

Linux基础

1.软件更新

1.源

Unix MAC APPStore 是闭源

`sudo apt-get update` 检测更新

`sudo apt-get upgrade` 更新

`sudo !!` 给前一条命令加权限

`.d` 文件 是目录

`sudo apt-get install *` 安装软件

`sudo apt-cache search *` 查找软件

2.命令

命令变量赋值时不能加空格

`ls` 查看当前目录文件

`cp` 拷贝

`mkdir` 创建目录

`touch` 创建空白文件

`cd ..` 返回上一层 `cd -` 返回上一次进入的文件

`echo` 输出 打印

`awk '{printf $第几个}'` （以空格换行为分割符分成几部分）输出第几部分

`cat` 强制读

`cut -d "" -f` `-d` 用来定义分隔符，默认为 `tab`键，`-f` 表示需要取得哪个字段

eg: `echo "a/b/c" | cut -d '/' -f 1` 执行结果是 `a`

先按 `/` 分段，第一个字段是 `a`，第二个字段是 `b`，第三个字段是 `c`，`-f` 就是取第几个字段，如果后面的是 `-f 1-`，就是取后一字段 `c`

`pwd` 输出路径

`/home` 不是家目录 `HOME` 家目录

`rm` 删除 `-f` 强制 `-r` 递归

删除特殊字符的文件

eg: `"*"` `rm *` 用 `\` 转义

`mv` 移动

`scp` 拷贝文件到远程主机 `scp 文件名 远程主机名@远程主机IP:文件名`

`port` 端口

`man` 查看使用手册

`tlldr` 查看使用手册（推荐使用 需要python-pip3 安装）

`grep` “内容” 文件 检索文件内容

`df -h` 查看文件系统占用空间

`which` 查看命令在哪 eg: `which ls`

`type` 查看命令类型 eg: `type ls`

`history` 查看历史命令

`!编号` 执行该命令

`ctrl+r` 匹配最近的历史名命令

`date` 查看时间

`free -m` 查看内存占用

`p` 管道 先进先出 出去就没了

`0` 标准输入 `1` 标准输出 `2` 标准错误输出

`2>&1 >/dev/null` 删除输出

3.ssh

下载 `openssh-server`

`ssh-keygen` 生成公用密钥

`ifconfig` 查看本机IP地址

将本主机的 `id_rsa.pub` 传到目标机器上的 `~/.ssh/authorized_keys` (使用 `scp` 传) (`ssh-copy-id -i ~/.ssh/id_rsa.pub 目标机器用户名@地址`)

4.vim

`Ctrl+Z` 挂起vim

`jobs` 查看挂起的有哪些

`fg 序号` 返回vim

`ps -ef | grep 内容` (`ps -ef` 是查看所有的进程的 然后用 `grep` 筛选出你要的信息)

`kill 名称` 杀死进程

`kill -9 名称` 强制杀死进程

`Ctrl+C` 终止进程

`Ctrl+Z` 挂起进程

`bg` 把被中断的任务放在后台执行

`jobs` 查看挂起的有哪些

`fg 序号` 返回进程

5.装ubuntu系统

创建主分区：

60G 主分区 空间起始位置 Ext4日志文件系统 /

创建swap分区：

4G 逻辑分区 空间起始位置 交换空间

创建boot分区：

500MB 逻辑分区 空间起始位置 Ext4日志文件系统 /boot

创建home分区：

10G 逻辑分区 空间起始位置 Ext4日志文件系统 /home

Linux系统信息

1. `uptime` 打印系统运行时长和平均负载

- `-s` 登录时间
- `-p` 显示更美观

2. `w` 当前登录用户列表及正在执行的任务

- `-h` 不打印头信息
- `-s` 使用短输出格式
- `-f` 切换显示FROM项，默认显示

3. `who` 显示当前登录系统的用户信息

- `-H` 显示各栏的标题信息列
- `-u` 显示空置时间
- `-q` 只显示登入系统的用户名称和总人数

4. `whoami` 打印当前有效的用户名称

- `whoami`
- `who am i`
- `logname`

5. `last` 显示用户最近登录信息

- `-f "file"` 指定记录文件
- `-n "num"` 指定输出记录数
- `-x` 显示系统关机，重新关机等信息

6. `uname` 打印当前系统信息

- `-a` 全部信息
- `-m` 电脑类型

`-n` 网络上主机的名称

`-v` os版本

`-p` 处理器类型

`-i` 硬件平台

`-o` 操作系统名称

7. `date` 显示或设置系统时间与日期

`date +%s` 显示当前时间戳

`date +%s -d "时间"` 将该时间转为时间戳

选项：

`-d "string"` 显示字符串所指的日期

`-s "string"` 设置时间

`-u` 显示GMT（如果是CST的情况下）

参数：

`+日期格式` 显示使用的日期格式

8. `cal` 显示日历

`-1, -3` 显示最近一个月和三个月的日历

`-j` 显示一年的第几天

`-y` 显示当前年

9. 给其他机主发信息

`w` 显示所有登录用户名

`wall "内容"` 给所有人发

`write 机主名` 给某人发

`echo "内容" >> /dev/设备/序号` 给某人发

10. `ping` 测试主机之间的连通性

`-d`：使用Socket的SO_DEBUG功能；`-c<完成次数>`：设置完成要求回应的次数；`-f`：极限检测；`-i<间隔秒数>`：指定收发信息的间隔时间；`-I<网络界面>`：使用指定的网络界面送出数据包；`-l<前置载入>`：设置在送出要求信息之前，先行发出的数据包；`-n`：只输出数值；`-p<范本样式>`：设置填满数据包的范本样式；`-q`：不显示指令执行过程，开头和结尾的相关信息除外；`-r`：忽略普通的Routing Table，直接将数据包送到远端主机上；`-R`：记录路由过程；`-s<数据包大小>`：设置数据包的大小；`-t<存活数值>`：设置存活数值TTL的大小；`-v`：详细显示指令的执行过程。

SHELL

`/dev/null` ubuntu垃圾桶

`/dev/zero` ubuntu垃圾桶

`#!/bin/bash` shell文件开头

代码后 一个空格# 后写注释

`bash` 文件 对. `sh` 文件作用同于 `./`

`chmod` 改变权限 `a+x` a所有人 给a加上x权限

`./可执行文件` 执行文件

`/` 目录文件拼接

``` 命令替换符

`if [[ 0 ]]` 执行

`if [[ 1 ]]` 不执行

### 1. `read [-选项] [变量...]`

`-a array` 把输入赋值到数组array中，从索引号零开始

`-d delimiter` 用字符串delimiter中的第一个字符指示输入结束，而不是一个换行符

`-e` 使用Readline来处理。这使得与命令相同的方式编辑输入

`-n num` 读取num个输入字符，而不是整行

`-p prompt` 为输入显示提示信息，使用字符串prompt

`-r` Raw mode. 不把 `\` 解释为转义字符

`-s` Silent mode.

`-t seconds` 超时

`-u fd` 使用文件描述符fd中的输入，而不是标准输入

## 2.echo string

`echo -e "\n"` 开启转义 (会将\n转化为换行)

`echo ""` (中文"" 会输出 "")

`echo "\"hello\""` 会在输出 "hello"

## 3.printf [格式化字符串] [arguments...]

`printf "hello $s\n" "hai zei"`

输出: hello hai zei

`$` 取值 `${较长的内容}` 取值

`echo $` 输出值

`$0` 获取当前执行shell脚本的文件名，包括路径

`$n` 获取当前执行脚本的第n个参数，n=1...9，如果n大于9，则需要将n使用大括号扩起来

`$*` 获取当前shell的所有参数，将所有命令行参数视为单个字符串，相当于 `$1$2$3`

`$#` 得到执行当前脚本的参数个数

`$@` 获取这个程序所有参数，并保留参数 之间的任何空白，相当于 `$1` `$2` `$3`，这是将参数传给其他程序的最好办法

`$?` 判断上一指令是否成功执行，0为成功，非零为不成功

`$$` 取当前进程的pwd

`$!` 上一指令的pwd

## 4.数组

可以这样定义数组 `a=(1 2 3)` 或 `declare -a a` 或 `a[下标]=数值`

`${}`

`${array[*]}` 输出数组内容

`${array[@]}` 输出数组内容

`${#array[@]}` 个数

`${!array[@]}` 下标

`array+=(a b c)` 数组追加

`sort` 数组排序

`unset` 删除数组与元素

## 数据提取操作

---

### 1. `tr` 对标准输入的字符替换，压缩，删除

`tr [cdst] <字符集> <字符集>`

`c` 取代所有不属于第一字符集的字符

`d` 删除所有属于第一字符集的字符

`s` 将连续重复的字符集的字符

`t` 先删除第一字符集较第二字符集多出的字符

### 2. `cut` 切分

`cut [-dfc] <file>`

`-d c` 以c字符分割

`-f num` 显示num字段的内容 `n-` `n-m` `-m`

`-b num` 字节

`-c num` 字符

### 3. `grep` 检索

`grep [-acinv] <string> <file>`

`-a` 将二进制文件—普通文件的形式搜寻数据

`-c` 统计搜寻到的次数

`-i` 忽略大小写

`-n` 顺序输出行号

`-v` 反向输出（输出没找到的）



## 4. sort 排序

```
sort [-fbMnrtuk] <file or stdin>
```

-f 忽略大小写

-b 忽略最前面的空格符

-M 以月份名称排序

-n 以纯数字方式排序

-r 反向排序

-u

-t 分隔符，默认[TAB]

-k 以那个区间排序

## 5. wc 统计字符，字数，行数

```
wc [-lwm] <file or stdin>
```

-l 仅列出行号

-w 仅列出多少字

-m 仅列出多少字符

-c 仅列出多少字符

## 6. uniq 去重

```
uniq [-ic]
```

-i 忽略大小字符的不同

-c 进行计数

## 7. tee 双向重导项

```
tee [-a] file
```

-a append

## 8. split 文件切分

```
split [-bl] <file> prefix
```

`-b size` 切分为size大小的文件

`-l num` 以num行为大小切分

## 9. `xargs` 参数代换

`xargs [-0pne] <command>`

`-0` 将特殊字符还原为普通字符

`-eEOF` 当xargs读到EOF时停止

`-p` 执行指令前询问

`-n num` 每次执行command时需要的参数个数

# 文件与目录基本操作

---

## 1. `cd` 切换工作目录

`cd /etc` 直接切换到/etc目录

`cd ..` 切换到上层目录

`cd .` 切换到当前目录

`cd` 回到自己的家目录

`cd ~` 回到自己的家目录

`cd -` 回到上次工作目录

## 2. `pwd` 打印当前工作目录

`-l` 显示逻辑工作目录

`-p` 显示物理工作目录

## 3. `mkdir` 创建目录

`-p` 自动创建父目录

`-m` 设置权限

## 4. `rmdir` 删除目录

`-p` 删除祖先

## 5. `cp` 拷贝

`-i` 若文件存在，询问用户

`-r` 递归复制

`-a` pdr的集合

`-p` 连同文件属性一起拷贝

`-d` 若源文件为连接文件属性，则复制连接文件的属性

`-s` 拷贝为软连接

`-l` 拷贝为硬连接

`-u` 源文件比目的文件新才拷贝

## 6. `rm` 删除

`-i` 互动模式

`-r` 递归删除

`-f` 强制

## 7. `mv` 移动

`-i` 互动模式

`-f` 强制

`-u` 源文件更新才会移动

## 8. `dirname` 最后一个 `/` 之前的内容

`basename` 最后一个 `/` 之后的内容

## 9. `cat` 正向连续读

`tac` 按行反向读

## 10. less 显示那页会给找的东西标出来

/string 向下查找 n 继续向下查找

/?string 反向查找 N 继续反向查找

## more 显示那页不会给找的东西标出来

/string 向下查找string关键字

:f 显示文件名称和当前显示的行数

q 离开

## 11. sed -n '行号, 行号p' 输出这些行

### n1 输出行号显示文件

-b 行号指定的方式

-b a 相当于 cat -a

-b t 相当于 cat -b

-n 列出行号的表示方法

-n ln 行号在屏幕最左边显示

-n rn 行号在自己字段的最右边显示

-n rz 行号在自己字段的最右边显示, 前面自动补全0

-w <num> 行号所占位数

## 12. od 以二进制方式查看文件内容

## 13. head 查看头几行

-n num 显示前num行

-n -num 除了后num行外, 其他的都显示

### tail 查看末尾几行

-n num 显示文件后num行

-n +num 除了前num行, 其他的都显示

## 14. 文件隐藏属性

```
chattr [+-=] [] <file or dir>
```

- A 不修改atime
- S 同步写入
- a 只能增加数据
- c 自动压缩，解压
- d 不会被dump程序备份
- i 不能删除，修改，建立连接
- s 文件删除时，直接从磁盘删除
- u 文件删除时，数据内容存在磁盘中

```
lsattr [-adR] <file or dir>
```

- a 打印隐藏文件的隐藏属性
- d 如果是目录，仅打印目录的信息
- R 递归

## 15. which 寻找可执行文件

查找path路径下所有的可执行文件

## 16. whereis 寻找特定文件

```
whereis [-bmsu] <file or dir>
```

- b 只查找二进制文件
- m 只查找manual路径下的文件
- s 只查找source源文件
- u 查找其他文件

## 17. locate 模糊定位

```
locate [-ir] keyword
```

`-i` 忽略大小写

`-r` 后面可接正则表达式

## 18. `find` 高级查找

`find [PATH] [option] [action]`

`find 路径 参数`

`find 路径 参数 -exec (命令) {} (命令) \`

在找到所需要的文件时 `-exec` 开始接下来的命令操作 `{}` 找到的内容 `\` 结束

## 19. 进入root权限后 建立用户 实现特殊权限目录

`sudo -i` 进入root权限

`su - 用户名` 进入其他用户

`groupadd 组名` 新建组

`useradd -G 组名 用户名` 在该组下新建用户

`groups 用户名` 查看用户在哪个组

`usermod -a -G 组名 用户名` 将该用户加入该组

`passwd 用户名` 给该用户设密码

`userdel 用户名` 删掉用户

`userdel -r 用户名` 删掉用户 删掉所有用户信息

`-R` 是递归（下同）

`chown -R 用户名 文件名或目录名（下同）` 将该用户赋予操作文件的权限

`mkdir /文件名`

`chgrp -R 组名 文件名` 将该组下所有用户赋予操作文件的权限

`chmod -R +w /文件名` 加写的权限

`chmod -R +t /文件名` 该目录下只能删除自己建的文件

`chmod -R +s /文件名` 用户在该目录里，有效组变为所属组

进行上列操作后在该组下的用户有读写的权限，但只能删除自己建的文件，其他组的用户只有读的权限

# 用户管理

---

## 一、用户管理的重要配置文件

---

`/etc/passwd` 用户名 密码位 用户编号 归属组编号 姓名 \$HOME Shell

`/etc/shadow` 用户名 已加密密码 密码改动信息 密码策略

`/etc/group` 群组名 密码位 群组编号 组内用户

`/etc/gshadow` 群组密码相关文件

`/etc/sudoers` 用户名 权限定义

## 二、用户管理相关的命令

---

### 1. `su` 命令

`-` | `-l` 重新登录

`-m` | `-p` 不更改环境变量

`-c comand` 切换后执行命令，并退出

### 2. `sudo` 命令

`-s` 切换为root shell

`-i` 切换到root shell，并初始化

`-u username` | `uid` 执行命令的身份

`-l` 显示自己的权限

### 3. `passwd` 命令

`-d` 清除密码

`-l` 锁定账户

`-e` 使密码过期

`-S` 显示密码认证信息

`-x days` 密码过期后最大使用天数

`-n days` 密码冻结后最小使用天数

`-s` 更改登录shell

`-f` 更改用户信息

## 4. `gpasswd` 命令

`-a username` 将用户加入群组

`-d username` 将用户从群组中删除

`-r` 删除密码

`-A username` 将用户设置为群组管理员

`-M username1,username2` 设置群组成员

## 5. `/etc/login.defs` 新建用户规则

`/etc/skel` 新建用户默认文件

## 6. `useradd` 命令

`-d dir` 指定\$HOME

`-m` 自动建立\$HOME

`-M` 不自动建立\$HOME

`-s shell` 设置用户登录shell

`-u uid` 设置用户编号

`-g groupname` 设定用户归属群组

`-G groupname` 设置用户归属附加群组

`-n` 不建立以用户名称为群组名称的群组

`-e days` 设置帐号过期时间

`-f days` 缓冲时间，days天后关闭帐号

`-c string` 设置用户备注

`-D [表达式]` 更改预设值

## 7. `userdel -r <username>` 删除用户相关文件和目录



## 8. usermod 命令

- c string 修改备注信息
- d dir 修改\$HOME
- e days 密码期限
- f days 密码过期后宽限的日期
- g groupname 修改用户所属群组
- G groupname 修改用户所属附加群组
- l username 修改用户账户名称
- L 锁定用户密码，使密码无效
- s shell 修改用户登录后所使用的shell
- u uid 修改用户ID
- U 解除密码锁定

## 9. id 命令

- g 下属所属组id
- G 显示附加组id
- n 显示用户，所属组码，或附加群组的名称
- u 显示用户id
- r 显示实际id

# 本地使用 github 命令

---

## 1.本地使用 github 命令

- git status 查看状态
- git add 文件名 交给管理
- git commit -m "" 交给管理
- git push 提交
- git pull 更新同步本地和github

`git log` 查看历史提交状态

`git reset --hard ID号` 退回到某个版本命令

`git --help` 帮助命令

`git pull origin master` 将远程仓库里面的项目拉下来

`dir` 查看有哪些文件夹

## 2.删除文件夹或文件

`git rm -r --cached 文件或文件夹名` 删除文件或文件夹

`git commit -m "说明"` 提交,添加操作说明

`git push -u origin master` 将本次更改更新到github项目上去

## 3.实现本地免密登录github

将本机的公匙添加到github

`git config --global user.name "自己的用户名"`

`git config --global user.email "自己的邮箱"`

在本机建立一个文件夹将github资料库拷贝过来 `git clone 自己的仓库链接`