

# Linux (和服务器) 基础操作分享

---

作者：陈肯

2020.10.15

## 准备工作：

---

- Windows: 在微软商店下载Ubuntu子系统
- Mac OS: 无需准备

## Note:

---

示例代码中有方括号的部分代表是需要按照需要更改的，如第一个例子

```
$ mkdir [foldername]
```

实际运用中可能是

```
$ mkdir ken
```

## Linux基本操作

---

### 1. 创建文件夹

```
$ mkdir [foldername]
```

### 2. 进入文件夹

```
$ cd [foldername/directory]
```

其中，\$ cd .. 代表返回上一级文件夹。

- Remark 1: Linux中，文件路径 . 代表当前文件夹， .. 代表上一级文件夹
- Remark 2: 路径名最前面有斜杠 / 的为绝对路径，没有的为相对路径

### 3. 查看当前路径下的文件和文件夹

```
$ ls
```

```
$ ls -R  
# 查看当前路径下文件和文件夹内所有文件信息
```

### 4. 打印出当前路径

```
$ pwd
```

## 5. 写文件/修改文件

```
$ vim [filename]
```

进入vim编辑器后，常用操作如下：

- 按 i 进入插入(insert)编辑模式，可以开始进行文件编辑。
- 编辑完成后，按 esc 退出编辑模式。
- 完成所有修改后，输入

```
:wq
```

按回车键退出并保存修改(write and quit)。或者输入

```
:q
```

按回车键不保存更改并退出(quit)

## 6. 查看文件内容

```
$ cat [filename]
```

## 7. 移动文件(和重命名文件)

```
$ mv [filename] [folder_path]  
# 移动文件
```

```
$ mv [filename] [newname]  
# 重命名
```

- Remark: 如果你想批量操作文件，如，你想操作的文件的格式为'20200101.csv','20200102.csv',..., 那么，如果你想批量移动1月的文件，你只需要输入

```
$ mv 202001*.csv [folder_path]
```

## 8. 复制文件(夹)

```
$ cp [filename] [folder_path]
```

- Remark 1: 与我们常说的'复制'不同，此处的复制会复制一个一模一样的文件，并放入新的文件夹中
- Remark 2: 在Linux中，对文件夹的操作需要在命令后加上 -r 如：

```
$ cp -r [foldername] [folder_path]
```

## 9. root权限(SuperUser Do):

```
$ sudo [command]
```

- Remark: 运行sudo后会需要输入密码，刷完密码按回车即可

## 10. 下载

```
$ wget [website_link]
```

## 11. 下载系统包

```
$ apt [options] [command] [package ...]
```

- Remark: 用于 Debian 和 Ubuntu 系统，此命令需要root权限。具体用到的时候可以上网查，此处不展开。

## 12. 查看系统状态

可查看运行的进程，系统cpu和内存情况等。

```
$ top
```

- Remark: 按键盘上的上下键滚动进程，按Q退出

## 13. 查看硬盘空间

```
$ df -h
```

- Remark: -h 命令表示显示为human-readable的形式，一般会使用 df -h

## 14. 查看当前文件夹下文件占硬盘空间

```
$ du -h
```

## 15. 比较两个文件的不同处

```
$ diff [file1] [file2]
```

- Remark: 需要先用apt/yum安装diffutils包

## 16. 查看历史输入过的命令

```
$ history
```

## 17. 命令详解

会输出命令的详细使用规则，包括可用参数等

```
$ man [command]
```

## 18. 压缩, 解压

```
$ zip [filename]
$ unzip [filename]
```

If you use gzip,

```
$ gzip [filename]
$ gunzip [filename]
```

- Remark 1: gzip压缩率会更高, 传输文件时建议大家先压缩再传输。 .gz格式文件适用范围会比较广, 基本所有的Linux/Unix based的系统都能有很好的支持。另外RAR格式无法在不安装额外包的情况下被Linux解压, 一般不会使用。
- Remark 2: pandas 可以直接读取压缩文件, 如 `df = pd.read_csv('example.csv.gz', compression='gzip')`

## 19. 退出正在执行的操作

按下键盘上的Control+C (Mac电脑也一样, 不是command)

## 20. 删除文件

```
$ rm [file_name]
```

# 服务器相关操作

## 1. 连接服务器

```
$ ssh [username]@[server_link]
```

然后输入密码

## 2. 文件传输

从本机到服务器

```
$ scp [filename] [username]@[serverlink]:[file_path]
```

Example:

```
$ scp test/helloworld.py chenken@10.20.253.10:ken/test
```

从服务器到本机: 在本机上运行,

```
$ scp [filename] [username]@[serverlink]:[file_path]
```

- Remark 1: 传输文件的命令一定要在本机上运行
- Remark 2: 由于Windows系统文件格式和Linux, Mac OS不同, 可以通过在Windows上安装Ubuntu子系统来完成文件传输的需求(或者使用Windows工具WinSCP, 但是我没有使用过不了解)。注意在Windows terminal中输入scp命令是无效的。但是在Mac OS和Linux系统电脑上都可以直接使用。Windows用户使用子系统时需要注意正反斜杠的区别。而且子系统下原来在Windows C盘的文件的path会变为 /mnt/c/...
- Remark 3: 文件较大时, 推荐使用gzip格式传输

### 3. 安装anaconda及相关操作

#### 1. 在清华镜像源上安装anaconda

```
$ wget https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/Anaconda3-2020.07-Linux-x86_64.sh
```

#### 2. 安装anaconda, 在anaconda所在的目录下运行

```
$ bash Anaconda3-2020.07-Linux-x86_64.sh
```

安装过程中一直按回车和输入yes

#### 1. 安装完毕后, 导入环境变量

##### 1. 打开配置文件

```
$ vim ~/.bashrc
```

##### 2. 在最后一行添加语句

```
export PATH=/home/XXX/anaconda3/bin:$PATH
```

XXX为你的用户名

##### 3. 更新配置文件

```
$ source ~/.bashrc
```

- Remark: 如果无法正常使用anaconda, 重连服务器。

#### 1. anaconda常用命令

##### 1. 创建虚拟环境

```
$ conda create --name [name] python=3.7
```

##### 2. 激活虚拟环境

以下两个命令都可以

```
$ conda activate [name]
$ source activate [name]
```

### 3. 安装python包

使用conda安装

```
$ conda install [package_name]
```

使用pip安装

```
$ pip install [package_name]
```

### 4. 更新当前环境所有包

```
$ conda update --all
```

### 4. 运行python文件

```
$ python [python_file_name]
```

### 5. 运行文件并且后台运行

使用这个命令后，就算本机上终端连接关闭后，程序还能继续运行，并且print或者报错内容会存到同级目录下nohup.out文件中。

```
$ nohup python [python_file_name] &
```

- Remark: 使用后会返回一个进程号(PID)，可以记录下来以便杀死程序。

### 6. 杀死进程

```
$ kill -9 [PID]
```

### 7. 根据文件名/文件类型查找运行中的进程

```
$ ps -aux|grep [key_word]
```

Examples:

```
$ ps -aux|grep python
$ ps -aux|grep hello_world
```

