1. 快捷键

搜索用过的命令： ctrl+r

2. 安装软件

2.1. Fedora 中安装

rpm

1. 安装一个包

rpm -ivh

2. 升级一个包

rpm -Uvh

3. 移走一个包

rpm -e

4. 安装参数

--force 即使覆盖属于其它包的文件也强迫安装

--nodeps 如果该 RPM包的安装依赖其它包，即使其它包没装，也强迫安装。

5. 查询一个包是否被安装

rpm -q < rpm package name>

6. 得到被安装的包的信息

rpm -qi < rpm package name>

7. 列出该包中有哪些文件

rpm -ql < rpm package name>

8. 列出服务器上的一个文件属于哪一个 RPM包

#rpm -qf

9. 可综合好几个参数一起用

rpm -qil < rpm package name>

10. 列出所有被安装的 rpm package

rpm -qa

11. 列出一个未被安装进系统的 RPM包文件中包含有哪些文件？

rpm -qilp < rpm package name>

yum

yum search *** 搜索匹配字符的 rpm 包

yum install *** 下载安装指定的包

yum remove *** 删除 *** 包，包括与该包有依赖性的包

yum update *** 更新指定的包

yum info updates　　列出已经安装的所有的　　rpm 包的信息

yum info installed　　列出已经安装的但是不包含在资源库中的　　rpm 包的信息

yum info ***　　列出所有以　mozilla　开头的　rpm 包的信息

只下载不安装：

yum install yum-downloadonly

yumdownloader　需要安装的包

在/var/cache/yum/InstallMedia/packages/　　下找

2.2.　Ubuntu 中安装

apt-get install　　软件包

在 Debian 中，软件包是通过一个数据库来管理的，通过这个数据库可跟踪系统中已安装、没有安装和现在可安装的软件包信息。如果软件包需其他软件包支持，　　apt-get　会搜索该数据库找到这种依赖关系一起下载相关软件。下载的软件包默认放在：　　/var/cache/apt/archives

如果安装指定版本：　apt-get install　　软件包 =版本

apt-get remove　　软件包

卸载没用的软件包，如果连配置文件一块删，则　　apt-get　-purge remove　软件包

aptitude

查看已安装或可用的软件包。

apt-cache showpkg　　软件包

显示软件包信息

apt-cache search　　字符串

在软件包列表中搜索字符串

apt-cache depends packagename

检查　packagename 依赖的包

3.　文件相关

3.1.　压缩解压：

gzip [　选项]　文件名

-d　将压缩文件解压（一般为　tgz　文件）

-v　对每一个压缩和解压的文件，显示文件名和压缩比。

tar　　- c　（ create,　建立一个压缩文件）

-x　（解压，不可与　-c ，-t ，同时使用）

-v　(　压缩过程中显示文件　)

-f　（使用文档名，其后立即接文档名，不要再加参数）

3.2.　查找文件：

find pathname　- name *.*

-type　b　(　　块设备文件　)

d　　（目录文件）

c （字符设备文件）
p （管道文件）
l （符号链接文件）
f （普通文件）

find /home/cailing/ -name "*.*" -print | xargs grep "MD5_DIGEST_LENGTH"

删除文件：
rm -rf （强制删除，不询问）

建立链接：
硬链接：只针对文件
符号链接：可针对目录
ln -s /var/ftp/public/ pub

3.3. 拷贝
scp -r 源 目的
如果目的为 windows 系统，则需先挂载 windows 下的共享文件夹，如：
mount -t cifs -o username=admin,passwd='a' //10.20.4.22/cl /usr/cl

4. 配置网络：
4.1. Fedora 系统：
4.1.1. 网络地址相关
找到这个文件 :/etc/sysconfig/network-scripts/ifcfg-eth0,
用文本编辑器打开修改
如果有 dhcp 服务动态配置就如下写：
auto eth0
iface eth0 inet dhcp
如果想使用静态 ip 地址及子网掩码就如下写：
Auto eth0
iface eth0 inet static
address 192.168.0.1
netmask 255.255.255.0
gateway 192.168.0.1
DEVICE=eth0 // 物理设备的名字
BOOTPROTO=none //表示静态配置，若为 dhcp 则为动态获得
HWADDR= // 物理地址
IPADDR=10.20.0.116
NETMASK=255.255.255.0
GATEWAY=10.20.0.7 // 网关地址
ONBOOT=yes // 该设备将在 boot 时被激活
4.1.2. 配置 NDS服务器地址
两个系统都是改这个文件： /etc/resolv.conf

在里面添加如下语句：

nameserver DNS 的 ip 地址

有几个添几个，一个单独占一行， linux 是按照从上到下的顺序查询的

编辑相应内容，保存退出，

运行 service network restart

或：

#ifconfig eth0 10.20.0.222/24 （这种方法在重启后失效）

#route add default gw 10.20.0.7

#echo nameserver 221.12.1.227

如果还不能上网，查看 /etc/nsswitch.conf(确保可用 DNS解析)

确认其中是否有此行： hosts: files dns

4.1.3. 防火墙

service iptables start （ stop ）

4.1.4. 路由相关

添加路由 ：

route add -net 10.0.0.0 netmask 255.0.0.0 dev eth0

删除路由：

route del -net 10.0.0.0 netmask 255.0.0.0 dev eth0

添加默认路由：

route add default gw 10.0.0.1

删除默认路由：

route del default gw 10.0.0.1

4.1.5. 网卡混杂模式

设置混杂模式 :ifconfig eth* promisc

取消混杂模式 :ifconfig eth* -promisc

4.1.6. 修改网卡设备名称

修改文件 /etc/udev/rules.d/70-persistent-net.rules

4.2. debian/ubuntu 系统：

4.2.1. 网络地址相关

IP 配置：

改这个文件： /etc/network/interfaces

内容同上，

保存后运行： sudo /etc/init.d/networking restart

4.2.2. 配置 NDS服务器地址

同 fedora 系统

4.2.3. 路由信息：

route -n

显示本机路由表，一般计算机路由表就两三条，去掉环回路由，只剩默认路由，一般下一条为网关。

手工增加 / 删除一条路由：

sudo route add -net 192.168.0.0 netmask 255.255.255.0 gw 172.16.0.1

跟踪路由信息：

tracert IP 地址

查询域名和 IP 地址的对应：

nslookup www.baidu.com

在网络邻居上隐藏你的计算机 （让人家看不见你！ ）：

net config server /hidden:yes

5. 串口

minicom

安装 minicom

minicom -s（进行 minicom 配置）

```
_[configuration]————¬ //配置
| Filenames and paths |//文件名和路径
| File transfer protocols|//文件传输协议
| Serial port setup |//串行端口设置
| Modem and dialing |//调制解调器和拨号
| Screen and keyboard |//屏幕和键盘
| Save setup as dfl |//设置保存到
| Save setup as.. |//储存设定为
| Exit |//退出
| Exit from Minicom |//退出minicom
└————————————┘
```

选择串行端口设置

```
|A-Serial Device(串口设备): /dev/ttyS0
|B-Lockfile Location(锁文件位置): /var/lock
|C-Callin Program(调入程序):
|D-Callout Program(调出程序):
|E-Bps/Par/Bits( ): 115200 8N1
|F-Hardware Flow Control(硬件数据流控制): No
|G-Software Flow Control(软件数据流控制): No
```

点击 A 设置串口为 /dev/ttyS0, 表示使用串口 1（com1）， /dev/ttyS1 表示串口 2

点击 E 设置 bps/par/bits(), 点 I 设置波特率为 115200，点 F 设置硬件流控制为 NO 回车

选中 Save setup as dfl ，回车，保存刚才的设置

选中 Exit 退出设置模式，刚才的设置保存到 /etc/minirc.dfl, 进入初始化模式。

启动开发板。运行 minicom 即可连接。

如果出现

Device /dev/ttyS0 lock failed: Operation not permitted

可能另一线程在用 minicom

ps auxf |grep minicom // 查看所用的线程

kill -9 进程号 // 杀死该进程

6. 服务：

/etc/init.d (可找到相应的服务，命令在 /sbin 中)

重启 apache：service httpd restart

acpi ：电源管理接口 (Advanced configuration and power Interface)

建议所有的笔记本用户开启它。一些服务器可能不需要 ACPI。

支持的通用操作有： “电源开关“， ”电池监视“， ”笔记本 Lid 开关“，“笔记本显示屏亮度“， “休眠”， “挂机”，等等。

cron, anacron, atd ：调度程序。建议开启 cron ，如果电脑将长期运行，就更应该开启它。

大都数情况下，应该关闭 atd 和 anacron 。对于服务器，应该深入了解以确定应开启哪个。

apmd: 一些笔记本和旧的硬件使用 apmd。

如果你的电脑支持 acpi ，就应该关闭 apmd。如果支持 acpi ，那么 apmd 的工作将会由 acpi 来完成

xinetd ：它是一个特殊的服务，可以根据特定端口收到的请求启动多个服务。

比如：典型的 telnet 程序连接到 23 号端口。如果有 telent 请求在 23 号端口被 xinetd 探测到，那 xinetd 将启动 telnetd 服务来响应该请求。为了使用方便，可以开启它。

运行 chkconfig －list ， 通过检查 xinetd 相关的输出可以知道有哪些服务被 xinetd 管理。

6.1. 开机启动服务

运行 ntsysv ，选中要重启时自启动的服务 (用空格键来选中或者取消选中)

用 Tab 键选择确定 (OK) 或取消 (cancel)

6.2. /etc/services 文件

它是记录网络服务名和他们对应使用的端口号及协议，文件中的每一行对应一种服务

它由 4 个字段组成，中间用 TAB或空格分隔，分别表示 —服务名称 ‖ —使用端口 ‖ —协议名称 ‖以及 —别名 ‖。

7. 查看硬盘、内存信息：

查看硬盘：

fdisk －l （查看分区）

df －hl （human-readable ， local ，查看剩余空间）

-a （查看全部文件系统）

查看内存：

cat /proc/meminfo

8. 进程：

ps －ef | grep ftp （查看 ftp 相关进程信息； |grep …，到… 找）

-u （进程所有者）

-x （显示无控制终端的进程）

-a （显示所有用户的所有进程）

ps 信息： PID TTY TIME CMD

（进程 ID）（终端名称）（进程执行时间） （命令行输入）

kill -9 [PID] (终止某进程)

./ … & (后台运行某程序)

jobs －l （查看）

9. VIM

fx ：移动光标到当前行的下一个 x 处。很明显， x 可以是任意一个字母，而且你可以使用 ； 来重复你的上一个 f 命令。

tx ：和上面的命令类似，但是是移动到 x 的左边一个位置。（这真的很有用）

Fx：和 fx 类似，不过是往回找。

w：光标往前移动一个词。
b：光标往后移动一个词。
0：移动光标到当前行首。
^：移动光标到当前行的第一个字母位置。
$：移动光标到行尾。
)：移动光标到下一个句子。
(：移动光标到上一个句子。
gg：到文件头
G：到文件尾
：n 到第 n 行
H：移动光标到屏幕上面
M：移动光标到屏幕中间
L：移动光标到屏幕下面

a：在当前字符的右边插入
o：在当前行下面插入一个新行

x：剪切当前字符到剪贴板。
dd：剪切当前行。
yy：拷贝当前行。
P：粘贴

9.1. Vimgdb

patch -d vim72 –backup -p0 < vimgdb/vim72.diff

打补丁的命令	-d		设置工作目录
	-b	（或） --backup	备份每一个原始文件
	-p		设置欲剥离几层路径名称；　后面的数字表示去掉路径的第几部分。　0，表示不去掉，为全路径

10. **Ethereal** 软件：

安装： rpm -ivh ethereal-0.99.0-1.src.rpm

命令： tethereal

可选参数： -V 、-f 、-i 、-d 、-c 、-a 、-w

如果只执行 tethereal ，那么将只抓取数据包的包头，不显示里边的内容。加上 -V 参数后，即可显示内容。

-f 参数用于过滤，默认情况下将抓取 tcp 和 udp 所有协议。

-F 指定写入文件的格式

-i 参数指定网络接口，如 lo ， eth0 等

-d 单数指定对那些端口的流量进行解码，如 -d tcp.port==8888,http 表示在 tcp 8888 端口的流量当成 HTTP协议来解码

-D 打印接口列表

-c 表示在实时捕包时要读取多少个包。

-a 指定一个规则，控制什么时候停止写 capture 文件。如 -a duration:2 表示 2 秒

-w 指定写入的文件，未指定则打印到 stdout 。默认的格式为 libcap

-r 表示读取一个保存好的 capture 文件如 -r all.capture

-q 安静，在远程时使用，否则会抓到自己 ssh 的报文

如果想抓取 UDP数据包并显示内容，则执行 tethereal -V -f udp 即可

另外还可以配合 grep 命令提取需要的关键内容

抓包：

tethereal -w filename -i eth0 -q

读包：

11. 登录相关

11.1. **root** 进行 **GUI** 登录

初次安装 Fedora 不能以 root 进行 GUI登录，设置如下：

以普通用户登录进入终端，运行 su，输入 root 密码，进入 root 账户

修改 /etc/pam.d/gdm 文件，找到如下的行：

auth required pam_succeed_ ：if.so user!=root quiet

将其注释掉（行前加 #）

修改 /etc/pam.d/gdm-password 文件中的相同的行

11.2. 远程桌面

linux 和 windows 之间的 vnc 配置过程：（以下是在 Fedora11 中）

1.在 linux 下安装 vncserver

[root@www ~]#yum search vnc-server

[root@www ~]#yum install tigervnc-server.i586

2. 运行

[root@www ~]#vncserver

出现密码提示，需要设置密码，这里设为 123456

注意： vncpasswd 可以更改这个密码，或者添加密码

设完密码后会生成一个终端编号，连接时端口号即为 5900 加桌面编号

3. 编辑文件 ：

更改为 kde 或者 gnome 等图形方式登录， Linux 上的 vnc server 内定的管理环境是 twm ，不好看，也不方便。第一次启动后 vnc 使用 twm 客户端（即只能显示个虚拟终端，可在终端里运行界面程序，但是任意时刻只能运行一个） ，这里把它改成 gnome 客户端。修改 $HOME/.vnc/xstartup

[root@localhost~]#vi /root/.vnc/xstsrtup

#unset SESSION_MANAGER

#exec /etc/X11/xinit/xinitrc

将上面两行的 #号去掉，使其生效

twm & 改为 gnome - session &

#vim /ect/sysconfig/vncservers

VNCSERVERS"1:root 10:cl "

4.重启 VNC

[root@www ~]#service vncserver restart

5.杀掉以前的连接号 :(＊＊＊＊此步骤相当重要＊＊＊＊＊)

[root@www ~]#vncserver -kill :连接号 (注：连接号第一个为 1,第二个为 2)

6 启动 vnc 连接号 ：

[root@www ~]#vncserver ： 10

7.在 windows 下运行 vncviewer,输入 ip+连接号 例如： 192.168.2.98 ： 5910

8. 如果连不上，查看是否已经关闭防火墙！ ！

Vim /etc/sysconfig/selinux

SELINUX=disabled

12. 系统相关

12.1. 主机名称

Linux ：

修改 /etc/hosts 文件，增加如下行

127.0.0.1 hostname hostname.com

修改 /etc/sysconfig/network 文件

Ubuntu ：

查看主机名称： hostname 或 uname -n

临时修改主机名： hostname 主机名

永久修改主机名：修改 /etc/hostname 文件，重启系统

12.2. 系统版本

Ubuntu ：

发行版本： cat /etc/issue

cat /etc/lsb-release

内核版本： uname -r

uname -a

12.3. 开机启动

vim /etc/rc.d/rc.local

进行添加

12.4. 环境变量

12.4.1. bash(GNU Bourne-Again Shell)

它是许多 Linux 平台的内定 Shell ，事实上还有许多传统 UNIX 上用的 Shell ，像 tcsh 、csh、ash、bsh、ksh 等等。

Shell Script 大致都类同，当您学会一种 Shell 以后，其它的 Shell 会很快就上手，大多数的时候，一个 Shell Script 通常可以在很多种 Shell 上使用。

12.4.2. 修改环境变量

有三中方法：直接用 export 命令

修改 profile 文件。例如在 /etc/profile 文件里面加一行：

export PATH="$PATH:/opt/au1200_rm/build_tools/bin"

修改 .bash.rc 文件。

12.4.3. source 命令

source FileName

作用： 在当前 bash 环境下读取并执行 FileName 中的命令。

注： 该命令通常用命令“ . ”来替代。如： source .bash_rc 与 . .bash_rc 是等效的。

source 命令与 shell scripts 的区别：

source 在当前 bash 环境下执行命令，而 scripts 是启动一个子 shell 来执行命令。这样如果把设置环境变量（或 alias 等等）的命令写进 scripts 中，就只会影响子 shell, 无法改变当前的 BASH, 所以通过文件（命令列）设置环境变量时，要用 source 命令。

12.4.4.

12.5. patch

它读取如何更改文件的源文件指示信息，然后应用这些更改。

源文件包含由 diff 命令产生的差别列表（或者 diff 列表）。

差异列表是比较两个文件和构建关于如何纠正差别的指示信息的结果。

缺省情况下， patch 命令使用从标准输入读入的源文件，但是使用 -i 标志和 PatchFile 变量可以覆盖此设置。

-p number

该标志使得无需手工编辑补丁文件就可以定制补丁文件到本地用户目录结构中。

* -p 0 导致使用完整路径名。
* -p 1 除去前导斜杠，留下 curds/whey/src/blurfl/blurfl.c 。
* -p 4 除去前导斜杠和前三个目录，留下 blurfl/blurfl.c 。

12.6.

13. configure

13.1. 宏

13.2. 参数

--help 输出帮助信息

--srcdir=DIR 源码的位置，一般不用指定，因为 configure 脚本一般和源码在同一目录下

--build=BUILD 指定软件包安装的系统平台，若没有指定，默认值是 —host 的值

--host=HOST 指定软件运行的系统平台，若没有指定，将会运行 cnfig.guess 来检测

如： –host=arm-linux 就是 arm 的交叉编译的选项

–host=mipsel-linux 就是 mipsel 的交叉编译选项

--target=GARGET 指定软件包面向的系统平台，这主要在程序语言工具如编译器汇编器上下文中起作用，如果没有指定，默认只用 —host 选项的值。

--prefix=PEWFIX 安装的位置

--disable-FEATURE 一些软件包可以选择这个选项来提供为大型选项的编译时配置，禁用某些特性

--cache-file=FILE configure 会在你的系统上测试存在的特征（或者 bug），为了加速随后进行的配置，测试的结果会存储在一个 cache file 里

--enable-FEATURE[=ARG] 跟上述上一条相反，启用默认被禁止的特性

如： –enable-buffers=128 ；

–enable-FEATURE=no 与上面提到的 '–disable-FEATURE' 是同义的

--with-PACKET[=ARG] 配置源码树时提供其他已安装软件包的信息

如： –with-tcl=/usr/local

–with-PACKAGE=no 与下面的 — without-PACKAGE 同义

交叉编译时有很多测试程序是不可以在 host 上运行的，就会出现错误：

error: cannot run test program while cross compiling

这类错误可以使用 cachefile 解决：

记下错误的地方，查找类似的结构，变量使用如：

echo ac_cv_have_abstract_sockets=yes>arm-linux.cache

添加 —cache-file=arm-linux.cache 。

14. Makefile

14.1. all

构造一个没有规则的终极目标

all ： target1 target2 …

target1: 依赖关系

规则。。。

target2: 依赖关系

规则。。。

14.2. install

14.3. clean

删除没用的中间文件

14.4. 规则

A：B
(tab)<command>　　// 每个命令行前都必须有　tab 符号。
g++编译 cpp 文件

14.5. 变量

$@:目标文件
$^: 所有的依赖文件
$<: 第一个依赖文件
$@:是 Makefile　预先定义的一个变量，表示目标命令，如：　　all ，install　，clean

14.6. gcc

-o ：后面的参数表示要输出的目标文件
-c ：表示仅编译，不连接，没有则表示仅连接
-l ：后面表示连接时所要的链接库

15. FTP

15.1. tftp　（fedora　中）

安装 tftp　：

yum search tftp
yum install ***　　安装 tftp　服务器或客户端

tftp　服务器：

启动：执行　ntsysv ，选中 tftp　服务，再执行　service xinetd restart　　命令。
配置：配置文件是　/etc/xinetd.d/tftp,　　如下所示

```
service tftp
{
disable = no
socket_type = dgram
protocol = udp
wait = yes
user = root
server = /usr/sbin/in.tftpd
server_args = -s /tftpboot
per_source = 11
cps = 100 2
}
```

其中 server_args=-s /tftpboot　　是 tftp　服务器运行时的参数。表示默认目录为　　/tftpboot

16. 内核相关

16.1. 信号量（ semaphore ）

本质上市一个整数值

和一对 P/V 函数联合使用

用于互斥时，应初始化为 1，任何时刻只能由单个进程或线程拥有。

此时也称为互斥体（ mutex ）mutual exclusion 。

内核中几乎所有的信号量均用于互斥。

使用信号量需包含 asm/semaphore.h 头文件

初始化： void sema_init(struct semaphore *sem, int val);

互斥模式下：

DECLARE_MUTEX(name) // 初始化为 1

DECLARE_MUTEX_LOCKED(name) // 带有 _LOCKED 是将信号量初始化为 0，即锁定

P 函数：

```
void down(struct semaphore *sem); /*          不推荐，会建立不可杀进程    */
int down_interruptible(struct semaphore *sem); /*          推荐，若操作被中断，返回非零值    */
int down_trylock(struct semaphore *sem); /*          带有 _trylock 的永不休眠，若不可获得，
                                                    返回非零值 */
```

V 函数：

```
void up(struct semaphore *sem); /*          释放信号量 */
```

读取者 / 写入者信号量 rwsem：

必须包含 linux/rwsem.h 头文件

只读任务可并行完成它们的工作而不需要等待其他读取者推出临界区。