

计算机与 IT 入门实验讲义

# Linux 基本操作

实 验 手 册

大连理工大学软件学院

实训基地

2016 年 7 月

# 目 录

实验说明.....	1
上机练习.....	1
预备知识 1 Linux 基础命令 .....	2
1.1 系统登录.....	2
1.1.1 登录 VMware Workstation .....	2
1.1.2 通过 telnet 登录 AIX.....	2
1.2 Linux 用户分类.....	3
1.3 shell .....	3
1.4 显示时间与日期命令 .....	4
1.4.1 显示日期的指令: date .....	4
1.4.2 显示日历的指令: cal .....	4
1.5 文件与目录的操作命令 .....	5
1.5.1 pwd 显示工作目录, 当前路径 .....	5
1.5.2 touch 新建文件 .....	5
1.5.3 ls (文件显示)—list directory contents .....	6
1.5.4 cd 改变当前目录 .....	6
1.5.5 cp (文件拷贝)—copy file .....	7
1.5.6 rm(删除文件或目录)—remove .....	8
1.6 权限改变命令.....	8
预备知识 2 vi 编辑器的使用 .....	10
2.1 vi 简介 .....	10
2.2 vi 三种模式.....	10
2.3 vi 的进入与退出.....	11
2.4 输入模式命令.....	12
2.4.1 在当前行移动光标的命令 .....	12
2.4.2 在行之间移动光标的命令 .....	12
2.5 命令模式下的文本删除.....	13
2.5.1 命令模式下的文本删除 .....	13
2.5.2 命令模式下的拷贝与粘贴命令 .....	13
预备知识 3 Linux 编译器 gcc.....	13
3.1 gcc 简介 .....	13
3.2 gcc 的使用 .....	14
3.3 gcc 文件扩展名规范 .....	14
3.4 使用 gcc 编译代码 .....	15

## 实验说明

本实验文档为软件学院 15 级《计算机与 IT 入门》课程编写，主要涉及 Linux 的常用操作，包括：Linux 的基础命令、vi 编译器的使用及 gcc 的使用等。

本次实验可以通过**登录 VMware Workstation 或远程连接 AIX 进行上机操作**，同学可**自行选择操作环境**。

文档中**红色部分为重点强调部分，请同学们务必认真查看**。红色部分包括常用命令、操作注意事项等。

## 上机练习

本次上机满分为 10 分，4 道练习全部完成最多可得满分 10 分。具体操作可参见**预备知识部分**。**严禁抄袭**（对文件进行操作时要进入到文件所在目录下）

- （1） 输入命令显示当前的日期、显示某年或某年某月的日历（2 分）
- （2） 新建一个文件（如 file.c）并能显示该文件的详细信息，实现对该文件的复制、删除操作（3 分）
- （3） 用 vi 编译器打开（2）的文件（如 file.c），编写一个简单地 C 程序（内容自拟），并保存退出（3 分）
- （4） 使用 gcc 对（3）的 C 程序文件进行编译并执行（2 分）

## 预备知识 1 Linux 基础命令

Linux 的命令很多，分为内部命令和外部命令。

内部命令是 shell 内置的命令，不需要在磁盘中有相应的可执行文件。

外部命令实际上就是一个程序或可执行文件，只是系统安装时它们已经被安装好。

Linux 的命令是大小写敏感的，一般为小写。

本节主要了解并掌握基础命令的语法及使用。

### 1.1 系统登录

#### 1.1.1 登录 VMware Workstation

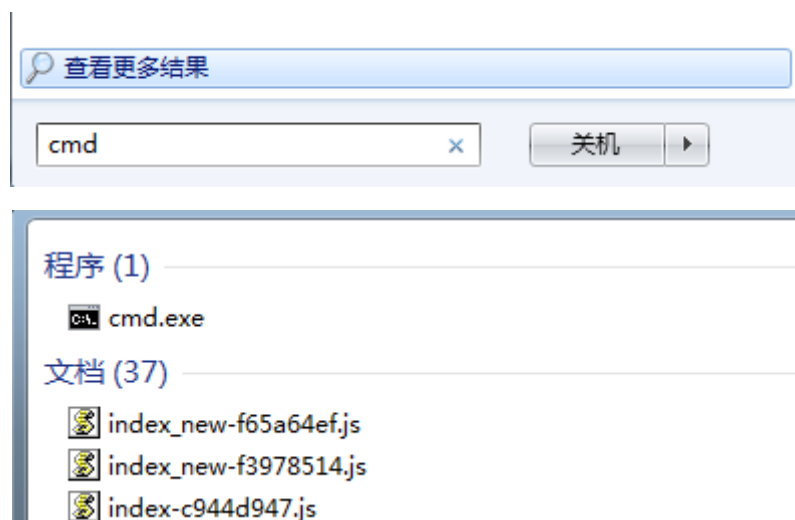
- 进入系统
- 打开 VMware Workstation
- 选择 Ubuntu -> 单击
- 用户名: ssuser
- 密码: ssuser



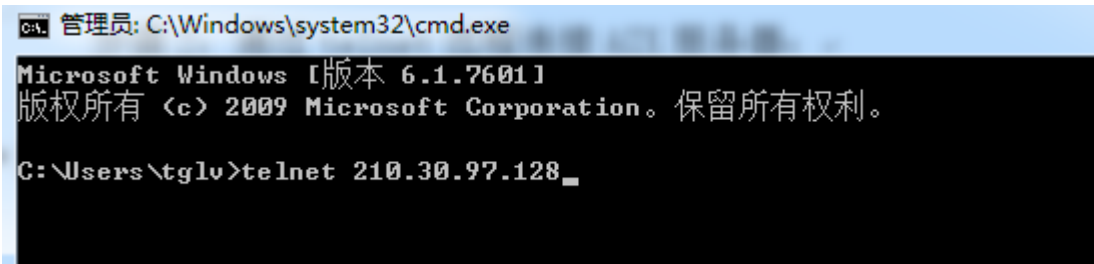
▶ Start this virtual machine

#### 1.1.2 通过 telnet 登录 AIX

步骤 1: 点击开始，在搜索栏输入 cmd，双击最上面的 cmd。



步骤 2: 通过 telnet 远程连接 AIX 服务器。IP :210.30.97.128

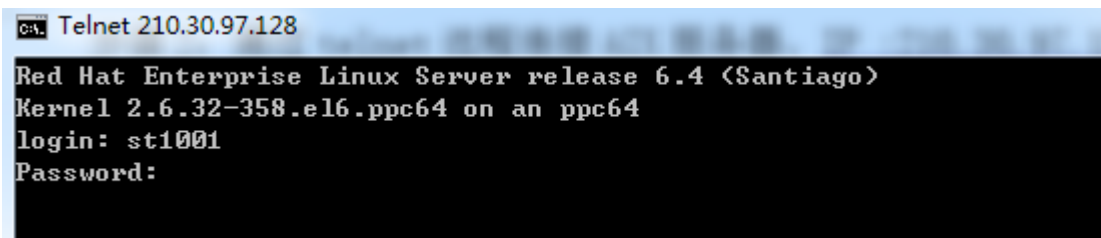


```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\tglu>telnet 210.30.97.128_
```

(注: 如果显示 telnet 不是内部或外部命令时参见 Telnet 的配置和用法:  
<http://jingyan.baidu.com/article/ae97a646b22fb6bbfd461d19.html>)

步骤 3: 输入账号和密码完成登录。(默认密码为: 123, 输入密码时不显示)



```
C:\ Telnet 210.30.97.128
Red Hat Enterprise Linux Server release 6.4 (Santiago)
Kernel 2.6.32-358.el6.ppc64 on an ppc64
login: st1001
Password:
```

## 1.2 Linux 用户分类

Linux 是一个多用户的操作系统, 注册用户要使用 Linux 系统资源, 首先必须登录系统。使用完系统后, 必须退出系统。

Linux 将用户分为普通用户和超级用户。一般的 Linux 使用者均为普通用户, 而系统管理员一般使用超级用户帐号完成一些系统管理的工作, 即 root 用户。

用户身份不同, 系统提示符也不相同。

#: 超级用户

\$: 普通用户

## 1.3 shell

Linux 提供给用户使用的命令行界面称为 Shell, 它本身是用 C 语言编写的程序, 给用户管理和使用系统提供了一个接口和桥梁。功能类似于 DOS 系统中的 commend.com, 但功能比后者强大的多。

### shell 命令特点

- **1.命令补齐功能**：当键入的字符足以确定目录中一个唯一的文件时，只须按 **Tab** 键就可以自动补齐该文件名的剩下部分。
- **2.后台执行功能**：在命令后加一个“&”符号。
- **3.输入输出重定向及通道**：command1|commend2，command1 的输出作为 command2 的输入。
- **4.一次输入多个命令**：中间用“;”隔开。
- **5.脚本功能**：支持编程，是 linux 命令的组合。
- **6.键盘的↑↓键**可以找到之前输入的命令。

## 1.4 显示时间与日期命令

- 显示日期与时间的指令： date
- 显示日历的指令： cal

### 1.4.1 显示日期的指令： date

如果想要知道目前 Linux 系统的时间，那么就直接在指令列模式输入 date 即可显示：

```
[st1001@cloud3 ~]$ date
Tue Jun 28 15:58:14 CST 2016
```

### 1.4.2 显示日历的指令： cal

如果想要列出目前这个月的日历，那么就直接在指令列模式输入 cal 即可显示：

```
[st1001@cloud3 ~]$ cal
      June 2016
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

[st1001@cloud3 ~]$
```

除了本月的日历之外，显示某一年的日历操作如下：

```
[st1001@cloud3 ~]$ cal 2008
                2008

   January                February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
   1  2  3  4  5              1  2              2  3  4  5  6  7  8
  6  7  8  9 10 11 12    3  4  5  6  7  8  9    9 10 11 12 13 14 15
 13 14 15 16 17 18 19    10 11 12 13 14 15 16    16 17 18 19 20 21 22
 20 21 22 23 24 25 26    17 18 19 20 21 22 23    23 24 25 26 27 28 29
 27 28 29 30 31          24 25 26 27 28 29    30 31

   April                  May                  June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
   1  2  3  4  5              1  2  3              1  2  3  4  5  6  7
  6  7  8  9 10 11 12    4  5  6  7  8  9 10    8  9 10 11 12 13 14
 13 14 15 16 17 18 19    11 12 13 14 15 16 17    15 16 17 18 19 20 21
 20 21 22 23 24 25 26    18 19 20 21 22 23 24    22 23 24 25 26 27 28
 27 28 29 30            25 26 27 28 29 30 31    29 30

   July                  August                September
```

显示某年某月的日历操作如下：

```
[st1001@cloud3 ~]$ cal 8 2008
      August 2008
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
[st1001@cloud3 ~]$
```

## 1.5 文件与目录的操作命令

### 1.5.1 pwd 显示工作目录，当前路径

```
[st1001@cloud3 ~]$ pwd
/home/st1001
```

### 1.5.2 touch 新建文件

```
[st1001@cloud3 ~]$ touch hello.c
[st1001@cloud3 ~]$ ls
hello.c
[st1001@cloud3 ~]$
```

### 1.5.3 ls (文件显示)— list directory contents

语法: `ls [选项] [目录或文件名]` **命令和选项之间有空格!!**

功能: 列出文件或子目录的信息 (默认字母排序)

选项: `ls` 当前目录下文件及文件夹

`ls -a` 隐藏文件

**`ls -l` 长格式, 详细信息**

`ls -d` 当前目录名

`ls -dl` 当前目录详细信息

`ls -c` 按修改时间排序

其他: `-b -f -F -i`

#### 例: `ls -l`

- `ls -l` 以长格式形式在每行显示一个目录或文件名, 如:

```
drwxr-xr-x  2 jjpr1  group   512 Nov 18 10:24 .
drwxrwxr-x 24 root   auth    512 Nov 16 10:33 ..
```

每一行的组成部分:

权限 链接数 文件主 组 长度 建立/修改时间 目录或文件名

`-rw- r-- r-- 1 jjpr jjpr 1299 Nov 200 9:33 hs`

所有者 同组用户 其他组用户 八进制表示为644

- 用颜色代表不同文件:

**蓝色: 目录**

**绿色: 可执行文件**

**红色: 压缩文件**

**浅蓝色: 链接文件**

**灰色: 其他文件**

- 在`ls -l`显示结果中, 第一个字符表示文件的类型

`-`: 普通文件

`d`: 目录

`c`: 字符设备

`b`: 块设备

`l`: 链接文件

### 1.5.4 cd 改变当前目录

语法: `cd dirName` **`cd` 和目录之间使用空格隔开**



作用：切换当前目录至 dirName

例子：\$cd usr/bin

```
[st1001@cloud3 usr]$ cd /usr/bin
[st1001@cloud3 bin]$
```

技巧：

\$cd 进入用户主目录；

\$cd / 进入根目录；

\$cd .. 返回上级目录（".."为上级目录的意思）；

```
[st1001@cloud3 usr]$ cd /usr/bin
[st1001@cloud3 bin]$
[st1001@cloud3 bin]$ cd ..
[st1001@cloud3 usr]$
```

\$cd ../../ 返回上两级目录；

\$cd - 返回进入此目录之前所在的目录；

### 1.5.5 cp (文件拷贝)—copy file

语法： cp [选项] 源文件 目标文件

cp [选项] 源目录 目标目录

功能：将一个文件复制为另一文件，或将数个文件复制到一个目录

选项： -p: 保持原始文件属性

-f: 覆盖已经存在目标文件

-i: 提示是否覆盖现有的目标文件

其他：-a,-d,-f,-r

例 1: \$cp f1.c f2.c

将文件 f1.c 复制(已存在)，并命名为 f2.c

```
[st1001@cloud3 ~]$ ls
f1.c
[st1001@cloud3 ~]$ cp f1.c f2.c
[st1001@cloud3 ~]$ ls
f1.c f2.c
```

例 2: \$cp \*.c ss

将所有 C 语言程序拷贝至 ss 目录中

例 3: \$cp -i f1.c ../f2.c

将当前目录下的文件 f1.c 复制到上层 目录中，并取名为 f2.c,若上层目录中已经存在 f2.c 文件，则在覆盖文件前询问用户。

### 1.5.6 rm(删除文件或目录)—remove

语法: rm [选项] 文件...      权限不足命令前添加: **sudo**    需输入密码  
(密码不显)

参数:

- r 将全部目录和子目录均删除。
- i 进行交互式删除。
- f 忽略不存在的文件，从不给出提示

例 1: #rm hello.c

删除当前目录下的文件 hello.c

```
[st1001@cloud3 ~]$ ls
hello.c
[st1001@cloud3 ~]$ rm hello.c
[st1001@cloud3 ~]$ ls
[st1001@cloud3 ~]$
```

例 2: #rm -i \*

交互式删除当前目录下的所有非隐藏文件

例 3: #rm -rf qq

强制删除当前目录下的 qq 子目录:

## 1.6 权限改变命令

文件的保护方式

(1)两种用户：超级用户、普通用户

(2)三个等级：文件拥有者(user)、同组用户(group)、其他用户(others)，所有用户(all)

(3)三种权限：读(r)、写(w)、执行(x)

ls -l 长格式显示目录内容：

权限 链接数 文件主 组 长度 建立/修改时间 目录/文件名

-rw-r--r-- 1 jjpr jjpr 1299 Nov 200 9:33 hs

所有者 同组用户 其他组用户

## 使用 Chmod(改变文件或目录的访问权限)

语法：chmod [who] [+|-|=] [mode] 文件或目录

选 项

Who	操作符	许可权
U 文件主	+ 增加权限	R 读
G 同组人	- 取消权限	W 写
O 其他人	= 从新设置	X 执行
A 所有人	(删除旧的权限)	

例 1：\$ chmod u=r f1

为 f1 文件的所有者只赋予读权限

例 2：\$ chmod u+w f1

为 f1 文件的所有者增加写权限

例 3：\$ chmod go-rwx f1

取消 f1 文件对其他用户的读、写、执行权限

## 预备知识 2 vi 编辑器的使用

本章将学习 vi 编译器的使用，介绍 Linux 系统中 vi 编译器的三种模式及相应的操作。

### 2.1 vi 简介

- visual interface，可视化编辑器
- vi 可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制.
- 只是一个文本编辑器，不能排版
- vi 没有菜单，只有命令
- vi 有三种基本工作模式，分别是：命令模式（command mode）、插入模式（insert mode）和底行模式（last line mode）

### 2.2 vi 三种模式

- 命令模式（进入时默认的模式）

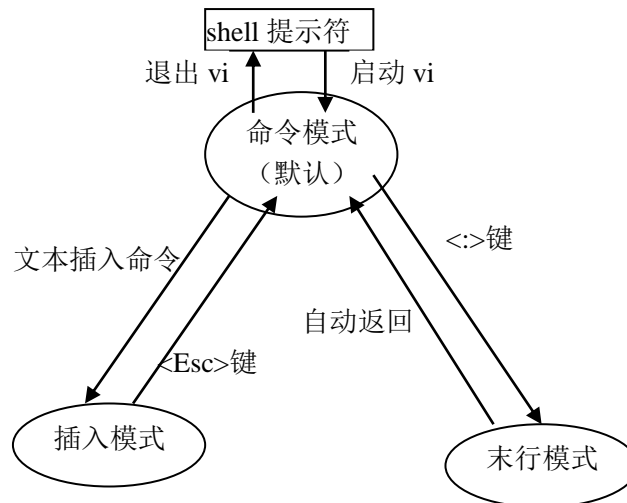
任何输入都会作为编辑命令，而不会出现在屏幕上，若输入错误，系统发出声音提示。

- 输入模式（编辑模式）：

任何输入的数据都置于编辑寄存器。在命令模式下输入（i, a, A 等），可以进入输入模式，输入模式下按 ESC，可返回命令模式。

- 末行模式（最后行模式）

以“:”或者“/”为前导的指令，出现在屏幕的最下一行，任何输入都被当成特别指令。（命令模式下输入 : 可进入末行模式，在该模式下输入命令可退出 vi 编译器）



## 2.3 vi 的进入与退出

- 进入 vi:
  - 命令“vi 文件名”
- 选项“+n”，表示希望在进入 vi 之后，光标处于文件中第 n 行上，
- 选项“+”，表示希望在进入 vi 之后光标处于文件最末行。
- 进入 vi 后屏幕左方会出现波浪符号，凡是列首有该符号就代表此列目前是空的。
- 在末行模式下，有四种方法可以退出 vi 返回到 shell:

输入 **:q** 系统退出 vi 返回到 shell。在用此命令时，若编辑的文件没有被保存，则 vi 在窗口的最末行给出提示信息。

输入 **:q!** vi 将放弃所作修改而直接退到 shell 下。

输入 **:wq** 先保存文件，然后再退出 vi 返回到 shell。

输入 **:x** 该命令的功能与命令模式下的 ZZ 命令功能相同

例 1:     : w  vitem.c

       : q

       : vi  vitem.c

: wq

例 2: vi file.txt (进入文件 file.txt, 保存修改并退出)

: wq

例 3: vi file.txt (进入文件 file.txt, 不保存退出)

: q!

## 2.4 输入模式命令

- i : 在光标前 输入文本
- I : 在当前行首 输入文本
- a : 光标后 输入文本
- A : 在当前行尾 输入文本

### 2.4.1 在当前行移动光标的命令

命令	含义
l 或者→	右移一个字符 (不会移到下一行)
h 或者←	左移一个字符 (不会移到上一行)
w	向右移动到下一个字的字首
<n>w	右移 n 个字
b	移到当前字的字首, 已经处于字首的, 则移到前一个字的字首
<n>b	左移 n 个字
e	移到当前字的字尾, 如果已经处于字尾, 则移到下一个字的字尾
0 (数字 0)	移到当前行的行首
\$	移到当前行的行尾

### 2.4.2 在行之间移动光标的命令

命令	含义
j 或者↓	移到下一行, 所在的列不变

+	移到下一行行首
k 或者↑	移到上一行，所在的列不变
-	移到上一行行首

## 2.5 命令模式下的文本删除

### 2.5.1 命令模式下的文本删除

- 字符删除命令:

**x** : 删除光标所在的字符（同输入模式下的退格键）

- 字和行删除命令

**dw** : 删除光标所在的单词

**dd**: 删除当前行

**d\$ (D)**: 删除光标至行尾所有字符

### 2.5.2 命令模式下的拷贝与粘贴命令

- **yw** 将光标所在单词拷入剪贴板
- **Y** 将光标至行尾的拷入剪贴板
- **yy** 将当前行拷入剪贴板
- **p** 将剪贴板中的内容粘贴在光标后
- **P** 将剪贴板中的内容粘贴在光标前

## 预备知识 3 Linux 编译器 gcc

本章将学习 Linux 编译器 gcc 的使用，以简单地 C 程序为例介绍 Linux 系统中 gcc 的操作。

### 3.1 gcc 简介

- 名称:

- GNU Compiler Collection

- 能够编译 c、c++、Object C
- 编译过程：
  - 用 gcc 编译程序时，分为四个阶段：
    - 预处理（Preprocessing）
    - 编译（Compilation）
    - 汇编（Assembly）
    - 链接（Linking）

程序员可以根据需要在任何一阶段停止

## 3.2 gcc 的使用

- 基本使用格式
  - `$ gcc [ 选项 ] <文件名>`
- 常用选项及含义

gcc 常用选项	
选项	含义
<code>-o file</code>	将经过 gcc 处理过的结果存为文件 <b>file</b> 。 这个结果文件可能是预处理文件、汇编文件、目标文件或者最终的可执行文件。 如果这个选项被省略了，那么生成的可执行文件默认名称为 <b>a.out</b> 。
<code>-c</code>	仅对源文件进行编译，不链接生成可执行文件。在对源文件进行查错时，或只需产生目标文件时可以使用该选项。

## 3.3 gcc 文件扩展名规范

gcc 文件扩展名规范



扩展名	类型	可进行的操作方式
<b>.c</b>	<b>c 语言源程序</b>	<b>预处理、编译、汇编、链接</b>
<b>.C, .cc, .cp, .cpp, .c++, .cxx</b>	<b>c++语言源程序</b>	预处理、编译、汇编、链接
<b>.i</b>	预处理后的 c 语言源程序	编译、汇编、链接
<b>.ii</b>	预处理后的 c++语言源程序	编译、汇编、链接
<b>.s</b>	预处理后的汇编程序	汇编、链接
<b>.S</b>	未预处理的汇编程序	预处理、汇编、链接
<b>.h</b>	头文件	不进行任何操作
<b>.o</b>	目标文件	链接

### 3.4 使用 gcc 编译代码

- 源代码

示例源程序——hello.c

```
#include <stdio.h>
int main()
{
    printf("hello gcc!\n");
    return 0;
}
```

- 生成可执行文件

- 命令:           Linux

• \$gcc hello.c -o hello

• 运行程序

• \$ ./hello

hello gcc!

AIX

\$ xlc -o hello hello.c

运行程序

\$ ./hello

hello gcc!