## Linux优势

服务器稳定强大、服务器软件免费，服务器上上之选；

便宜、速度快；

嵌入式系统：家电、手机、数字相机。。。；

## X Window System

Linux上一套软件而已，XFree86计划，X Server管理显卡屏幕鼠标键盘等；

快捷键

Ctrl + Alt + F1~F6，切换终端；

Win键：工作空间；

VirtualBox增强工具：自动调整分辨率、鼠标捕获；

调整图标大小：右键图标resize icon；

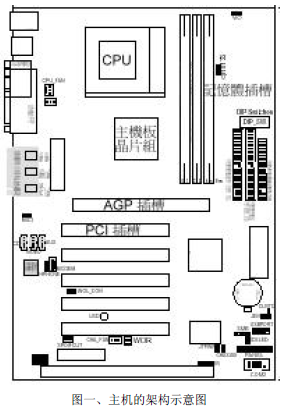
修改桌面背景：右键桌面；

添加输入法：system tools，setting，添加语言；

设置快捷键：setting，keyboard，拉到最下面添加，手动找程序路径，如：/usr/bin/gnome-terminal；

## 硬件

### 介绍



PCI、AGP、SATA。。。随时代变化，推陈出新；

CPU

Intel和AMD；

频率：运算次数/单位时间，外频：CPU与接口设备传输/运算，超频：工作在额定功率之上，更快但不稳定，够快了不用，而且PC的速度和内存、CPU、显卡速度，硬盘的速度以及其它相关的输入输出接口速度有关；

显卡

GPU，显卡上处理耗时的画面数据，减轻CPU负担；

显存，画面质量；

BIOS

主板上的软件；

系统宕机怎么办？

软件-》硬件，硬件排查：超频？机壳、CPU等硬件工作温度过高？硬件损坏、兼容？等等；

### 选购

仁者见仁智者见智；

一般小型主机且不含 X Window 系统：

o 用途：家庭用 NAT 主机或小型企业之非图形接口小型主机。

o CPU：大于 Pentun 133 以上等级即可。

o RAM：至少 32MB ，不过还是大于 64MB 以上比较妥当！

o 网络卡：一般的 10/100 Mbps 即可应付。

o 显示卡：随便！只要能够被 Linux 捉到即可，例如 S3 或 Sis 6326

o 硬盘：2GB 以上即可！

• 桌上型 Linux 系统/含 X Window ：

o 用途：Linux 的练习机或 Office 工作机。

o CPU：最好等级高一点，例如 P-III 或 K7 以上等级。

o RAM：一定要大于 256MB 比较好！否则容易有停顿的现象。

o 网络卡：普通的 10/100 Mbps 就好了！

o 显示卡：使用 32MB 以上内存的显示卡！

o 硬盘：越大越好，最好有 20GB 。

• 中型以上 Linux 服务器：

o 用途：中小型企业/学校单位的 FTP/mail/WWW 等网络服务主机。

o CPU：最好等级高一点，例如 P4 或 K7 以上等级。甚至可以考虑使用双 CPU 系统。

o RAM：最好能够大于 512MB 以上，大于 1GB 更好！

o 网络卡：知名的 3Com 或 Intel 等厂牌，比较稳定效能较佳！注意，也可选购 10/100/1000 Mbps 的速度。

o 显示卡：如果有使用到图形功能，则一张 64MB 内存的显示卡是需要的！

o 硬盘：越大越好，如果可能的话，使用 SCSI 或者磁盘阵列，或者网络硬盘等等的系统架构， 能够具有更稳定安全的传输环境，更佳！

### 在Linux中表示

每个装置都是一个“文件（档案）”；，/dev目录下

|  |  |
| --- | --- |
| IDE 硬盘机 | /dev/hd[a-d] |
| SCSI 硬盘机 | /dev/sd[a-p] |
| USB 随身碟 | /dev/sd[a-p](与 SCSI 硬盘一样) |
| CDROM | /dev/cdrom |
| 软盘机 | /dev/fd[0-1] |
| 打印机 | /dev/lp[0-2] |
| 鼠标 | /dev/mouse |
| 磁带机 | /dev/ht0 (IDE)或 /dev/st0 (SCSI 界面) |

## GNU计划

史托曼发起的：建立一个自由、开放的Unix操作系统；gcc（GNU C

Compliler c语言编译器）、glibc（GNU C Library c函数库）、Bash shell等；

为了避免自由软件（开放源码）被利用成为专利软件，史托曼草拟了GPL（General Public License通用公共许可证），也就是说，你可以任意修改发行，但你的软件不能取消GPL授权，而商家包括自由软件工作者如何获利，大部分都是在以“服务”为生，毕竟不是每个人都会写软件的；

为了方便用户使用，Linux distribution（【Kernel + Softwares + Tools + OS安装程序】，网上共享或光盘安装）出现了，商业公司卖这个东东，包括Red Hat、Ubuntu、Fedora。。。，他们都基于Linux GPL，所不同的可能就是自家所开发的界面、管理工具、管理模式等不同；

## 虚拟机

虚拟机，就是个软件，但是它能使用计算机的物理资源（cpu，内存，硬盘。。。），能虚拟出几个逻辑独立的操作系统，互不影响；

virtualbox轻便免费；VMware先进商业；

### VirtualBox虚拟机网络设置（四种方式）

https://www.douban.com/group/topic/15558388/

VirtualBox的提供了四种网络接入模式，它们分别是：   
1、NAT 网络地址转换模式(NAT,Network Address Translation)   
2、Bridged Adapter 桥接模式   
3、Internal 内部网络模式   
4、Host-only Adapter 主机模式   
  
第一种 NAT模式   
解释：   
NAT模式是最简单的实现虚拟机上网的方式，你可以这样理解：Vhost访问网络的所有数据都是由主机提供的，vhost并不真实存在于网络中，主机与网络中的任何机器都不能查看和访问到Vhost的存在。   
虚拟机与主机关系：   
只能单向访问，虚拟机可以通过网络访问到主机，主机无法通过网络访问到虚拟机。   
虚拟机与网络中其他主机的关系：   
只能单向访问，虚拟机可以访问到网络中其他主机，其他主机不能通过网络访问到虚拟机。   
虚拟机与虚拟机之间的关系：   
相互不能访问，虚拟机与虚拟机各自完全独立，相互间无法通过网络访问彼此。   
IP:10.0.2.15   
网关：10.0.2.2   
DNS：10.0.2.3   
一台虚拟机的多个网卡可以被设定使用 NAT， 第一个网卡连接了到专用网 10.0.2.0，第二个网卡连接到专用网络 10.0.3.0，等等。默认得到的客户端ip（IP Address）是10.0.2.15，网关（Gateway）是10.0.2.2，域名服务器（DNS）是10.0.2.3，可以手动参考这个进行修改。   
NAT方案优缺点：   
笔记本已插网线时： 虚拟机可以访问主机，虚拟机可以访问互联网，在做了端口映射后（最后有说明），主机可以访问虚拟机上的服务（如数据库）。   
笔记本没插网线时： 主机的“本地连接”有红叉的，虚拟机可以访问主机，虚拟机不可以访问互联网，在做了端口映射后，主机可以访问虚拟机上的服务（如数据库）。   
  
第二种 Bridged Adapter模式   
解释：   
网桥模式是我最喜欢的用的一种模式，同时，模拟度也是相当完美。你可以这样理解，它是通过主机网卡，架设了一条桥，直接连入到网络中了。因此，它使得虚拟机能被分配到一个网络中独立的IP，所有网络功能完全和在网络中的真实机器一样。   
虚拟机与主机关系：   
可以相互访问，因为虚拟机在真实网络段中有独立IP，主机与虚拟机处于同一网络段中，彼此可以通过各自IP相互访问。   
虚拟机于网络中其他主机关系：   
可以相互访问，同样因为虚拟机在真实网络段中有独立IP，虚拟机与所有网络其他主机处于同一网络段中，彼此可以通过各自IP相互访问。   
虚拟机于虚拟机关系：   
可以相互访问，原因同上。   
IP：一般是DHCP分配的，与主机的“本地连接”的IP 是同一网段的。虚拟机就能与主机互相通信。   
笔记本已插网线时：（若网络中有DHCP服务器）主机与虚拟机会通过DHCP分别得到一个IP，这两个IP在同一网段。 主机与虚拟机可以ping通，虚拟机可以上互联网。   
笔记本没插网线时：主机与虚拟机不能通信。主机的“本地连接”有红叉，就不能手工指定IP。虚拟机也不能通过DHCP得到IP地址，手工指定IP后，也无法与主机通信，因为主机无IP。   
这时主机的VirtualBox Host-Only Network 网卡是有ip的，192.168.56.1。虚拟机就算手工指定了IP 192.168.56.\*，也ping不能主机。   
  
第三种 Internal模式   
解释：   
内网模式，顾名思义就是内部网络模式，虚拟机与外网完全断开，只实现虚拟机于虚拟机之间的内部网络模式。   
虚拟机与主机关系：   
不能相互访问，彼此不属于同一个网络，无法相互访问。   
虚拟机与网络中其他主机关系：   
不能相互访问，理由同上。   
虚拟机与虚拟机关系：   
可以相互访问，前提是在设置网络时，两台虚拟机设置同一网络名称。如上配置图中，名称为intnet。   
IP: VirtualBox的DHCP服务器会为它分配IP ，一般得到的是192.168.56.101，因为是从101起分的，也可手工指定192.168.56.\*。   
笔记本已插网线时：虚拟机可以与主机的VirtualBox Host-Only Network 网卡通信   
这种方案不受主机本地连接（网卡）是否有红叉的影响。   
  
第四种 Host-only Adapter模式   
解释：   
主机模式，这是一种比较复杂的模式，需要有比较扎实的网络基础知识才能玩转。可以说前面几种模式所实现的功能，在这种模式下，通过虚拟机及网卡的设置都可以被实现。   
我们可以理解为Vbox在主机中模拟出一张专供虚拟机使用的网卡，所有虚拟机都是连接到该网卡上的，我们可以通过设置这张网卡来实现上网及其他很多功能，比如（网卡共享、网卡桥接等）。  
虚拟机与主机关系   
默认不能相互访问，双方不属于同一IP段，host-only网卡默认IP段为192.168.56.X 子网掩码为255.255.255.0，后面的虚拟机被分配到的也都是这个网段。通过网卡共享、网卡桥接等，可以实现虚拟机于主机相互访问。   
虚拟机与网络主机关系   
默认不能相互访问，原因同上，通过设置，可以实现相互访问。   
虚拟机与虚拟机关系   
默认可以相互访问，都是同处于一个网段。   
虚拟机访问主机 用的是主机的VirtualBox Host-Only Network网卡的IP：192.168.56.1 ，不管主机“本地连接”有无红叉，永远通。   
主机访问虚拟机，用是的虚拟机的网卡3的IP： 192.168.56.101 ，不管主机“本地连接”有无红叉，永远通。   
虚拟机访问互联网，用的是自己的网卡2， 这时主机要能通过“本地连接”有线上网，（无线网卡不行）   
  
通过对以上几种网络模式的了解，我们就可以灵活运用，模拟组建出我们所想要的任何一种网络环境了。   
比如我想模拟出来一个一台主机，监控一个局域网上网情况的网络环境。   
首先我开启了两台虚拟机vhost1与vhost2，当然如果硬件允许，我同样可以再增加vhost3、vhost4…   
所有的vhost我都设置成internat内网模式，网络名称为intnal，网关为192.168.56.100，意思就是通过 192.168.56.100网卡上网。其中有一台vhost1我设置为双网卡，一张为内网模式（192.168.56.100），一张为网桥模式（192.168.1.101）。两张网卡设置双网卡共享上网   
虚拟机之间为局域网，其中有一台虚拟机vhost1通过与外网相连，所有局域网中的虚拟机又通过vhost1来实现上外网。这样vhost1就可以监控整个虚拟机局域网上网情况了。 

## 安装Linux主机

必须要针对distributions的特性、服务器软件的能力、未来的升级需求、硬件扩充需求等等来考虑，进行合理的主机规划，《硬件-选购》；

Windows是安装在C盘里的，Linux也是，安装在磁盘上的；

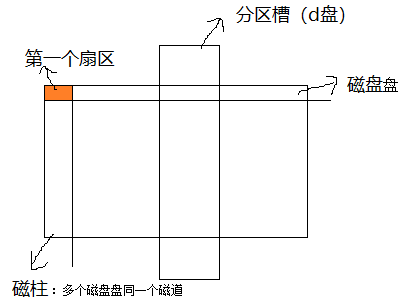
### 磁盘分区

虚拟机下，磁盘档案：/dev/vd[a-p]；

P83主板6个SATA插槽，PC有2个SATA磁盘（装在1，5槽）和1个USB磁盘，SATA1：sda，SATA5：sdb，USB（开机才捕获）：sdc;

一块磁盘，数据记录在磁盘盘上面，磁盘盘又细分“扇区”和“磁道”两种单位，Sector大小有512bytes和4bytes两种；

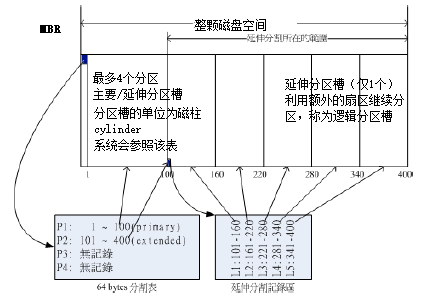
第一个扇区记录整颗磁盘的重要信息，早期叫MBR（Master Boot Record），现在大于2TB的，称为GPT（GUID partition table），它们就是“分区表”，最初磁盘像“原木”，切割后才能使用；

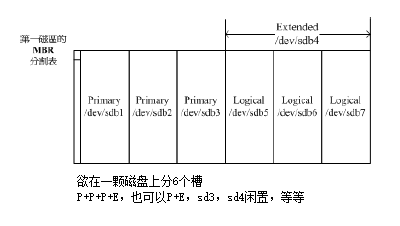


第一个扇区512bytes会有两种数据：

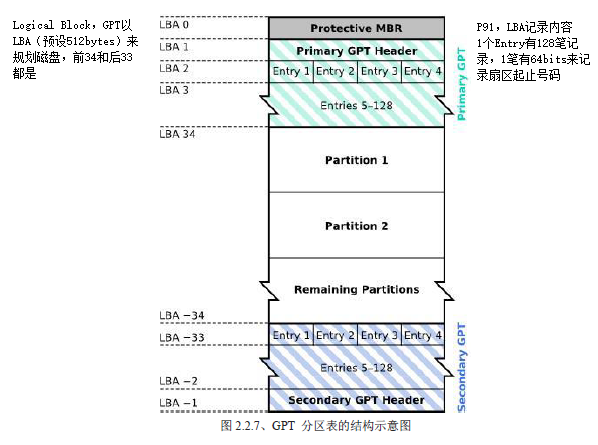
1）主要启动记录区（MBR安装开机管理程序的地方）；

2）分区表，64bytes，磁盘最多有4个分区；





GPT，磁盘容量 > 2TB时；



### 开机程序

BIOS与UEFI机制；

CMOS是记录各项硬件参数且嵌入在主板上的存储器，BIOS则是写入主板的软件程序（韧体，计算机执行的第一个程序）；

大致流程：

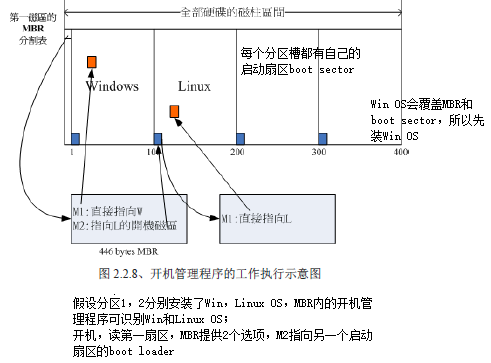
开机-》BIOS-》读取硬件设备，如硬盘-》读取第一个扇区MBR（开机管理程序Boot loader）-》加载核心文件-》开始操作系统的功能；

BIOS与MBR由硬件支持，Boot loader则是由OS写入MBR的一套软件，它包括：

提供选单：用户可以选择不同的开机项目，这也是多重引导的重要功能！

 载入核心文件：直接指向可开机的程序区段来开始操作系统；

 转交其他**loader**：将开机管理功能转交给其他loader负责。



UEFI，定位和BIOS一样，P94；

### 安装系统

了解Linux《文件系统》；

《VirtualBox详细安装使用教程(图文)》、百度文库

虚拟机安装Centos

下载VirtualBox -》下载IOS镜像，Centos官网-》磁盘分区，P124；

安装VirtualBox增强工具；

安装报错：

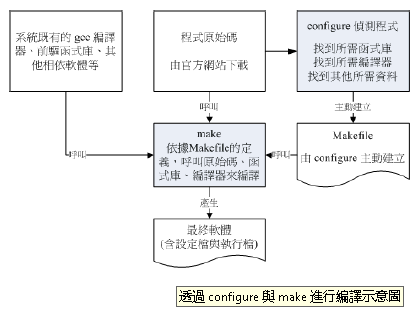
The virtual machine **'Centos7-Everything'** has terminated unexpectedly during startup with exit code 1 (0x1). More details may be available in **'C:\Users\dmxjMao\VirtualBox VMs\Centos7-Everything\Logs\VBoxHardening.log'**.

### 软件安装

Linux上的软件几乎都是经过GPL授权的，都提供源代码，你可以随意定制；

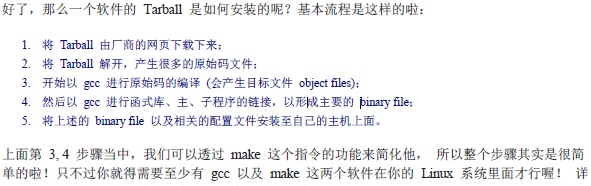
make简化编译过程，搜索Makefile文件（如何编译）；

软件开发商会写一支侦测程序来侦测用户的作业环境，主动建立Makefile规则文件，它叫configure；



#### Tarball软件

就是将源码用tar打包，再gzip压缩，tarball文件 \*.tar.gz之类的；



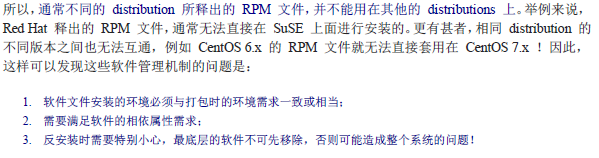
#### RPM

直接拿厂商提供的可执行文件；

Fedora，CentOS，SuSE等使用；

RPM机制就是定义一些管理软件的规则，比如：1，软件依赖的话，安装A需要B，那有网络就去下载B先；2，在线升级；3，终端指令； 等等；

RPM文件在安装时，先读取文件参数，对比Linux环境，找依赖（如果依赖无法安装即失败），安装完毕后，软件信息会写入？/var/lib/rpm目录下（升级查询等都靠它）



SRPM（带Source的RPM，同RPM一起Release），编译它生成适合环境的RPM；

查询和校验

rpm [–q查询软件是否安装a查询所有] , /var/lib/rpm

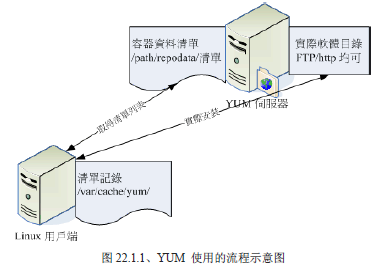
rpm [-ql路径i详细c配置文件d说明文件R依赖文件]

rpm [-qf文件所属软件]

rpm –qp[licdR]未安装的文件信息

yum是Red Hat系统的软件管理工具；

CentOS将软件Release到YUM服务器上



yum指令

查询

yum [-y用户输入自动yes --installroot=/some/path自定义路径] [search查找软件list列出yum server上的软件info软件详细provides提供文件的软件]

yum list updates 列出server可供升级的软件

安装

yum [install安装update升级]

卸载

yum remove 程序名

yum配置文件

CentOS的映射站点可能选错，导致下载非常慢

vim /etc/yum.repos.d/CentOS-Base.repo

baseurl-》软件库的实际网址

软件群组

yum [grouplist可用群组groupinfo组内软件名groupinstall安装所有groupremove移除]

#### Samba

rpm -qi samba 安装信息

报错：Cannot find a valid baseurl for repo: base/7/x86\_64

无法联网：setting-》Network；

yum -y install samba samba-client

cd /etc/samba/

mv smb.conf smb.conf.origin备份smb.conf

vim smb.conf 内容参考已存；

smbpasswd -a mao -e smb123

firewall-cmd --list-services

firewall-cmd --permanent --add-service=samba

 firewall-cmd –reload

重启smb服务

报错：Failed to start Samba SMB Daemon.

testparm查看/etc/samba/smb.conf是否正确，发现security=share已经废弃，改为user，重启服务；

windows中输入：\\虚拟机ip，即可访问虚拟机；

systemctl status network

报错：Failed to start LSB: Bring up/down networking.

不用network，NetworkManager集成了它，可以更方便的切换网卡的配置；

nmcli device status(ifconfig -a) 查看网络配置

nmcli device show ens39

nmcli connection show

enp0s3没有配置文件，所以无法连接网络，因为是桥接，所以要手动配置虚拟机ip

配置见：/etc/sysconfig/network-scripts

BOOTPROTO=dhcp 是自动获取ip，改为static

ONBOOT=yes

HWADDR=08:00:27:80:28:28

MACADDR=08:00:27:80:28:28

IPADDR=192.168.0.144

NETMASK=255.255.255.0

GATEWAY=192.168.0.1

DNS1=192.168.0.22

DNS2=114.114.114.114

DNS3=8.8.8.8

PEERDNS="0"

主机ip: 192.168.0.143，DNS3，PEERDNS好像没什么用；

如果虚拟机要ping通windows，要把windows防火墙关了；

要拷贝内容到share目录，还得关闭SELinux，vim /etc/selinux/config；临时关闭：**setenforce** 0；

VirtualBox的网卡一定要选择正确，无线和有线是不一样的；

VirtualBox 网络配置：NAT，它是单向的，主机无法ping通虚拟机，改用桥接；

### apache+php+mysql

#### mysql

[www.mysql.com](http://www.mysql.com/) → MySQL Community Server → 参考官网手册：2.2 Installing MySQL on Unix/Linux Using Generic Binaries 下面的都是过时的（教训）

bin/mysqld –initialize –user=mysql 会有一个初始密码，要记住

重新初始化报错：data目录不为空

找my.cnf，把data删了；

bin/mysqld\_safe --user=mysql & 手动后台运行mysql服务，ps -aux|grep mysql查看；

报错：mysqld\_safe error: log-error set to '/var/log/mariadb/mariadb.log', however file don't exists. Create writable for user 'mysql'.

修改my.cnf，把mariadb配置去掉；

登录mysql客户端：bin/mysql -u root -p，报错：无法使用/tmp/mysql.sock-》配置my.cnf，[mysql]

service mysql(.service) status -> SUCCESS! MySQL running (5342)，它好像会启动/etc/init.d/mysql

如何用systemctrl 设置mysql.service

vim /etc/systemd/system/mysql.service

[Unit]

Description=MySQL Server

Documentation=man:mysqld(8)

Documentation=http://dev.mysql.com/doc/refman/en/using-systemd.html

After=network.target

After=syslog.target

[Install]

WantedBy=multi-user.target

[Service]

User=mysql

Group=mysql

ExecStart=/usr/local/mysql/bin/mysqld --defaults-file=/etc/my.cnf

LimitNOFILE = 5000

#Restart=on-failure

#RestartPreventExitStatus=1

#PrivateTmp=false

修改ExecStart即可，然后就能用systemctrl管理了；

systemctrl enable mysql开启启动

Created symlink from /etc/systemd/system/multi-user.target.wants/mysql.service to /etc/systemd/system/mysql.service.

没有/etc/my.cnf？可能是用了mysqlnd,mysqli什么的，在/etc/my.cf.d下，拷贝一个到/etc/mysql/my.cnf

export PATH=$PATH:/usr/local/mysql/bin，重启就失效了；

mysql -u root -p -> set password='123456'; mysql123

开机作为服务启动（废弃）

cp support-files/mysql.server /etc/init.d/mysql.server

在/etc/rc.local里添加：systemctl start mysql.server.service

chkconfig --add mysql.server

#### httpd

rpm -ql apache；

安装Development Libraries与Development tools

yum grouplist

yum -y groupinstall development tools

提示没有包可下载，就是不用下了。

httpd编译也依赖pcre-devel软件包

yum -y install pcre-devel

openssl version 查看版本

yum install -y libxml2-devel

yum install openssl openssl-devel； --with-ssl

下载apr和apr-util

[https://apr.apache.org](https://apr.apache.org/)

/usr/local/src下解压编译安装

tar -xjvf apr-xxx.tar.bz2

cd apr-xxx

./configure –prefix=/usr/local/apr

make; make install

apr-util同理，不过安装过错报错：

xml/apr\_xml.c:35:19: fatal error: expat.h: No such file or directory

#include <expat.h>

原因可能是CentOS.ios缺少依赖

yum install expat-devel；再重新make；

这样/usr/local/下就有apr和apr-util两个文件夹了。

<https://httpd.apache.org> -》 安装：<https://httpd.apache.org/docs/2.4/>

./configure –with-apr=/usr/local/apr –with-apr-util=/usr/local/apr-util

默认–prefix=/usr/local/apache2，默认开启—enable-so

bin/apachectrl -k start报错：

httpd: Syntax error on line 66 of /etc/httpd/httpd.conf: Cannot load modules/mod\_mpm\_event.so into server: /usr/local/apache/modules/mod\_mpm\_event.so: cannot open shared object file: No such file or directory

安装的之前，把/etc/httpd删掉，那是之前安装留下的

使用systemctrl管理

同mysql，但是不行；

apache开机启动

cp /usr/local/apache/bin/apachectl /etc/init.d/httpd

vi httpd 在#!/bin/sh下加：#chkconfig:345 85 15 #description: …

chkconfig --add httpd chkconfig httpd on 作为服务并开机启动。

#/etc/rc.local 添加 /usr/local/apache2/bin/apachectrl -k start，开机启动。

httpd: Could not open configuration file /etc/httpd/httpd.conf: No such file or directory

删掉/etc/httpd报错。

httpd就用conf/httpd.conf下的；

(98)Address already in use: AH00072: make\_sock: could not bind to address [::]:80

(98)Address already in use: AH00072: make\_sock: could not bind to address 0.0.0.0:80

no listening sockets available, shutting down

bin/apachectl -k start报错，端口占用

ps -ef | grep httpd 查看包含httpd的进程信息

netstat -lnp|grep 80 查看80端口是否被占用

-> tcp6 0 0 :::80 :::\* LISTEN 15205/httpd

可以看到，其实httpd已经起来了，你也可以kill -9 15205强制杀死httpd进程再重启一遍;

#### php

[www.php.net](http://www.php.net/)

/usr/local/src → tar -xjvf → cd php-xxx → ./configure … make & make install-s

同样吸取教训，按官网来。

./configure --with-apxs2=/usr/local/apache/bin/apxs

--with-mysqli //mysql的增强版

--with-pdo-mysql

--with-zlib( symfony )

--with-openssl

报错：Cannot find OpenSSL’s <evp.h>

yum install openssl-devel，继续报错，yum被弄坏了

yum clean all

rm -f /var/lib/rpm/\_\_db\*

rpm –rebuilddb

yum update（应该可以不用）

yum install openssl-devel

php无法使用mysql，报错：找不到驱动

php -m |grep -i "pdo"

配置mysql

<http://www.runoob.com/php/php-mysql-connect.html>

<http://php.net/manual/en/mysqli.installation.php>

php-mysql包，ext/mysql, ext/mysqli, and pdo\_mysql PHP extensions.

yum install php-mysql

安装PDO

<http://php.net/manual/en/pdo.installation.php>

./configure只是配置，扩展还是要自己去下载的

http://pecl.php.net/get/PDO\_MYSQL-1.0.2.tgz 官网下载

<https://www.cnblogs.com/metasequoia/p/3968081.html>

php自带很多扩展：

/usr/local/src/php-5.6.36/ext/pdo\_mysql

安装php扩展

cd phpxxx/ext/=》 whereis phpize，如果没有yum install php-devel

xx/phpize ->生成config文件

./configure --with-php-config=(whereis php-config)

make & make install

编译完之后会生成：apache2/modules/libphp5.so

cp php.ini-development /usr/local/lib/php.ini

apache加入php模块

httpd.conf

LoadModule php5\_module modules/libphp5.so

<FilesMatch \.php$>

SetHandler application/x-httpd-php

</FilesMatch>

重启apache，测试

apaceh/htdocs，test.php 输入 <?php phpinfo() ?> -》 localhost/test.php

test.php原样输出

因为httpd.conf里加了这个，把它去掉

<FilesMatch "\.phps$">

SetHandler application/x-httpd-php-source

</FilesMatch>

Symfony框架

Symfony中国，php symfony，报错：

PHP Fatal error: Uncaught exception 'PharException' with message 'zlib extension is required for gz compressed .phar file "/usr/local/web/symfony"' in /usr/local/web/symfony:9

Stack trace:

#0 /usr/local/web/symfony(9): Phar::mapPhar('defaiult.phar')

phpinfo() 页面：

|  |  |
| --- | --- |
| gzip compression | disabled (install ext/zlib) |
| bzip2 compression | disabled (install pecl/bz2) |
| OpenSSL support | disabled (install ext/openssl) |

Whereis zlib → cd php-xxx, ./configure –help , --with-zlib→ 三步走

symfony new prj，报错：

Unable to find the wrapper "https" - did you forget to enable it when you configured PHP?

Cd php-xxx, ./configure –with-openssl

安装php扩展：zlib和curl，yum install curl curl-devel；

caching\_sha2问题

修改my.cnf，[mysqld]

default... = mysql\_native\_password

mysqli\_real\_connect() 字符集问题

[mysqld]

collation-server=…

character-set-server:utf8mb4

## Linux操作系统

OS的概念都是一样的，连接着硬件与软件的桥梁；Linux就是GNU GPL授权模式的一个“软件”，任何人均可取得源码并修改发行；

### 寻找帮助

g + tabtab，200多条！

【--help】选项，但是bc的scale，+-\*/^%等用法就不会用，--help常用于协助查询【指令的选项与参数】信息；

【man 指令】manual，使用手册；

如man date，注意有出来的页面叫“man page”, DATE(1)这个1是有意义的，man man区段；

man page常用快捷键：

Page Down/空格、Home、/(?)word向下/上搜索、n/N继续搜索、q推出man page；

与指令相关的说明文档可能有多个，如man -f man可能出现 man(1)，man(7)，默认的man page是数字较小的，man 7 man可以指定，也可以修改/etc/man\_db.conf调整顺序；

man -k 关键词，包含关键词的文档；

man -f等价于 whatis ，man -k 等价于 apropos，要先用root身份建立相关db，即执行# mandb；

【info 指令】manual的链接版，按“h“看快捷键；

其他

可以参考下面三个地方的文档，软件开发者制作的“帮助文件”一般也会放在这里；

/usr/share/man，/usr/share/info，/usr/share/doc，。。。

### 关机

who：谁在线；

netstat –a： 网络联机状态；

ps –aux：后台执行的程序；

sync：数据写入磁盘，reboot/shutdown内联；

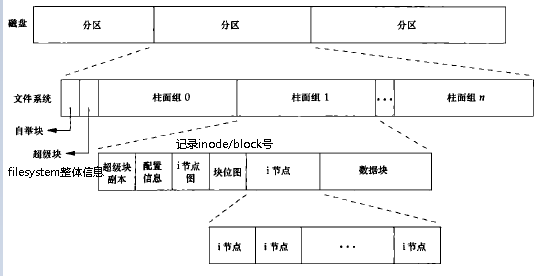
reboot：重启；

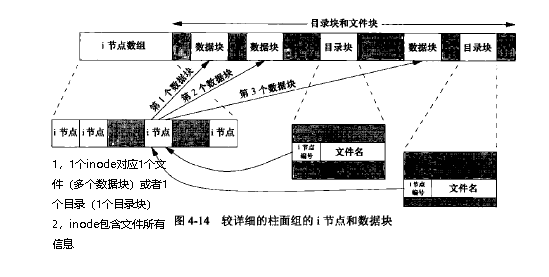
shutdown：关机；

### 文件系统

#### 结构

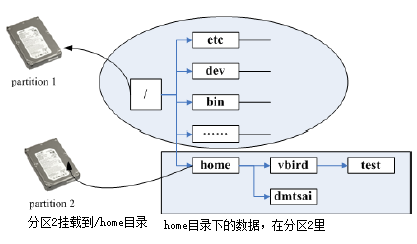
文件系统是针对分区(partition)的，本地、网络上的filesystem都可以挂载，对分区格式化以建立fs；





#### 挂载

将目录作为“进入点”（称为挂载点），进入磁盘分区槽的数据所在位置；



#### 权限

任意文件都有【User，Group，Other】三种身份的权限；

/etc/passwd，系统账号信息；/etc/shadow，个人密码；/etc/group，群组信息；

ls -al (ll –a)

drwxr-xr-x.（权限） 2（连接数） Morris（ower） Morris（group） 4096（bytes） Mar 10 19:23（最后修改时间） Documents（档案名）

##### 指令

/etc/group，群组信息中要有；

chgrp [-R]递归 users群组名 x.cfg目录或文件名；

/etc/passwd，拥有者要有；

chown [-R] myn:users x.cfg 可以连组一起改；owner为myn，组为users；

r:4 w:2 x:1，chmod [-R] 664 x.cfg；

-rw-rw-r—664， -rwxr-xr-x755，-rwxr-----740其他人不可见；

另一种方式：u,g,o,a,+,-,=；

chmod u=rwx,go=rx .xcfg，chmod a+w x.cfg所有人添加w权限；

在linux下，只要文件具有x属性，就可以被执行，与其后缀名无关，但是通常我们会加上后缀名，如\*.sh脚本，\*.Z,\*.tar,\*.tar.gz,\*.zip,\*.tgz压缩包，\*.html，\*.php。。。；

##### 目录权限

r 可以读（ls）目录结构列表；

w 可以修改目录结构列表（创建文件或目录、删除无论权限为何、重命名、移动）；

x 可以进入（cd）成为工作目录；

##### 默认权限

umask [-S符号表示] ，掩码：默认值要减掉的权限；umask 0000 修改；

默认建立文件：-rw-rw-rw-，目录：drwxrwxrwx；

##### 隐藏属性

chattr [+-=] [Aatime不改S同步写磁盘a只能增加数据c保存时压缩读取时解压d dump i不能动s永久删除u缓存删]

lsattr [-a隐藏文件d目录本身R递归]

##### 特殊权限

Set UID，s在owner的x位置上；

SUID 权限仅对二进制程序(binary program)有效，如cat、passwd；

 执行者对于该程序需要具有x 的可执行权限；

 本权限仅在执行该程序的过程中有效(run-time)；

 执行者将具有该程序拥有者(owner) 的权限，如cat的owner是root，但cat没有SUID权限。

Set GID，s在group的x位置上；

类似SUID，执行者将具有该程序群组 的权限，同时，一个目录也可以设定SGID，用户进入此目录，有效群组变为该目录群组，这样用户新建文件时，新文件群组为该目录群组；

Sticky Bit，t在other的x位置上；

只对目录有效，目录有wx权限，其下的文件/目录，只有自己与root能删除；

4 SUID，2 SGID，1SBIT，

chmod 4775 .。。。

chmod u=rwxs,g+s,o=x test;

#### 目录

目录树的根目录是“/“，“~”家目录，“~Morris”Morris的家目录，”-”上一个工作目录；

##### 目录配置

依据FHS（Filesystem Hierarchy Standard）标准，规范特定目录下应该放置什么样的数据，它的三层主目录是：/根目录 , /var运行时，/usr(Unix Software Resourse)应用程序；

《鸟叔 P225》建议的目录配置；

##### 指令

cd，变换目录；

pwd [-P]，显示当前目录；

mkdir [-mp]，建立目录；

rmdir，删除目录；

#### 文件

##### 指令

ls [d目录i”inode”F分类R递归St排序]

cp [i覆盖询问l硬链接s软链接p非默认属性r递归u目标旧或不存在复制d软链接属性a”drp”组合] 源1 (源2...) 目标

rm [f强制i询问r递归]

mv [fiu源较新才更新] 源1(源2...) 目标

==================文件内容=============================

cat [b行号空白行无E显示断行字符n行号T标记tab v特殊字符A”vET“组合] 显示全文内容

more 文件

快捷键：空格，翻一页；Enter，翻一行；/string，当前页搜索；b，回翻；

less 文件

快捷键：空格/PageDown，翻一页；/(?)string；n/N，继续（反向）搜索；g，去第一行；G，去最后一行；

head [-n 数字，负数是不要前n行] 文件，显示前n行； tail类似

od [-t cASCII字符dox[size]十八十六进制，每个整数占size位] 二进制文件；

touch [-acm修改atime ctime mtime dt欲修改的时间] 文件，无选项会新建文件；

====================查找============================

file 文件，查看文件类型；

which [-a所以PATH的指令] 指令，查找指令完整路径；

whereis [l列出查找目录b只找二进制m只找manual下 s源文件u除了bms] 档案名；

locate [-i忽略大小写r正则l 数字，列出] 关键词，whereis和locate从已建立的db找，find从硬盘找，所以有时候找不到，P273更新db；

find [PATH]

[option

–a/c/mtime n n天前的24h内 +n n天前不含n –n n天内 –newer file比file还要新的

-uid/gid n –user/ group name –nouser/nogroup

-name filename –size[+大-小][ck]，如-size +50k比50k大的文件 –type[f普通b设备d目录l链接s套接字 p管道 –perm [-完全包含/包含任意]mode =权限

] [action]

find / -mtime 0，0代表当前时间；

find / -name “\*passwd\*”，通配符是通用的；

find / -perm /7000 –exec ll {}\; action，{}代表查找内容；

==================备份还原、光盘刻录=========================

《鸟叔Chap8.4、8.5、8.6》

====================================================

##### 压缩

\*.Z compress 程序压缩的文件，过时；

\*.zip zip 程序压缩的文件，兼容win；

\*.gz gzip 程序压缩的文件，GNU计划；

\*.bz2 bzip2 程序压缩的文件；

\*.xz xz 程序压缩的文件，bzip2，xz比gzip好，配合tar；

\*.tar tar 程序打包的数据，并没有压缩过；

\*.tar.gz tar 程序打包的文件，其中并且经过 gzip 的压缩

\*.tar.bz2 tar 程序打包的文件，其中并且经过 bzip2 的压缩

\*.tar.xz tar 程序打包的文件，其中并且经过 xz 的压缩

gzip [-c压缩数据输出屏幕d解压t校验一致性v压缩比#1-9，1最快压缩最差，默认6] 档案名，

gzip –c services > services.gz，保留源；

zcat/zmore/zless/zgrep，针对gzip；

bzip2 [-k保留源，其他同gzip]，bzcat/bzmore/bzless/bzgrep；

xz [-l列出压缩信息，其他同bzip2]，xzcat...；

tar [-z用gzip压缩j用bzip2 –J用xz -c打包v查看档案详细t查看打包档案x解压C指定目录解压-f目标档案名，和f紧挨] 目标 源

压缩：tar -jcv -f filename.tar.bz2 要被压缩的文件或目录名称

 查询：tar -jtv -f filename.tar.bz2

 解压缩：tar -jxv -f filename.tar.bz2 -C 欲解压缩的目录

解压单个文件

tar -jtv -f /root/etc.tar.bz2 | grep 'shadow'，查询压缩包里的文件

tar -jxv -f /root/etc.tar.bz2 etc/shadow，解压

除去一些文件

tar -jcv -f /root/system.tar.bz2 --exclude=/root/etc\* \ > --exclude=/root/system.tar.bz2 /etc /root

#### 磁盘

df [-a所有kmh bytes格式T列出fs名称i 以inode数量显示]，文件系统整体磁盘使用量；

du [-a所有s总量S不含子目录总量kmh易读] 档案名，档案使用量；

lsblk [-d磁盘f磁盘fs名-i ASCII输出m权限p完整路径t详细] 设备，list block device，P312；

blkid 设备UUID；

parted 设备 print，分区信息；

磁盘分区、格式化、校验、挂载，《鸟叔Chap7.3》；

#### 链接

1. 软link，快捷方式；

独立的文件；

2）inode连结，硬link；

不能跨Filesystem；

 不能link 目录；

ln [-s软f目标覆盖] 源 目标

### vi编辑器

Unix系统内建vi，vim是vi的增强版，是个【程序开发工具】；

常用vi操作

ctrl+d/u，翻半页；

n空格，移动n个字符；H/M/L当前页第一/中间/最后行的第一个字符；

G文件最后，nG第n行，gg文件开始，n回车，向下移动n行；

u复原，ctrl+r重做；

(n)x(del)，X(backspace)；(n)dd删除整行；d$删除光标到行最后；dG光标到文件最后；

(n)yy复制行；p，P光标之前、之后行粘贴；

/(?)word查找；

:n1,n2s/word1/word2/g(c提示用户)，[n1,n2]行之间将word1替换为word2；:1,$s/word1/word2/g;

.小数点，重复上一个动作；

区块操作

v选择列，V选择行, Ctrl+v区块选择, y复制反白, d删除反白,p粘贴，配合n，方向键；

批量注释

V选择行，ctrl + v, 大写I，输入//，然后连按两下Esc；

vim复制粘贴到外部

vim --version|grep “clipboard”

如果-clipboard前面是-号，说明不支持粘贴到外部

多文本编辑

vim a.txt b.txt...

:n 下一个文件, :N, :files列出开启的文件，也可以打开多个vim，用鼠标复制粘贴；

:sp vim中打开多个窗口，:sp filename；

ctrl+w+上下，切换窗口；

ctrl+alt+t，新建tab页；

ctrl+t，设置快捷键，打开Terminal；

插件

<https://github.com/Vim/vim> 注意Vim大写

Vundle, github.com/VundleVim

<https://www.vim.org/>

pathogen，不要拷贝到/usr/share/vim/vim74/autoload下！它在~/.vim里！

[Universal Ctags](https://ctags.io/) ：<https://github.com/universal-ctags/ctags>

taglist

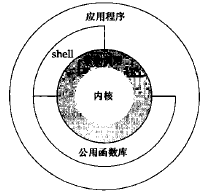
NERD tree :git clone https://github.com/scrooloose/nerdtree.git ~/.vim/bundle/nerdtree

~/.vim/bundle/nerdtree/doc/， 其实并没有用，还是要拷贝到vim74里；

vim-nerdree-tabs

### BASH shell

操作OS的基本接口，它也是一支程序，称为“壳”’ ，Linux预设的shell为Bash shell；



指令格式：命令 [选项] 参数，区分大小写，’\’指令太长换行；

#### 快捷键

tab：补齐功能，命令/文件名+两次tab；

ctrl+c：终止程序；

ctrl+d：输入结束，相当于exit；

shift + PageUp/PageDown，终端翻页；

\Enter 换行输出；

ctrl+u、k删除光标前、后指令串；

atl 1~9终端分页切换，shift+ctrl+T新建分页，shift+ctrl+N新建窗口；

#### 通配符

\*、?、[]、[-]、[^]，和regex一模一样；

#注释、\转义、;连续指令、!非

&指令后台执行

|管道

>,>> 流重定向，取代和累加

<,<< 流重定向

‘’，$是纯文本，””，$保留功能

`` 先执行的指令

() 子shell的起始结束

{} 命令区块的组合

#### 正则

语系影响匹配，默认LANG=C，兼容POSIX标准；

[:alnum:] [0-9][a-z][A-Z]

[:alpha:] [a-z][A-Z] [:lower:] , [:upper:]

[:blank:] 空格和tab

[:cntrl:] 控制键，ctrl，上下左右，tab，del，bs。。。

[:digit:] [0-9]

[:graph:] 除了[:blank:]

[:print:] 可打印字符

[:punct:] 标点符号

[:space:] 能产生空白的字符，ctrl

[:xdigit:] [0-9A-Fa-f]

^$行首行尾，grep ‘^$’ 空白行

.\*

{m,n}，’o\{2,5\}’

延伸正则

egrep

+ {1,n}

? {0,1}

| 或

() 组

()+ 组

grep [-A后跟数字，after后面几行B before几行—color=auto –n行号] ‘查找字符串’ filename

sed [-n处理过的行才列出ef将sed动作写到文件内r延伸语法i修改文件] ‘[n1,n2行$最后一行] action，分析stdin，取代删除新增撷取特定行，管道命令；

function：a新行新增，\再新增一行，c取代，d删除，i插入，p打印，s取代可搭配正则；

nl /etc/passwd | sed ‘2,5d’ 删除2,5行

sed –n ‘5,7p’ = head –n 7 | tail –n 4

sed ‘s/旧/新/g’

awk ‘{条件}. {动作..}.‘ filename，sed行处理，awk倾向于一行的分段处理；

last –n 5 | awk ‘{print $1 “\t” $3 “lines:” NR第几行” \t cols:” NF共几列 FS当前分隔符 }’ 打印1,3列，默认列以空格或tab分隔，$0整行

条件： >,<,>=,<=,==, !=

cat /etc/passwd | awk ‘{FS=”:”} $3 < 10 {print $1 “\t” $3}’

|  |
| --- |
| cat pay.txt | \  > awk '{if(NR==1) printf "%10s %10s %10s %10s %10s\n",$1,$2,$3,$4,"Total"}  > NR>=2{total = $2 + $3 + $4  > printf "%10s %10d %10d %10d %10.2f\n", $1, $2, $3, $4, total}' |

多条指令动作，分号或Enter；

变量直接用；

#### 文件比对

diff [-b多个空白相同B多行空白相同i大小写] file\_src file\_des，以行为单位比对

cmp [-l列出所有不同字节，默认第一个]，字节比对

#### 变量

##### 规则

echo $变量名，查看变量；

变量名=变量值（不能有空格）；

[a-zA-Z0-9]，开头不能是数字；

单引号，特殊字符为一般字符，双引号，var=”lang is $LANG”，保留特性，有些；

可以用’\’转义特殊字符，如[Enter],$,\,空格,’等；

赋值指令结果：var=$(指令)或`指令`；

代表变量值：”$变量名”或$(变量名)；

export 变量名，变成环境变量，共享；

unset 变量名，取消变量；

##### 变量修改

read [-p提示t等待秒数] 变量，读取变量；

declare [-a数组i整型x同export-r同const-p列出变量类型+aixr取消] 变量，声明变量，默认字符串；

删除

${变量#/\*local/bin:} ${变量}格式，#前面开始删，最短，##最长；%从后面开始删，%%；

替换

${变量/旧字符串/新字符串} 替换第一个；

${变量//旧字符串/新字符串} 全部替换；

变量测试

新变量=${旧变量-值}，没有则设置；

新变量=${旧变量:-值}，有冒号，变量没有或空字符串则设置；

+、=、：、-、？，《鸟叔P449》

#### 别名

alias lm=’ls –al | more’ 简单；

alias rm=’rm –i’ 安全；

alias ls=’ls –color=auto’ 功能；

alias，查看别名；

unalias，取消别名；

#### 环境配置

env，查看环境变量；

set，查看env+自定义；

?， 上一个指令的回传值；

/etc/issue 欢迎信息，/etc/issue.net telnet的主机登入画面，/etc/motd给登入者提示，P455；

login shell（登入bash，要走流程的） 和 non-login shell

login shell流程：读取/etc/profile系统设定，读取~/.bash\_profile或~/.bash\_login或~/.profile，个人设定，在CentOS下个人设定:~/.bash\_profile->~/.bashrc；

/etc/profile主要内容：环境变量、/etc/profile.d/\*.sh包含其他脚本等；

然后读取个人配置；

source

不用注销，使配置生效；

source 配置文件，或 .小数点 配置文件，将配置读进当前shell；

non-login shell 只读取~/.bashrc；

其他配置：/etc/man\_db.conf、~/.bash\_history、~/.bash\_logout。。。

stty(set tty) [-a所有]

列出终端机的按键配置，^代表ctrl，intr中断erase向后删除字符。。。；

stty erase ^h 设置；

set设定tty配置，P463‘’

还有/etc/inputrc、/etc/DIR\_COLORS\*。。。

#### 数据重定向

指令 > 文件，>前面默认是1（stdout），

command > file1 2> file2 ，stdout和stderr到不同文件，2>没有空格；

/dev/null

黑洞装置，吃掉任何导入的信息，如 2> /dev/null；

command > file1 2>&1，stdout和stderr写到一个文件中；

<,<<

文件内容代替键盘输入

cat > file < filein ，把filein输入到file；

<<代表“结束的输入字符”

cat > file << “eof” 输入内容+eof，而不需要ctrl+d；

#### 管道

处理前一个指令传来的正确信息；

cmd1 | cmd2，要求：1）|后肯定是cmd，2）cmd2能处理cmd1的正确stdout信息；

撷取

cut [-d分隔字符f取第几段-c取出区间字符]

echo ${PATH} | cut –d ‘:’ –f 3,5 -f 1-

grep [-a二进制以text方式搜索c计算找到次数i忽略大小写n输出行号v反向选择] [--color=auto] 搜索字符串 文件名

排序

sort [-f忽略大小写b忽略前导空格M月份缩写排序n数字排序r反向t分隔符默认tab –u相同数据取一个k区间] 文件或stdin

cat /etc/passwd | sort，默认以第一列的字符形态排序；

cat /etc/passwd | sort –t’:’ –k 3 以第三列排序；

uniq [-ic计数]

去掉重复数据；

双向重定向

tee [-a追加] filename，一份文件，一份stdout；

字符转换

tr [-d删除s取代] 字符串...，删除讯息文字；

last | tr [a-z] [A-Z] 取代

col [-x将tab转成对等的空格b]

join [-t分隔符，比对第一列，相同则两个文件相同行合并i大小写12哪个文件用第一列分析] file1 file2，分析多个文件，类似sql 联合查询；

join –t’:’ /etc/passwd /etc/shadow | head –n 3

paste [-d分隔符，默认tab] file1 file2(-代表stdin)直接粘贴两行；

expand [-t几个空格] file 将tab转空格；

xargs

xargs [-O去特殊字符含义p询问e结束工作字符串n使用几个参数] cmd ，产生指令参数；

cut -d':' -f 1 /etc/passwd | head -n 3 | xargs -n 1 id（因为id不是管道指令，xargs是）

#### 未归档指令

可执行文件，系统会在环境变量PATH目录下找，echo $PATH；

echo $PATH查看；

PATH=”${PATH}:/Xxx“ 添加PATH；

locale查看修改 语言；P175

date、cal日历、bc计算器，scale=3；P175

quit退出程序；

su，切换root身份；su Morris；sudo 指令；

type [-t类型p外部指令完整路径a所有] name，指令是否为bash内建；

ulimit [-SHacdfltu] [配额]，限制用户访问资源，P443；

history [-n列出n条c取消缓存r将histfiles内容读到当前shell的history缓存中a追加history指令到histfiles中w写history到文件]，登入读取~/.bash\_history，环境HISTFILESIZE（1000）条，！n执行、！command（执行最近开头为command的指令）；

last [-n] 最近登录信息；

wc [-l多少行w多少单词m多少字符]，列出文件数据统计；

split [-b欲分区的文件大小l以行分区] file PREFIX文件前缀

split –b 300k /etc/services services（生成的小文件自动以xxxxaa/ab/ac命名）

cat services\* >> servicesback 合成一个文件

ll –a / | split –l 10 – lsroot ，10行一个文件；

#### Shell Scripts

脚本，批处理文件；

##### 语法

逐行，从左到右执行指令，遇Enter（CR）执行；

n个空白，tab等等同；

#是注释，#!/bin/bash声明script使用的shell名称；

##### 执行

xx.sh 需要rx权限；

1）/home/../xx.sh；2）./xx.sh；3）PATH，sh xx.sh；

##### 习惯

script文件头记录

 script 的功能；

 script 的版本信息；

 script 的作者与联络方式；

 script 的版权宣告方式；

 script 的History (历史纪录)；

 script 内较特殊的指令，使用『绝对路径』的方式来下达；

 script 运作时需要的环境变量预先宣告与设定。

##### 实例

###### hello，world

#!/bin/bash

#Program:

echo –e “Hello,World!”

exit 0 #$0返回值

###### 用户输入

read，P528

###### 随日期变化

date，P528

###### 数值运算

declare –i total=${val1}+-\*/(整数)${val2}

小数：echo “123.123\*44.9” | bc

#### BASH操作

##### 终端连续指令

1，分号；

2，$? && ||

$?=0指令执行成功，

cmd1 && cmd2，cmd1正确则开始cmd2，否则不执行；

cmd1 || cmd2，cmd1正确则cmd2不执行，否则执行；

/tmp/abc不管存在否，都建立hehe文件

ls /tmp/abc || mkdir /tmp/abc && touch /tmp/abc/hehe &&和||只和上一条cmd $? 有关；

## make

### 概念

make指令会搜寻Makefile（or makefile）文件，makefile记录了编译的所有信息！而make指令能够增量编译！

configure

软件开发商写的，用来侦测用户环境，然后自动生成Makefile文件。

是否有适合的编译程序可以编译本软件的程序代码；

 是否已经存在本软件所需要的函式库，或其他需要的相依软件；

 操作系统平台是否适合本软件，包括Linux 的核心版本；

 核心的表头定义档(header include) 是否存在(驱动程序必须要的侦测)。

### makefile

目标：目标文件1-N

[tab] gcc –o 执行文件 目标文件1-N

#注释

比如:

main: main.o haha.o sin\_value.o cos\_value.o

gcc -o main main.o haha.o sin\_value.o cos\_value.o -lm

clean:

rm -f main main.o haha.o sin\_value.o cos\_value.o

make main / make clean / make clean main

makefile支持shell script

变量

VAR = -lm

左边不能有空格tab等，

${变量} 或 $(变量) 使用，

shell的环境变量可以直接使用，

取用规则是：make指令后的优先；makefile里的其次；shell的第三；

$@代表目前的目标，

LIBS = -lm

OBJS = main.o haha.o sin\_value.o cos\_value.o

CFLAGS = -Wall

main: ${OBJS}

gcc -o $@ ${OBJS} ${LIBS} <==那个 $@ 就是 main ！

clean:

rm -f main ${OBJS}