实验一 Linux 操作

实验环境：

Linux 平台

实验时间：

4 小时

实验目的：

要求在熟悉 Linux 系统的基础上，了解用 C 语言编写文本处理程序的具体过程。

实验目标：

熟悉 Linux 操作系统，在 Linux 中使用 C 语言进行文本处理。完成下面问题：
使用 C 语言编写一个反向打印程序，使之能够按与输入文件中文本行相反的次序来打印（即后出现的文本行先打印）。

实验准备：

（1）阅读 Linux 的使用手册（可从网络下载电子书），包括图形界面使用、系统管理、Shell 命令和目录结构。
（2）熟悉 gcc 的使用方法。

实验步骤：

Linux 操作系统是一个通用的多用户分时操作系统，现已成为高档微机、工作站及若干小型机系统上的主要操作系统，并在许多领域中获得了广泛的应用。Linux 系统具有简单、高效、易懂等特点，同时具有良好的可移植性，它代表着多用户操作系统发展的方向。C 语言最初是为开发 Linux 操作系统而设计，Linux 核心和所有的用户程序几乎都是用 C 语言编写，是 Linux 的标准语言。

[提示]
1、登录进入 Linux 系统，可以通过用户图形界面中的菜单创建用户组并为用户分配注册号，也可以通过 shell 命令创建用户组，创建用户组的 shell 命令格式如下：

groupadd -g group_ID group_name

注意：如果命令没有权限不能执行，在命令前加“sudo ”并输入命令。

为用户分配注册号的 shell 命令格式如下：

useradd －u user_number －g primary_group_ID
－G supplementary_group_ID －c comments
-d home-directory －s program －m login_ID

其中-u 选项用来表示用户 ID 号， -g 选项用来表示主用户组名， -G 选项用来 表示可由“,”隔开的多个 supplementary_group_ID，-c 选项用来表示注释信息，d 选项用来表示主目录名，-s 选项用来表示注册 shell 程序，-m 选项不带参数， 它用来将/etc/skel 目录中的内容（如.profile 文件和标准目录文件）拷入新的注册 号下，login_ID 是该命令必需的用户注册号。

例：groupadd －g 51 newuser

useradd －u 203 －g newuser －c “new user” －d /user/liu －s /bin/sh －m liu

如果采用缺省设置，也可以只使用如下 shell 命令：

useradd liu

分配完注册号后，也可以为用户设置口令。用来设置口令的 shell 命令格式如下：

passwd options login_ID

其中任选项 options 可为如下内容： -n days 表示口令的有限时间。 -x days 表示口令从设置到修改的时间。

例：passwd －n 7 －x 6 liu 或者：

passwd liu

系统在提示输入和重新输入后，确认口令设置成功。

修改用户属性的 shell 命令格式如下：

usermod options login_ID

其中，任选项 options 可以为如下内容：

-c comment 表示注释信息

-d pathname 表示当前目录

-g group_ID 表示主用户组名

-G supplementary_group_ID 表示主用户组的增补组名

-l login 表示用户注册号

-m 表示将/etc/skel 目录中的内容拷入新的注册号下

-s program 表示注册用 shell 程序

u user_ID_number 表示用户 ID 号

-e days 表示口令从设置到修改的天数

-f days 表示注册起作用的天数

作为一个例子，我们修改注册 shell 程序，这时可使用如下 shell 命令：

usermod －s /bin/csh liu

修改用户组属性的 shell 命令格式如下：

groupmod options

其中，任选项 options 可为如下内容：

-g group _ID 表示为用户组分配一个新的组 ID 号

-o group _ID 表示为用户组分配一个重复的组 ID 号

-n group_name 表示为用户组分配一个新名字

作为一个例子，我们将 newuser 组名修改为 test，这时可使用如下 shell 命 令：

groupmod －n test newuser

只删除用户注册号的 shell 命令格式如下：

userdel login_ID

实例如下：

userdel liu

删除用户注册号及其所在目录和文件的 shell 命令格式如下：

userdel －r login_ID

实例如下：

userdel －r liu

删除用户组的 shell 命令格式如下：

groupdel group_ID

实例如下：

groupdel newuser

2、从新的子目录中开始工作 使用 mkdir 命令创建新的子目录 c_progs。

```
$ mkdir c_progs
$
```

使用 cd 命令进入到新创建的子目录中。

```
$cd c_progs
$
```

3、输入 C 程序

输入 C 程序可以用任意的文本编辑工具来完成。下面输入 C 程序 cowpact.c。

```
/*   Remove newlines */
#include <stdio.h>
main() {
    int c,n=0,max=1;
    while((c=getchar())!=EOF)
    {
        if (c==’\n’)
          n++;
        else
          n=0;
        if (n<=max)
            putchar( c);
    }
 }
```

保存到当前用户的用户目录下，一般为 home/username

4、编译 C 程序

由于 C 语言是一种高级语言，所以输入完 C 程序后就要对它进行编译。cc 命令可以用来编译 C 程序。如果在 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。为方便起见，我们使用带-o 任选项的 cc 命令 来进行编译。（也可以使用 gcc，用法相同。）

```
$cc  – o compact compact.c
$
```

这样，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。

5、执行 C 程序

由于存放编译结果的文件本身就是可执行文件，所以可以在 shell 提示符下 敲入该文件的名字来执行它，在有些情况下还要提供输入内容。下面给出 compact 的执行过程。

首先创建测试文件 testfile。

```
$cat > testfile
this is line 1
this is line 2

this is line 4
<ctrl+d>
$
```

对 testfile 文件使用 compact。

```
$./compact < testfile
   this is line 1
   this is line 2
   this is line 4
$
```

这样就从输入文件中删除了多余的空行。

实验报告

（1）尝试实验步骤 1 的每个部分。
（2）在/usr 目录下创建自己的目录（可以用姓名，或者学号）。
（3）利用文本编辑器步骤 3 中的 c 程序，并编译和执行它。
（4）完成实验目标，绘制该程序的流程图。