

d. 输入`pwd`, 再按<Enter>键。这时屏幕将如下图所示:

e. 输入`:wq`, 并按<Enter>键;

f. 在shell提示符下, 输入`bash firscrip`并按<Enter>键;

g. 观察结果。当前的工作目录中有多少个文件? 他们的名称和大小? 还有

谁在使用你的计算机系统？当前的工作目录是什么？

三、 实验内容

(一) 文件编辑器 vi

vi 是在 UNIX 上被广泛使用的中英文编辑软件。vi 是 visual editor 的缩写，是 UNIX 提供给用户的一个窗口化编辑环境。

进入 vi，直接执行 vi 编辑程序即可。

例：\$vi test.c

显示器出现 vi 的编辑窗口，同时 vi 会将文件复制一份至缓冲区 (buffer)。vi 先对缓冲区的文件进行编辑，保留在磁盘中的文件则不变。编辑完成后，使用者可决定是否要取代原来旧有的文件。

1、vi 的工作模式

vi 提供二种工作模式：输入模式 (insert mode) 和命令模式 (command mode)。使用者进入 vi 后，即处在命令模式下，此刻键入的任何字符皆被视为命令，可进行删除、修改、存盘等操作。要输入信息，应转换到输入模式。

(1) 命令模式

在输入模式下，按 ESC 可切换到命令模式。命令模式下，可选用下列指令离开 vi：

: q!	离开 vi，并放弃刚在缓冲区内编辑的内容
: wq	将缓冲区内资料写入磁盘中，并离开 vi
: ZZ	同 wq
: x	同 wq
: w	将缓冲区内资料写入磁盘中，但并不离开 vi
: q	离开 vi，若文件被修改过，则要被要求确认是否放弃修改的内容，此指令可与 : w 配合使用

(2) 命令模式下光标的移动

H	左移一个字符
J	下移一个字符
K	上移一个字符
L	右移一个字符
O	移至该行的首
\$	移至该行的末
^	移至该行的第一个字符处
H	移至窗口的第一列
M	移至窗口中间那一列
L	移至窗口的最后一列
G	移至该文件的最后一列
W, W	下一个单词 (W 忽略标点)
B, B	上一个单词 (B 忽略标点)
+	移至下一列的第一个字符处
-	移至上一列的第一个字符处

(移至该句首
)	移至该句末
{	移至该段首
}	移至该段末
NG	移至该文件的第 n 列
N+	移至光标所在位置之后第 n 列
n-	移至光标所在位置之前第 n 列

(3) 输入模式

输入以下命令即可进入 vi 输入模式:

a(append)	在光标之后加入资料
A	在该行之末加入资料
i(insert)	在光标之前加入资料
I	在该行之首加入资料
o(open)	新增一行于该行之下, 供输入资料用
O	新增一行于该行之上, 供输入资料用
Dd	删除当前光标所在行
X	删除当前光标字符
X	删除当前光标之前字符
U	撤消
.	重做
F	查找
s	替换,例如:将文件中的所有"FOX"换成 "duck",用":%s/FOX/duck/g"
ESC	离开输入模式

(二) GCC 编译器

LINUX 上可用的 C 编译器是 GNU C 编译器,它建立在自由软件基金会编程许可证的基础上,因此可以自由发布。

LINUX 上的 GNU C 编译器 (GCC) 是一个全功能的 ANCI C 兼容编译器,而一般 UNIX (如 SCO UNIX) 用的编译器是 CC。

1、使用 GCC/CC

通常后跟一些选项和文件名来使用 GCC 编译器。GCC 命令的基本用法如下:

gcc [options] [filenames]

命令行选项指定的编译过程中的具体操作

2、GCC 常用选项

GCC 有超过 100 个的编译选项可用,这些选项中的许多可能永远都不会用到,但一些主要的选项将会频繁使用。很多的 GCC 选项包括一个以上的字符,因此必须为每个选项指定各自的连字符,并且就像大多数 LINUX 命令一样不能在一个单独的连字符后跟一组选项。

当不用任何选项编译一个程序时, GCC 将建立 (假定编译成功) 一个名为

a.out 的可执行文件。例如，

```
gcc test.c
```

编译成功后，当前目录下就产生了一个 a.out 文件。

也可用 -o 选项来为即将产生的可执行文件指定一个文件名来代替 a.out。例如：

```
gcc -o count count.c
```

此时得到的可执行文件就不再是 a.out，而是 count。

3、CC 编译

在 cc 命令后面直接跟上文件名，则编译后的输出结果将存放在标准的 a.out 文件中。如果 cc 命令使用 -o 任选项，则可以将编译结果存放在自己命名的文件中。

```
$cc -o compact compact.c
```

```
$
```

编译结果就存放在 compact 文件中。如果出现编译错误，则可以利用 vi 命令来对程序进行修改。

4、执行 C 程序

格式： ./可执行文件名

例： ./a.out

./count

四、实验结果（包括程序或图表、结论陈述、数据记录及分析等）

五、实验总结（包括心得体会、问题回答及实验改进意见）

六、实验报告要求

实验结束后，完成《实验报告 1》。

实验报告要求：

0.文件名：实验 1 -2016xxxx-名字

1.字体：宋体；字号：五号；首行缩进两格；行间距为 1.5 倍。

2.表格位置和内容居中，字体字号同要求 1，表格必须有表题，居于表的顶部居中；表格必须有说明。

3.图片居中显示，单张图片长宽不超过均不超过 5 厘米，图必须有图名，居于图下居中位置；图必须有不少于 10 字的说明。

4.请将实验内容、步骤截图保存，记录好实验中的各种数据，体现在报告里。

5.请删除报告书内括号及括号内的内容。请勿修改报告书其余格式和文字。

6.完成实验报告后，请先交给班级学委，由学委打包好发给任课老师。