# Unix系统概述

## 1.Solaris

Solaris是Sun Mincrosystems研发的计算机操作系统，被认为是Unix系统的衍生版本之一。目前Solaris属于混合开源软件。

## 2.操作系统Kernel和Shell

### 2.1 Kernel

操作系统可以分为Kernel和Shell，其中Kernel是操作系统内核，它由操作系统中用于管理存储器，文件，外设和系统资源的那些部分组成，操作系统内核通常运行进程，并提供进程间的通信（IPC）的机制

### 2.2 Shell

Shell，俗称外壳，指“提供使用界面的软件”(命令解析器),类似于DOS下的command.com，接收用户命令，然后调用相应的应用程序。同时它又是一种程序设计语言，作为命令语言，它交互式地解释和执行用户输入的命令或者自动地解释和执行预先设定好的一连串命令。

作为一种程序语言，它定义了各种变量和参数，并提供了许多在高阶语言中才具有的控制结构，包括循环和分支。

Shell的分类

Bourne Shell:

1) 由斯蒂芬伯恩为AT&T UNIX编写

2）是最早和最知名的Unix shell

3）拥有大多数shell脚本的共同点

4）提示符 “#”为根用户，“$”为普通用户

5）下一个版本是bash，增强了b-sh

C shell:

1) 由文本编辑器vi的作者比尔乔伊在美国加州大学伯克利作为BSD

UNIX的一部分创建

2）提供命令行的历史和编程环境

3）类似于C编程语言

4）不兼容Bourne shell语法

5）其中tcsh是csh的增强，完全兼容

6）提示符“#”为根用户，“%”为普通用户

# UNIX文件系统

## 1 Linux文件系统

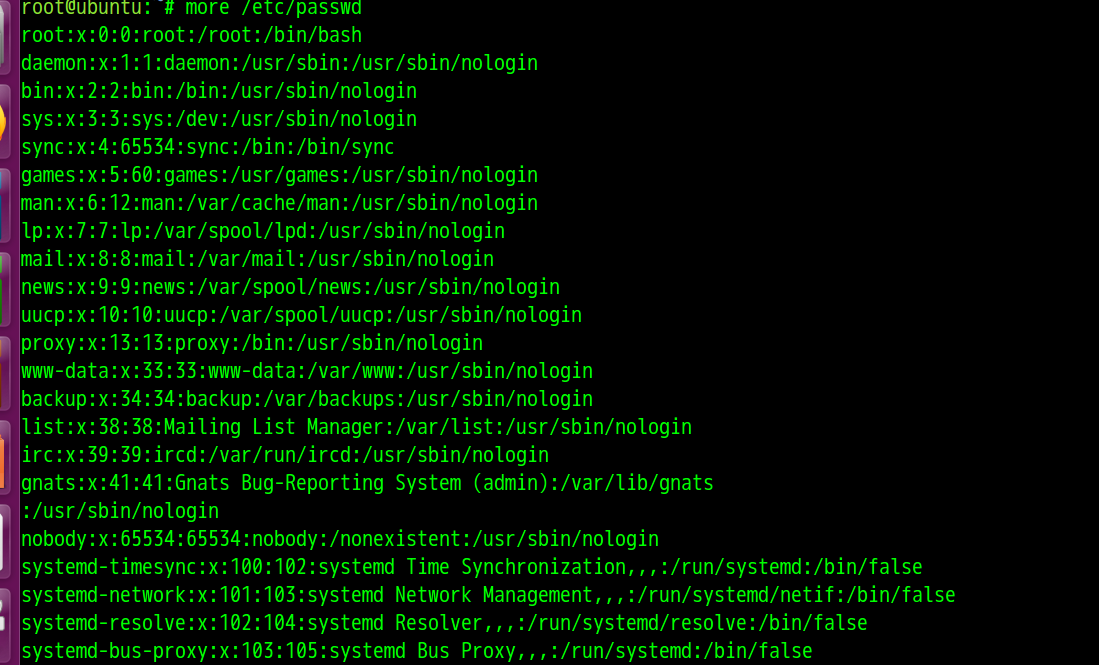
文件系统是操作系统用于在磁盘分区上的组织文件的方法和数据结构

文件系统由三部分组成：与文件管理有关软件，被管理文件以及实施文件管理所需的数据结构。

文件系统也就是对文件存储器空间进行组织和分配，负责文件存储并对存入的文件进行保护和检索的系统

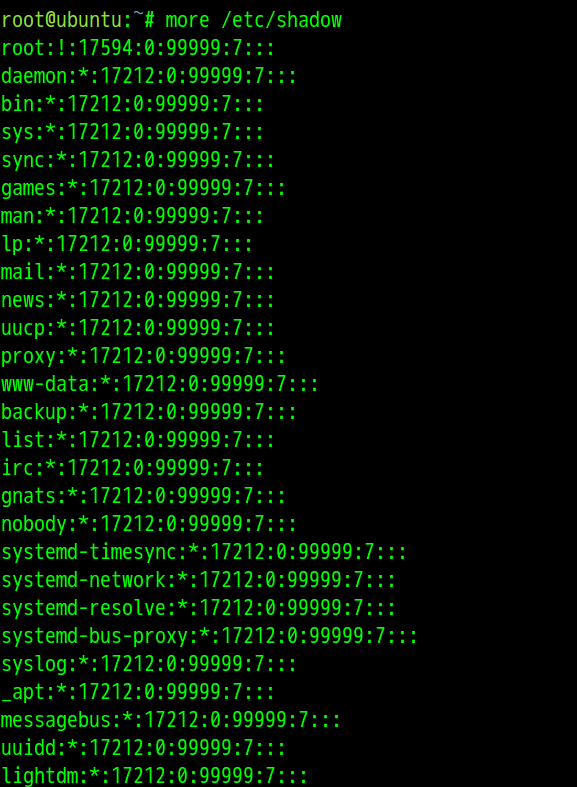
## 2 Unix用户账户文件

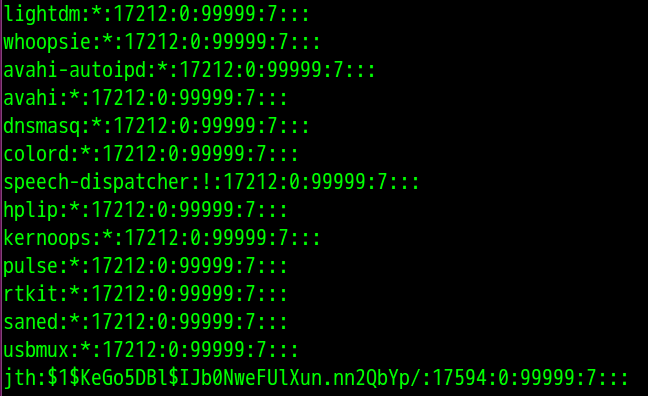
### 1 用户信息文件



### 2用户密码文件

用户密码文件保存在/etc/shadow





### 3用户组配置文件

可以查看etc/group获取



# UNIX常用命令

## 1.命令格式

Unix/Linux系统下的所有命令大小写敏感，格式如下：

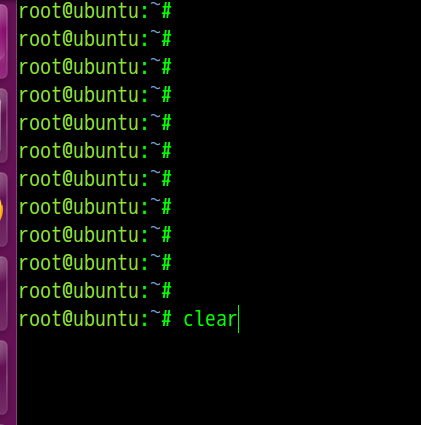
命令 [-选项] [参数]

其中，选项和参数都是可选的，中间空格隔开

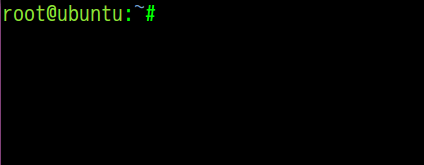
## 2.基本命令

### 2.1清屏 clear

执行前：

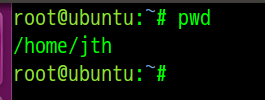


执行后：



即清除屏幕，将当前行变为第一行

### 2.2 打印当前路径 pwd



### 2.3 改变路径 cd

路径的描述方式：

相对路径： 从当前所在目录出发描述路径的方式，不以/开始的就

是相对路径

绝对路径： 从根目录出发描述路径的方式，以/开头的就是绝对路

径

特殊的路径符：

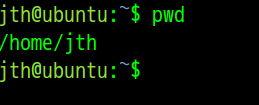
/ 根目录

.. 上一级目录

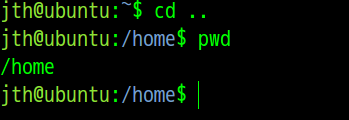
. 当前目录

~ 用户主目录

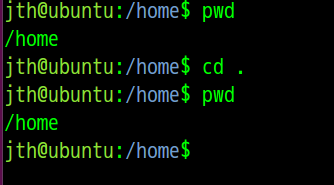
使用 pwd获取当前目录



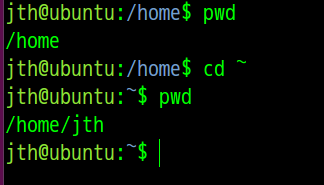
使用cd.. 返回上级目录



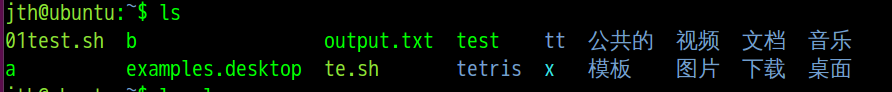
使用当前路径cd .



进入用户主目录



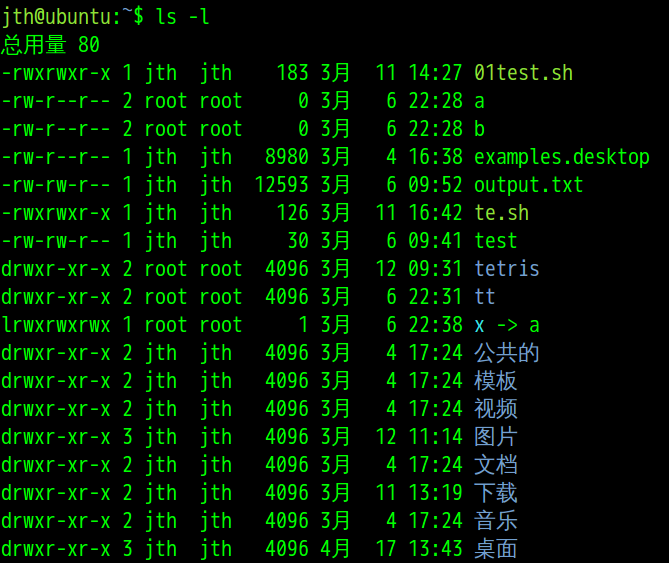
2.4 显示目录下的内容 ls



ls -a 显示全部的内容，包括隐藏文件



ls-l 详细形式显示文件



d rwx r-x r-x 2 root root 4096 3月12 09:31 tetris

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩

① 文件类型，“d”表示目录，“-”表示普通文件 ，“l”表示链接文件

② 文件拥有者对该文件的访问权限（r:读，w:写，x:执行，“-”无权限）

③ 同组者对该文件的访问权限

④ 其他人对该文件的访问权限

⑤ 文件的硬链接数

⑥ 文件的拥有者

⑦ 文件的属组

⑧ 文件大小

⑨ 文件的最后一次修改时间

⑩ 文件名

### 2.5 修改权限 chmod

1.符号模式

chmod u/g/o/a =/+/-r/w/x filename

其中

u:所有者

g:同组用户

o:其他用户

a:所有用户

+:增加权限

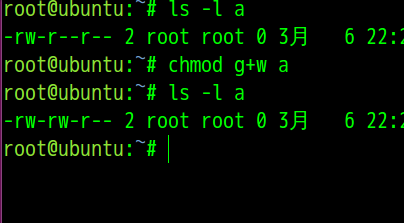
-:减去权限

=:最终权限

r:读取权限

w:写入权限

x:执行权限



### 

2.绝对模式

相对应的文件或目录的权限由三位数字表示，从左向后，第一位代表所

者的权限，第二位代表同组其他用户的权限，第三位是其他用户的权限，而

一个数字都是1,2,4三位数字组成之和，可以是0,1,2,3,4,5,6,7

1 执行权限

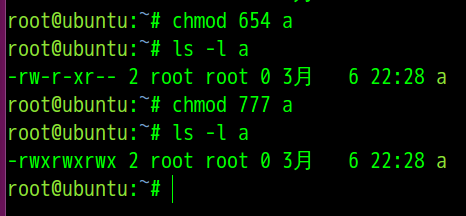
2 写入权限

4 读取权限

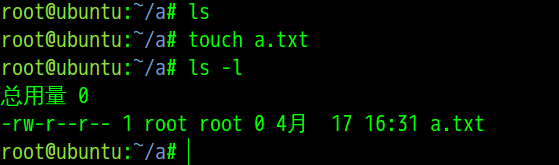
故 rwx = 7

r-x =5

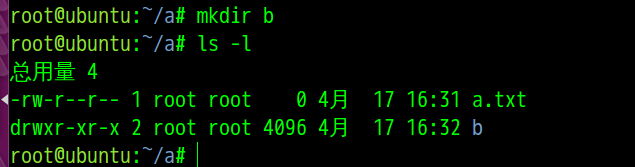
……



### 2.6 文件创建命令 touch



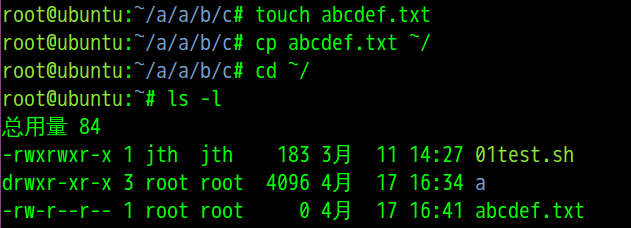
### 2.7 创建目录 mkdir



mkdir -p A/B/C/D可以创建多个层级的目录

这样就可以创建 文件夹A，A中有B,B中有C，C中有D的四个层级的文件夹

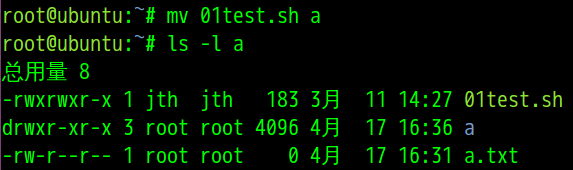
### 2.8 文件拷贝 cp



cp 文件名 要拷贝的路径

cp 文件夹名 要拷贝的路径 -r

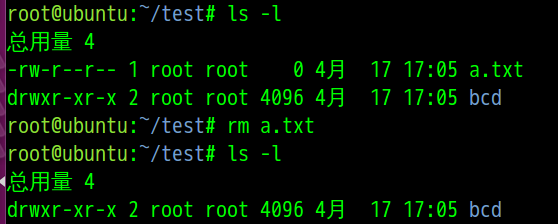
### 2.9 文件移动命令 mv



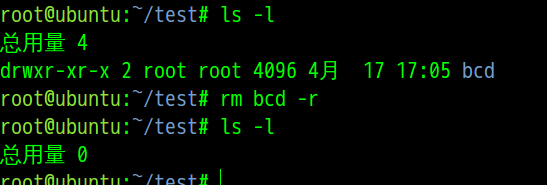
### 2.10 删除命令

rm 命令

删除文件

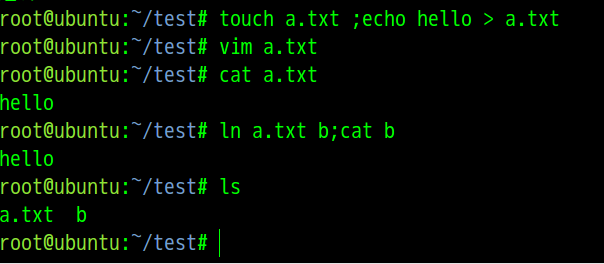


删除文件夹



### 2.11 创建连接 ln

ln用来创建一个硬链接或软连接



touch a.txt；echo hello > a.txt

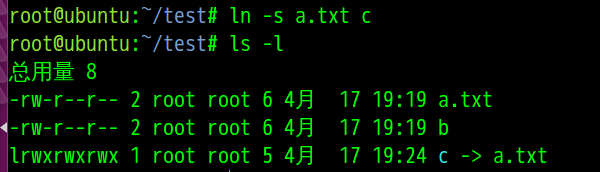
这一句将创建a.txt文件，并写入hello

ln a.txt b ;cat b

这一句创建了一个连接b指向a.txt，使用cat能查看b中的内容

可知其与a.txt中的内容相同

创建软连接



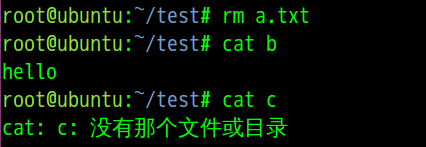
硬链接和软连接的区别：

硬链接本质上是磁盘上文件数据的另外一个访问路径，删掉本来的

a.txt系统就关闭了a.txt对磁盘数据访问的路径，不影响硬链接b，而软

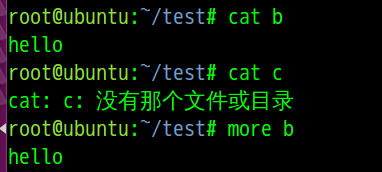
连接里面只保存了目标文件的位置，当删掉了目标文件，软连接文件也

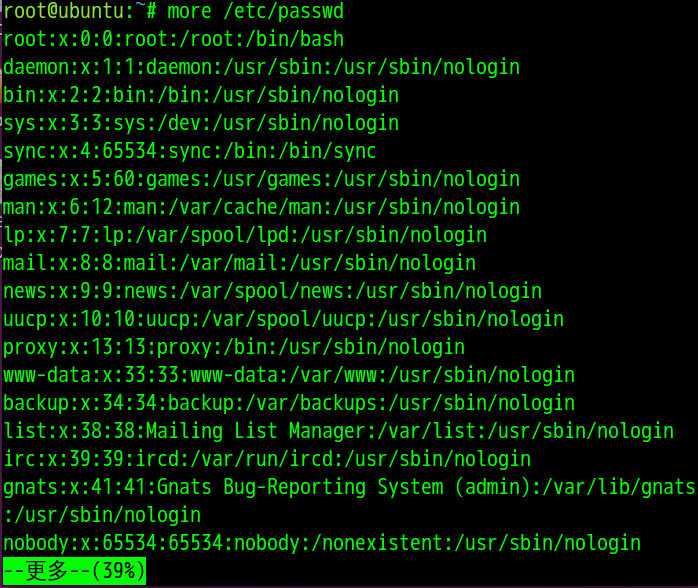
失去了访问磁盘数据的通路



### 2.12 文件查看命令 cat more head tail

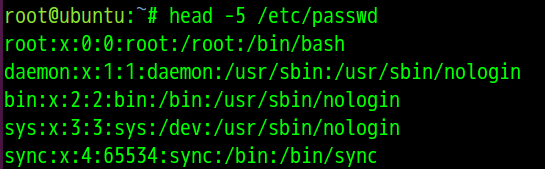
命令cat是用来查看文件内容，more也是，但more是分屏查看文件的内容

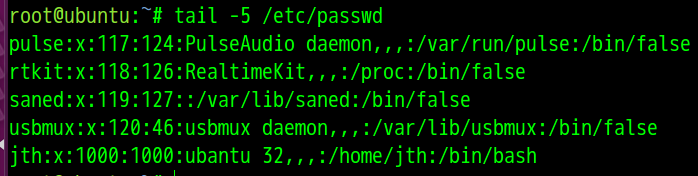




这种情况下 回车键查看下一行，空格键查看下一屏，q退出

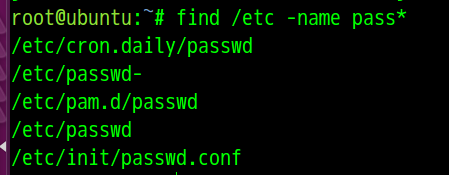
命令head和tail查看文件的开头n行和结尾n行





### 2.13 文件查找命令 find

find命令可以根据指定的条件在指定目录下查找文件



find的一些常用参数

-name 按照文件名称查找

-perm 按照文件权限查找

-prune 可以使find不在当前指定的目录中查找，如果同时使

用-depth选项，那么-prune将被find命令忽略

-user 按照文件属主查找文件

-group 按照文件所属的组来查找文件

-mtime -n +n 按照文件的更改时间查找文件，-n表示文件

更改时间距现在n天以内，+n表示文件更改

时间距现在n天以前

-nogroup 查找无有效所属组的文件，即该文件所属的组

/etc/groups中不存在

-newer file1 ! file2 查找更改时间比file1新但比文件file2旧的文件

-type 查找某一类型的文件

-b 块设备文件

-d 目录

-c 字符设备文件

-p 管道文件

-l 符号链接文件

-f 普通文件

-size n :[c] 查找文件长度为n块的文件，带有c时表示文件长度以字节计

-depth: 在查找文件时，首先查找当前目录中的文件，然后再在其子目录中查找

-fstype: 查找位于某一类型文件系统中的文件，这些文件系统类型通常可以在配置文件/etc/fstab中找到，该配置文件包含了本系统中有关文件系统的信息

-mount: 在查找文件时不跨越文件系统mount点

### 2.14 内容查找命令 grep

grep是一种文件搜索工具，它能使用正则表达式搜索文件，并把匹配的行打印出来。

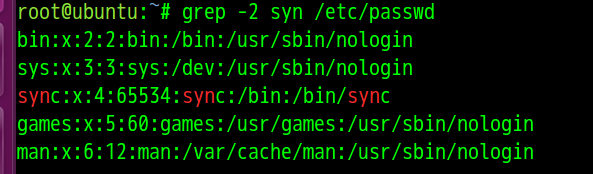
如在/etc/passwd中搜索和root相关的内容



命令的选项：

-？ 同时显示匹配行上下的?行，如 grep -2 pattern filename

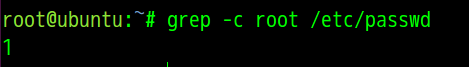
同时显示匹配行的上下2行



-b 打印匹配行前面打印该行所在的块号码



-c --count ,只打印匹配的行数，不显示匹配的内容



-f File --file=File 从文件中提取模板，空文件中包含0个模板，什么都不匹配

-i 忽略大小写

-n 在匹配的行前面打印行号

-s 不显示关于不存在或者无法读取文件的错误信息

### 2.15 查看进程 ps

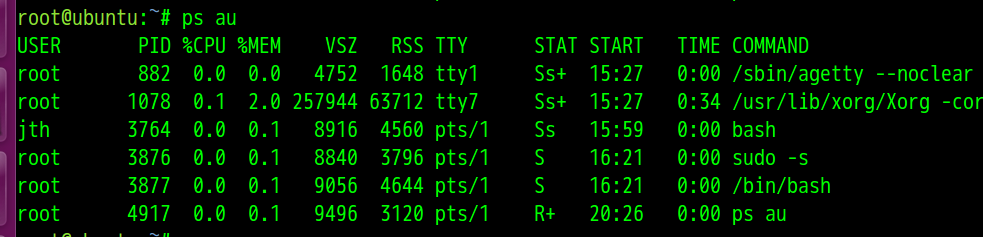
常用选项：

-A 列出所有进程

-w 显示加宽，可以显示较多的资讯

-au 显示详细的资讯

-aux 显示所有包含其他使用者的行程



USER 进程拥有者

PID pid进程号

%CPU 占用的CPU

%MEM 占用的内存使用率

VSZ 占用的虚拟内存大小

RSS 占用的内存

TTY 使用的终端设备号

STAT 进程状态（D：不可中断，R：正在执行中，S：休眠状态，

T：暂停执行，Z:不存在但暂时无法消除（僵尸进程），

W：没有足够的内存分页可分配，<:高优先序的进程，N:低优先序的进程，L：有内存分页分配并锁在记忆内

s:包含子进程，+:位于后台的进程组，I：多线程，克隆线程）

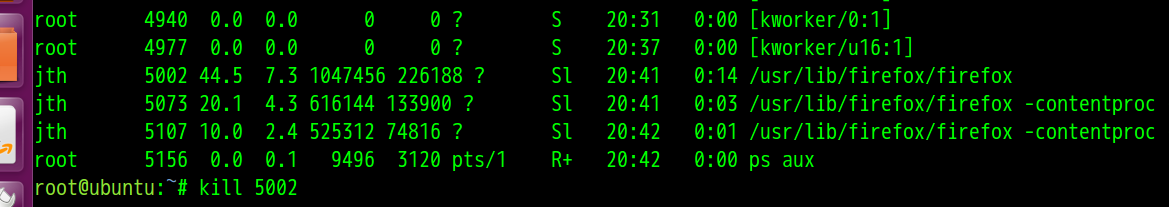
START 进程开始的时间

TIME 执行的时间

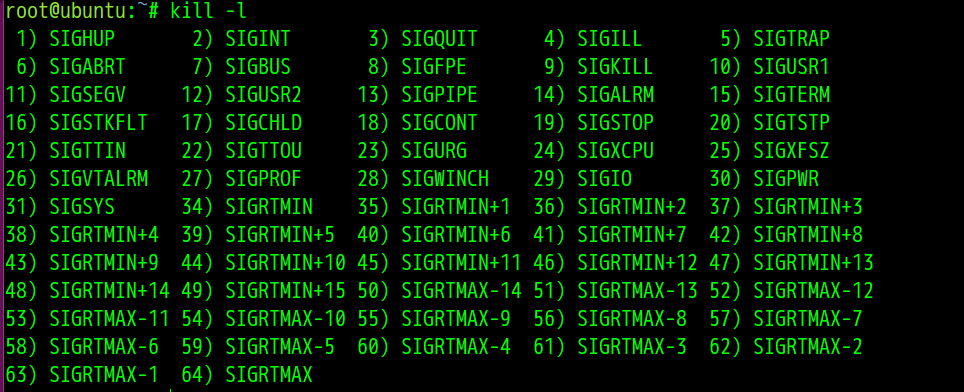
COMMAND 所执行的指令

### 2.16 杀掉指定的进程 kill

kill 可以用来终结一个进程，还可以发送信号



kill -s 还可以发送信号

主要的信号包括

其中（1-31）号信号为不可靠信号(非实时的)，后面的为可靠信号(实时信号)，区别在于前者不支持排队，可能造成信号丢失，而后者不会

1)SIGHUP

这个信号在用户终端连接(正常或非正常)结束时发出，通常是在终端的控制进程结束时，通知同一session内的各个作业，这时它们与控制终端不再关联。

2）SIGINT

程序终止信号，在用户键入INTR字符(通常是ctrl+C)时发出，用于通知前台进程终止进程

3）SIGQUIT

和SIGINT类似，但由于QUIT字符(Ctrl+\)来控制，进程因收到SIGQUIT退出时会产生core文件，类似一个程序错误信号

4）SIGILL

执行了非法指令，通常是因为可执行文件本身出现错误，或者试图执行数据段，堆栈溢出时也可能产生这个信号

5）SIGRAP

由于断点指令或其他trap指令产生，由debugger使用

6) SIGABRT

调用abort函数产生的信号

7) SIGBUS

非法地址，包括内存地址对齐错误，比如访问一个四个字长的整数，但其地址不是4的倍数，它与SIGSEGV的区别在于后者是由于合法存储地址的非法访问触发的(如访问不属于自己存储空间或只读存储空间)

8）SIGFPE

在发生致命的算术运算错误时发出，不仅包括浮点运算错误，还包括溢出以及除数为0等其他所有的算术的错误

9） SIGKILL

用来立即结束程序的运行，本信号不能被阻塞，处理和忽略，如果某个进程无法终止，可尝试发送这个信号

10）SIGUSR1

供用户使用

11）SIGSEGV

试图访问未分配给自己内存，或试图往没有写权限的内存地址写数据

12）SIGSUR2

供用户使用

13）SIGPIPE

管道破裂，这个信号通常在进程间通信产生。比如采用FIFO管道通信的两个进程，读管道没打开或者意外终止就往管道写，写进程会收到SIGPIPE信号。此外用Socket通信的两个进程，写进程在写Socket的时候，读进程已经终止

14）SIGALRM

时钟定时信号，计算的是实际的时间或时钟时间。alarm函数使用该信号

15）SIGTERM

程序结束信号，与SIGKILL不同的是该信号可以被阻塞和处理，通常用来要求程序自己正常退出，shell命令kill缺省时产生这个信号，如果进程终止不了，才发送SIGKILL

16) SIGCHLD

子进程结束时，父进程会收到这个信号

如果父进程没有处理这个信号，也没有等待子进程，子进程虽然终止，但是还是会在内核进程表中占用表项，这时的子进程称为僵尸进程

17) SIGCONT

让一个停止的进程继续执行，本信号不能被阻塞，可以用一个handler来让程序在由stopped状态变为继续执行时完成特定的工作，如重新显示提示符

18）SIGSTOP

停止进程的执行，该进程还没结束，只是暂停执行，本信号不能被阻塞，处理或忽略

19）SIGTSTP

停止进程的运行，但该信号可以被处理货忽略，用户键入SUSP字符（Ctrl+Z）发出这个信号

20）SIGTTIN

当后台作业要从用户终端读取数据时，该作业中的所有进程会收到SIGTTIN信号，缺省时这些进程会停止执行

………….

# Shell 脚本编程

## 1变量

## 2 数组

### 初始化

arr=(math English Chinese)

### 引用

${arr[0]}

### 数组个数

${#arr[\*]}

### 所有元素

${arr[\*]}

### 赋值

arr[0]=chemical

## 3 输入输出

read

read var 获取输入到var

read 获取输入到内置变量REPLY

read –a arr 读入一组词，依次赋值给数组arr

read –p “please input 5 digits” –t 10 –a arr

-p输出提示信息please input 5 digits

-t 超时，如果没有输入，到10秒就自动返回

echo

echo “Hello World” 输出一行文本

echo $num,或echo ${num} 输出一个变量值

echo –n “Hello World” 输出后不会换行

echo –e “\t”A 可以解释转义字符

4 算术扩展

‘’ 忽略所有特殊字符

“”忽略大部分特殊字符(无法忽略$, `)

`` 命令替换(将一个命令的标准输出插入到命令的任何位置)

$() 命令替换