

知识点1【linux概述】（了解）

知识点2【linux命令概述】

- 1、--help 帮助信息（只能查看命令 中文）
- 2、man 帮助信息（查看命令、库函数、系统调用等）
- 3、tab自动补全
- 4、history查看历史命令
- 5、重定向 >
- 6、管道 |
- 7、tree树状显示
- 8、clear清屏
- 9、cd切换目录
- 10、pwd显示路径
- 11、cat查看文本内容
- 12、rm删除文件或目录
- 13、cp拷贝
- 14、mv 移动文件或目录（剪切文件或目录）
- 15、mkdir创建文件夹
- 16、touch创建文件 不能编辑文件
- 17、grep：在指定文件中查找指定内容
 - 1、grep hehe在终端中查找字符串hehe
 - 2、在文件中查找内容
- 18、ln链接文件（重要）

1、软链接

2、硬链接

19、tar压缩和解压

1、gzip格式

2、bz2格式

3、暴力解压

知识点3【编辑器之神vim】(了解)

1、安装vim环境

知识点1【linux概述】（了解）

linux以树状 管理目录结构。

根目录：/


```
edu@edu: ~$ cd /
edu@edu:/$ cd ..
edu@edu:/$ cd ..
edu@edu:/$ ls
bin      dev      initrd.img  lost+found  opt      run      srv      usr
boot     etc      lib         media       proc     sbin     sys      var
cdrom    home     lib64       mnt         root     snap     tmp      vmlinuz
edu@edu:/$ _
```

```
edu@edu:/$ cd home
edu@edu:/home$ ls
edu
edu@edu:/home$
```

当创建linux用户的时候 系统会自动以用户名 在/home/创建一个文件夹。

cd ~进入家目录

```
edu@edu: ~  
edu@edu: ~$ ls  
examples.desktop  vmware-tools-distrib  
VMwareTools-10.3.10-13959562.tar.gz  公共的  
edu@edu: ~$ mkdir tools work
```



```
1 cd work  
2 mkdir c/day01 -p  
3 cd c/day01/
```

知识点2 【linux命令概述】

命令 选项 参数

命令：具体的linux命令

选项：扩展命令的功能

参数：命令作用的对象

ls 查看当前目录下 没有隐藏的文件

ls -a显示所有文件（包含隐藏文件）

ls -l以列表的方式显示文件信息

ls -h和-l一起使用 以更人性化的方式显示文件大小

```

edu@edu: ~/work/c/day01$ ls
edu@edu: ~/work/c/day01$ touch a.txt
edu@edu: ~/work/c/day01$ ls
a.txt
edu@edu: ~/work/c/day01$ ls -a
.  ..  a.txt
edu@edu: ~/work/c/day01$ ls -l
总用量 0
-rw-rw-r-- 1 edu edu 0 7月 20 14:23 a.txt
edu@edu: ~/work/c/day01$ ls -a -l
总用量 8
drwxrwxr-x 2 edu edu 4096 7月 20 14:23 .
drwxrwxr-x 3 edu edu 4096 7月 20 14:18 ..
-rw-rw-r-- 1 edu edu 0 7月 20 14:23 a.txt
edu@edu: ~/work/c/day01$ ls -a -l -h
总用量 8.0K
drwxrwxr-x 2 edu edu 4.0K 7月 20 14:23 .
drwxrwxr-x 3 edu edu 4.0K 7月 20 14:18 ..
-rw-rw-r-- 1 edu edu +0 7月 20 14:23 a.txt
edu@edu: ~/work/c/day01$

```

ls的参数 常常修饰的是路径

```

edu@edu: ~/work/c/day01$ ls
a.txt
edu@edu: ~/work/c/day01$ ls /
bin  dev  initrd.img  lost+found  opt  run  srv  usr
boot  etc  lib  media  proc  sbin  sys  var
cdrom  home  lib64  mnt  root  snap  tmp  vmlinuz
edu@edu: ~/work/c/day01$

```

1、--help 帮助信息（只能查看命令 中文）

命令 --help 查看该命令的帮助信息

2、man 帮助信息（查看命令、库函数、系统调用等）

man 是 Linux 提供的一个手册，这个手册有很多个章节

第一章中放的是 命令 的帮助信息

第二章中放的是系统调用（操作系统提供的函数 open）

第三章中放的是库函数（c 库提供的函数 printf）的帮助信息

man 1 ls 从第一章节中查找ls命令

3、tab自动补全

4、history查看历史命令

5、重定向 >

ls > a.txt将ls显示的结果 输出到文件中

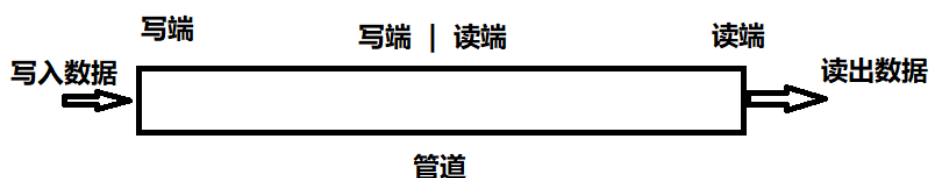
- 1 echo "hello world"
- 2 往终端上输出 字符串"hello world"

```
edu@edu: ~/work/c/day01$ echo "hello world" > a.txt
edu@edu: ~/work/c/day01$ cat a.txt
hello world
edu@edu: ~/work/c/day01$
```

echo "hello world" > a.txt将 "hello world" 字符串 输出到文件中（覆盖）

echo "world" >> a.txt将 "world" 追加到文件末尾（追加）

6、管道 |



ps -A | grep ssh 从ps -A的结果中 使用grep命令查找字符串ssh

ps -A将系统的所有进程 都显示到终端上

grep xxx重终端上查找字符串"xxx"

```
edu@edu: ~/work/c/day01$ ps -A | grep ssh
17152 ?          00:00:00 sshd
17772 ?          00:00:00 sshd
17833 ?          00:00:00 sshd
19848 ?          00:00:00 sshd
19914 ?          00:00:00 sshd
edu@edu: ~/work/c/day01$
```

7、tree树状显示

提前安装tree命令

- 1 sudo apt-get install tree
- 2 mkdir a
- 3 touch ./a/b.txt

```
edu@edu: ~/work/c/day01$ tree
.
├── a
│   └── b.txt
└── a.txt

1 directory, 2 files
edu@edu: ~/work/c/day01$ _
```

用法: tree -L 层数

```
edu@edu: ~/work/c/day01$ tree / -L 1
/
├── bin
├── boot
├── cdrom
├── dev
├── etc
├── home
├── initrd.img -> boot/initrd.img-4.4.0-21-generic
├── lib
├── lib64
├── lost+found
├── media
├── mnt
├── opt
├── proc
└── root
```

只显示一层

8、clear清屏

9、cd切换目录

- 1 cd 目录 进入目录
- 2 cd .. 返回上一级目录
- 3 cd ~ 回到家目录
- 4 cd - 返回上一次目录

10、pwd显示路径

11、cat查看文本内容

将文件的内容显示到终端上

12、rm删除文件或目录

rm a.txt 删除文件名为a.txt的文件

```

edu@edu: ~/work/c/day01$ touch a.txt b.txt c.txt
edu@edu: ~/work/c/day01$ ls
a  a.txt  b.txt  c.txt
edu@edu: ~/work/c/day01$ rm *.txt
edu@edu: ~/work/c/day01$ ls
a
edu@edu: ~/work/c/day01$

```

删除所有.txt

rm 目录 -r 删除目录一定要加上-r

```
1 sudo rm /* -rf 别尝试
```

13、cp拷贝

cp a.txt test

```

edu@edu: ~/work/c/day01$ ls
a.txt
edu@edu: ~/work/c/day01$ touch a.txt
edu@edu: ~/work/c/day01$ ls
a.txt
edu@edu: ~/work/c/day01$ mkdir test
edu@edu: ~/work/c/day01$ ls
a.txt  test
edu@edu: ~/work/c/day01$ tree
.
├── a.txt
└── test

1 directory, 1 file
edu@edu: ~/work/c/day01$

```

```

edu@edu: ~/work/c/day01$ tree
.
├── a.txt
└── test

1 directory, 1 file
edu@edu: ~/work/c/day01$ cp a.txt test
edu@edu: ~/work/c/day01$ tree
.
├── a.txt
└── test
    └── a.txt

1 directory, 2 files
edu@edu: ~/work/c/day01$

```

如果cp拷贝的是一个文件夹 必须加-r

```

edu@edu: ~/work/c/day01$ ls
a.txt  b  test
edu@edu: ~/work/c/day01$ cp test b -r
edu@edu: ~/work/c/day01$ tree
.
├── a.txt
├── b
│   └── test
│       └── a.txt
└── test
    └── a.txt

3 directories, 3 files
edu@edu: ~/work/c/day01$ _

```

cp 文件 目的目录

cp 目录 目的目录 -r

cp 文件1 文件2 将文件1备份成文件2


```

edu@edu:~/work/c/day01$ ls
a.txt  b  test
edu@edu:~/work/c/day01$ echo "hello" >> a.txt
edu@edu:~/work/c/day01$ cat a.txt
hello
edu@edu:~/work/c/day01$ cp a.txt b.txt
edu@edu:~/work/c/day01$ ls
a.txt  b  b.txt  test
edu@edu:~/work/c/day01$ cat b.txt
hello
edu@edu:~/work/c/day01$

```

14、mv 移动文件或目录（剪切文件或目录）

mv 源文件 目的目录 将 源文件 移动到目的目录中

```
1 mv a.txt test文件夹
```

mv 源目录 目的目录 将文件夹 移动到 另一个文件夹里面（注意别同名）

mv 源文件 目的文件 将源文件名 重命名为 目的文件

```

edu@edu:~/work/c/day01$ ls
b  b.txt
edu@edu:~/work/c/day01$ mv b.txt c.txt
edu@edu:~/work/c/day01$ ls
b  c.txt
edu@edu:~/work/c/day01$ _

```

mv 源目录 目的目录 如果没有目的目录名 那就是将源目录 重命名 为目的目录

15、mkdir创建文件夹

mkdir 目录 创建目录

mkdir 创建多级目录 如果某级目录不存在 需要加-p

```
edu@edu:~/work/c/day01$ mkdir f/d -p
edu@edu:~/work/c/day01$ tree
.
├── c
│   ├── c
│   │   └── a.txt
│   └── test
│       └── a.txt
├── c.txt
├── f
│   └── d
└── test

6 directories, 3 files
edu@edu:~/work/c/day01$
```

16、touch创建文件 不能编辑文件

17、grep：在指定文件中查找指定内容

1、grep hehe在终端中查找字符串hehe

```
edu@edu: ~/work/c/day01
edu@edu:~/work/c/day01$ grep hehe
ni hao haha xixi hehe lala heihei
ni hao haha xixi hehe lala heihei
```

2、在文件中查找内容

1 grep 查找信息 文件名 参数-n 显示行号

```

edu@edu:~/work/c/day01$ ls
c  c.txt  f  test
edu@edu:~/work/c/day01$ echo "hello world" >> c.txt
edu@edu:~/work/c/day01$ echo "world hello" >> c.txt
edu@edu:~/work/c/day01$ echo "world hehe" >> c.txt
edu@edu:~/work/c/day01$ echo "hello hehe" >> c.txt
edu@edu:~/work/c/day01$ cat c.txt
hello
hello world
world hello
world hehe
hello hehe
edu@edu:~/work/c/day01$ _

```

```

edu@edu:~/work/c/day01$ ls
c  c.txt  f  test
edu@edu:~/work/c/day01$ cat c.txt
hello
hello world
world hello
world hehe
hello hehe
edu@edu:~/work/c/day01$ grep world c.txt -n
2:hello world
3:world hello
4:world hehe
edu@edu:~/work/c/day01$ _

```

在某个路径及其子目录下的所有文件中查找字符串 需要加上-R

```

edu@edu:~/work/c/day01$ grep world ./* -R -n
./c.txt:2:hello world
./c.txt:3:world hello
./c.txt:4:world hehe
edu@edu:~/work/c/day01$ _

```

18、In链接文件（重要）

1、软链接

```
1 ln 源文件名字 链接文件名字 -s
```

源文件和链接文件 内容实时同步 链接文件等价于源文件的快捷方式 如果删除源文件 链接文件将无效

2、硬链接

```
1 ln 源文件名字 链接文件名字
```

源文件和链接文件 内容实时同步 链接文件是一个独立的普通文件 如果删除源文件 链接文件不受影响

19、tar压缩和解压

打包和压缩的区别：

打包：将多文件 封装在一起

压缩：将多文件 封装在一起 通过特定的算法 将冗余的数据 进行删除

tar默认是打包命令，如果想用tar进行压缩 必须加选项

1、gzip格式

压缩：

```
1 tar zcvf 压缩包名 文件 1 文件2 文件3.....
```

```
edu@edu:~/work/c/day01$ ls
c  c_h  f  test
edu@edu:~/work/c/day01$ tar zcvf test.tar.gz *
c/
c/c/
c/c/a.txt
c/test/
c/test/a.txt
c_h
f/
f/d/
test/
edu@edu:~/work/c/day01$ ls
c  c_h  f  test  test.tar.gz
edu@edu:~/work/c/day01$
```

压缩当前路径下的所有文件

解压：

```
1 tar zxvf 压缩包名称
```

```
edu@edu:~/work/c/day01$ tar zxvf test.tar.gz
c/
c/c/
c/c/a.txt
c/test/
c/test/a.txt
c_h
f/
f/d/
test/
edu@edu:~/work/c/day01$ ls
c  c_h  f  test  test.tar.gz
edu@edu:~/work/c/day01$
```

```
1 tar zxvf 压缩包名称 -C 目的目录
```

将压缩包 解压到 目的目录中

```
1 tar zxvf test.tar.gz -C ./a
```

2、bz2格式

压缩:

```
1 tar jcvf 压缩包包名 文件1 文件2.....
```

解压:

```
1 tar jxvf 压缩包名称
```

```
1 tar jxvf 压缩包名称 -C 目的目录
```

3、暴力解压

```
1 tar xvf 压缩包
```

```
edu@edu: ~/work/c/day01$ ls
test.tar.bz2
edu@edu: ~/work/c/day01$ tar xvf test.tar.bz2
a/
a/f/
a/f/d/
a/c_h
a/c/
a/c/c/
a/c/c/a.txt
a/c/test/
a/c/test/a.txt
a/test/
edu@edu: ~/work/c/day01$ ls
a  test.tar.bz2
edu@edu: ~/work/c/day01$ _
```

知识点3 【编辑器之神vim】（了解）

1、安装vim环境

```
1 sudo apt-get install vim
2 sudo apt-get install ctags
```

step1: 将 vim_configure 拷入当前用户的目录下
使用 samba 或共享文件夹完成

step2: 打开终端，执行以下命令：

```
cd vim_configure
./copy_con.sh
```

```

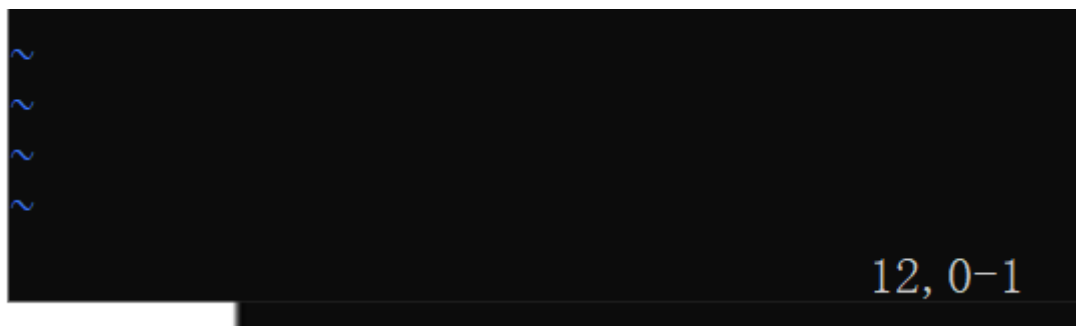
edu@edu:~/tools$ ls
vim_configure
edu@edu:~/tools$ cd vim_configure/
edu@edu:~/tools/vim_configure$ ls
copy_con.sh  _vim  _vimrc
edu@edu:~/tools/vim_configure$ chmod +x copy_con.sh
edu@edu:~/tools/vim_configure$ ls
copy_con.sh  _vim  _vimrc
edu@edu:~/tools/vim_configure$ ./copy_con.sh
start copy
/home/edu
copy successful
edu@edu:~/tools/vim_configure$ _

```

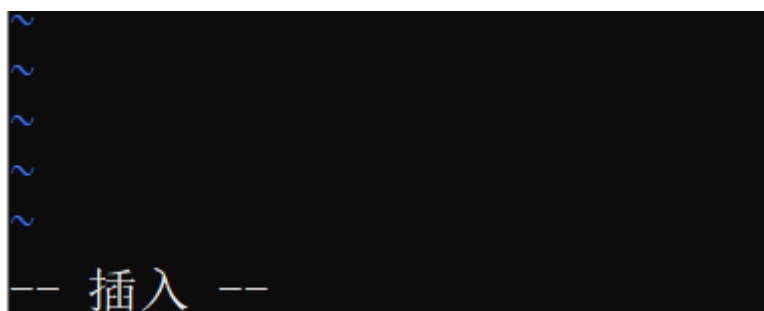
2、vim的3种模式

编辑模式、插入模式、命令行模式（最后一行模式）

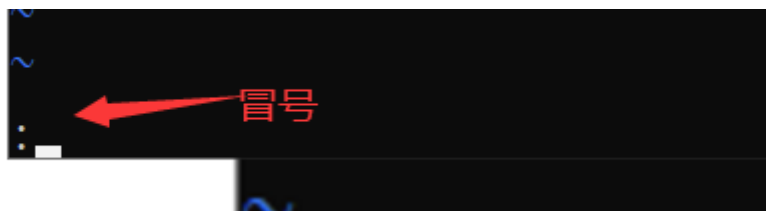
编辑模式：对代码的修改（复制、粘贴、剪切等等）



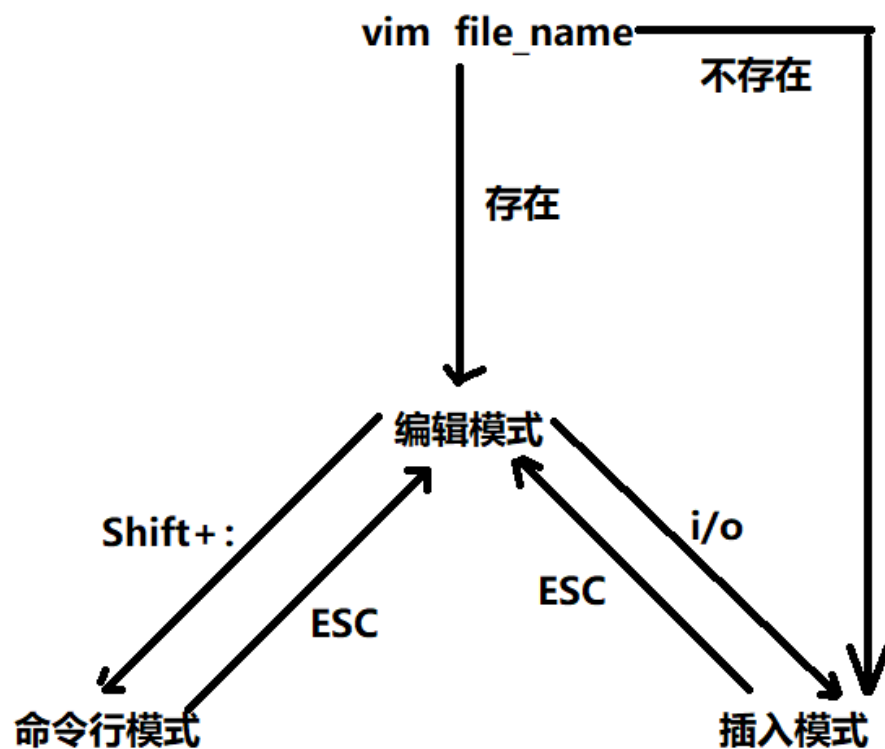
插入模式：用户编辑代码



命令行模式：保存 退出 另存为等



3、vim3种模式的切换



4、最后一行模式命令（保存 退出）

w保存 q退出 !强制执行
:w 保存文件
:wq 保存文件并退出 vi
:x 保存文件并退出 vi
:q! 不保存文件，强制退出 vi
:w filename 另存为 filename

5、编辑模式

u 撤消前面多次修改。ctl +r 反撤销
[n]x 删除光标后 n 个字符。 3x
[n]X 删除光标前 n 个字符。 6X
[n]dd 删除（剪切）从当前行开始的 n 行。 4dd
dw 删除（剪切）一个单词
p粘贴
[n]yy 复制从当前行开始的 n 行。 4yy
yw复制一个单词
. 执行上一次操作
shift +zz(按住 shift 同时按下两次z) 保存并退出

[n]G：将光标定位到第 n 行开始处

5G 将光标移动第 5 行开始处

G: 将光标定位到文件结束处

gg:将光标定位到文件开始处

/字符串：从光标开始处向文件尾查找字符串。

n: 同一方向重复上一次查找命令。

N: 反方向重复上一次查找命令

:nohl 取消高亮

:set hl 设置高亮

知识点4 【gcc】(了解)

```
#include <stdio.h>

int main(int argc, char *argv[])
{
    printf("hello world\n");
    return 0;
}
```

gcc 00_code.c -o main 将源文件00_code.c生成可执行文件main

gcc 00_code.c 默认生成可行性文件a.out