

# 实验零：Linux 初识

Design by W.H Huang | Direct by Prof Feng

## 1 实验目的

✦ 本次实验并非系统介绍Linux系统理论知识，注重**实操**，而且是**针对后续实验需要用到Linux相关知识**，而特别设计的实验。

通过本次实验，你应该：

- 安装Linux系统环境，了解云服务器相关知识
- 掌握Linux基本知识，如：`vim` 的操作使用、Linux系统常用命令、用户与权限相关知识等
- 相关工具 `FTP`、`SSH` 等使用

或许你是第一次接触Linux，相信会给你带来不一样的体验。接下来让我们正式进入实验环节吧。

## 2 实验准备

在正式开始接触Linux前，我们需要搭建好Linux平台。

一般而言，我们有三种方式选择安装Linux系统：

- 购买云服务器，极速搭建（推荐）
- 安装双系统（推荐）
- 安装Linux虚拟机

考虑到实际后续实验需求，我们推荐采用前两种方式来完成Linux系统搭建。本次实验出于时间考虑只详细介绍 **第一种：云服务器** 方式，课后大家可选择其它方式。

### 2.1 云上Linux

云服务器搭建Linux工作流程如下：

1. 腾讯云/阿里云购买学生10元优惠云服务器
2. 搭建可视化图行界面

#### 2.1.1 购买云服务器

购买数量为N，N为组员人数。

腾讯云/阿里云服务器都有学生优惠10元/月，以下是撸羊毛详细过程。

1. 进入学生优惠购买界面，以腾讯云为例：[腾讯云学生优惠](#)

地区可选 `上海三区` & `广州四区`，广州离重庆更近一点所以选择 `广州四区`

选择操作系统为：`CentOS 7.6.64`



登录后依次输入账号，密码：

- 初始账号为root，密码是你购买云服务器所设置的。

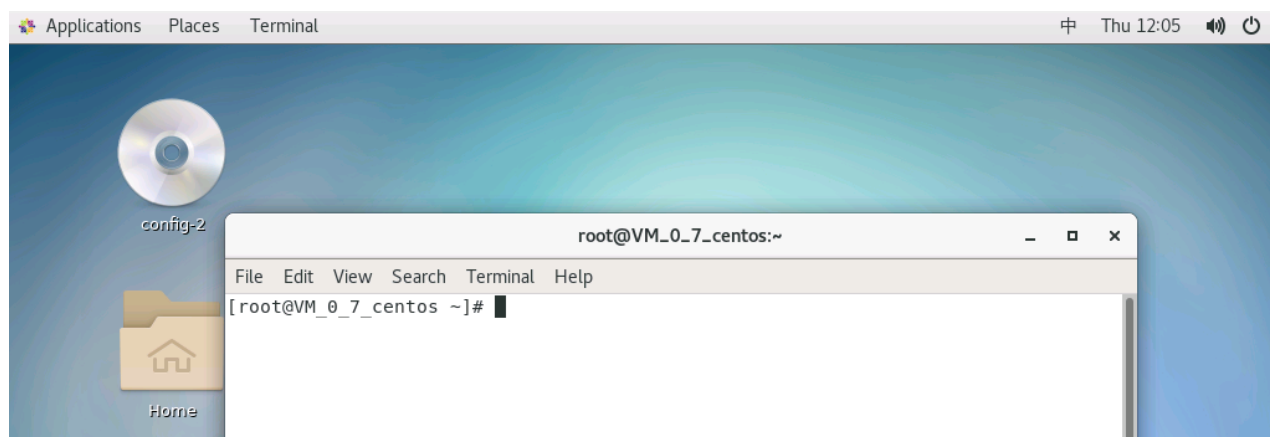
## 2. 安装图形界面

```
yum groupinstall "GNOME Desktop" "Graphical Administration Tools"
```

## 3. 启动图形界面

```
startx # 进入图形界面
```

一切顺利，你应该看到下面图形界面：



## 2.2 云下Linux

我们依旧建议你在云上Linux完成本次及后续实验学习，当然你也可以选择：

- 双系统安装Linux（推荐）
- 虚拟机安装Linux

相关安装你可以在课后完成与实践，出于篇幅及时间考虑这里不再赘述。

## 2.3 SSH工具---Xshell

每次在浏览器连接云服务器终究还是不太方便，我们可以安装SSH工具如Xshell在Windows界面下用来访问远端不同系统下的服务器，从而达到方便操作远程控制终端的目的。

以下是安装使用简单教程：

### 1. 下载Xshell

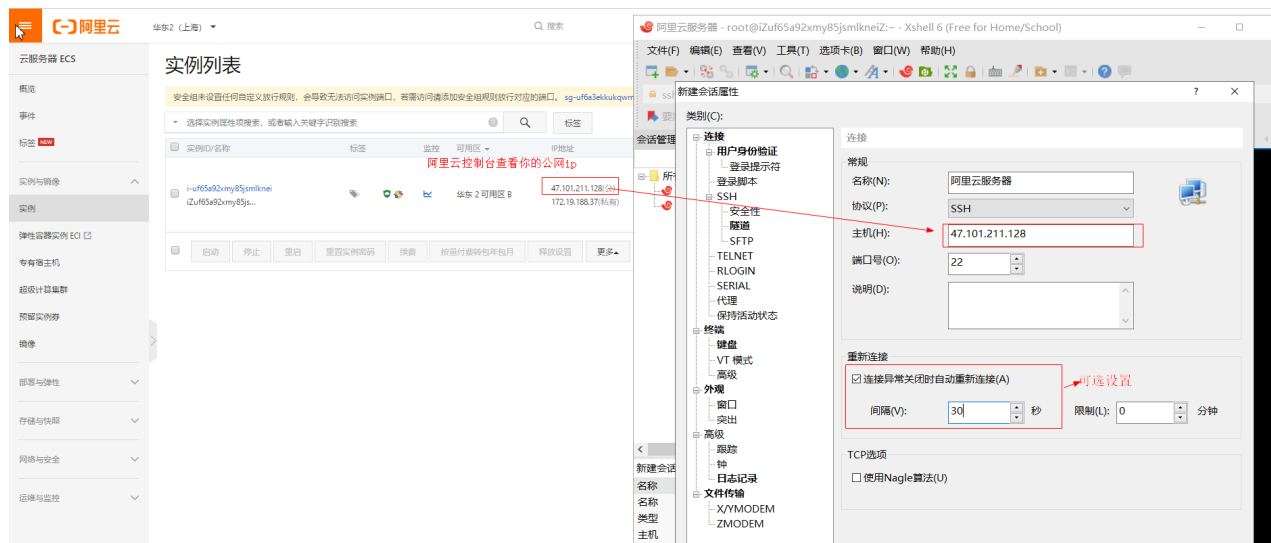
Xshell 下载地址：[Xshell腾讯高速下载](#)

安装一直点下一步傻瓜似安装，最后可以看到Xshell界面。

### 2. 连接服务器

选择文件 -- 新建：

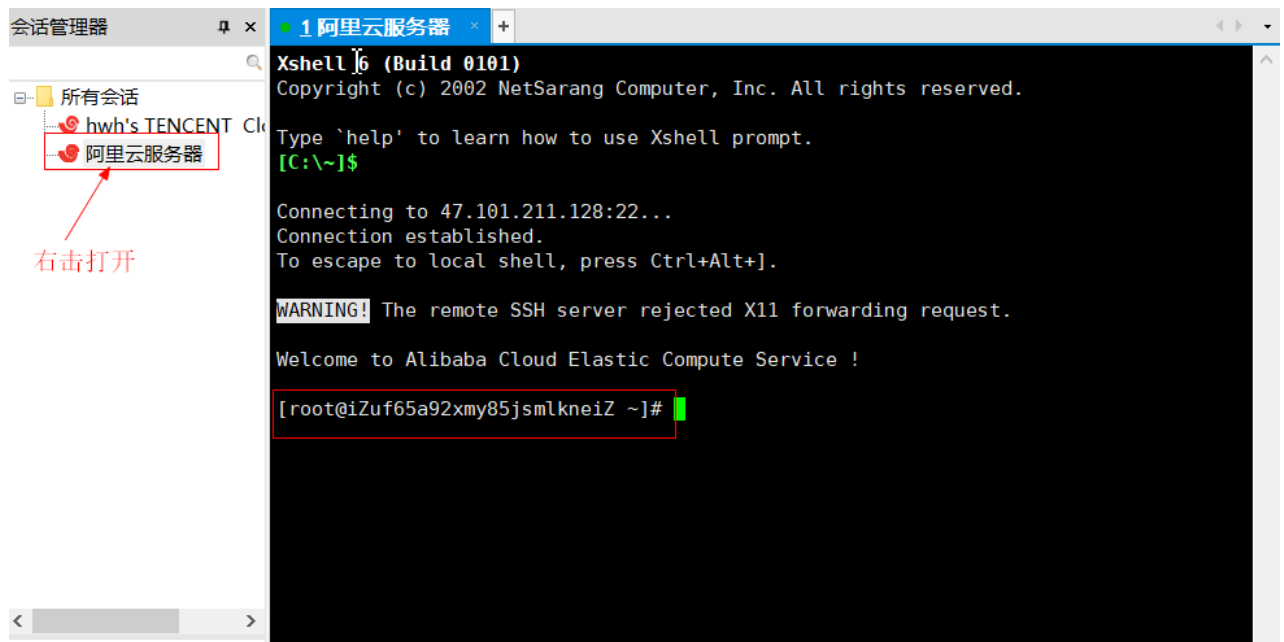
☺ 下图查看公网IP是在[阿里云上实例控制台](#)，如果是腾讯云也同样登陆[腾讯云控制台](#)查看即可。



按照上图填好信息，再点击 用户身份验证，输入登陆账号密码（就是我们在购买云服务器时设置的）：



点击确定。然后在Xshell界面选择：会话管理 - 阿里云服务器 右击 - 打开，便连接到我们的服务器了。



### 3. 测试

接下来你便可以使用 xshell 而非浏览器登陆，愉快的连接使用我们的Linux服务器了。

- 注：如果新建会话连接不了，重新打开Xshell即可解决。

### 4. 修改主机名

为了方便区分，请同学们将主机名修改为自己：**姓名首字母+学号后四位**。

如，张三：`zs4321`

```
vim /etc/hostname
```

按下 `i` 进入插入模式，删除所有内容，然后编辑你的主机名：

```
[root@master /]# vim /etc/hostname  
zs1234
```

按下 `ESC` 键进入命令模式，输入 `:wq!` 保存并退出

最后重启服务器：

```
reboot
```

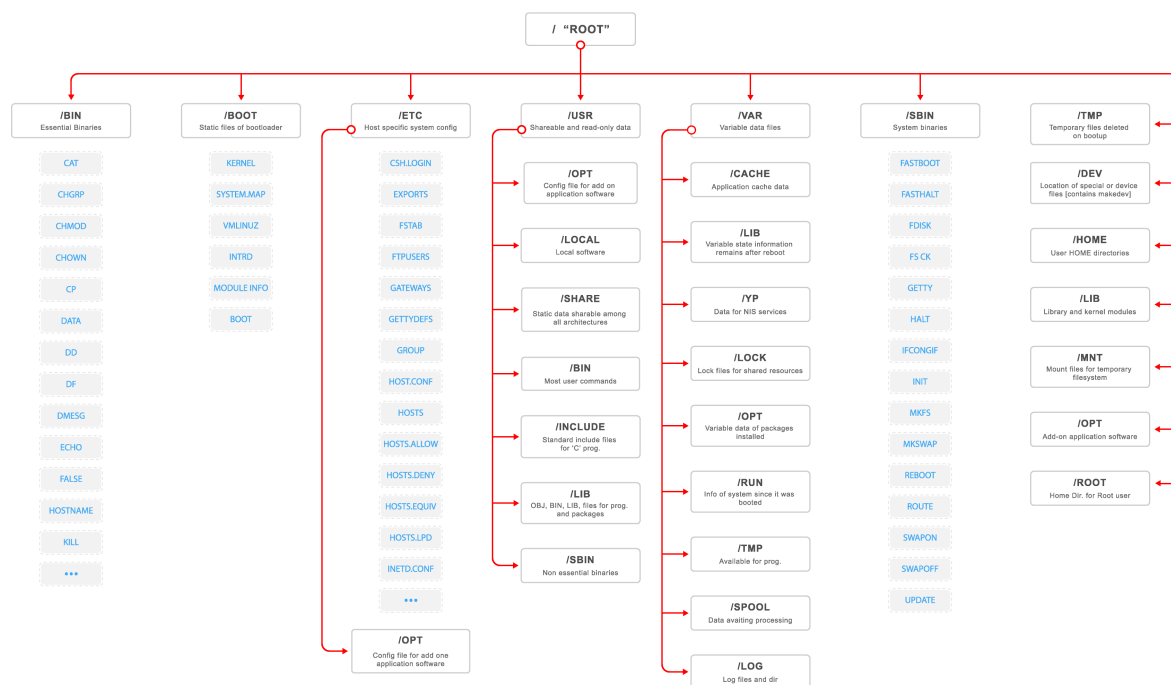
等待约1分钟左右，重新连接可发现主机名已经被修改。

## 3 Linux相关知识

通过本节你将掌握后续实验所必须要掌握的Linux相关知识。

### 3.1 Linux 系统目录结构

Linux目录结构如下图：



我们必须要知道的根目录 / 相关目录作用：

- **/bin**: binary缩写，保存可执行文件，我们敲的命令都在bin中
- **/boot**: 引导目录，操作系统需要引导启动的都在其下
- **/etc**: 所有的配置文件保存其下，一般以 **.conf** 结尾
- **/home**: 所有用户家目录（**root**除外），每个用户都在其下有个对应文件夹保存对应信息。
- **/root**: root用户家目录。
- **/var**: 保存一些经常变换的信息，如**服务器网站**信息，**操作系统日志**信息
- **/tmp**: 临时目录，会被隔几天自动删除
- **/proc**: 系统的实时的信息，不存在硬盘，在内存中。

```
[root@iZuf65a92xmy85jsmlkneiZ ~]# cd /proc
[root@iZuf65a92xmy85jsmlkneiZ proc]# ls
1      20512  258   29723  39    761    devices  keys      pagetypeinfo  timer_stats
10     21072  259   3      40    765    diskstats key-users  partitions     tty
105    215    26    30427  41    767    dma      kmsg      sched_debug    uptime
12     216    26651 30515  443   779    driver   kpagecount schedstat      version
13     236    27    30702  445   8      execdomains kpageflags  scsi           vmallocinfo
```

## 3.2 文件系统相关操作

以下命令，为了更好实践巩固，请务必亲自验证。

### 1. **ls** 显示文件

命令格式：**ls** <参数(可选)> <目录(可选，默认当前)>

常用参数解释：

- **ls**: 显示文件，但不显示隐藏文件

```
[hadoop@master /]$ ls
bin  data  etc  lib  lost+found  mnt  proc  run  srv  tmp  var
boot dev  home lib64 media  opt  root  sbin sys  usr
```

- `ls -a`: 可显示隐藏文件
- `ls -l`: 详细列出文件信息, 不显示隐藏文件 (加上参数 `-a` 可以)
- `ls -R`: 递归显示目录结构
- `ls -ld`: 显示目录和链接信息

## 2. `cd` 切换目录

常和 `pwd` 命令配合显示当前目录:

```
[hadoop@master ~]$ pwd
/
```

命令格式: `cd <参数(可选)> <目录(可选, 默认家目录)>`

常用参数解释:

- `cd .`: 切换为当前目录
- `cd ..`: 切换到上级目录
- `cd ~`、`cd`: 切换到当前用户 **家目录**
  - 家目录: 普通用户在 `/home/用户名` 下, `root` 用户在 `/root` 下
- `cd -`: 切换到上一个工作目录

## 3. `touch/mkdir` 创建文件/文件夹

我们通常使用:

- `touch <文件名>`: 创建文件
- `mkdir <目录名>`: 创建目录

实践: 我们切换到根目录下创建相应文件夹/文件 `/test/readme.md`

根目录创建文件夹:

```
mkdir /test # 加了/指定在根目录下, 不加/默认在当前目录
```

创建文件:

```
touch /test/readme.md
```

我们切换到根目录下进行查看:

```
cd /
ls -R test
```

```
[hadoop@master ~]$ cd /
[hadoop@master ~]$ ls -R test
test:
readme.md
```

## 4. `cp` 复制文件

命令格式: `cp <参数(可选)> <源文件/文件夹> <目标文件/文件夹(没有会创建)>`

常用参数解释:

- `cp -r`: 递归复制整个目录树 (**复制文件夹时必须加**)

- `cp -v` : 显示详细信息, 复制的详细过程

实践: 复制 `/test/readme.md` --> `/test/readme1.md`

```
cp /test/readme.md /test/readme1.md
```

切换到 `/test` 下查看如下:

```
cd /test
ls -l
```

```
[hadoop@master ~]$ cd test/
[hadoop@master test]$ ls -l
total 0
-rw-r--r-- 1 root root 0 Feb  4 20:09 readme1.md
-rw-r--r-- 1 root root 0 Feb  4 20:03 readme.md
```

## 5. `mv` 移动文件

命令格式: `mv` <参数(可选)> <源文件/文件夹> <目标文件/文件夹 (没有会创建)>

△ 如果是在当前目录移动, 则相当是 **重命名** 文件/文件夹!

实践: 利用 `mv` 命令重名 `/test/readme1.md` 为 `/test/readme2.md`

```
mv /test/readme1.md /test/readme2.md
```

```
[hadoop@master test]$ ll
total 0
-rw-r--r-- 1 root root 0 Feb  4 20:09 readme2.md
-rw-r--r-- 1 root root 0 Feb  4 20:03 readme.md
```

## 6. `rm` 删除文件

命令格式: `rm` <参数(可选)> <目标文件/文件夹 (没有会创建)>

常用参数解释:

- `rm -i`: 交互式, 会提醒你是否删除
- `rm -r`: 递归删除所有目录中所有内容 (**删除文件夹一定要**)
- `rm -f`: 无任何提示, 直接删除

实践: 删除我们此前创建的 `/test` 文件夹, 并要求交互式提醒。

```
rm -ir /test
```

请再次查看是否还存在 `/test` 目录。

# 3.3 Linux常用命令/技巧

## 3.3.1 Linux常用命令

### 1. `useradd` 创建用户

命令格式: `useradd` <参数> <新建用户名>



常用参数解释：

- o `useradd -m`：创建新用户同时还在 `/home` 创建用户同名文件夹

实践：创建用户 `huihui`，并修改密码为 `123456`。

```
useradd -m huihui    # 创建用户
passwd huihui # 修改密码
```

△ 修改密码时，Linux上不会有任何字符提示输入，输入完毕直接回车就好。

我们还需把用户 `huihui` 添加到 `sudo` 配置文件中：

```
visudo
```

进入 `vim` 编辑器，按下 `i` 进入插入模式，输入下面红框字符。

```
## Allow root to run any commands anywhere
root    ALL=(ALL)        ALL
hadoop  ALL=(ALL)        ALL
huihui  ALL=(ALL)        ALL
```

输入完毕，按下 `ESC`，然后输入 `:wq!` 保存文件即可。

## 2. `su` 切换用户

命令格式：`su <用户名（可选，默认root用户）>`

```
su huihui    # 切换用户
id           # 显示用户信息
```

```
[huihui@master ~]$ id
uid=1001(huihui) gid=1001(huihui) groups=1001(huihui)
```

特别的，我们切换到 `huihui` 用户是个普通用户，有些命令只能在 `root` 用户权限下执行，我们可以在前加上 `sudo`，例如：

```
[huihui@master ~]$ touch /test.txt
touch: cannot touch '/test.txt': Permission denied
```

```
sudo touch /test.txt
```

会提示输入 `root` 用户密码，输入正确命令便可以正确执行了。

最后，你可以切换回 `root` 用户：

```
su # 会要求输入root用户密码
```

## 3. `date/cal` 日期时间

命令格式：`date <参数（可选）>`，显示时间

```
[huihui@master ~]$ date
Tue Feb  4 20:52:11 CST 2020
[huihui@master ~]$ date +%y/%m/%d
20/02/04
```

命令格式：`cal <参数（可选）>`，显示日历

```
[huihui@master ~]$ cal
      February 2020
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
```

#### 4. 查看文件

- `cat <文件名>` : **全部显示**
- `more <文件名>` : **部分显示**, 回车一直往下查看
- `less <文件名>` : **部分显示**, ↑、↓ 键进行查看
- `head <参数 (可选)> <文件名>` : 显示文件头部信息
  - 参数, `-n 3` : 指定显示文件头3行
- `tail <参数(可选)> <文件名>` : 显示文件尾部部分
  - 参数, `-n 3` : 显示3行
  - 参数, `-f` : 一般用于查看日志, 命令不退出, 不断显示更新的内容

#### 5. zip/tar 压缩/打包/解压

`zip` 如果没有安装, 需要先安装:

```
yum instal zip
```

**[zip]命令格式:** `<zip> <参数 (可选)> <目标文件名> <源文件名>`

- 压缩后源文件会被保留

实践: 家目录下创建文件夹 `/test`, 并对其进行压缩。

```
su huihui
mkdir ~/test
zip ~/test.zip ~/test
```

查看家目录下文件:

```
[huihui@master root]$ ls ~
test test.zip
```

**[tar] 命令格式:** `<tar> <参数 (可选)> <目标文件名> <源文件名>`

- `tar` 命令常用户文件 **打包/压缩/解压**

实践: 打包并压缩目录 `~/test` 下所有 `txt` 文档

创建2个 `txt` 文档

```
touch ~/test/1.txt
touch ~/test/2.txt
```

打包--> 压缩所有文档 (**打包只是整理不等于压缩**) :

```
# -c 表示打包文件
# -z 表示打包后在调用gzip进行压缩
# -f 必要参数, 表示使用档案名字
cd ~/test
tar -czf alltxt.tar.gz *.txt # *表示匹配0个或多个字符
```

```
[huihui@master test]$ ls
1.txt 2.txt alltxt.tar.gz
```

实践: 解压 ~/alltxt.tar.gz

```
# -x 表示解压文件
# -z 表示使用gzip解压, 因为解压的文件被gzip压缩过
# -f 必要参数, 表示使用档案名字
# -C 指定解压路径, 下面指定解压到家目录下
tar -xzf ~/test/alltxt.tar.gz -C ~
```

```
[huihui@master test]$ cd ~
[huihui@master ~]$ ll
total 12
-rw-rw-r-- 1 huihui huihui  0 Feb  4 21:25 1.txt
-rw-rw-r-- 1 huihui huihui  0 Feb  4 21:25 2.txt
-rw-rw-r-- 1 huihui huihui 142 Feb  4 21:25 alltxt.tar.gz
```

## 6. locate/find 查找文件

locate 命令如果无法使用, 请先安装:

```
yum -y install mlocate
```

[locate] 命令格式: locate <关键字>

实践: 查找此前创建的 1.txt 在哪

```
sudo updatedb # 先更新下数据库
locate 1.txt
```

```
[huihui@master root]$ locate 1.txt
/etc/brltty/brl-ts-pb65_pb81.txt
/etc/pki/nssdb/pkcs11.txt
/home/huihui/1.txt
/home/huihui/test/1.txt
/usr/lib/firmware/brcm/brcmfmac4330-sdio.Prowise-PT301.txt
```

[find] 命令格式: find <查找位置> <查找参数> <需要查找的文件>

常用参数解释:

- <查找位置>: . 表示从当前目录查找; / 表示从根目录全盘查找
- <查找参数>: 指定以什么方式查找
  - -name: 按文件名查找
- <需要查找的文件>: 支持正则表达形式

实践: 查找此前创建的 1.txt 在哪

```
sudo find / -name *.txt
```

```
[huihui@master root]$ sudo find / -name 1.txt
[sudo] password for huihui:
/home/huihui/1.txt
/home/huihui/test/1.txt
```

## 7. jobs/bg 作业管理

很多时候，我们会使用 `ctrl+z` 中断当前命令后台运行。比如，我们输入 `sudo visudo` 进入编辑模式：

```
[huihui@master root]$ sudo visudo
[sudo] password for huihui:

## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particular
## users.
```

但是，这个时候我们又想切换回中端，于是按下 `ctrl+z`。

使用 `jobs` 命令可以查看后台运行的命令：

```
[huihui@master root]$ jobs
[5]+  Stopped                  sudo visudo
```

这个时候我们如果还想继续编辑，只需键入：

```
fg 5 # 把后台命令前台运行，注意序号可能不是5是1
```

当然，如果不需要再编辑，可以直接杀死该进程：

```
kill %5 # 注意序号可能不是5是1
```

## 3.3.2 常用技巧

本节将介绍linux最常用的技巧。

### 1. TAB自动补全

TAB可使得我们只需键入 **命令/文件/文件夹** 一部分，便可直接按 `TAB` 键自动补全。

如果你快速双击两次 `TAB` 还会显示当前可自动补全的全部选择：

```
cd /home/h
```

```
[huihui@master root]$ cd /home/h
hadoop/ huihui/
```

### 2. 重复命令技巧

- `↑ + 回车`：执行上一条命令
- `!字符`：重复前一个以指定“字符”开头命令
- `!num`：按照历史序号执行
- `!?abc`：重复之前包含abc的命令
- `ESC + .`：复制上一个命令参数

### 3. 命令搜索

- `history`：显示之前命令
- `ctrl+r`：键入关键字后，会自动搜索符合的命令

```
(reverse-i-search)`txt': find / -name l.txt
```

#### 4. 获取帮助

非常常见的事，你会经常不记得一个命令的参数、用法。除了立即谷歌/百度/冥思苦想/...，你还可以借助Linux系统自带命令来查看相关命令用法。

我们将主要解释 `help` / `man` 两种相关方法查看命令帮助。

- `help`

几乎所有命令都可以使用 `--help` 参数获取使用方法、参数信息等。

例如：

```
ls --help
```

```
[huihui@master root]$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                do not ignore entries starting with .
  -A, --almost-all        do not list implied . and ..
      --author              with -l, print the author of each file
  -b, --escape             print C-style escapes for nongraphic characters
      --block-size=SIZE    scale sizes by SIZE before printing them; e.g.,
                          '--block-size=M' prints sizes in units of
                          1,048,576 bytes; see SIZE format below
  -B, --ignore-backups     do not list implied entries ending with ~
  -c                       with -lt: sort by, and show, ctime (time of last
                          modification of file status information);
                          with -l: show ctime and sort by name;
                          otherwise: sort by ctime, newest first
  -C                       list entries by columns
      --color[=WHEN]       colorize the output; WHEN can be 'never', 'auto',
                          or 'always' (the default); color is never
                          forced, unless WHEN is 'always' or the
                          equivalent long option
  -d, --directory           list only directories
  -D, --dired               for -l, list in dired (ls-like) format
  -f, --full                list all files, including the implied . and ..
      --file-separator=SEP use SEP as separator for file names
      --format=FORMAT       with -l, use FORMAT to print file names
                          (the default is to use fill-in formatting)
      --full-time            with -l, print the full ctime
      --full-size            with -l, print the full file size
      --full-time-size      with -l, print the full ctime and file size
      --full-time-size-perc with -l, print the full ctime, file size,
                          and percentage of disk used
      --help               display this help message
  -h, --human-readable      with -l, print sizes in human readable
                          format (e.g., 1K, 1M, 1G)
  -l, --long                list with long format
      --long-dired          for -l, list in dired (ls-like) format
  -L, --logical             list files as logical, not physical
      --long-help           display this help message in long format
  -m, --vertical            list files in vertical (portrait) format
  -M, --vertical-dired      for -l, list in dired (ls-like) format
  -n, --no-pipe             do not use pipes between programs
  -N, --no-sort             do not sort
      --no-truncate         do not truncate output lines
  -o, --other                do not ignore implied . and ..
      --one-line            for -l, list in one-line format
  -O, --show-ownership      with -l, show ownership information
  -P, --show-protective     with -l, show protective permissions
  -q, --hide-control-chars  replace control characters with \C
      --hide-larger-files   with -l, hide files larger than SIZE
      --hide-unreadable     with -l, hide files with unreadable
                          permissions
  -r, --reverse             reverse the order of the sorted list
  -R, --recursive           list the contents of the specified
                          directories recursively
      --recursive-dired     for -l, list in dired (ls-like) format
  -s, --sort=TYPE           sort by TYPE
                          TYPE: -t, time of last modification; -u,
                          time of last access; -g, group; -k,
                          size; -o, ownership; -x, extension; -S,
                          size; -b, block size; -f, file type
  -t, --time                with -l, sort by, and show, time of last
                          modification of file status information
  -u, --time                with -l, sort by, and show, time of last
                          access of file status information
  -v, --version              display the version number
  -w, --width=COLS           set terminal width to COLS
  -x, --horizontal          list files in horizontal (landscape)
                          format
  -X, --horizontal-dired    for -l, list in dired (ls-like) format
  -y, --classify            with -l, print file types as classes
      --classify-dired      for -l, list in dired (ls-like) format
  -Z, --show-security        with -l, show security context
  -z, --zero                with -l, print file names separated by
                          zero characters
      --zero-dired          for -l, list in dired (ls-like) format
```

- `man`

`man` 命令是 Linux **最为常用** 的帮助命令。

命令格式：`man <参数(可选)> <命令>`

常用参数：

- `-k`：此时 `<命令>` 可以不全，搜索相关相关命令

```
man ls
```

```
LS(1)                                     User Commands                               LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters
```

😊 快速定位技巧：进入 `man` 文档 --> 输入 `/ <keywords>` 快速搜索/高亮指定关键字。

例如输入： `/ -a` 再回车

```
LS(1)                                     User Commands                               LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

    Mandatory arguments to long options are mandatory for short options too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file
```

## 3.4 vim 基础入门

后续实验多次需要使用 `vim` 文本 编辑功能，请仔细按照下述步骤实操。

`vim` 编辑器有三种模式：

- 命令模型：默认进入时就是命令模型，此模式下只接受命令对文本进行操作
- 插入模式：命令模式下按下 `i`、`o` 键可进入，此模式下可对文本进行编辑、插入
- EX模式：命令模式下输入 `:` 便可进入EX模式，用户保存修改或退出 `vim`

### 3.4.1 vim实践

进入命令模式有两种方式：

- 初始进入编辑文件时，默认就是命令模式
- 在其它模式下按下 `ESC` 键便可进入命令模型

常用**命令模式**下命令如下：

`vim` 启动后默认进入的就是命令模式，只接受命令，如输入：

- **i**: 进入插入模式 (**Esc** 退回到命令模式, 下同)
- **dd**: 删除整行, 还是命令模式
- **yy**: 复制鼠标当前行, 还是命令模式
- **p**: 粘贴复制的行
- **u**: 撤销上一个操作
- **/**: 查找关键字, 按下 **n** 可以不断切换

按下 **:** 可进入EX模式, 用户保存修改或者退出 **vim**。

常用**EX模式**下命令如下:

- **:q** / **:q!**: 退出 / 强制退出 (比如文件只读修改时)
- **:w**: 保存当前文件
- **wq** == **:x**: 保存并退出
- **:set number**: 显示行号
- **:sh**: 切换回命令行, **ctrl+d** > 返回vim

实践: 在 **~** 目录下创建 **hello.txt**, 使用 **vim** 编辑。

```
touch ~/hello.txt
vim ~/hello.txt
```

### 1. 插入模式

按下 **i** 键插入模式, 输入以下字符

```
[huihui@master root]$ vim ~/hello.txt
hello
hello linux!
```

### 2. 疯狂复制

按下 **ESC** 键进入命令模式 --> 光标移动到 **hello linux** 那行 --> 按下 **yy** 进行复制 --> 按下 **p** 进行多次复制:

```
[huihui@master root]$ vim ~/hello.txt
hello
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
hello linux!
```

### 3. 显示行号

按下 **ESC** 键进入命令模式 --> 输入 **:set number** --> 显示行号

```
[huihui@master root]$ vim ~/hello.txt
```

```
1 linux
2 hello linux!
3 hello linux!
4 hello linux!
5 hello linux!
6 hello linux!
7 hello linux!
8 hello linux!
```

#### 4. 保存退出

按下 **ESC** 键进入命令模式 --> 输入 **:wq!** --> 退出

查看是否保存成功：

```
cat -n ~/hello.txt    # 参数n表示同时显示行号
```

```
[huihui@master root]$ cat -n ~/hello.txt
1  linux
2  hello linux!
3  hello linux!
4  hello linux!
5  hello linux!
6  hello linux!
7  hello linux!
8  hello linux!
```

## 3.5 用户权限基础

### 3.5.1 用户与组

[用户] 限制使用者或者进程 可以使用，不可以使用哪些资源。

- **用户种类**：root用户（ID: 0）；系统用户（ID:1~499）；普通用户（ID:500以上）
- **用户与组**：每个用户属于一个**主组**，一个或多个**附属组**
- **用户与shell**：每个可登陆用户有一个指定**shell**
- **用户相关配置文件**：
  - `/etc/passwd`：保存用户信息

```
[root@iZuf65a92xmy85jsmlkneiZ ~]# vim /etc/passwd
名字      ID
root:x:0:0:root:/root:/bin/bash  shell
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

- `/etc/shadow`：保存用户密码（加密的）

```
[root@iZuf65a92xmy85jsmlkneiZ ~]# vim /etc/shadow
root:$6$UeZEmZFu9xrJNAFY$5XKonKK3gsy/CPrqmzuC4vkYzJGiANZ8QVNZ4EQws0
80:18044:0:99999:7:::
bin:!:17110:0:99999:7:::
daemon:!:17110:0:99999:7:::
adm:!:17110:0:99999:7:::
用户密码，没有密码用！！
```

- `/etc/group`：保存组信息

请自行查看，并截图在保存。



**[组]用来管理用户**，每个组拥有一个 `GroupID`。

独立完成以下用户创建及相关操作。

## 1. 创建用户

命令格式: `useradd <参数(可选)> <用户名>`

执行 `useradd` 命令，会执行以下默认操作：

1. 在 `/etc/passwd` 添加用户信息
2. 为用户建立一个家目录 `/home/<username>`
3. 将 `/etc/shell`（用户刚建立的初始文件）复制到用户家目录

常用参数解释：

- `-u`：指定userID
- `-g`：指定主组，默认**会建立一个和用户同名的组**，用户默认属于这个组。
- `-G`：指定附属组

实践：创建一个用户 `lxSmile`，并指定 `userID=555` (普通用户id)，所属组为 `testGroup`。

```
su root                # 切换到root权限
groupadd testGroup     # 创建组testGroup
useradd -u 555 -g testGroup lxSmile
```

查看创建用户信息：

```
id lxSmile
```

```
[root@master ~]# id lxSmile
uid=555(lxSmile) gid=1002(testGroup) groups=1002(testGroup)
```

```
cat -n -E /etc/passwd    # 在/etc/passwd配置文件查看用户信息
```

```
46 huihui:x:1001:1001::/home/huihui:/bin/bash$
47 lxSmile:x:555:1002::/home/lxSmile:/bin/bash$
```

## 2. 修改用户

命令格式: `usermod <参数(可选)> <用户名>`

常用参数解释：

- `-l`：修改用户名
- `-u`：新userID
- `-g`：用户所属组
- `-G`：用户所属附属组

实践：修改用户 `lxSmile`，修改 `userID=666` (普通用户id)，所属组为 `testGroup1`。

```
groupadd testGroup1
usermod -u 666 -G testGroup1 -g testGroup1 lxSmile
```

```
[root@master ~]# id lxSmile
uid=666(lxSmile) gid=1003(testGroup1) groups=1003(testGroup1)
```

### 3. 删除用户

命令格式: `userdel <参数(可选)> <用户名>`

常用参数解释:

- `-r` : 同时删除用户家目录 (默认不删除)

实践: 删除刚刚创建的用户 `lxSmile` , 并同时删除其家目录。

```
userdel -r lxSmile
```

查看是否还存在:

```
[root@master ~]# id lxSmile
id: lxSmile: no such user
```

## 3.5.2 文件权限实操

[文件权限]用来管理文件读、写、执行, 每个文件都有特定权限、所属组、所属用户。

- **权限类型**: 权限分为读(r)、写(w)、执行(x)
- **UGO权限控制**: Linux权限基于UGO模型进行控制
  - **UGO**: **U**ser、**G**roup、**O**ther, 每一个文件权限都基于UGO设置 (即用户、所属组、和其它用户能操作权限)
  - `ls -l <文件/文件夹名>` : 可查看权限

根据下图我们可以知道 **test** :

类型为目录(d), 所属用户权限为读写执行(rwx), 所属组权限为读写执行(rwx), 其它用户权限为读执行(r-x), 所属用户为huihui, 所属组为huihui, 大小为4096B, 修改时间为21:26。

```
[huihui@master ~]$ ll
total 16
-rw-rw-r-- 1 huihui huihui 0 Feb 4 21:25 1.txt
-rw-rw-r-- 1 huihui huihui 0 Feb 4 21:25 2.txt
-rw-rw-r-- 1 huihui huihui 142 Feb 4 21:25 alltxt.tar.gz
-rw-rw-r-- 1 huihui huihui 244 Feb 4 22:49 hello.txt
d-rwxrwxr-x 2 huihui huihui 4096 Feb 4 21:26 test
-rw-rw-r-- 1 huihui huihui 184 Feb 4 21:08 test.zip
```

独立完成以下文件权限相关操作。

首先我们创建相关测试文件:

```
su
cd /
mkdir test5
touch /test5/5.txt
```

#### 1. 修改文件所属用户

命令格式: `chown <参数(可选)> <用户名> <文件/文件夹>`

常用参数:

- `-R` : 递归的修改文件夹下所有子文件/文件夹的所属用户

```
chown -R huihui /test5 # 修改文件夹test5所属用户为huihui
```

```
[root@master /]# ll -d /test5
drwxr-xr-x 2 huihui root 4096 Feb  5 11:58 /test5
```

## 2. 修改文件所属组

命令格式: `chgrp <参数(可选)> <组名> <文件/文件夹>`

常用参数:

- `-R` : 递归的修改文件夹下所有子文件/文件夹的所属组

```
chgrp -R huihui /test5 # 修改文件夹test5所属组为huihui
```

```
[root@master /]# ll -d /test5
drwxr-xr-x 2 huihui huihui 4096 Feb  5 11:58 /test5
```

## 3. 修改文件权限

命令格式: `chmod <参数(可选)> <模式> <文件/文件夹>`

常用参数:

- `-R` : 递归的修改文件夹下所有子文件/文件夹的所属用户

修改 `/test5/5.txt` , 权限从 `rw-r--r--` ----> `r--rwxrwx`

```
[root@master /]# ll -d /test5/5.txt
-rw-r--r-- 1 huihui huihui 0 Feb  5 11:58 /test5/5.txt
```

# 用户 (u) 权限[减去写]权限

```
chmod u-w /test5/5.txt
```

# 组 (g) /其他用户 (o) 权限[加上写、执行]

```
chmod go+wx /test5/5.txt
```

```
[root@master /]# ll -d /test5/5.txt
-r--rwxrwx 1 huihui huihui 0 Feb  5 11:58 /test5/5.txt
```

# 4 实验小结

本次实验虽然只是简单地对接下来实验所需的linux操作进行初步熟悉,但是或许第一次接触linux的你还是觉得颇为困难。但是不用担心,后续所有有关linux操作,你在本次实验中已经全部学习过了,相关操作也有详细的实验指导。

接下来,你将正式开始 `spark/hadoop` 的学习,希望你能收获满满: )。