# 实验02 Linux下的C程序开发

## 一、实验目的

1．掌握vi编辑器的启动和退出方法；

2．掌握vi编辑器中录入文件内容的方法；

3．掌握vi编辑器的各种编辑命令的使用；

4．熟悉字符模式下的C程序编译和运行；

5．熟悉在Linux的字符模式下，编辑、编译及调试一个C程序的基本方法。

## 二、实验环境

安装了 Linux系统的计算机一台

## 三、准备知识

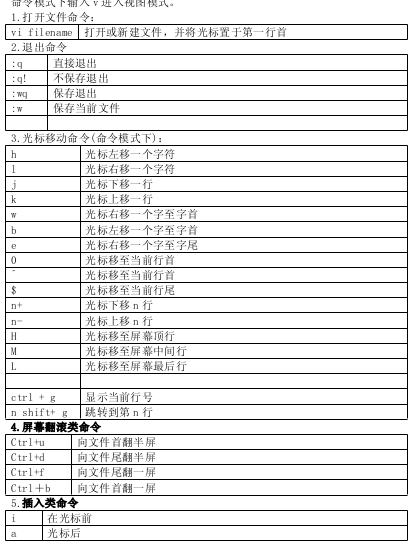
**vi编辑器的使用**

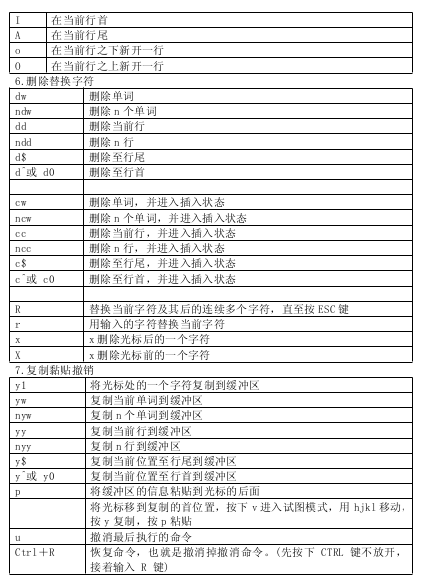
vi提供二种工作模式：输入模式（insert mode）和命令模式（command mode）。使用者进入vi后，即处在命令模式下，此刻键入的任何字符皆被视为命令，可进行删除、修改、存盘等操作。要输入信息，应转换到输入模式。

**进入vi的命令**

vi filename :打开或新建文件，并将光标置于第一行首   
vi +n filename ：打开文件，并将光标置于第n行首   
vi + filename ：打开文件，并将光标置于最后一行首   
vi +/pattern filename：打开文件，并将光标置于第一个与pattern匹配的串处   
vi -r filename ：在上次正用vi编辑时发生系统崩溃，恢复filename   
vi filename....filename ：打开多个文件，依次进行编辑

在输入模式下，按ESC可切换到命令模式。命令模式下，按insert键或i可切换到输入模式。





## 四、GNU C编译器

在Unix操作系统中，专门提供了CC命令来帮助系统工程师编译C语言；Linux操作系统中，采用的是gcc命令；如果是c++ 直接将gcc改为g++即可。

1. 编译的命令是：

gcc -c test.c // -c 选项告诉 GCC 仅把源代码编译为目标代码。缺省时 GCC 建立的目标代码文件有一个 .o 的扩展名。

gcc -o test test.c // -o 编译选项来为将产生的可执行文件用指定的文件名。

gcc test.c //a.out可执行文件

格式：

gcc filename.c [-o outputfilename.out]

其中gcc是c的编译器。参数：filename.c 是要编译的源文件的名称，outputfilename表示输出文件名称，中括号表示括号内部的内容可输入也可以不输入（中括号本身不再命令行中出现）。如果不输入outputfilename.out，默认的输出文件是a.out 。

例如，

gcc test.c

g++ test.cpp

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

g++ –o count count.cpp

此时得到的可执行文件就不再是a.out，而是count。

1. 最后一步是运行程序，方法如下：（**注意加上前面的 ./** ）

./ count

或 ./ a.out

**(3) c/c++语言多文件编译**

如果有多个源文件，基本上有两种编译方法：

[假设有两个源文件为test.c和testfun.c]

1. 多个文件一起编译

用法：#gcc testfun.c test.c -o test

作用：将testfun.c和test.c分别编译后链接成test可执行文件。

2. 分别编译各个源文件，之后对编译后输出的目标文件链接。

用法：

#gcc -c testfun.c //将testfun.c编译成testfun.o

#gcc -c test.c //将test.c编译成test.o

#gcc -o testfun.o test.o -o test //将testfun.o和test.o链接成test

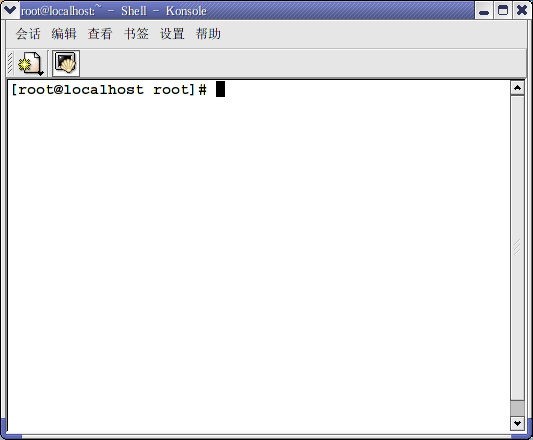
以上两种方法相比较，第一种方法编译时需要所有文件重新编译，而第二种方法可以只重新编译修改的文件，未修改的文件不用重新编译。

**如果是c++ 直接将gcc改为g++即可。**

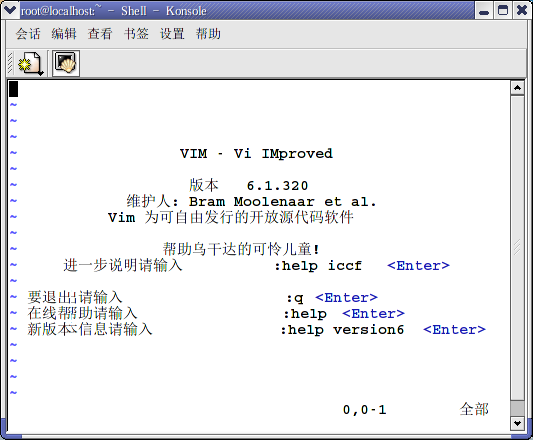
## 五、实验内容

* 1. vi编辑器的启动和退出

首先启动shell终端，如下图



在shell提示符下输入vim或vi,将启动vi编辑器



若要退出Vi，在输入模式下按ESC键可切换到命令模式然后输入:wq或:q！即可退出。

* 1. 使用键盘在命令模式和输入模式之间进行切换

进入输入模式： a追加，i插入，o当前行后插入一空行.

注意其他三个对应命令：A I O

* 1. 本次实验以一个简单的C程序为例，请用GCC进行编译并运行。例子由两个文件组成：main.c, calc.c，其中，main.c文件内容为：

#include <stdio.h>

int calc(int);

int main(int argc, char\* argv[])

{

int iInput=0, iOutput=0;

int i=0;

scanf("%i",&iInput);

while(iInput < 0)

{

printf("Please input a positive integer!\n");

scanf("%i",&iInput);

}

iOutput = calc(iInput);

printf("Result is:%i\n", iOutput);

return 0;

}

calc.c的文件内容为：

#include <stdio.h>

int calc(int iIn)

{

int i = 0, iOut = 0;

iOut = iIn;

for(i = 0; i < iIn; i++)

{

iOut += iOut \* iIn;

}

return iOut;

}

* 1. 复制正文：在命令模式下，yy 是复制当前行， nyy复制n行。然后在命令模式下，将光标移动到将要粘贴的行处，按“p”进行粘贴。
  2. 删除正文：x删除一个字符， dd 删除当前行，ndd 删除n行， dw 删除一个单词。删除内容均放在缓冲区。在命令模式下，按“U”键为撤消命令。
  3. 替换正文：p 将缓冲区的内容粘贴到当前光标处。r n用字符n替换当前字符。
  4. 查找定位：nH将光标定位到第n行开头，其中n为1、2、3……。而且H必须是开启大写状态下使用。
  5. 文件操作:w 写盘，:wq写盘退出，:q!不存盘退出，:w filenam写文件名。

## 六、实验报告（请在课外完成以下内容并提交实验报告）

**(一)、用VI编辑器输入以下程序，用gcc编译、运行并截图。**

**程序1：createFork.c**

#include <stdio.h>

#include<unistd.h>

#include<sys/types.h>

int main( )

{

int p1,p2;

while((p1=fork( )) == -1); /\*创建子进程p1\*/

if (p1= =0) putchar('b');

else

{

while((p2=fork( )) = = -1); /\*创建子进程p2\*/

if(p2 ==0) putchar('c');

else putchar('a');

}

return 0;

}

**程序 2：anotherFork.c**

#include <stdio.h>

#include<unistd.h>

#include<sys/types.h>

int main( )

{

int p1,p2,i;

while((p1=fork( )) == -1); /\*创建子进程p1\*/

if (p1 ==0)

for(i=0;i<10;i++)

printf("daughter %d\n",i);

else

{

while((p2=fork( )) == -1); /\*创建子进程p2\*/

if(p2= =0)

for(i=0;i<10;i++)

printf("son %d\n",i);

else

for(i=0;i<10;i++)

printf("parent %d\n",i);

}

return 0;

}

**(二)、用VI编辑器输入以下程序，用g++编译并运行。**

**程序 1：helloCPP.cpp**

#include<iostream>

using namespace std;

int main()

{

cout<<"hello!"<<endl;

cout<<"Welcome to c++!"<<endl;

return 0;

}

**程序 2：isLeapYear.cpp**

#include <iostream>

using namespace std;

int main()

{

int year;

bool IsLeapYear;

cout << "Enter the year: ";

cin >> year;

IsLeapYear = ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0));

if (IsLeapYear)

cout << year << " is a leap year" << endl;

else

cout << year << " is not a leap year" << endl;

return 0;

}

**(三)、完成以下命令：**

**使用上述的四个源代码文件完成：createFork.c anotherFork.c helloCPP.cpp isLeapYear.cpp**

1. 把这四个文件的后缀名去掉；
2. 在vi编辑**createFork**状态下查看当前编辑文件的文件名、行数以及字符数等信息，使用搜索字符串(putchar)、删除(前三行)和粘贴(前五行到最后的位置)功能，使用替换(while替换成forloop)命令及存盘退出、存盘不退出和不存盘退出命令；
3. 从文件读取输入，并输出到显示器；
4. 把标准输出定向到文件**newcreateFork**中，记录文件内容；
5. 把（1）中的四个文件合并为一个文件myMergeFile，并用命令wc显示其行数、单词数和字符数；
6. 使用管道将myMergeFile中内容逐屏显示出来；
7. 构造一个”>>”使用符号追加新内容的例子；
8. 查找这五个文件中有没有包含Welcome，IsLeapYear、cd的行，有几个文件包含IsLeapYear。

**(四)、思考：如何把一个hello.c文件复制到虚拟机Ubuntu系统当中？**