

## 脚本

### 随机点名的脚本

```
1  #!/bin/bash
2
3  if [[ $# -ne 1 ]]; then
4      echo "Usage : bash List.sh num"
5      exit
6  fi
7
8  Cnt=$1
9
10 Names=(`cat names`)
11
12 function rand(){
13     Min=1
14     Max=${#Names[*]}
15     Num=`cat /dev/urandom | head -n 10 | cksum | cut -d ' ' -f 1`
16     echo $[ $Num % $Max + $Min ]
17 }
18
19 for i in `seq 1 $Cnt`; do
20     RandNum=`rand`
21     echo "${Names[$RandNum]} "
22 done
23
24 Ans=`rand`
```

### 线性筛脚本

```
1  #!/bin/bash
2  declare -a prime
3  End=1000
4  Sum=0
5
6  function init() {
7      S=$1
8      E=$2
9      for (( i = $S; i <= $E; i++ )); do
10         prime[$i]=0
11     done
12 }
13
14 init 0 1000
15
16 for (( i = 2; i <= ${End}; i++ )); do
17     if [[ ${prime[$i]}x == x ]]; then #如果数组元素为空，空并不是0! !
18         prime[0]=$[ ${prime[0]} + 1 ]
19         prime[${prime[0]}]=$i
20         Sum=$[ ${Sum} + $i ]
21     fi
22     for (( j = 1; j <= ${prime[0]}; j++ )); do
```

```
23     if [[ ${i} * ${prime[$j]} ] -gt ${End} ]]; then
24         break
25     fi
26     prime[${i} * ${prime[$j]} ]=1
27     if [[ ${i} % ${prime[$j]} ] -eq 0 ]; then
28         break
29     fi
30 done
31 done
32
33 echo ${Sum}
34
```

## Linux帮助系统

即man手册。

在学习使用Linux时，我们会遇到很多以前没有用过的命令和功能，这个时候最好的解决办法就是求助于man

man 的使用方法很简单，例如查看ls的帮助手册，可以直接使用命令 `man ls` 即可查看ls的命令帮助

LS(1)

User Commands

LS(1)

NAME #命令全名, 简单的说明及用法

`ls` - list directory contents

SYNOPSIS #基本语法

`ls` [OPTION]... [FILE]...

DESCRIPTION #详细说明语法中参数的用法

List information about the FILES (the current directory by default).

Sort entries alphabeti-

cally 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`

#### REPORTING BUGS #bug提交联系方式

Report **ls** bugs to [bug-coreutils@gnu.org](mailto:bug-coreutils@gnu.org)

GNU coreutils home page: [<http://www.gnu.org/software/coreutils/>](http://www.gnu.org/software/coreutils/)

General help using GNU software: [<http://www.gnu.org/gethelp/>](http://www.gnu.org/gethelp/)

Report **ls** translation bugs to [<http://translationproject.org/team/>](http://translationproject.org/team/)

#### COPYRIGHT #版权保护

Copyright © 2011 Free Software Foundation, Inc. License GPLv3+:

GNU GPL version 3 or later

[<http://gnu.org/licenses/gpl.html>.](http://gnu.org/licenses/gpl.html)

This is free software: you are free to change and redistribute it.

There is NO WARRANTY, to

the extent permitted by law.

#### SEE ALSO ##在哪里可以看到更多关于该命令的文档

The full documentation for **ls** is maintained as a Texinfo manual. If the **info** and **ls** programs

are properly installed at your site, the command

```
info coreutils 'ls invocation'
```

should give you access to the complete manual.

GNU coreutils 8.12.197-032bb

September 2011

LS(1)

man手册中开头和结尾的地方写了LS(1)了，那么这是什么意思呢？

它代表的是一般用户可使用的命令。

在man中，常见的几个数字的含义如下表所示

代号	代表的含义	举例
1	用户在shell环境下可以操作的命令或可执行文件	<code>man 1 ls</code>
2	系统内核可调用的函数和工具	<code>man 2 reboot</code>
3	一些常用的函数与函数库，大部分C的函数库	<code>man 3 readdir</code>
4	设备文件的说明，通常是在 <code>/dev</code> 下的设备	<code>man 4 null</code>
5	配置文件或某些文件的格式	<code>man 5 interfaces</code>
6	游戏	<code>man 6 lol</code>
7	惯例与协议等，例如linux文件系统，网络协议等	<code>man 7 tcp</code>
8	系统管理员可用的命令	<code>man 8 reboot</code>
9	跟kernel有关的文件	
n	旧文档	
o	新文档	
l	本地文档	

在man手册中，我们可以用到的快捷键如下

快捷键	功能
Ctrl + f(oward)	向下翻一页
Ctrl + b(ackward)	向上翻一页
/	查找
Ctrl + d(own)	向下翻半页
Ctrl + u(p)	向上翻半页
q(uit)	退出

`man` 的常用选项

1	<code>man [-fk]</code>
2	<code>-f #what is</code> 精确查找
3	<code>-k #a propos</code> 即关于这个命令的所有东西 即模糊查找

在Linux中还有一种在线求助的方式 `info`

# 通配符(bash中的)

- 1. `?`  
代表单个 任意字符
- 2. `*`  
代表任意几个 任意字符
- 3. 例子

```
suyelu@HaiZei-Tech:~$ ls
a.log  code  HaiZei  helloworld  你好world
suyelu@HaiZei-Tech:~$ ls *ld
helloworld  你好world
suyelu@HaiZei-Tech:~$ ls ??world
你好world
suyelu@HaiZei-Tech:~$ ls ?????world
helloworld
suyelu@HaiZei-Tech:~$
```

- 4. linux中还有下面几种通配符

通配符	含义	举例
[list]	匹配list中的任意单一字符	a[xyz]b a与b之间有且只有一个字符, 且只能是x或y或z, 如: axb, ayb, azb。
[!list]	匹配除list中的任意单一字符	a[!0-9]b a与b之间有且只有一个字符, 但不能是数字, 如axb, aab, a-b等
[c1-c2]	匹配c1-c2中的任意单一字符	a[0-9]b a与b之间有且只有一个字符, 该字符是0-9之间的数字, 如a0b, a1b, ... , a9b。
{string1,string2,...}	匹配 sring1 或 string2 (或更多)其一字符串	a{abc,xyz,123}b a与b之间只能是abc或xyz或123这三个字符串之一。

# 任务管理

- 1. `&`  
在命令的后面加上 `&` 表示后台执行的意思  
`command &`
- 2. `;`  
在命令之间以 `;` 相连, 表示顺序执行的意思  
`command1;command2`

### 3. &&

命令之间以 && 相连时，只有第一个命令成功执行，第二个命令才会执行

```
command1 && command2
```

### 4. ||

命令之间以 || 相连时，如果前一个命令执行失败，则执行第二个命令

```
command1 || command2
```

### 5. ``

命令中如果包含另一个命令，则用符号 `` 将他包括起来，在执行的时候系统将优先执行 `` 中的子命令，然后将其结果代入父命令继续执行

### 6. Ctrl + z

在shell中执行命令时，同时按下 Ctrl + z 可以将其暂时挂起，那么该进程所占用的内存就会被置换出来，置换到swap交换区中

### 7. jobs

查看所有在后台执行和挂起的任务以及任务的编号

```
suyelu@HaiZei-Tech:~/code$ jobs
[1]-  Stopped                  vim a.c  (wd: ~)
[2]+  Stopped                  vim b.c  (wd: ~)
[3]   Running                  ./a.out &
```

### 8. bg

使用bg，可以将挂起的命令后台运行

```
suyelu@HaiZei-Tech:~$ vim helloworld.c

[1]+  Stopped                  vim helloworld.c
suyelu@HaiZei-Tech:~$ bg
[1]+  vim helloworld.c &
suyelu@HaiZei-Tech:~$
```

### 9. fg

可以将后台执行的命令转为前台执行

bg和fg再加上任务编号，表示让这个任务后台/前台执行

## 管道与重定向

### 1. |

管道命令，例如 p1 | p2 将p1的标准输出作为p2的标准输入

### 2. >

将标准输出重定向到文件

3. >>

将标准输出追加到文件

4. <

从文件中获取标准输入

5. <<

```
cat >> /etc/sysconfig/network << EOF
HOSTNAME=$HOST_NAME
EOF
```

<< xxx 即将xxx作为EOF(结束符)

## Linux系统信息

命令	功能
uptime	打印系统运行时长和平均负载
w	当前登录用户列表及正在执行的任务
who	显示当前登录系统的用户信息
whoami	打印当前有效的用户名称
last	显示用户最近登录信息
uname	打印当前系统信息
date	显示或设置系统时间与日期
cal	显示日历

1. 为什么要有w和who这些命令？

因为linux是个多用户系统！在绝大多数情况下，一个系统是存在很多用户的！

## uptime

- 1 uptime [-phsv]
- 2 打印系统运行时长 和 平均负载(最近1min,5min,15min)
- 3 -s 开机的时间
- 4 -p 开机后经过多少时间



```

root@HaiZei-Tech:~# uptime
12:19:00 up 120 days, 18:17, 2 users, load average: 0.00, 0.00, 0.00
root@HaiZei-Tech:~# uptime -p
up 17 weeks, 1 day, 18 hours, 17 minutes
root@HaiZei-Tech:~# uptime -s
2017-07-28 18:01:48

```

## W

```

1 w [-husfV] <user> <...>
2   #获取当前登录用户和正在执行的进程
3   #PS:包含了uptime的信息!!!
4   -h 不打印头信息
5   -s 使用短输出格式
6   -f 切换显示FROM项, 默认显示

```

```

suyelu@HaiZei-Tech:~$ w
11:57:25 up 120 days, 17:55, 2 users, load average: 0.00, 0.00, 0.00
USER      TTY      FROM          LOGIN@      IDLE   JCPU   PCPU WHAT
suyelu    pts/0    1.189.203.207 11:42      14:37   0.04s  0.00s pager
suyelu    pts/1    1.189.203.207 11:57       2:00s  0.03s  0.00s w

```

USER	TTY	FROM	LOGIN@	IDLE	JCPU	PCPU	WHAT
登录用户名	登录后系统分配的终端号	远程主机名	登录时间	空闲时间	①	当前进程所占用时间	当前进程正在执行的命令

①和该终端相连的所有进程占用的时间, 包含当前正在执行的后台作业

## who

```

1 who [-Huq] <file | arg1 arg2>
2   #显示当前登录系统的用户信息
3   -H 显示各栏的标题信息列
4   -u 显示空置时间
5   -q 只显示登入系统的用户名称和总人数

```

```

suyelu@HaiZei-Tech:~$ who -H
NAME      LINE      TIME          COMMENT
suyelu    pts/0      2017-11-26 15:47 (1.189.203.207)
suyelu    pts/2      2017-11-26 15:57 (1.189.203.207)
suyelu@HaiZei-Tech:~$ who -q
suyelu suyelu
# users=2

```

## whoami

```
1 | #可以这么看，whoami是不严肃地问，who am i是严肃地问
```

```
192:~ chenbin$ whoami
chenbin
192:~ chenbin$ who am i
chenbin ttys000 Mar 14 23:44
```

```
root@HaiZei-Tech:~# id -un
root
root@HaiZei-Tech:~# id root
uid=0(root) gid=0(root) groups=0(root)
```

## last

```
1 | last [-fnx] <username...> <tty...>
2 |   #显示用户最近登录信息
3 |   -f "file" 指定记录文件
4 |   -n "num" 指定输出记录数
5 |   -x 显示系统关机，重新关机等信息
```

```
root@pil:~# last -x -n 10
runlevel (to lvl 5)  4.9.41-v7+      Sun Nov 26 08:18   still running
root pts/0          192.168.1.10      Sun Nov 26 08:18   still logged in
reboot system boot  4.9.41-v7+      Thu Jan  1 00:00   still running
shutdown system down 4.9.41-v7+      Sun Nov 26 08:18 - 00:00 (-17496+-8:-18)
root pts/1          192.168.1.10      Sun Nov 26 08:15 - down (00:02)
root pts/2          192.168.1.10      Sun Nov 26 07:30 - 07:47 (00:17)
pi pts/1           192.168.1.10      Sun Nov 26 07:25 - 07:37 (00:11)
pi pts/0           192.168.1.10      Sun Nov 26 03:52 - 08:18 (04:25)
pi pts/0           192.168.1.10      Sat Nov 25 12:34 - 12:35 (00:00)
pi pts/0           192.168.1.10      Sat Nov 25 11:41 - 12:24 (00:43)
```

## uname

```
1  uname [-amnvpio]
2      #打印当前系统信息
3      -a 全部信息
4      -m 电脑类型
5      -n 网络上主机的名称
6      -v os版本
7      -p 处理器类型
8      -i 硬件平台
9      -o 操作系统名称
```

```
192:~ chenbin$ uname
Darwin
192:~ chenbin$ uname -a
Darwin 192.168.0.102 18.7.0 Darwin Kernel Version 18.7.0: Tue Aug 20 16:57:14 PDT 2019; root:xnu-4903.271.2~2/RELEASE_X86_64 x86_64
192:~ chenbin$ uname -p
i386
192:~ chenbin$ uname -n
192.168.0.102
```

## date

---

```
1  date [-dsu] <参数>
2      #显示或设置系统时间与日期
3      -d "string" 显示字符串所指的日期
4      -s "string" 设置时间
5      -u 显示GMT(如果是CST的情况下)
6      参数
7      <+日期格式> 显示使用的日期格式
```

```
192:~ chenbin$ date
2020年 3月15日 星期日 11时49分39秒 CST
192:~ chenbin$ date -u
2020年 3月15日 星期日 03时49分41秒 UTC
192:~ chenbin$ date +%Y
2020
192:~ chenbin$ date +"%H:%M:%S"
11:50:37
```

## cal

---

```
1 cal [option] <参数>
2   #显示日历
3 选项
4   -1,-3 显示最近一个月和三个月的日历
5   -j 显示一年的第几天
6   -y 显示当前年
7 参数
8   月：指定月份
9   年：指定年份
```

```
[suyelu@HaiZei-Tech:~$ cal -j
```

November 2017

Su	Mo	Tu	We	Th	Fr	Sa
			305	306	307	308
309	310	311	312	313	314	315
316	317	318	319	320	321	322
323	324	325	326	327	328	329
330	331	332	333	334		

```
[suyelu@HaiZei-Tech:~$ cal -1
```

November 2017

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	30		

Linux目录与文件

## 目录

---

命令	功能
cd	切换当前工作目录
pwd	打印当前工作目录
mkdir	创建目录
rmdir	删除目录

绝对路径：从/开始的路径

相对路径：从.或..开始的路径

## cd

```
1 cd #回到家目录
2 cd ../#切换到上层目录
3 cd - #回到上次工作目录
4
5 #为什么可以回到上次工作目录呢？
6 #因为环境变量中有个OLDPWD，cd - 其实就相当于cd ${OLDPWD}
7 #但如果你刚打开一个shell，就使用cd -
8 #会报错：OLDPWD not set
```

## pwd

```
1 pwd [-LP]
2 -L 显示逻辑工作目录
3 -P 显示物理工作目录
```

1. L和P的区别主要还是由于link引起的

2. 例子

比如/var/run是个链接文件

```
[suyelu@HaiZei-Tech:~$ cd /var
[suyelu@HaiZei-Tech:/var$ ls -al run
lrwxrwxrwx 1 root root 4 Feb 20 2017 run -> /run
[suyelu@HaiZei-Tech:/var$ cd run
[suyelu@HaiZei-Tech:/var/run$ pwd -L
/var/run
[suyelu@HaiZei-Tech:/var/run$ pwd -P
/run
suyelu@HaiZei-Tech:/var/run$
```

## mkdir

- 1 `mkdir [-pm] <dir>`
- 2 `-p` 自动创建父目录
- 3 `-m` 设置权限

1. 比如说你想创建 `test/test/test/test`，但是最后一个目录的上级目录就不存在，那么这个时候就可以-p了。
2. 如果你在刚刚那个命令中加上 `-m 777`，也只是给予最后创建的那个目录777的权限，它的父目录没有这个权限。

3. 

```
192:~ chenbin$ mkdir test/test/test/test
mkdir: test/test/test: No such file or directory
192:~ chenbin$ mkdir -p -m 777 test/test/test/test
192:~ chenbin$ ls test
test
192:~ chenbin$ ls -al test/
total 0
drwxr-xr-x  3 chenbin  staff   96  3 14 23:52 .
drwxr-xr-x@ 60 chenbin  staff 1920  3 14 23:52 ..
drwxr-xr-x  3 chenbin  staff   96  3 14 23:52 test
192:~ chenbin$ ls -ald test/test/test/test/
drwxrwxrwx  2 chenbin  staff   64  3 14 23:52 test/test/test/test/
```

## rmdir

- 1 `rmdir [-p] <dir>`
- 2 `-p` 删除祖先
- 3 #相对于rm -rf来说比较安全，但很少用，因为rm -rf更加简单粗暴

## 思考

1. 我们自己写的 程序在编译后，使用命令 `./a.out` 执行，这个命令是什么意思？  
就是把当前目录下的a.out给这个shell让他去执行，shell拿到他之后发现是个可执行程序，然后shell就新开一个子进程去执行他
2. `PATH="$PATH":/home/chenzheyu/bin` 的含义是什么？  
其实就是字符串拼接，把后面那个路径加到PATH路径中去

## 文件与目录的管理



命令	功能
ls	显示文件及目录信息
cp	拷贝
rm	删除
mv	移动
basename	取文件名
dirname	取目录名

## ls

```
1 ls [-aldh]
2 -a 全拼是--all
3 -l 即use a long listing format
4 -h 全拼是--human-readable, 即人类可读的模式
5 -d 全拼是--directory, 即list directories themselves, not their contents
```

## cp

```
1 cp [-irapdsu] <sour> <des>
2 -i 若文件存在, 询问用户是否要替代
3 -r 递归复制
4 -a -pdr的集合 #较为常见的写法
5 -p 连同文件属性一起拷贝
6 -d 若源文件为连接文件的属性, 则复制连接文件的属性
7 -s 拷贝为软连接
8 -l 拷贝为硬连接
9 -u 源文件比目的文件新才拷贝 #常用于日志系统
10
```

## rm

```
1 rm [-irf] <dir_or_file>
2 -i 互动模式
3 -r 递归删除
4 -f force
5 #f 可能是force、file、format
6 #rm -rf一定要慎用!!!!
```

## mv

```
1 mv [-ifu] <source> <dest>
2 mv source1 source2 source3 dir
3 -i 互动模式
4 -f force
5 -u 源文件更新才会移动
```

## 随堂练习

自己动手实验一下dirname和basename的用法

### basename

1. 不加选项时，basename就是根据目录分隔符求出最后一个目录分隔符后面的字符串，并不关注这个文件是否存在！！

```
192:~ chenbin$ basename a.out
a.out
192:~ chenbin$ basename /a.out
a.out
192:~ chenbin$ basename ~/a.out
a.out
```

### dirname

1. 不加选项时，dirname就是提取最后一个字符串前面的目录名，并不关注这个目录是否存在！！

```
192:~ chenbin$ dirname a.out
.
192:~ chenbin$ dirname /a.out
/
192:~ chenbin$ dirname ~/a.out
/Users/chenbin
```

## 注意点

1. 这俩命令只是做的字符串操作，并不检查目录或文件是否存在！！



2. 用处：用在脚本中！！

## 文件内容的查阅

命令	功能
cat	正向连续读
tac	反向连续读
nl	输出行号显示文件
more	一页一页的显示文件内容
less	与more相似，但是可以上下翻看
head	只看头几行
tail	只看末尾几行
od	以二进制方式查看

### cat

```
1 cat [-AbEnTv] <file>
2 -v 列出不可见字符
3 -E 显示断行符为$
4 -T 显示tab为^T
5 -A 相当于-vET
6 -b 列出行号
7 -n 列出行号，连同空行也编号
```

### tac

刚好与cat相反，从最后一行开始打印。

linux中，命令对文件的处理都是按行来的！

### nl

```
1 nl [-bnw] <file>
2   -b 行号指定的方式
3   -b a 相当于 cat -a
4   -b t 相当于 cat -b
5   -n 列出行号的表示方法
6   -n ln 行号在屏幕最左边显示
7   -n rn 行号在自己字段最右边显示
8   -n rz 行号在自己字段的最右边显示，前面自动补全0
9   -w <num> 行号所占位数
```

## more

```
1 more <file>
2   /string 向下查找string关键字
3   :f 显示文件名称和当前显示的行数
4   q 离开
```

## less

```
1 less <file>
2   /string 向下查找 n:继续向下查找
3   /?string 方向查找 N:继续反向查找
```

## head

```
1 head [-n num] <file>
2   -n num 显示前num行
3   -n -num 除了后num行，其他都显示
```

## tail

```
1 tail [-n num] <file>
2   -n num 显示文件后num行
3   -n +num 除了前num行，其他都显示
```

## 随堂练习

如何查看一个文件的第101行到120行？

```
man ls | nl -b a -n rz -w 3 | head -n 120 | tail -n 20
```