2017、07、18

 1：Alt+F1 组合键： 打开菜单

2：Ctrl+Alt+[F1~F6] : 可以切换到1~6号控制台

3：Ctrl+Alt+F7 : 可以返回图形界面

4：sudo halt: 退出命令（halt：停止）；

5：路径分为绝对路径和相对路径

绝对路径：绝对路径的起始点为根目录 / ，例如 /usr/local/bin 就是绝对路径，它指向系统中一个绝对的位置，不受其它因素影响。

相对路径：相对路径的起始点为当前目录，如果现在位于 /usr 目录，那么相对路径 local/bin 所指示的位置为 /usr/local/bin

（注： ubuntu绝对路径：例如在目录/home/lzj/temp下有一个文件filename.txt那么filename.txt文件的绝对路径就是/home/lzj/temp

Ubuntu相对路径：如果你当前路径是在/home/lzj下,那么filename.txt文件的相对路径就是./temp

ubuntu中更目录”/“；当前目录“./”；当前目录的上一级目录(如果有上一级目录的话)“../”

个人总结：所谓绝对路径，就是指不受当前位置影响的路径；而相对路径，则是以当前路径为基点/相对点开始算起，而当前目录用./表示，当前目录的上一目录用../表示）

6：路径相关命令

cd (change directory) 更改目录。

pwd (print working directory)显示当前路径。

ls (list) 显示当前目录中的文件列表。

请尝试以下操作：

cd /etc 进入“/etc”目录，这里使用的是绝对路径

pwd 显示当前路径，这个命令返回结果“/etc”

cd init.d 进入“/etc”目录的子目录“init.d”，这里使用的是相对路径

cd .. 进入上一级目录“/etc”

cd ../home “/etc” 目录的上一级目录为“/”，它的子目录“home”为“/home”

cd - 回到上一次的目录，我们在“/etc”目录跳转到“/home”目录，所以这次是回到“/etc”目录

cd ~ “~”代表当前用户的“$HOME”目录，也就是“/home/{用户名}”目录。

ls 在任何时候，您都可以使用“ls”命令，来了解当前目录下都有哪些文件。

### 7：[Ubuntu下使用sshfs挂载远程目录到本地](http://blog.csdn.net/netwalk/article/details/12952719)

**8：**软件

Linux中没有**注册表**这个概念。安装软件，理论上讲，只要拷贝所有相关文件，并运行它的主程序就可以了。

按照传统，一个软件通常分别拷贝到同级目录下的 bin、etc、lib、share等文件夹。

**Bin**            可执行文件，程序的可执行文件通常在这个目录下。在环境变量中设定搜索路径，就可以直接执行，而不需要定位其路径。

**Etc**配置文件，大部分系统程序的配置文件保存于 /etc 目录，便于集中修改。

**Lib**库文件，集中在一起，方便共享给不同程序。相较不同的软件单独保存库文件，能够节约一些磁盘空间。

**Share**程序运行所需要的其它资源，例如图标、文本。这部分文件是专有的，不需要共享；而且目录结构相对复杂，混放在一起比较混乱，所以单独存放。

9：隐藏文件

Linux下，名称中第一个字符为 . 的文件或者文件夹，系统默认情况下将它们隐藏起来

10：cd ~ 进入您的用户目录

ls 查看当前目录下的文件列表

ls -a 查看所有文件的文件列表（包括隐藏文件） （注：ls和-a之间有空格）

ls –A：只查看隐藏的文件

11：ls –l：查看详细信息格式的文件列表，例如：

drwxr-xr-x 2 jianghu jianghu 4096 Jul 17 20:05 Desktop

（权限） （文件数）（归属用户）（归属群组）（文件大小）（创建日期）（文件名称）

关于权限行：

d rwx r-x r-x

### (第一组1) （第二组3） （第三组3） （第四组3）

### 第一组的格式： d：文件夹

### -：普通文件

### |：链接

### b：块设备文件

### c：字符设备文件

### 第二组、第三组、第四组：分别表示归属用户、归属群组、其他用户或群组对于该文件的权限。

### rwx rwx rwx

### 其中，r：可读 w：可写 x：可执行

### 每一组存在几种字符，就表示该用户可以对该文件可以执行的权限范围。例如：rwx，就表示可以同时进行读、写、执行三种操作；r-x则表示只能进行读和执行，但是不能进行写入的操作；而-wx则表示可以进行写和执行的操作，但是不能进行读取。

它们的顺便不能颠倒，某一位置为空(-)，则表示不具有相应的权限。

说明：Linux下的可执行文件并不是由扩展名（例如 .exe ）决定的，而是由其可执行权限位决定。

12： chmod：更改文件的权限；

chown：更改文件的归属；

13：当命令输入提示符末尾是$时：表示将以普通用户的身份执行命令。

su：（switch user）切换用户到其他用户。例如：

su root：切换到root用户

如果su命令后面没有切换目标，那么表示默认切换到root用户。

14：sudo apt-get install rxvt-unicode：安装rxvt终端

urxvt：启动rxvt终端；

**urxvtd ：启动一个守护进程daemon（支持控制台）**



15：对rxvt进行简单的设置，使其更好看：

16：Ctrl+a（head）：使光标移动到行首；

Ctrl+e（nd）：光标移到行末；

17：wget加空格加音乐地址链接：下载音乐；

18：Ctrl+c ：强行中止

Ctrl+s：冻结shell，可以使用Ctrl+q尝试恢复正常；

19：移动命令：

\C-a 移动到行首 Aheah               \C-e 移动到行末 End                   \C-f 向前移动一个字符 Forward

\C-b 向后移动一个字符 Backward               \M-f 向前移动一个单词               \M-b 向后移动一个单词

\C-l 清空屏幕 cLear           这两个命令也可以理解为移动命令               \C-p 上翻，前一条命令 Previous

\C-n 下翻，后一条命令 Next               编辑命令：

\C-d 删除光标后的一个字符 \M-d 删除光标后的一个单词 Delete

\BackSpace 删除光标前的一个字符 \M-BackSpace 删除光标前的一个单词

\C-k 删除光标至行末的部分 Kill                  \C-u 删除光标至行首的部分 Unix-line-discard

\C-w 删除光标前的一个单词 Word              \C-y 粘贴（最后删除的对象） Yank            \C-- 撤消

搜索历史纪录：

\C-r 连续使用 ``C-r`` 可以查找下一个                \M-p           \M-n

补全：

\Tab 使用频率最高的功能！                \C-o 遍历补全 （未定义）

\M-? M-= 列出所有可能选项，相当于按两次Tab键（M-？ 实际按键为\A+\S+/）

\M-# 注释掉当前命令，用于将当前命令暂存于历史纪录列表（\A+\S+3）

\M-! 补全命令，通常用来补全子命令，例如 ``sudo`` 的子命令（\A+\S+1）

\M-~ 补全用户名（\A+\S+`）              \M-@ 补全主机名（\A+\S+2）

\M-$ 补全变量（\A+\S+4）                  \M-\_ 补全历史纪录中的纪录（\A+\S+-）

\M-\* 将所有可能选项放到命令行中（\A+\S+8）

20：通配符

使用 ? 代表任意单个字符。例如 ???lo ，表示 lo 前有三个字符，它可以匹配 Hello

使用 \* 代表随意几个任意字符。例如 \*.iso ，代表所有iso格式的文件。

说明：您可以将遍历补全和通配符结合使用，以提高效率。

例如：

cd \*/ 则遍历补全只补全文件夹

chmview \*.chm 则遍历补全只补全chm文件

21：任务管理

&--------------------------------在命令的末尾加上一个 & 符号，表示背景任务，例如：

wget http://www.download.net/xxx/mp3 &

;----------------------------------使用 ; 将多个命令连结起来，则表示任务按顺序执行

&&------------------------------使用 && 将多个命令连结起来，则表示只有前面的命令执行成功，后面的命令才能得以执行

``-----------------------------------`<命令>` ，如果一个命令中包含以 `` （Esc键下方的按键）括起来的子命令，那么子命令将被优先执行，执行结果被代入上一级命令继续执行

22：Ctrl+z

将当前Shell中的任务挂起

这个时候任务的状态为

[1]+ Stopped xxx

Bg-------------------------------------------------将挂起的任务背景运行。这时它的状态为[1]+ xxx &

Fg-------------------------------------------------将背景任务调到前台执行jobs

方括号中的数字为命令的任务编号，您可以使用 jobs 命令来查看所有背景任务

如果后台运行多个任务，您可以在 bg 或者 fg 后跟任务编号，作为操作对象，例如：bg 2

23：管道、重定向

>-----------------------重定向符号，它的作用是将命令的输出结果重定向到一个文件中。比如我们想把命令 ls 的结果保存为 FileList 文件，作一个清单，我们可以使用重定向符号来完成它：

ls -l > FileList

（注：ls(空格)-l(空格)>(空格)filelist）在Shell中，无论多少个空格，都将被解析为一个分隔符

>>----------------------作用与 > 基本相同，不同点在于， >> 以追加的方式，将命令的输出写入文件的末尾。

<-----------------------是从文件到命令的重定向，将文件的内容作为命令的输入。

|------------------------为管道符号，它的作用是将前一个命令的输出，作为下一个命令的输入。假设一个目录下的文件太多，使用 ls命令不能够在屏幕中完全显示，这个时候您可以将 ls 命令的输出，通过管道符号，作为浏览器 less 的输入。就可以使用浏览器的功能翻页、查找：ls -al | less

24：

脱字符 \

它能够将一个具有特殊涵义的字符转换普通字符

Shell中的一些功能是通过特殊符号作为控制字符来实现的

这产生一个问题，如果一个文件名中，刚好包含了这些字符，比如 ; ，就很难对它进行操作。使用 less 浏览这个文件

less ;xxx

less 会很快返回一个错误信息，因为并没有一个文件名作为操作对象。接着，Shell会报告，系统中没有 xxx 这个命令。

这是因为Shell将文件名中的 ; 解析为按顺序执行命令。

上面的两个任务，可以在文件名中每个特殊字符前加一个 \ ，像这样

less \;xxx

less \ \xxx

less \;\ \&\xxx

说明：也可以使用 " 将文件名括起来，例如 less "; &xxx" ，在很多情况下，这样甚至更方便。

脱字符在Shell中也可以作为换行符，在一个命令的末尾添加一个 \ ，然后回车，在下一行继续输入命令剩余的部分，将一个命令拆分为多行且不影响它的执行（如果执行一个很长的命令，请将它拆分为多行以便于阅读）

事实上换行符也符合脱字符的定义。回车键有两个涵义，一个是 执行 （Enter），另一个 换行 （折线箭头）。在Shell中它作为控制字符 执行 ，使用脱字符后，它便代表排版字符 换行 了。

24：另一种shell“fish”：具有强大的语法亮点；可以考虑学习一下，以作为环境调节备用；

25：dirh （dir history）就可以显示当前会话中进入的文件夹纪录

使用 prevd 和 nextd 跳转

假如曾进入过1 2 3 4 5 这几个文件夹， prevd 4 可以让你在 5 中直接跳到 1

26：whereis xxx 命令，来查找 xxx 程序的安装位置，详见搜索；

27：定命令的搜索路径

使用 echo $PATH ，可以显示 $PATH 变量，输出如下：

/usr/local/sbin /usr/local/bin /usr/sbin /usr/bin /sbin /bin /usr/bin/X11 /usr/games /usr/X11R6/bin

它是一个环境变量，代表执行命令时，Shell的搜索路径。

执行一个命令时，Shell会到 $PATH 变量定义的路径去搜索，并运行与命令同名的可执行文件。如果程序、脚本等可执行文件并不在上面的路径中，就必须使用绝对路径或者相对路径定位可执行文件。

28：

**Ubuntu系统简介**

Ubuntu系统目录结构

以下为Ubuntu目录的主要目录结构，

/ 根目录

│

├boot/ 启动文件。所有与系统启动有关的文件都保存在这里

│ └grub/ Grub引导器相关的文件

│

├dev/ 设备文件

├proc/ 内核与进程镜像

│

├mnt/ 临时挂载

├media/ 挂载媒体设备

│

├root/ root用户的$HOME目录

├home/

│ ├user/ 普通用户的$HOME目录

│ └.../

│

├bin/ 系统程序

├sbin/ 管理员系统程序

├lib/ 系统程序库文件

├etc/ 系统程序和大部分应用程序的全局配置文件

│ ├init.d/ SystemV风格的启动脚本

│ ├rcX.d/ 启动脚本的链接，定义运行级别

│ ├network/ 网络配置文件

│ ├X11/ 图形界面配置文件

├usr/

│ ├bin/ 应用程序

│ ├sbin/ 管理员应用程序

│ ├lib/ 应用程序库文件

│ ├share/ 应用程序资源文件

│ ├src/ 应用程序源代码

│ ├local/

│ │ ├soft/ 用户程序

│ │ └.../ 通常使用单独文件夹

│ ├X11R6/ 图形界面系统

│

├var/ 动态数据

│

├temp/ 临时文件

├lost+found/ 磁盘修复文件

29：

启动流程

Linux系统主要通过以下步骤启动：

1.读取MBR的信息，启动Boot Manager

Windows使用NTLDR作为Boot

Manager，如果您的系统中安装多个版本的Windows，您就需要在NTLDR中选择您要进入的系统。

Linux通常使用功能强大，配置灵活的GRUB作为Boot Manager，我们将在启动管理章节中向您介绍它的使用方式。

2.加载系统内核，启动init进程

init进程是Linux的根进程，所有的系统进程都是它的子进程。

3.init进程读取 /etc/inittab 文件中的信息，并进入预设的运行级别，按顺序运行该运行级别对应文件夹下的脚本。脚本通常以 start 参数启动，并指向一个系统中的程序。

通常情况下， /etc/rcS.d/ 目录下的启动脚本首先被执行，然后是 /etc/rcN.d/ 目录。例如您设定的运行级别为3,那么它对应的启动目录为 /etc/rc3.d/ 。

4.根据 /etc/rcS.d/ 文件夹中对应的脚本启动Xwindow服务器 xorg

Xwindow为Linux下的图形用户界面系统。

5.启动登录管理器，等待用户登录

Ubuntu系统默认使用GDM作为登录管理器，您在登录管理器界面中输入用户名和密码后，便可以登录系统。（您可以在 /etc/rc3.d/ 文件夹中找到一个名为 S13gdm 的链接）

更改运行级别

在 /etc/inittab 文件中找到如下内容：

# The default runlevel.

id:2:initdefault:

这一行中的数字 2 ,为系统的运行级别，默认的运行级别涵义如下：

0 关机       1 单用户维护模式         2~5 多用户模式         6 重启

### Ctrl+Alt+t：快速打开terminal终端窗口；

### touch main.c ：在当前目录下创建一个名称为main的c文件；

### gcc main.c ：编译名称为main的c文件，如果没有错误或者警告的信息打印出来，则说明编译通过，并且在同样的目录下会生成一个.out文件；然后./a.out就会运行该文件；

### gcc main.c –o zjh ：用来指定生成的可执行的程序的文件名；（在该实例中，将源文件main.c生成的可执行文件的名称指定为zjh）

### gcc main.c –lm –o zjh ：如果程序中用到了不是gcc默认的库函数的时候，比如程序中用到了math.h这个头文件。这个头文件不是gcc默认的头文件，所以在编译的时候要加入-l库的首字母来进行编译。比如这里是：gcc hello.c -lm -o zjh；

### gcc -c hello.c -o hello.o ：将为.c 的源程序编译成.o文件，该命令要用到-c参数。

### rm 文件名 ：删除某个文件；（remove）

### rm -r 文件夹名 ：删除某个文件夹；

### sudo vim /etc/vim/vimrc：修改vim编辑器的相关配置；

### 30：修改root密码：

### sudo passwd root：输入新的密码

### su root：获取用户权限

### 

### 31：关于在Ubuntu和Windows之间设置共享文件夹：

### Windows和虚拟机Ubuntu之间可以通过共享文件夹的方式来进行文件的互换操作。通常是在Windows下创建一个共享文件夹，然后在Ubuntu上对该文件夹进行挂载然后再在虚拟机上同样创建一个不同名的文件夹，并将该文件夹和Windows下的文件夹进行链接，这样在Windows下的文件夹上的操作可以通过挂载来同步到Ubuntu上，反过来Ubuntu上的对该文件的修改也可以直接同步到Windows下。

### 第一步：Windows下创建文件夹

### 第二步：Ubuntu上添加该共享文件夹

### 第三步：Ubuntu上创建不同命的文件夹

### 第四步：将二者通过mount进行链接，使其具有共享同步功能

### 第五步：下次打开虚拟机之后进行命令挂载便可正常使用

### 教程如下：<http://blog.csdn.net/skylake_/article/details/53132499>

### 补充与更正：设置共享文件夹的时候，在Windows下的文件夹命名可以和Ubuntu上的共享文件夹名称一致，在Ubuntu上进行将Windows下的文件夹添加上去的时候，此时的名称也可以和Ubuntu上的一致，而稍后在Ubuntu上通过命令挂载文件夹的时候，所用到的两个名称就是此时的名称和Ubuntu上的名称：

### 

### 例如：

### 在Windows下创建：A

### 在Ubuntu上添加文件夹时设置共享文件夹名称：a

### 在Ubuntu上创建文件夹：share

### 但是在挂载时：

### mount –t vboxsf a /mnt/share /\*将添加进去的文件夹与Ubuntu上的文件夹进行共享链接\*/

### 总结：命名无所谓，可以全部相同，也可以全部不同，但是作为两个环境下搭建共享桥梁的文件夹添加模块，一定要把路径添加正确，另外就是在命令挂载时要分得清哪个是Windows下的，哪个是Ubuntu下的。

### 每一个共享文件夹的挂载都需要用到各自的命令行，当有多个共享文件夹时，可以考虑使用脚本来将所有的挂载一次性完成：

### 

### 这样当需要挂载时，只需要运行该脚本即可：

### 

### 挂载成功：

### 

### 32：关于.sh脚本用法。

### 编写.sh脚本（vim test.sh）：

### 

### 之后并不能马上运行，还需要使其可执行，即命令行：chmod +x test.sh

### 

### 

### 然后才可以运行该脚本：

### 

### 33：进入控制台：alt + ctrl + Fn(n = 1~6);

### 进入有图形界面的控制台（即terminal）：alt + ctrl + F7；

### 两者是一样的，但是有图形界面的控制台更占用内存和CPU。

### 34：进入root用户：su root；

### 退出root用户：exit；logout；ctrl + D;

### 35：关于vim使用技巧（长期更新）。

### 跳到指定行：

### 编辑模式下：n+gg(n：行数)；

### n+G(n：行数)；

### 命令模式下：n+回车；

### 跳到尾行：

### 命令模式下：shift+g；

### 跳到行尾：命令模式下：$

### 36：关于shell脚本语言。

### 01：在shell中，所有的变量都以字符串的形式保存，当没有空格存在时，不在乎是否加“”，但是当有空格时，必须要用“”：

### 

### 

### 02：赋值号（即=）周围不能有空格，否则就是语法错误。

### 03：单双引号的区别：

### :

### 37：关于gcc的两个参数（以add.c sum.c main.c为例，其中main需要链接add和sum）：-c 和 –o

### 单独将某一个.c文件不进行链接，只是简单的编译成目标文件.o：

### (如果不需要对该目标文件进行特殊命名)：gcc –c add.c

### （如果需要特殊命名）：gcc –c XXX add.c

### 当生成各自的目标文件以后需要链接生成最终的可执行文件：-o

### gcc –o main.o add.o sum.o

### 如果需要在编译的同时进行链接直接生成最终可执行文件，那么：

### gcc -o main main.c add.c sum.c

### 38：在同一目录下写的.h文件，在相应的.c文件中包含时，必须是#include””,否则会报错找不到该头文件。因为<>是在根目录下寻找，而””是在当前目录下寻找。

### 38：进程与线程的区别

### 39：make与makefile的应用

### 40：线程的概念（pthread库）