Linux文件目录和分区

文件目录：

/：根目录

/root：存放root用户的相关文件

/home：存放普通用户的相关文件

/bin：存放普通用户的可执行命令

/sbin：存放需要具备一定权限才可以使用的命令

/mnt：默认挂载光驱（iso）的挂载点

/etc：存放系统配置相关的文件

/var：存放经常变化的文件

/usr：存放系统的应用程序，比如apt-get install xxx安装的软件

/tmp：存放临时文件、日志文件

/dev：存放硬盘设备信息

/boot：存放Linux操作系统启动时的引导文件

Linux设备命名规则：

Linux中IDE硬盘，驱动标识符为“hdx-”，其中hd表示设备类型为IDE硬盘，x为盘号，表示第几块硬盘（a为基本盘，b为基本从属盘，c为辅助主盘，d为辅助从属盘），“-”代表分区，前4个分区用数字1到4表示，它们是主分区或扩展分区，从5开始就是逻辑分区（扩展分区不能直接使用，需切割出逻辑分区，逻辑分区的数量没有限制）

Linux中SATA或SCSI硬盘，驱动标识符为“sdx-”，sd表示设备类型为SATA硬盘，如：

hda2：表示第一个IDE硬盘上的第二个主分区或扩展分区

sdb5：表示第二个SATA硬盘上的第一个逻辑分区

通过命令**fdisk -l**：查看Linux系统分区的具体情况

通过命令**df -h 目录**： 查看目录在哪个分区和磁盘使用情况，下图说明“/home”对应第一块逻辑分区

挂载点和分区：

在Linux系统里，“分区”被称作挂载点，“挂载点”的意思是：把一部分硬盘容量映射成为一个文件夹，这个文件夹的名字就叫做“挂载点”，因此Linux中只有文件夹形式存在的挂载点，而不会有C盘，D盘，E盘等，Linux中分区（硬盘容量）映射为文件夹（挂载点）

挂载命令：mount [-参数] [设备名称] [挂载点]

卸载命令：umount 挂载点

比如：mount  /dev/sda1  /test/ 将主分区挂载到test目录；umount /test/ 卸载

Linux常见挂载点的容量设置：

/boot：100M

swap：交换分区，通常为物理内存的两倍

/：根分区，容量尽可能大

/home：容量尽可能大

Linux常用命令（有问题Google）

**Tab：**双击，命令补全

**sudo su**：进入root用户

**apt-get install package：**安装程序

**apt-get remove --purge softname：**卸载并清除配置

**./test.bin &**：在当前目录下安装bin文件，./指当前目录，&指程序以后台方式运行

**startx**：进入Linux图形界面

**shutdown –h now**：立刻关机

**reboot；shutdown–r now**：重启

**dos2unix**：Windows文件（\r\n）转化为Linux文件（\r）

**unix2dos**：Linux文件（\n）转化为Windows文件（\r\n）

**exit**：退出

**clear**：清空

**pwd**：显示当前目录

**cd**：改变目录

cdnull：切换到当前用户（如zzjie）的主目录

**ls**：列出文件和目录

ls -a：显示隐藏文件

ls -l：显示长列表格式

**ll**：显示文件详细内容

**mkdir**：建立目录

**rmdir**：删除空目录

**touch**：建立空文件

**cp**：复制

cp-r 文件夹 目标目录：拷贝文件夹

**scp**：在Linux主机间复制文件

scplocal\_file remote\_username@remote\_ip:remote\_folder

scplocal\_file remote\_username@remote\_ip:remote\_file

scplocal\_file remote\_ip:remote\_folder

scplocal\_file remote\_ip:remote\_file

scp–r：复制文件夹

scp：从远程到本地同理

**rsync：**远程数据同步

rsync–option src dest

rsync–option src remote\_username@remote\_ip:remote\_file

rsync–option src remote\_username@remote\_ip:remote\_fodler

**mv**：移动文件和改文件名

**rm**：删除文件和目录

rm-rf：递归、强制删除目录和内容，fr：recursive force

**ln**：链接，指向源文件

ln-s 源文件 目标文件：目标指向源文件

创建软连接的时候没有使用-s创建的是硬连接

硬连接就像是文件的一个别名，在linux中文件名跟文件数据是分开存放的

**more**：分页显示文件内容，下一页：空格 上一页：Shift + PageUp

**less**：分页显示文件内容

**grep**：在文本中查询内容

grep–n“xxx” Hello.java，其中-n显示行数

**|**：管道命令

在Linux系统中，| 就是管道，把上一个命令的结果交给 | 后面的命令处理，比如：

ls/etc/ | more：列出etc目录内容，分页显示

find 目录 –name “条件”

find/ -name \*.html | grep bookmark：在根目录，按照文件名\*.html查找，再在文件中查找bookmark

**find**：搜索文件（夹）

1）在特定的目录下搜索并显示指定名称的文件（夹）

find/ -name Hello.java：从根目录，按照文件名查找Hello.java

2）搜索一段时间内被存取、变更的文件（夹）

findhome/ -amin -10：10分钟内存取的文件或目录

findhome/ -atime -10：10小时内存取的文件或目录

findhome/ -cmin -10：10分钟内更改过的文件或目录

findhome/ -ctime +10：10小时前更改过的文件或目录

3）搜索指定大小的文件

findhome/ -size +10k：查找home下，大小为10k的文件

**>：**管道定向命令，比如：

grep“main”-nam Hello.java > 1.bak：结果输送到1.bak保存起来（覆盖）

grep“main”-nam Hello.java >> 1.bak：结果输送到1.bak保存起来（添加到文件末尾）

>filename 2>&1：“>&”is the syntax to redirect a streamto another file descriptor，“&”indicates that what follows is a file descripter and not a file，0 is stdin，1 is stdout，2 is stderr

**chown**：改变文件（夹）的所有者

chown 用户名 文件名/目录 如：chown Dave hello.c

chown Dave：Dave hello：改变文件（夹）的属主和属组

**chgrp**：改变文件（夹）所在组

chgrp –R 组名 文件名/目录 如：chgrpDave hello.c

**chmod**：改变文件（夹）的权限，比如：chmod -r 777 Hello.java

其中777中的每个数字分别代表拥有者，组，其他用户对文件的权限

R权限为4，w权限为2，x权限为1

**jobs**：查看所有的作业列表

Linux用户（组）管理命令（只有root用户能添加用户）

useradd用户名：添加用户

useradd[-u uid] [-g group] [-d home] [**-s shell**]user\_name：添加用户

useradd–m 用户名：自动建立用户的登入目录

useradd–d 用户名：指定用户登陆时的起始目录

passwd用户名：给用户设密码

userdel用户名：删除用户

userdel–r 用户名：删除用户和用户相关文件

userdel–r 用户名：删除用户和户主目录

groupadd组名：添加用户组，

useradd-g 组名 用户名：创建用户，并指定组，该命令还能用于改变用户所在组

usermod –G 组名 用户名 //附加组，要重新登陆才能生效

whoam i：显示用户信息

Note：

**查看linux中所有用户信息**：cat/etc/passwd | more，假如显示如下：

along：x：1001：1001：：/home/along：/bin/sh

对应格式：用户名：密码：用户id：组id：：用户目录：用户使用的shell解析器

**root用户查看用户密码**：cat /etc/shadow | more

**root用户查看群组信息**：cat /etc/group | more

**查看linux中所有组的信息：cat /etc/group | more**

**mount [-参数] [设备名称] [挂载点]**：挂载设备

**umount 挂载点**：卸载设备

**fdisk -l：**查看Linux系统分区的具体情况

**df -h 目录**：查看目录在哪个分区和磁盘使用情况

du –h 目录：显示目录下的文件大小

**env**：查看操作系统的环境

**history**：显示历史命令

history10：显示最近使用的10个命令

!5：执行历史编号为5的命令

!ls：执行最后一次以“ls”开头的命令

**ping**：ping通网络

**tracepath**：trace网络节点

**route**：显示路由表

**ifconfig**：查看ip情况

ifconfigeth0 xxx.xxx.xxx.xxx：更改设备的IP地址

ifconfigeth0 x.x.x.x netmask x.x.x.x. up：配置eth0 IP地址和子网掩码，同时激活设备

**hostname name**：更改主机名称

**netstat**：显示网络统计信息

netstat–anp：按端口显示网络信息，并显示进程号

**date**：显示系统时间

date月月日日时时分分年年年年.秒秒

**cal 月 年：**显示月历 –y(显示一年的月历)

**ps -aux**：显示正在运行的进程

ps–ef | grep zzjie：显示父进程ID（PPID）

ps–aux | grep zzjie：显示所有包含其他使用者的进程

-a：显示所有用户的所有进程

-u：按用户名和启动时间的顺序来显示进程

-x：显示无法控制终端的进程

TTY：终端 “?”是守护进程，deamon无终端

VSZ：占用虚拟内存大小

RSS：占用内存大小

ps–axf：查看进程树

ps–axm：列出线程

pspid：显示pid进程

**pstree**：显示即进程树

**top**：动态监控进程

时间系统已启动时间  用户数 当前系统负载情况 …僵尸进程

96.0id：CPU资源空闲比例；内存和Swap

监视特定用户：输入u，输入用户名，回车

终止指定进程：输入k，输入要结束的进程号

指定系统状态更新时间：top -d 10：10秒更新1次

top运行中可以通过 top 的内部命令对进程的显示方式进行控制：

l：开启或关闭第一部分第一行top信息

t：开启或关闭第一部分第二行Tasks和第三行Cpus信息

m：开启或关闭第一部分第四行Mem和第五行Swap信息

N：以PID的大小顺序排列进程

P：以CPU占用率大小排列进程

M：以内存占用率大小排列进程

q：退出top

top–p pid：只监控pid线程

**vmstat：**Linux系统监控工具，是查看虚拟内存vm使用状况的工具

在Linux内存管理中，主要是通过“调页Paging”和“交换Swaping”,Paging：将内存中不常使用的页面换到磁盘上，Swaping：将整个进程交换到磁盘上

Page-out：分页写入磁盘；Page-in：分页回到内存

vmstat2 10：

r：run queue运行队列中进程数量，超过CPU数目，出现CPU瓶颈

b：blocked等待IO的进程数量，阻塞的进程

swpd：当前使用的虚拟内存KB数，如果大于0，表示机器内存不足

free：当前空闲的物理内存KB数

buff：缓冲的内存大小

cache：缓存的内存大小

si：swapin 每秒从交换区读入虚拟内存的大小，如果大于0，表示物理内存不够用

so：swapout 每秒从虚拟内存写入交换区的大小，如果大于0，表示物理内存不够用

bi：blockin 每秒从块设备读入内存的数据总量（读磁盘）

bo：blockout 每秒从内存写入块设备的总量（写磁盘）

in：每秒中断数

cs：每秒上下文切换数

us：user 用户进程执行时间

sy：system系统进程执行时间

id：idle  cpu空闲时间

wa：wait  cpu等待IO时间

通过**/proc/cpuinfo**：查看CPU性能

grep‘model name’/proc/cpuinfo：查看CPU和CPU核数

**jstack：**

jps：查出java进程的pid

top –p pid：观察进程情况

Shift + h：显示该进程的所有线程

将线程id转化为16进程，比如线程id 3720 == 0x7d0，注意小写

jstack pid | grep -A 10 0x7d0：查询具体线程的状态，基本可以定位程序的问题所在

需要的时候再了解其他JVM监控工具

**kill 进程号**：终止进程，本质是协商退出，并不是强制退出

kill-9（KILL） 进程号：发送SIGKILL信号，强制终止进程

killall进程号：终止进程和子进程

kill–STOP pid：发送SIGSTOP信号，将后台进程暂停

kill–CONT pid：使后台暂停的进程恢复运行

kill–l：list available signals

kill信号说明：

**2）SIGINT**：**^C** sends a **SIGINT** to all the processes in theforeground process group. To do the equivalent with kill，**kill –SIGINT pid**（OS-level concept）or to the job，**kill –SIGINT %**（shell-level concept，the pipeline ended with&）

**man command**：命令手册

**info command**：在线查看命令文档

输入**/**：Gegexp search

**stty –a**：get all the terminal controlcharacter assignments

比如，intr = ^C，susp = ^Z等

**free -m：**查看内存使用情况

内存描述文件：cat /proc/meminfo：机器的内存使用情况

/proc/pid/maps：显示当前进程所占用的虚拟地址

/proc/pid/statm：进程所占用的内存

Linux空闲内存：free+ buffers + cached

**export**：设置或显示环境变量（临时）

exportPAHT=&PATH：&HOME/bin：root/test：设置PATH环境变量

如果将export写入：/etc/profile或/etc/bashrc等全局，那么针对所有用户，环境变量永久生效

如果将export写入：/home/zzjie/.bashrc或.bash\_profile，那么仅对zzjie登陆的用户环境变量有效

echo$PATH：查看PATH环境变量

/etc/profile：为系统的每个用户设置环境信息，当用户第一次登陆时，该文件被执行

/etc/bashrc：为每个运行bashshell的用户执行此文件，当bash shell被打开时，该文件被读取

/zzjie/profile：zzjie用户的环境设置

/zzjie/bashrc：zzjie用户的bash shell配置，给bash初始化，如bash代码补全，bash别名等

/zzjie/bash\_logout：当用户每次退出bash shell时，该文件被执行

**alias**：指定别名

aliasdump=’mysqldump –u root –p db > /home/date.bak’：指定数据库备份操作别名dump

aliassubl="/home/zzjie/Downloads/Sublime\ Text\ 2/Sublime\ Text\2/sublime\_text"

**shell通配符**：\* ？

**reptyr**：reptyr is a utilityfor taking an existing running program and attaching it to a new terminal

**iptraf**：网络测量工具

iptraf–g：

iptraf–d eth0：

Notes：命令最终在linux的内核执行，但先传给shell解析器，再传给Kernel

Linux文件和文件夹权限

llHello.java和test文件夹显示文件详细信息

**-rw-rw-r-- 1 zzjie zzjie 96  7月 1322:34 Hello.java**

**drwxrwxr-x 2 zzjie zzjie 4096  7月 1511:53 test/**

权限说明：r：可读，用4表示；w：可写，用2表示；x：可执行，用1表示

文件类型：

分析：-rw-rw-r--；drwxrwxr-x

第一部分-表示Hello.java是普通文件，d表示test是文件夹；第二部分rw-和rwx表示文件（夹）所有者zzjie对该文件具有的权限；第三部分rw-和rwx表示文件（夹）所在组对该文件的权限；第四部分r--和r-x表示其他组的用户对该文件（夹）的权限

Linux压缩、解压、RPM

zip（zip --help）

zipxxx.zip 文件1 文件2 …：将多个文件压缩

zip–r xxx.zip 目录 文件1 文件2：将目录下所有文件和文件夹，文件1，文件2压缩

zip–m xxxx.zip 文件：向压缩文件xxx.zip中添加文件

unzip–od /home/zzjie xxx.zip：把xxx.zip文件解压到/home/zzjie

tar（tar --help）：可调用gzip，bzip2，compress

-c：--create

-r：--apend

-t：--list

-u：--update

-x：--extrate

-f：--file

-v：--verbose

-z：--gzip，--gunzip

-j：--bzip2

-z：--compress，--uncompress

tar –cvf 打包文件名.tar 被打包的文件/路径..（多个文件空格隔开） //打包

tar –xvf 打包了的文件名.tar //解包

tar –zcvf 打包文件名.tar.gz 被打包的文件/路径..（多个文件空格隔开）//压缩

tar –zxvf 打包了的文件名.tar.gz （–C 目标路径） //解压，如果用-C，目标路径必须存在

tar –jcvf 打包文件名.tar.bz2 被打包的文件/路径..（多个文件空格隔开）//压缩

tar –jxvf 打包了的文件名.tar.bz2 （–C 目标路径） //解压，如果用-C，目标路径必须存在

压缩：

**tar –cf archive.tar 1.jpg**：将1.jpg文件压缩打包成tar文件

**tar –uf archive.tar 1.jpg**：更新原来tar包中的1.jpg文件

**tar –cvfarchive.tar \*.jpg**：将目录里所有jpg文件压缩

**tar –czf**archive**.tar.gz\*.jpg**：将目录里所有jpg文件打包成jpg.tar，再用gzip压缩，生成一个gzip压缩过的包，命名为archive.tar.gz

**tar –cjf archive.tar.bz2\*.jpg**：将目录里所有jpg文件打包成jpg.tar，再用bzip2压缩，生成一个bzip2压缩过的包，命名为jpg.tar.bz2

**tar –cZfjpg.tar.Z \*.jpg**：将目录里所有jpg文件打包成jpg.tar，再用compress压缩，生成一个compress压缩过的包，命名为jpg.tar.Z

解压：

**tar –xvffile.tar**：解压tar包

**tar -xzvf file.tar.gz（file.tgz）**：解压tar.gz或者tgz包

**tar -xjvf file.tar.bz2**：解压 tar.bz2包

**tar –xZvffile.tar.Z**：解压tar.Z包

**gunzip（gzip –d） \*.gz**：解压gz包

**bunzip（bzip –d） \*.bz2**：解压bz2包

**uncompress \*.Z**：解压Z包

源文件安装：

./configure（--prefix=path）：针对当前系统，配置好安装参数（配置软件安装目录）

make：将源代码文件编译为二进制文件

makeinstall（runas root）：将编译好的二进制文件安装到系统中

结合：& ./configure && make && sudo make install

rpm

rpm就是软件包管理工具，可以安装、删除、更新以rpm包形式发布的软件，类似Windows里面的“添加/删除程序”，通常包括Binary和Source两种rpm包

rpm常用命令：

rpm-q 软件名 //查询软件是否安装

rpm-qa | grep samba //在所有已经安装的软件包中查找samba

rpm-ql samba　//查询rpm包安装位置

rpm-qi samba  //查询rpm包的信息

**对未安装软件包的查看：**

rpm–qpi file.rpm  //查看一个软件包的用途、版本

rpm–qpl file.rpm  //查看一个软件包所包含的文件

rpm-ivh file.rpm  //安装rpm包并显示进度

rpmivh file.rpm --nodes --force：强制安装，忽略依赖关系

rpm-ivh --relocate /=/opt/gaim gaim-1.3.0-1.fc4.i386.rpm  //指定安装目录

rpm-Uvh --oldpackage gaim-1.3.0-1.fc4.i386.rpm //新版本降级为旧版本

rpm–e --nodeps file.rpm  //强制删除rpm包

Linux Shell编程

Shell基本概述：

壳（Shell）保护内部的核（Kernel），Shell是一种具备特殊功能的程序，是用户和Linux Kernel间的接口，为了屏蔽内核的复杂性，为了保护内核以免用户误操作造成损害，在内核的周围建了一个外壳，**用户向shell提出请求，shell解释并将请求转发给内核**，比如ls /home命令，首先由shell解析，最终转发给内核，有内核处理

Shell的本身就是一个解释型程序设计语言，shell程序设计语言支持绝大多数在高级语言中能见到的程序元素，如函数、变量、数组、程序控制结构等，任何在提示符中能键入的命令都能放到一个可执行的shell程序中，本质上，shell脚本：是一个文本文件，是命令的集合，有执行的权限，执行方式（./script.sh）

Shell种类有：Bourne shell（sh），Korn Shell（ksh），C shell（csh），BourneAgain Shell（bash）等，通过echo$SHELL查看当前shell

Shell的使用：

1）用户登陆后自动执行的shell脚本文件：

.bashrc位于主目录下，执行系统的脚本/etc/bashrc：配置系统基本数据

.bash\_profile位于主目录下，执行系统的脚本/etc/profile：配置系统环境变量

（比如PATH，CLASSPATH等都能在这里配置）

2）在/home/zhengzj目录下有用户的环境变量：.bashrc、.bash\_profile等

Linux任务调度：crontab

在root用户下添加crontab定时计划任务（修改/etc/crontab文件）

时间格式：\* \* \* \* \* 对应：分钟 小时 月 年 星期，比如：

0 2\* \* \*  root date >> /home/date：每天凌晨2：00，root用户，待执行命令

crontab调度多个任务：

1）vi /etc/crontab中写多个命令

2）shell编程：将多个任务组织到sh文件中，再写入/etc/crontab文件，比如：

0022 \* \* \* root /home/xx/mysql\_backup.sh：每晚22:00备份mysql数据库

crontab-r：终止任务

crontab–l：列出当前调度任务

Linux网络环境配置

**1、临时配置：**重启机器，配置消失

ifconfigeth0 x.x.x.x netmask x.x.x.x. up：配置eth0 IP地址，同时激活设备

route adddefault gw x.x.x.x：配置网关

hostnamexxxx：配置主机名

dhcclienteth0：DHCP动态获取IP

**2、修改配置文件（Google）**：

Ubuntu网络配置涉文件：

/etc/network/interfaces：网络接口配置，包括：IP地址、子网掩码、网关等

/etc/resolv.conf：DNS服务器设置

/etc/hostname：主机名设置

/etc/hosts：域名解析映射

/etc/hosts.allow：IP访问允许规则

/ect/hosts.deny：IP访问禁止规则

注意：修改网络配置文件后，要重启网络接口：/etc/init.d/networking restart

Linux vi编辑器

1）vi Hello.java

2）输入i：进入到插入模式

3）输入esc：进入命令模式

4）输入冒号：

5）wq：保存退出或者q!：退出不保存

Linux Java开发

在Linux上建立CookieService和TaskService，并且运行爬虫：

目录说明：

设置CLASSPATH：export CLASSPATH=.:/home/zzjie/service/lib/je-5.0.73.jar

或者修改vi /home/zzjie/.bashrc文件（用户级别）：

进入src目录下，编译源文件：

$javac com/crawl/common/SystemProperties.java

$javac hhf/crawl/master/InitLoadTask.java

进入src目录下，运行InitLoadTask.class文件：

$java hhf/crawl/master/InitLoadTask：

此时，任务数据库建立完毕

PS：保证文件的目录都正确，每次修改源文件都需要重新编译

————————————————————————————————————

启动Cookie服务和Task服务：

问题：要导入的Jar包太多，如何处理？

exportCLASSPATH=.：$CLASSPATH:/home/zzjie/service/lib/log4j-1.2.17.jar

进入src目录下，编译源文件：

$javac hhf/crawl/master/MasterServer.java

进入src目录下，运行程序：

$java hhf/crawl/master/MasterServer -start CookieService

$ javahhf/crawl/master/MasterServer -start UserTaskService

启动服务成功：

查看service.log：ss

————————————————————————————————————

下面运行爬虫程序：

在/home/zzjie目录下建立distributed\_crawler文件夹：

将爬虫程序打包成Jar包：

MANIFEST.MF文件如下：

一定要遵循固定的格式

运行jar文件：

$ java-Dfile.encoding=utf-8 -jar distributed\_crawler\_test.jar

（注意检查配置文件是否都已经正确）

程序正常启动：

查看日志文件：

登陆日志：

Cookie和Task分配日志：

微博下载日志：

可以看出多个线程同时爬取微博

查询mongodb：

因此，多线程爬虫已经正常工作

————————————————————————————————————

服务器后台运行：

服务器后台运行cookie和task服务：使用nohup和&，并且重定向控制台输出到文件

服务器后台运行爬虫：

Linux远程登录，让程序后台运行

nohup**：让提交的命令忽略hangup信号**

“> filename 2>&1”来更改缺省的重定向文件名，通常配合“&”使命令成为后台运行的任务

catfilename；cat filename | more；cat filename | less查看文件

举例：

**nohup java-Dfile.encoding=utf-8 -jar distributed\_crawler.jar > log/44 2>&1&**

nohupcommand &：控制台logout，进程仍然运行，起到守护进程的作用，脱离终端的控制

原理说明：

进程组（process group）：每个进程组有唯一的GID，即进程组长进程的ID

会话期（session）：有唯一一个会话期首进程（session leader），会话期ID为首进程ID，会话期可以有一个单独的控制端，**与控制终端连接的会话期首进程叫做控制进程**

当网络断开或者终端窗口关闭时，终端接口将挂断信号送给控制进程（会话期首进程），控制进程收到SIGHUP信号退出，使得会话期内其他进程退出

&**：在结尾加上“&”使命令成为作业，并放入后台运行，能通过jobs查看作业情况**

setsid**：run a program in a new session**

我们知道如果事先在命令前加上nohup或者setsid就可避免HUP信息的影响，但如果我们未加任何处理就已经提交了命令，该如何补救才能让它避免HUP信号的影响：

disown：detachthe process from its parent（the shell）

**$ disown –h jobspec**：使某个作业忽略HUP信号

**$ disown –ah**：使所有作业都忽略HUP信号

**$ disown –rh**：使正在运行的作业忽略HUP信号

**disown示例1：如果提交命令时**未使用&**将命令放入后台运行**

如果没有把命令当做作业来运行，能通过**CTRL-z**：将当前的进程挂起到后台暂停运行，然后通过**jobs**查看它的作业号，用**fg（foreground）**将挂起的进程重新放回前台，**bg（background）**将挂起的进程放在后台继续运行，这样我们就能在一个终端内灵活切换多个任务，但是如果挂起会影响进程的运行结果，得慎用此方法（使用fg和bg能够在一个终端内灵活的切换作业）

**disown示例2：如果提交作业时已经使用&命令放入后台运行**

我们已经知道如何让进程避免HUP信号的影响，但如果有大量这种命令需要稳定在后台运行，如何避免对每条命令都做这样的操作？

screen：

Screen是一个可以在多个进程之间多路复用一个物理终端的窗口管理器，screen中有会话的概念，用户可以在一个session会话中创建多个screen窗口（**^a + a**），在每一个screen窗口中就像操作一个真实的SSH连接窗口那样：

**$ screen**：登陆Linux，输入screen命令，创建screen会话，同时创建第一个screen窗口（**^a + w**）

（子进程、父进程的关系非常清楚）

screen还有更高级的功能，我们可以在不中断screen各窗口中程序的运行而暂时断开（detach）screen会话，并在随后时间重新连接（attach）,重新控制各窗口中运行的程序

比如，我们在一个screen会话中打开4个窗口（/bin/bash），输入**^a + d** ，断开会话：

看出screen会话中保存了4个screen窗口，screen会话首进程父进程为1

如果退出screen窗口，当只剩余那一个窗口时，screen会话结束，否则回到上一个窗口

在一个screen窗口中输入screen命令，相当于^a +c：创建一个新窗口

（注意到给screen发送命令使用了特殊的组合键：^a）

**$ screen –ls**：列出所有会话

**$ screen –rsession\_name**：重新连接指定的会话

**$ screen –dmSsession\_name**：建立一个处于断开模式下的会话

**$ screen –wipe**：同list，但删除那些无法连接的会话

当我们用“-r”连接到screen会话后，我们就可以在这个伪终端里执行各种操作，再也不用担心HUP信号会对我们的进程造成影响，也不用给每个命令都加上“nohup”或者“setsid”

原因：

未使用screen时新进程的进程树：

使用screen后新进程的进程树：

screen常用键绑定：

总结：

nohup/setsid/&：临时需要

disown：补救当前已经在运行了的作业

screen：大批量操作命令

Windows下Linux开发

1）samba服务器：实现Windows和Linux文件共享

2）ssh：Windows客户端远程登录Linux服务器，并且上传、下载（sftp）文件到Linux

下图是Windows用户远程登录到Linux服务器：

3）Xshell：集成ssh客户端

自己的笔记

linux下隐藏文件以.开头 如.123.txt

属性可以连着写 如 ls -lha

通配符:

\*(代表任意个数个字符)

？(代表任意一个字符，至少一个)

[](表示可以匹配字符组中的任意一个)

[abc]（表现匹配abc中的任意一个）

[a-f]匹配从a到f范围内的任意一个字符(a,b,c,d,e,f,)

ls /查看当前目录文件 -a(显示所有文件，包括隐藏) -l(显示详细信息) -h(配合-l显示文件大小)

tree /以树状图形式查看文件目录信息

pwd /查看当前目录路径

clear /情况屏幕

touch 文件名称 /创建文件

cd 目录名称 /切换目录 .(当前目录) ..(上一级目录)

mkdir 目录名称 /新建目录 -p(强制创建文件)

rm 文件名称 //删除文件，目录 删除的文件不会放到回收站 -r(强制删除)

command --help /显示帮助信息

man command /显示帮助信息，man manual缩写

cp 源文件url 目标文件url /复制文件，如果不需要改名，后面的url可以不加文件名 -r(强制复制)

mv 源文件url 目标文件url /移动文件，如果不需要改名，后面的url可以不加文件名 如果路径不变就相当于重命名

cat 文件名 /打开文件，如：cat 123.txt -n,-r(显示行号)

more 文件名 /打开文件(分屏显示)，如：cat 123.txt

grep 关键词 文件名 /在文件中查找相应关键词，高亮显示，如grep as 123.txt -n(显示行号) -v(显示不含有关键词的行) -i(忽略大小写)

echo /在终端窗口显示紧接着的文本 >(把内容输出到文件，覆盖) >>(把内容输出到文件，覆盖) 如：echo hello > 123.txt

> >> /配合其他命令使用，把将要输出到命令行的数据输出到指定文件，如:echo hello > 12.txt ,ls -hl > 123.txt

管道| /将一个命令的输出作为另一个命令的输入 如:ls -hla | more

shutdown /关机 默认六十秒后，-c(取消关机) -r(重启) -r now（马上重启）+10(十分钟后关机) 20:50(20:50关机)

ifconfig /相当于windows的ipconfig

ssh -p 端口号 用户名@远程端ip /远程登陆 //高级技巧：免密码登录，配置别名

scp /远程文件拷贝

scp -P 端口号 本地文件路径 用户名@远程端ip:目标文件路径

srp -P port 01.py user@remote:Desktop/01.py

scp -P port user@remote:Desktop/01.py 01.py

scp -P port -r demo user@remote:Desktop

scp -P port -r user@remote:Desktop demo

FileZilla /window下ftp服务软件

chmod +/- rwx 文件名·目录名 /修改访问权限 如：chmod -w 1.txt

id 用户名 //查看用户所对应的id

cat -n /etc/passwd | grep 用户名/组名 //获取对应用户/组的信息

who //查看当前所有登陆的用户列表

whoami //查看当前登陆用户的账户名

要在命令前加sudo

groupadd 组名 //添加组

groupdel 组名 //删除组

cat /etc/group //确认组信息，/etc/group 是存放组信息的目录

chgrp -R 组名 文件/目录名 //递归修改文件/目录的所属组

用户

useradd 用户名 //添加用户 -m(自动建立用户家目录) -g(用户所属组)

password 用户名 //添加密码

userdel -r 用户名 //删除用户 -r(自动删除用户家目录)

which 命令名 //查看命令的执行器在哪里

su – 用户名 //切换用户且切换相应目录 exit退出当前用户

apt linux下的一个安装包管理工具

sudo apt install 软件包 //安装

sudo apt remove 软件名 //卸载

sudo apt upgrade //更新