一、vim编辑器

有高亮显示，还错误行提示，可以脱离鼠标

进入vim编辑器

vi/vim 文件名

命令行模式

不能输入，可以进行复制、粘贴、剪切、格式化、行间跳转

nyy 从光标所在行开始复制n行

p 粘贴到光标所在的下一行

ndd 从光标所在行开始剪切n行

格式化 gg = G

鼠标选中要格式化的部分，再按=

gg 跳转到首行

G 跳转到尾行

ngg 跳转到第n行

0 跳到光标所在行的行首

/ word 查找单词

n 向下查找

N 向上查找

插入行模式

在命令行模式下按i/a/o/I/A/O进入插入行模式

i：从光标所在的前一位进行插入

a：从光标后一位进行插入

o：从光标所在下一行插入

I：从光标所在行首插入

A：从光标所在行尾插入

O：从光标所在上一行插入

回到命令行模式 按一下esc

底行模式

命令行模式下进入底行模式：

shift + ；

按空格

插入行模式下进入底行模式，先按esc进入命令行模式，再从命令行模式进入底行模式。

保存，退出，显示行号，取消行号，取消高亮

:w 保存

:q 退出

:q! 强制退出

:set nu 显示行号

:set nonu 取消行号

:noh 取消高亮

底行模式回到命令行模式 按两下esc，

二、终端指令

命令行提示符

ubuntu@ubuntu:~/22081/day3$

ubuntu：用户名

@：分割符

ubuntu：计算机名/主机名

~/22081/day3 :表示用户当前所在路径

$:用户权限 代表普通用户权限

#:超级用户 root用户权限

su切换用户

su 用户名

sudo 用管理员权限 执行后面的指令

切换不到root用户，因为root用户没有密码，先给root用户设置一个密码

sudo passwd root

exit

退出当前用户，返回上一个用户。

如果当前用户没有上一个用户了，就会当前的终端。

cd切换路径

..上一级路径

.当前路径

ls

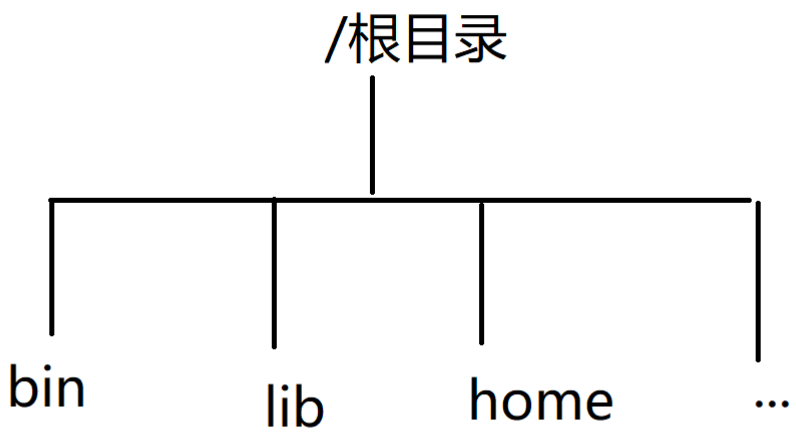
ls 显示当前路径下的文件

ls -a显示当前路径下的文件包括隐藏文件

mkdir

mkdir 目录名　创一个目录

mkdir -p 目录名/目录名/目录名····　创建有层级关系的目录。



Linux文件系统结构，是一个倒插树结构。

rmdir

rmdir 目录名

rmdir -p 目录名/目录名/目录名····　删除有层级关系的目录。

只能删除空目录

rm

删除文件/目录

rm 文件名

rm -r 目录名

touch

touch创建一个文件

mv

mv file1 file2　当我们file2不存在的时候，表示把file1重命名为file2

当file2存在的时候，表示把file1重命名为file2，并之前的file2

mv file1 c2(c2是个目录)　把file1 移动到c2里面

mv file1 c2/file2　把file1移动到c2里面，　重命名为flie2，如果c2中原来就有file2，把原来的file2覆盖。

mv C1 c2　把目录C1移动到c2里面

cp

cp file1 file2

把file1，复制了一份，放到file2中，如果没有file2，就创建了一个file2，如果file2存在，就覆盖之前的file2.

cp file1 路径名

把file1，复制了一份，放到目标路径中，如果目标路径中又和file1重名的文件，就会被覆盖掉。

cp -r dir1 dir2

拷贝一份dir1，放到dir2中，如果当前路径没有dir2目录，就创建一个dir2目录，如果当前路径有dir2目录，把之前的dir2覆盖掉

cat

cat 文件名　查看文件的内容

echo

echo 打印内容　　后面跟什么内容，就在终端打印什么出来。

三、第一个C语言程序

```
#include <stdio.h> #预处理标识符  include包含头文件的标准格式  <>从C语言的标准库中找头文件
""从当前路径找头文件 stdio.h要包含的头文件
int main(int argc, const char *argv[])
int返回值类型  main函数名，主函数，每一个c程序有且仅有一个main函数  ()参数列表  int argc,
const char *argv[]main函数的外部传参
{
    printf("hello world!\n");  //标准输出函数 " "中原封不动的输出，除了占位符和转义字符
    return 0;   return返回函数的返回值，函数结束
}
;是语句结束的标志，没一个独立的语句后面都要加;
```

四、gcc编译器

4.1一步编译

gcc 文件名.c　会默认生成一个a.out文件

gcc 文件名.c -o 文件名　编译文件，并重命名生成的可执行文件

```
ubuntu@ubuntu:~/22081/day1$ ls
a.out  hello.c
ubuntu@ubuntu:~/22081/day1$ ./a.out
1
ubuntu@ubuntu:~/22081/day1$ gcc hello.c
ubuntu@ubuntu:~/22081/day1$ gcc 1.c
ubuntu@ubuntu:~/22081/day1$ gcc hello.c -o hello
ubuntu@ubuntu:~/22081/day1$ ls
1.c  a.out  hello  hello.c
ubuntu@ubuntu:~/22081/day1$ ./hello
1
ubuntu@ubuntu:~/22081/day1$ ./a.out
ubuntu@ubuntu:~/22081/day1$
```

**4.2分步编译ESc-->iso

预处理---编译---汇编----链接

4.2.1预处理

展开头文件、替换宏定义、删除注释、生成编译文件，不会检查语法错误。

gcc -E hello.c -o hello.i

4.2.2编译

检查语法错误,生成汇编文件

gcc -S hello.i -o hello.s

4.2.3汇编

生成二进制文件

gcc -c hello.s -o hello.o

4.2.4链接

链接库，生成可执行文件

gcc hello.o -o hello

五、宏定义

5.1无参宏定义

```
#define N 3  //宏名一般大写
#define 宏名 宏体
宏体是对宏名的简单替换。
```

5.2带参宏定义

```
#define ADD(参数1,参数2,参数3,···)  对参数进行的操作
```

```c
#include <stdio.h>//引入头文件
#define N 3
#define ADD(a,b) a+b
int main(int argc, const char *argv[])//主函数
{
    printf("1\n");//标准输出函数
    printf("%d\n",N);//宏定义
    printf("%d\n",ADD(4,5));//带参宏定义
    return 0;
    printf("3");
}
```

六、计算机中的数据存储

1bit

1Byte = 8bit

1KB = 1024Byte

1MB= 1024KB

1GB= 1024MB

1TB = 1024GB

七、常量

7.1整型常量

0，1，100…　%d

printf("%d",0);

7.2浮点型常量

3.14 …　%f

7.3字符型常量

'a','b','c'…　%c

7.4字符串常量

“hello”，……　%s

7.5宏定义常量

```c
#include <stdio.h>//引入头文件
#define N 3.0
#define ADD(a,b) a+b
int main(int argc, const char *argv[])//主函数
{
    printf("1\n");//标准输出函数
    printf("%f\n",N);//宏定义
    printf("%d\n",ADD(4,5));//带参宏定义
    return 0;
    printf("3");
}
```