目录

[记录 2](#_Toc2262176)

[推荐书籍 2](#_Toc2262177)

[乌班图系统开启 2](#_Toc2262178)

[环境变量 2](#_Toc2262179)

[进程 4](#_Toc2262180)

[常用命令 4](#_Toc2262181)

[/ 查找 4](#_Toc2262182)

[常用 4](#_Toc2262183)

[文件系统跳转 5](#_Toc2262184)

[文件目录操作 5](#_Toc2262185)

[cat 6](#_Toc2262186)

[less 7](#_Toc2262187)

[归档解压缩 7](#_Toc2262188)

[zip格式 7](#_Toc2262189)

[tar 7](#_Toc2262190)

[其他格式 8](#_Toc2262191)

[重定向 8](#_Toc2262192)

[>重定向标准输出 8](#_Toc2262193)

[重定向标准错误输出 9](#_Toc2262194)

[重定向标准输入< 9](#_Toc2262195)

[管道线 9](#_Toc2262196)

[权限 9](#_Toc2262197)

[chown 10](#_Toc2262198)

[添加用户 10](#_Toc2262199)

[process 11](#_Toc2262200)

[后台管理 11](#_Toc2262201)

[查找 12](#_Toc2262202)

[网络操作 13](#_Toc2262203)

[ssh 13](#_Toc2262204)

[rsync 14](#_Toc2262205)

[上传 14](#_Toc2262206)

[下载 14](#_Toc2262207)

[安装软件 14](#_Toc2262208)

[手动安装 14](#_Toc2262209)

[自己编译源码 15](#_Toc2262210)

[包管理器 15](#_Toc2262211)

[deb包（apt-get） 15](#_Toc2262212)

[rpm（yum） 16](#_Toc2262213)

[脚本编程 18](#_Toc2262214)

[系统相关 19](#_Toc2262215)

[整理 19](#_Toc2262216)

[文件 19](#_Toc2262217)

[less 19](#_Toc2262218)

# 记录

## 推荐书籍

推荐书籍 linux命令行

## 乌班图系统开启

com键打开快速启动栏。

搜索terminal

## 环境变量

终端配置

export PATH=${PATH}:~/Library/Android/sdk/platform-tools:~/Library/Android/sdk/tools

仅仅保证对此次会话有效

配置文件方式

Mac系统的环境变量，加载顺序为：

/etc/profile /etc/paths ~/.bash\_profile ~/.bash\_login ~/.profile ~/.bashrc

当然/etc/profile和/etc/paths是系统级别的，系统启动就会加载，后面几个是当前用户级的环境变量。后面3个按照从前往后的顺序读取，如果~/.bash\_profile文件存在，则后面的几个文件就会被忽略不读了，如果~/.bash\_profile文件不存在，才会以此类推读取后面的文件。~/.bashrc没有上述规则，它是bash shell打开的时候载入的。

1）/etc/paths （全局建议修改这个文件 ）

编辑 paths，将环境变量添加到 paths文件中 ，一行一个路径

Hint：输入环境变量时，不用一个一个地输入，只要拖动文件夹到 Terminal 里就可以了。

2）/etc/profile （建议不修改这个文件 ）

全局（公有）配置，不管是哪个用户，登录时都会读取该文件。

3）/etc/bashrc （一般在这个文件中添加系统级环境变量）

全局（公有）配置，bash shell执行时，不管是何种方式，都会读取此文件。

规则

export PATH=$PATH:<PATH 1>:<PATH 2>:<PATH 3>:------:<PATH N>

${}为占位符，${PATH} 和 $PATH等效

=前后不能有空格

: 为分隔符（等同于windows环境变量中的 ; ）

PATH=$PATH:xxx 可以理解为 " PATH+=xxx "（当然不能这么写）

PATH=xxx:$PATH这种写法亦可，只是前后顺序的问题；

以上命令可以拆为两行：

把adb所在路径~/Library/Android/sdk/platform-tools:~/Library/Android/sdk/tools加入到PATH中去；

生效(重载配置文件，或者重启终端)

source ~/.bash\_profile 或者 . ~/.bash\_profile

## 进程

mac查找杀死进程

lsof -i :端口

kill -9 进程ID

# 常用命令

ls

ctrl+d 退出其他环境

ctrl+l 清屏

mkdir

rm

rm –r 删除文件夹

man rm

## / 查找

比如/-r

n 查找下一处

q 退出手册

### 常用

fdisk –f

查看系统盘

df –h

查看硬盘使用情况

zsh

设置快捷命令

## 文件系统跳转

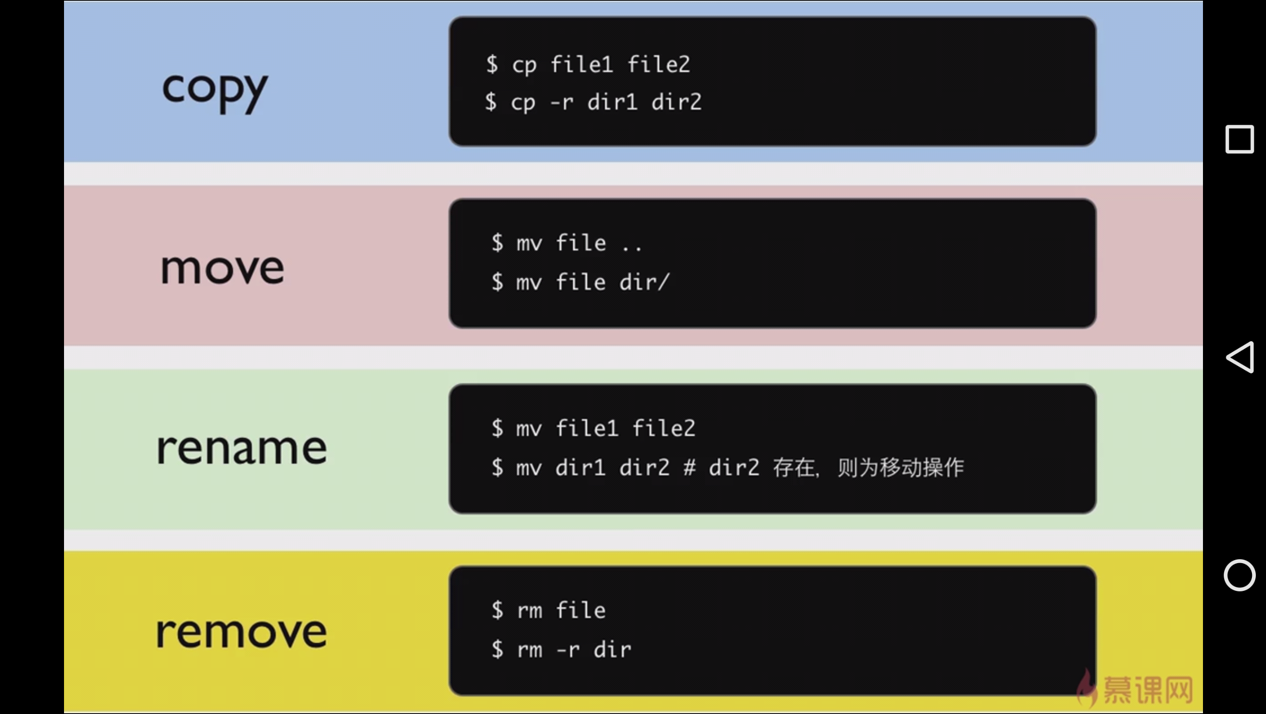
分区 sda1 sda2 但是只有一棵树

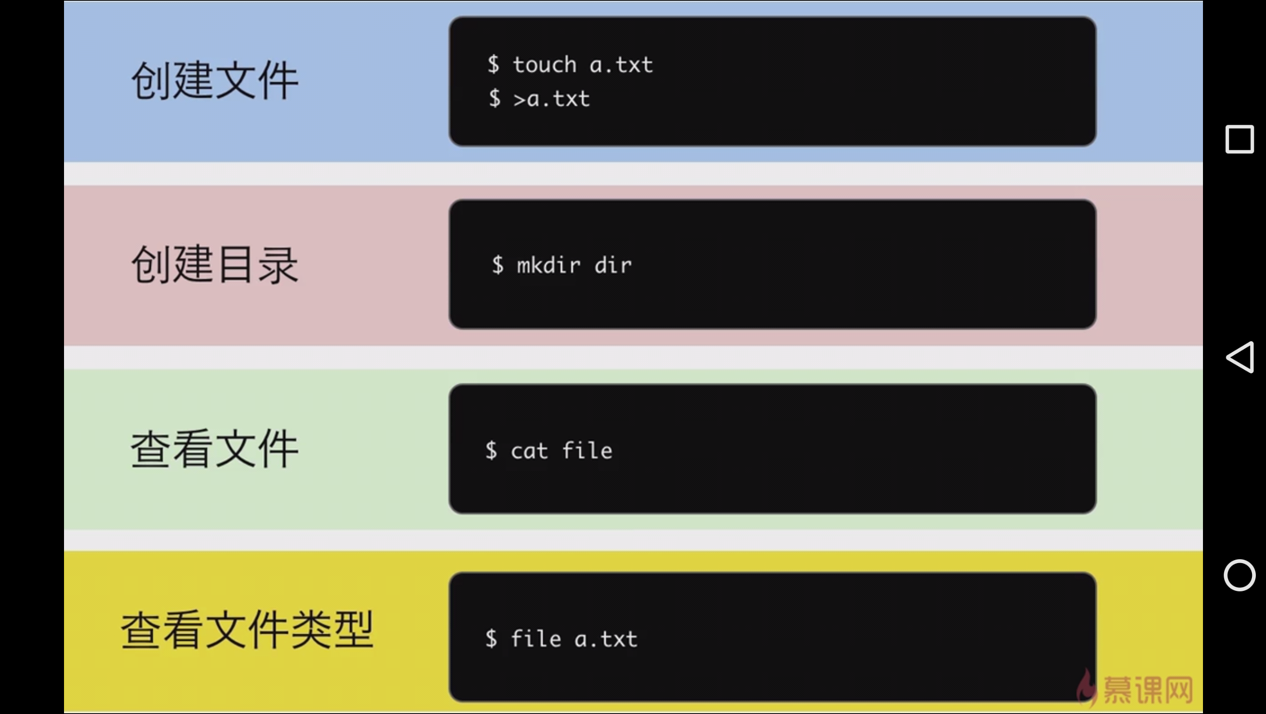
pwd 用户挂在的目录（进入cmd后默认是用户所在目录）

..表示上层目录(mac可以用)

tab补齐 两次tab查看所有选项

# 文件目录操作





rename 和 mv都可以移动目录

-a 查看隐藏文件，目录

. 表示当前目录，移动复制时方便

-rf 递归 强制

\* 通配符

wget 下载(mac下没有)

## cat

不分页

1.一次显示整个文件。$ cat filename  
2.从键盘创建一个文件。$ cat > filename    
   只能创建新文件,不能编辑已有文件.  
3.将几个文件合并为一个文件： $cat file1 file2 > file

参数：  
-n 或 --number 由 1 开始对所有输出的行数编号  
-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号  
-s 或 --squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行  
-v 或 --show-nonprinting

## less

分页 j向上滚屏 k向下滚屏

gg 文件头

G 文件尾

# 归档解压缩

## zip格式

unzip

zip –r

压缩文件夹

zip -r myfile.zip ./\*

将当前目录下的所有文件和文件夹全部压缩成myfile.zip文件,－r表示递归压缩子目录下所有文件

unzip 默认解压到当前目录

## tar

.tat.gz

tar -zxvf解压缩

tar -zcvf压缩

-c: 建立压缩档案

-x：解压

-z：有gzip属性的 .tar.gz gzip压缩

-j .tar.bz2 bzip2压缩

-Z .tar.Z compress压缩

代表三种不同的压缩方式

-v：显示所有过程

-f: 使用档案名字，切记，这个参数是最后一个参数，后面只能接档案名。

tar -zcvf 压缩文件名 文件或目录

tar -zcvf one.tar.gz n/

tar -zxvf 压缩文件名 （解压成原来的文件或文件夹，文件夹名字是压缩前的名字）

## 其他格式

.tar.bz2格式

tar jxvf

tar jcvf

## 重定向

存在的文件

>

0 stdin 1 stdout 2 sterr文件标示

### >重定向标准输出

每次会清空

如果file2有内容

cat file1 > file2

file2原本的内容会清空

解决 cat file1>>file2

或者 cat file1 file2 > file2

### 重定向标准错误输出

2>

### 重定向标准输入<

# 管道线

（数据传输，前一个命令的输出是下一个命令的输入）

| 管道符

uniq grep sort

# 权限

用户在用户主目录下有写权限

sudo 使用超级用户

安装软件

ctrl+a 到行首

读 写 执行

owner

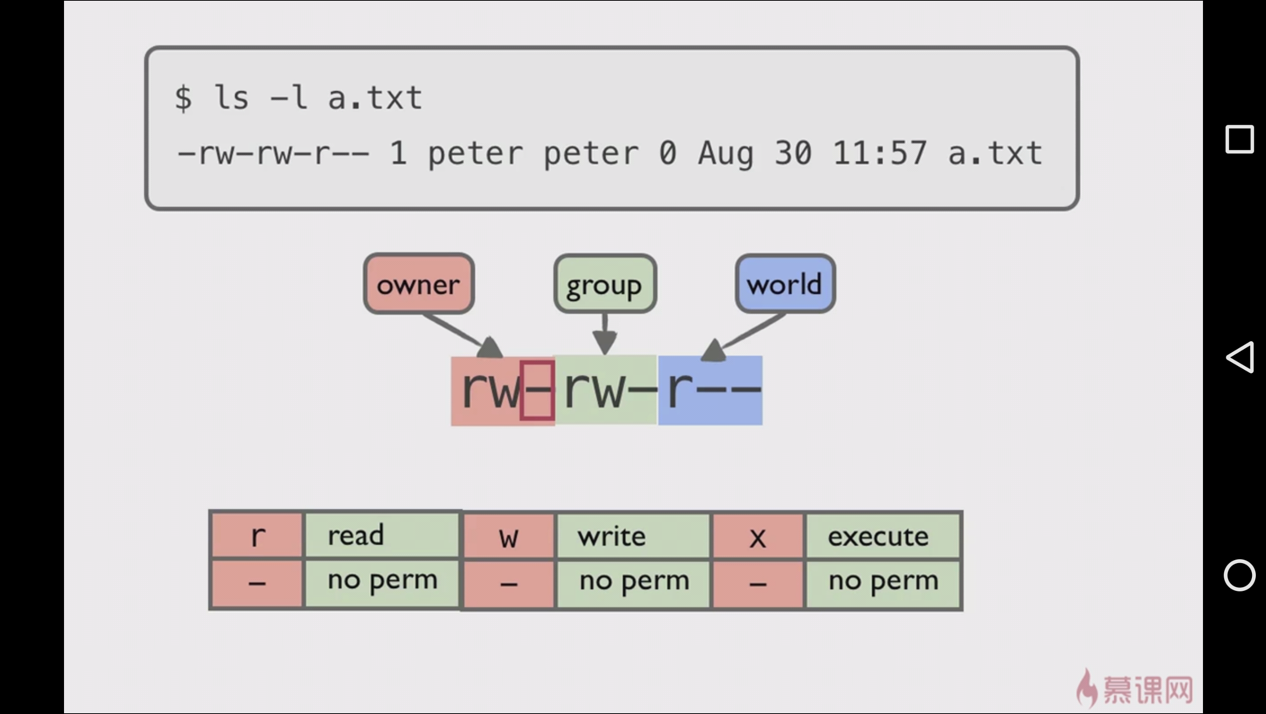
group

world

ls –l

查看带权限的列表

### chown



第一个字母表示文件类型

chmod +w 对应文件文件夹 赋予写权限

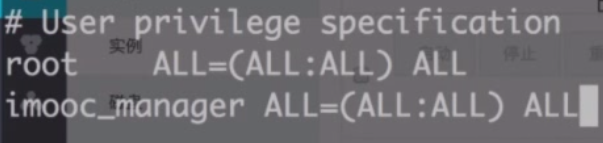
可以同时赋予三个权限 对应8进制数

## 添加用户

adduser aaa

授权 gpasswd –a aaa sudo 添加到sudo组

sudo visudo



all

1 这条规则对所有sudo生效

2 aaa可以以任何身份执行命令

3 aaa可以以任何组执行命令

4 规则适用于所有命令

# process

获取进程号

pid

ps

ctrl+shift+d 新打开一个窗口

ps aux|less 查看所有进程并分页

ps aux|grep vim

复制粘贴

双击选中

ctri+shift+c

ctrl+shift+d

alt+tab 切换命令

kill 对应pid 杀死进程

ctrl+上箭头 上一次命令

打开程序 ：直接输入进程名字 如firefox 会进入对应的shell

## 后台管理

后台打开 firefox &

暂停对应程序的shell ctrl+z bg 进入后台 fg回到前台 ctrl+c 结束

kill

kill -15 pid 杀死

kill -9 pid 强制杀死

工作台

linux同时运行七个工作台

ctri+alt+f1 进入第一个工作台

控制台某个进程失去响应无法操作时，可以进入其他工作台杀死进程，再回到对应工作台（一半f7）

# 查找

locate 从数据库中查找，从数据库中，update一般一天运行一次，所以locate找不到新创建的文件

updatedb

全局查找

find

目录范围内查找

find .

find .|grep .text 会标示text后缀

find . –type f 只看文件

find . –type d 只看目录

exec 操作查找到的文件

find . –type –f –exec ls –l ‘{}’’;’

‘{}’代表已经查到的文件名

‘；’结束

echo 放入数据

-n 打印出行号

-i 忽略大小写

-print

ack 类似grep

grep

字符串查找

e 打开选择项

# 网络操作

## ssh

远程操作协议 22端口

需要openssh-server(云服务器一般装了) openssh-client（比如ubuntu desktop）

连接

ssh name@ip或者域名

如果用户名相同 ，可以直接ssh ip或者域名

ctrl+d 退出服务器

公钥私钥(不用每次登陆都输密码)

ssh-keygen创建

默认保存在.ssh文件夹中

id\_rsa id\_rsa.pub

然后将公钥上传到服务器

可以手动粘贴到/home/name/.ssh/authorized-keys

自动ssh-copy-id 地址

tmux 一个管理工具，可以保护现场，比如

## rsync

scp

同步目录

### 上传

目录

rsync –r mydir address:路径（即使默认路径也要有:）

rsync -v weekly.zip [root@10.8.250.84:/home/frontend/](mailto:root@10.8.250.84:/home/frontend/)（没有连接的情况下）

目录下的文件

-a

-r 递归调用

-v 打印信息

rsync mydir/ address:mydir/

--delete 本地删除的文件，服务器也会删掉

--dry-run

### 下载

反过来即可

# 安装软件

## 手动安装

编译好的程序

解压

移动到自己规划的目录（最好用.开头将目录隐藏）

添加到系统路径echo $PATH（用：隔开）

也可以创建符号链接

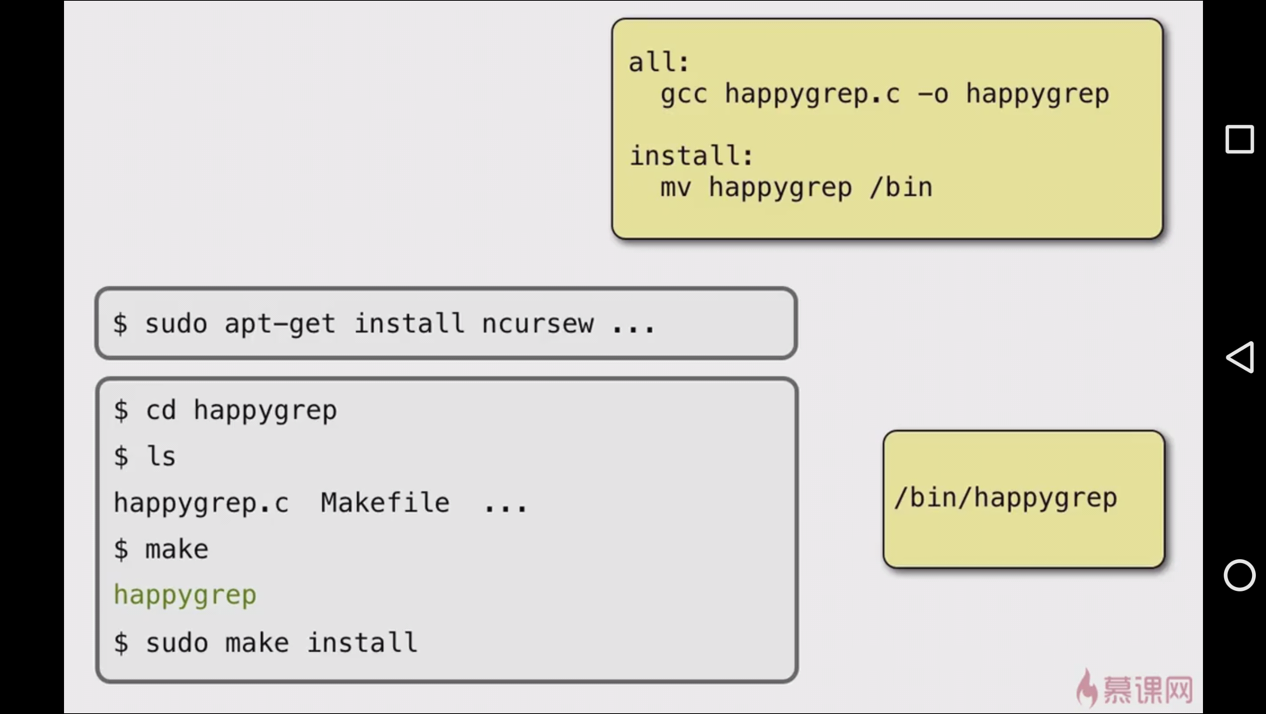
ln –s ~/.sublime/sublime\_text ~/bin/subl(bin本身在系统路径中)

## 自己编译源码

./config

make

sudo make install



## 包管理器

### deb包（apt-get）

deb后缀的软件包是for Debian系的（包括Ubuntu），不是给centos安装的。  
rpm后缀的软件包才是for Redhat系的（包括CentOS）。

装包

sudo dpkg –I google-chrome-stable-current\_i386.deb

常用命令

dpkg –l

dpkg –L 包名（不用输完） 查看这个包安装了哪些文件

dpkg –S 文件绝对路径 查看文件属于哪个包

从apt-get仓库安装

底层包裹了dpkg

安装

sudo apt-get install 软件（如git）

删除

sudo apt-get purge git

purge 同时删除配置文件

搜索

apt-cache serch 软件部分名字| less 分页

### rpm（yum）

CentOS的软件安装工具不是apt-get  是yum

-y: 自动选择y,不用手动确定

yum -y install 包名（支持\*） ：自动选择y，全自动

yum install 包名（支持\*） ：手动选择y or n

yum remove 包名（不支持\*）

rpm -ivh 包名（支持\*）：安装rpm包

rpm -e 包名（不支持\*）：卸载rpm包

0查找

yum 包名

yum -y list java\*

1 安装  
yum install 全部安装  
yum install package1 安装指定的安装包package1  
yum groupinsall group1 安装程序组group1

2 更新和升级  
yum update 全部更新  
yum update package1 更新指定程序包package1  
yum check-update 检查可更新的程序  
yum upgrade package1 升级指定程序包package1  
yum groupupdate group1 升级程序组group1

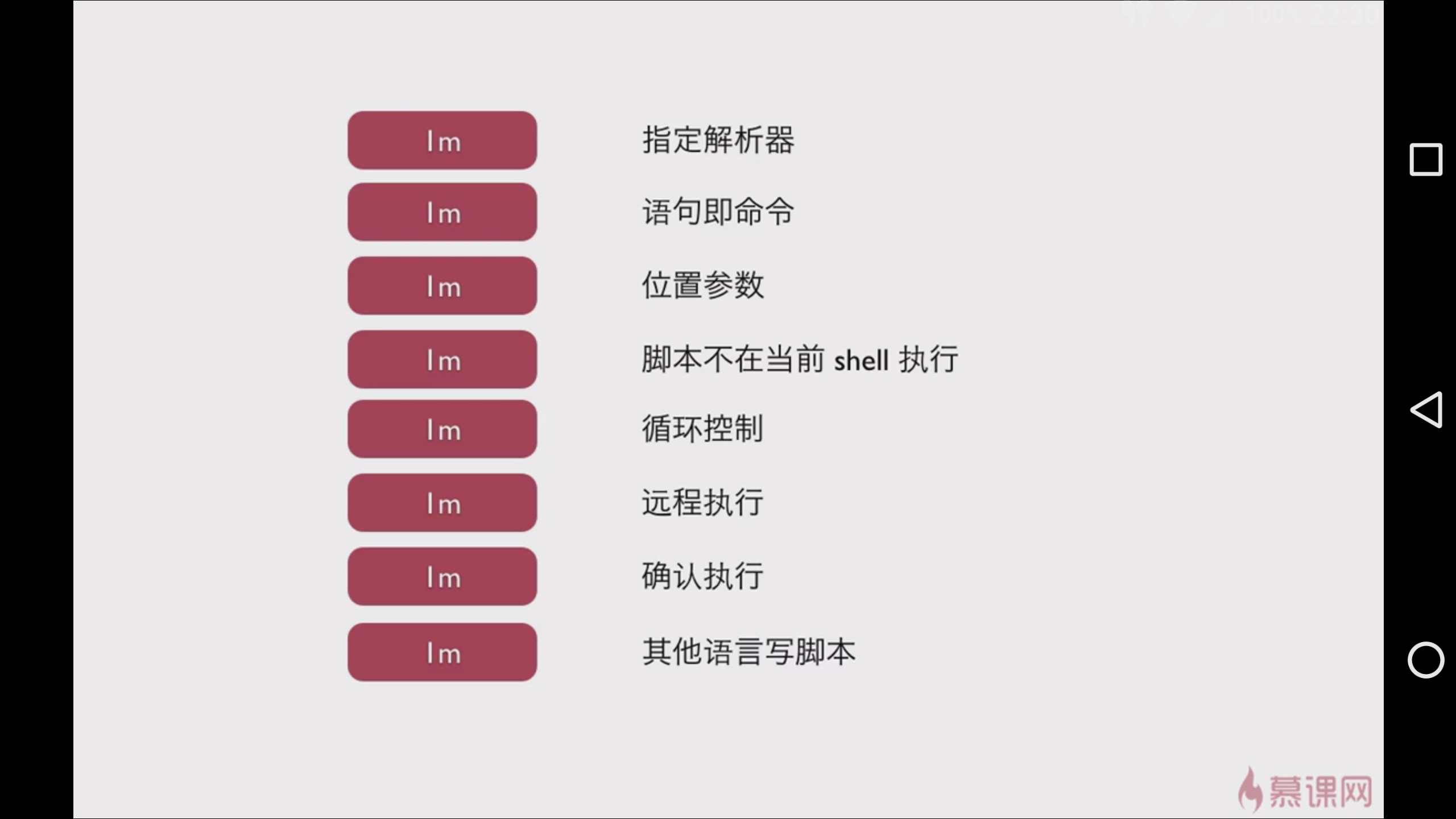
3 查找和显示  
yum info package1 显示安装包信息package1  
yum list 显示所有已经安装和可以安装的程序包  
yum list package1 显示指定程序包安装情况package1  
yum groupinfo group1 显示程序组group1信息yum search string 根据关键字string查找安装包

4 删除程序  
yum remove &#124; erase package1 删除程序包package1  
yum groupremove group1 删除程序组group1  
yum deplist package1 查看程序package1依赖情况

5 清除缓存  
yum clean packages 清除缓存目录下的软件包  
yum clean headers 清除缓存目录下的 headers  
yum clean oldheaders 清除缓存目录下旧的 headers  
yum clean, yum clean all (= yum clean packages; yum clean oldheaders) 清除缓存目录下的软件包及旧的headers

# 脚本编程

常用8个小点



1

第一行 env shell

2

对空格敏感

3

$0 脚本本身

$1 第一个参数

4

新开一个shell，只是看不见

sourece 在当前shell执行

资源：自由网站基金会

# 系统相关

# 整理

## 文件

### less

**1 搜索**

当使用命令 less file-name 打开一个文件后，可以使用下面的方式在文件中搜索。搜索时整个文本中匹配的部分会被高亮显示。

1) 向前搜索

/ ： 使用一个模式进行搜索，并定位到下一个匹配的文本

n ： 向前查找下一个匹配的文本

N ： 向后查找前一个匹配的文本

2) 向后搜索

? ： 使用模式进行搜索，并定位到前一个匹配的文本

n ： 向后查找下一个匹配的文本

N ： 向前查找前一个匹配的文本

**2 全屏导航**

ctrl + F ：向前移动一屏

ctrl + B ：向后移动一屏

ctrl + D ：向前移动半屏

ctrl + U ：向后移动半屏

**3 单行导航**

j ： 向下移动一行

k ： 向上移动一行

**4 其它导航**

G ： 移动到最后一行

g ： 移动到第一行

按空格：向下翻一页

b：向上翻一页

d：向下翻半页

u：向上翻半页

q / ZZ ： 退出 less 命令

**5 编辑文件**

v ： 进入编辑模式，使用配置的编辑器编辑当前文件

**6 标记导航**

当使用 less 查看大文件时，可以在任何一个位置作标记，可以通过命令导航到标有特定标记的文本位置。

ma ： 使用 a 标记文本的当前位置

'a ： 导航到标记 a 处

**7 浏览多个文件**

方式一，传递多个参数给 less，就能浏览多个文件。

less file1 file2

方式二，正在浏览一个文件时，使用 :e 打开另一个文件。

less file1

:e file2

当打开多个文件时，使用如下命令在多个文件之间切换

:n - 浏览下一个文件

:p - 浏览前一个文件

**8  less 版 tail -f**

在 Linux 动态查看日志文件常用的命令非 tail -f 莫属，其实 less 也能完成这项工作，使用 F 命令。

使用 less file-name 打开日志文件，执行命令 F，可以实现类似 tail -f 的效果。