

## unix 常用命令

a. 關於檔案 / 目錄處理的指令：

### 1. ls

這是最基本的檔案指令。ls 的意義為 "list"，也就是將某一個目錄或是某一個檔案的內容顯示出來。

如果你在下 ls 指令後頭沒有跟著任何的檔名，它將會顯示出目前目錄中所有檔案。

也可以在 ls 後面加上所要察看的目錄名稱或檔案的名稱，如

```
% ls /home2/X11R5
```

```
% ls first
```

ls 有一些特別的參數，可以給予使用者更多有關的資訊，如下：

-a：在 UNIX 中若一個目錄或檔案名字的第一個字元為 "."，則使用 ls 將不會顯示出這個檔案的名字，我們稱此類檔案為隱藏檔。如 tcsh 的初設檔 .tcshrc；如果我們要察看這類檔案，則必須加上參數 -a。

-l：這個參數代表使用 ls 的長 ( long ) 格式，可以顯示更多的資訊，如檔案存取權，檔案擁有者 ( owner )，檔案大小，檔案最後更新日期，甚而 symbolic link 的檔案是 link 那一個檔等等。如下

```
% ls -l
```

```
drwx--x--x 2 jjtseng 512 Aug 8 05:08 18
drwx--x--x 2 jjtseng 512 Aug 8 22:00 19
-rw----- 1 jjtseng 566 Aug 8 05:28 makefile
```

### 2. cp

cp 這個指令的意義是複製 ("COPY")，也就是將一個或多個檔案複製成另一個檔案或者是將其複製到另一個目錄去。

cp 的用法如下：

cp f1 f2：將檔名為 f1 的檔案複製一份為檔名為 f2 的檔案。

cp f1 f2 f3 ... dir：將檔案 f1 f2 f3 ... 都以相同的檔名複製一份放到目錄 dir 裡面。

cp -r dir1 dir2：將 dir1 的全部內容全部複製到 dir2 裡面。

cp 也有一些参数，如下：

-i：此参数是当已有档名为 f2 的档案时，若逐自使用 cp 将会将原来 f2 的内容掩盖过去，因此在要盖过之前必须先询问使用者一下。如使用者的回答是 y(yes)才执行复制的动作。

-r：此参数是用来做递归复制用，可将一整颗子树都复制到另一个目录中。

### 3. mv

mv 的意义为 move，主要是将一档案改名或换至另一个目录。如同 cp，它也有三种格式：

mv f1 f2：将档名为 f1 的档案变更成档名为 f2 的档案。

mv dir1 dir2：将档名为 dir1 的目录变更成档名为 dir2 的目录。

mv f1 f2 f3 ... dir：将档案 f1 f2 f3 ... 都移至目录 dir 里面。

mv 的参数有两个，-f 和 -i，其中 -i 的意义与 cp 中的相同，均是 interactive 询问之意。而 -f 为强迫 ( force )，就是不管有没有同名的档案，反正我就是要搬过去，所有其他的参数遇到 -f 均会失效。

### 4. rm

rm 的意义是 remove，也就是用来杀掉一个档案的指令。在 UNIX 中一个被杀掉的档案除非是系统恰好有做备份，否则是无法像 DOS 里面一样还能够救回来的。所以在做 rm 动作的时候使用者应该要特别小心。

rm 的格式如下：

rm f1 f2 f3 .....

而 rm 的参数比较常用的有几个：-f，-i，与 -r

-f：将会使得系统在删除时，不提出任何警告讯息。

-i：在除去档案之前均会询问是否真要除去。

-r：递归式的删除。

小心不要随便使用 rm -rf，否则有一天你会“欲哭无泪”.....

### 5. mkdir

mkdir 是一个让使用者建立一个目录的指令。你可以在一个目录底下使用 mkdir 建立一个子目录，使用的方法如下：

mkdir dirname1 [ dirname2 ... ]

如此你就可以建立一个或多个目录。

#### 6. chdir ( cd )

这是让使用者用来转移工作目录用的。

chdir 的用法如下：

```
chdir dirname
```

如此你就可以将目前的目录转移到 `dirname` 这一个目录去。

或使用 "`chdir ..`" 来转移到上一层目录。

#### 7. mdir

相对於 `mkdir` , `mdir` 是用来将一个 "空的" 目录杀掉的。如果一个目录下面没有任何档案, 你就可以用 `mdir` 指令将其除去。 `mdir` 的用法如下：

```
mdir dirname1 [ dirname2 .... ]
```

如果一个目录底下有其他的档案, `mdir` 将无法将这个目录杀掉, 除非使用 `rm` 指令的 `-r` 选项。

#### 8. pwd

`pwd` 会将目前目录的路径 ( `path` ) 显示出来, 例如：

#### 9. cat/more/less

以上三个指令均为察看档案内容的指令。 `cat` 的意义是猫 ... 不不不, 是 concatenate , 在字典上的意思是 "连结, 将...串成锁状" ( 语出资工电子词典 `cdict` ), 其实就是把档案的内容显示出来的意思。 `cat` 有许多奇怪的参数, 较常为人所使用的是 `-n` 参数, 也就是把显示出来的内容加上行号。 `cat` 的用法如下：

`cat [-n]` : 自标准输入读进内容, 你可以用 `pipe` 将别的程式的输出转向给 `cat` .

`cat [-n] filename` : 将 `filename` 的内容读进来, 显示在标准输出上。

问题在於 `cat` 它是不会停下来的, 因此并不好用 ( 试想如果一个萤幕二十四行, 而一个档案四百行, `cat` 一出来将会劈哩啪啦不断的卷上去, 使用者很难据此得到他们所需的资讯。 ) 所以才有人又写了 `more` 出来。

`more` , 跟据蔡文能老师的说法是 "再多一点就好", `more` 可以将所观察的档案跟据终端机的形态一页页的显示出来, 再根据使用者的要求换页或卷行。如果使用者要在某一个档案中搜寻一个特定的字串, 则按 `/` 然後跟著打所要搜寻的单字即可进行搜寻。 `more` 也可以找得到。 `more` 的用法如下：

```
more filename
```

如果你在使用中觉得已经看到了所要看的部份，可以按 'q' 离开 more 的使用。  
在使用中按 'v' 亦可以使用编辑器来编辑所观看的档案。

less 的用法与 more 极类似，原先它就是为了弥补 more 只能往前方卷页的缺点而设计。less 的用法如下：

```
less filename
```

其与 more 不同的是它可以按 y 来往上卷一行，并且可以用 "?" 来来回搜寻你所要找的单字。

#### 10. chmod

chmod 为变更档案模式用 ( change mode )。这个指令是用来更改档案的存取模式 ( access mode )。在 UNIX 一个档案上有可读 (r) 可写 (w) 可执行 (x) 三种模式，分别针对该档案的拥有者 ( owner )、同群者 ( group member ) ( 你可以 ls -lg 来观看某一档案的所属的 group )，以及其他的人 ( other )。一个档案如果改成可执行模式则系统就将其视为一个可执行档，而一个目录的可执行模式代表使用者有进入该目录之权利。chmod 就是用来变更一些档案的模式，其使用方式如下：

```
chmod [ -fR ] mode filename ...
```

其参数的意义如下：

- f Force. chmod 不会理会失败的动作。
- R Recurive. 会将所有子树下的所有子目录及档案改为你所要改成的模式。

mode 可以为一个三位或四位的八进制数字，来表示对某些对象的存取权。详情可参阅 chmod(1) 的 manual page 中有关 Absolute Modes 的说明。

或是用一个字串来表示，请参考 chmod(1) 的说明。

b. 关于 Process 处理的指令：

#### 1. ps

ps 是用来显示目前你的 process 或系统 processes 的状况。

以下列出比较常用的参数：

其选项说明如下：

- a 列出包括其他 users 的 process 状况。
- u 显示 user - oriented 的 process 状况。
- x 显示包括没有 terminal 控制的 process 状况。
- w 使用较宽的显示模式来显示 process 状况。

我们可以经由 ps 取得目前 processes 的状况，如 pid，running state 等。

## 2. kill

kill 指令的用途是送一个 signal 给某一个 process。因为大部份送的都是用来杀掉 process 的 SIGKILL 或 SIGUP，因此称为 kill。kill 的用法为：

```
kill [ -SIGNAL ] pid ...
```

```
kill -l
```

SIGNAL 为一个 signal 的数字，从 0 到 31，其中 9 是 SIGKILL，也就是一般用来杀掉一些无法正常 terminate 的讯号。其余讯号的用途可参考 sigvec(2) 中对 signal 的说明。

你也可以用 kill -l 来察看可代替 signal 号码的数目字。kill 的详细情形请参阅 man kill。

## c. 关于字符串处理的指令：

### 1. echo

echo 是用来显示一串字符串在终端机上。echo -n 则是当显示完之后不会有跳行的动作。

### 2. grep/fgrep

grep 为一过滤器，它可自一个或多个档案中过滤出具有某个字符串的行，或是自标准输入过滤出具有某个字符串的行。

fgrep 可将欲过滤的一群字符串放在某一个档案中，然后使用 fgrep 将包含有属于这一群字符串的行过滤出来。

grep 与 fgrep 的用法如下：

```
grep [-nv] match_pattern file1 file2 ....
```

```
fgrep [-nv] -f pattern_file file1 file2 ....
```

-n 把所找到的行在行前加上行号列出

-v 把不包含 match\_pattern 的行列出

match\_pattern 所要搜寻的字符串

-f 以 pattern\_file 存放所要搜寻的字符串

## d. 网络上查询状况的指令：

### 1. man

man 是手册 ( manual ) 的意思。UNIX 提供线上辅助 ( on-line help ) 的功能，man 就是用来让使用者在使用时查询指令、系统呼叫、标准程式库函数、各种表

格等的使用所用的。man 的用法如下：

```
man [-M path] [[section] title] .....  
man [-M path] -k keyword ...
```

-M path man 所需要的 manual database 的路径。  
我们也可以用设定环境变量 MANPATH 的方式来取代 -M 选项。  
title 这是所要查询的目的物。  
section 为一个数字表示 manual 的分类，通常 1 代表可执行指令，  
2 代表系统呼叫 ( system call )，3 代表标准函数，等等。

像下面 man 查询的片段：

```
SEE ALSO  
apropos(1), cat(1V), col(1V), eqn(1), lpr(1), more(1),  
nroff(1), refer(1), tbl(1), troff(1), vgrind(1), vtroff(1),  
whatis(1), eqnchar(7), man(7), catman(8)
```

我们如要参考 eqnchar(7) 的资料，则我们就输入 man 7 eqnchar，  
便能取得我们所要的辅助讯息。

```
-k keyword
```

用来将含有这项 keyword 的 title 列出来。

man 在 UNIX 上是一项非常重要的指令，我们在本讲义中所述之用法均仅只是一个大家比较常用的用法以及简单的说明，真正详细的用法与说明还是要请你使用 man 来得到。

## 2. who

who 指令是用来查询目前有那些人在线上。

## 3. w

w 指令是用来查询目前有那些人在线上，同时显示出那些人目前的工作。

## 4. ku

ku 可以用来搜寻整个网路上的 user，不像 w 跟 who 只是针对 local host 的查询。而且 ku 提供让使用者建立搜寻特定使用者名单的功能。你可以建立一个档案 information-file 以条列的方式存放你的朋友资料，再建立一个档案 hosts-file 来指定搜寻的机器名称。ku 的指令格式可由 ku -h 得到。

## e. 网路指令：

UNIX 提供网路的连接，使得你可以在各个不同的机器上做一些特殊的事情，如你可以在系上的 iris 图形工作站上做图形的处理，在系上的 Sun 上读 News，甚至到学校的计中去找别系的同学 talk。这些工作可以利用 UNIX 的网路指

令，在你的位子上连到各个不同的机器上工作。如此一来，即使你在寝室，也能轻易的连至系上或计中来工作，不用像以前的人必须泡在冷冰冰的机房面。

这些网路的指令如下所述：

#### 1. rlogin 与 rsh

rlogin 的意义是 remote login，也就是经由网路到另外一部机器 login。

rlogin 的格式是：

```
rlogin host [ -l username ]
```

选项 -l username 是当你在远方的机器上的 username 和 local host 不同的时候，必须输入的选项，否则 rlogin 将会假设你在那边的 username 与 local host 相同，然后在第一次 login 时必然会发生错误。

rsh 是在远方的机器上执行某些指令，而把结果传回 local host。rsh 的格式如下：

```
rsh host [ -l username ] [ command ]
```

如同 rlogin 的参数 -l username，rsh 的 -l username 也是指定 remote host 的 username。而 command 则是要在 remote host 上执行的指令。如果没有指定 command，则 rsh 会去执行 rlogin，如同直接执行 rlogin。

不过 rsh 在执行的时候并不会像一般的 login 程序一样还会问你 password，而是如果你没有设定 trust table，则 remote host 将不会接受你的 request。

rsh 须要在每个可能会做为 remote host 的机器上设定一个档案，称为 .rhosts。这个档案每一行分为两个部份，第一个是允许 login 的 hostname，第二个部份则是允许 login 的 username。例如，在 ccsun7.csie.nctu.edu.tw 上头你的 username 为 ysjuang，而你的 home 下面的 .rhost 有以下的一行：

```
ccsun6.cc.nctu.edu.tw u8217529
```

则在 ccsun6.cc.nctu.edu.tw 机器上的 user u8217529 就可以用以下的方法来执行 rsh 程式：

```
% rsh ccsun7.csie.nctu.edu.tw -l ysjuang cat mbox
```

将 ysjuang 在 ccsun7.csie.nctu.edu.tw 上的 mbox 档案内容显示在 local host ccsun6.cc.nctu.edu.tw 上。

而如果 .rhost 有这样的一行，则 ccsun6.cc.nctu.edu.tw 上的 user u8217529 将可以不用输入 password 而直接经由 rsh 或 rlogin login 到

ccsun7.csie.nctu.edu.tw 来。

注意：

.rhost 是一个设定可以信任的人 login 的表格，因此如果设定不当将会让不法之徒有可以乘机侵入系统的机会。如果你阅读 man 5 rhosts，将会发现你可以在第一栏用 + 来取代任何 hostname，第二栏用 + 来取代任何 username。

如一般 user 喜欢偷懒利用 " + username " 来代替列一长串 hostname，但是这样将会使得即使有一台 PC 上跑 UNIX 的 user 有与你相同的 username，也可以得到你的 trust 而侵入你的系统。这样容易造成系统安全上的危险。因此本系禁止使用这样子的方式写你的 .rhost 档，如果发现将予以停机直到你找中心的工作人员将其改正为止。同理，如果你的第二个栏位为 +，如 " hostname + " 则你是允许在某一部机器上的 "所有" user 可以不用经由输入 password 来进入你的帐号，是壹种更危险的行为。所以请自行小心。

## 2. telnet

telnet 是一个提供 user 经由网路连到 remote host。

telnet 的格式如下：

```
telnet [ hostname | ip-address ] [ port ]
```

hostname 为一个像 ccscun1 或是 ccscun1.cc.nctu.edu.tw 的 name address，ip-address 则为一个由四个小於 255 的数字组成的 ip address，如 ccscun1 的 ip-address 为 140.113.17.173，ccscun1.cc.nctu.edu.tw 的 ip-address 为 140.113.4.11。你可以利用 telnet ccscun1 或 telnet 140.113.17.173 来连到 ccscun1。

port 为一些特殊的程式所提供给外界的沟通点，如资工系的 MUD 其 server 便提供一些 port 让 user 由这些 port 进入 MUD 程式。详情请参阅 telnet(1) 的说明。

## 3. ftp

ftp 的意义是 File Transfer Program，是一个很常应用在网路档案传输的程式。ftp 的格式如下：

```
ftp [ hostname | ip-address ]
```

其中 hostname | ip-address 的意义跟 telnet 中的相同。

在进入 ftp 之後，如果与 remote host 连接上了，它将会询问你 username



与密码，如果输入对了就可以开始进行档案传输。

在 ftp 中有许多的命令，详细的使用方式请参考 ftp(1)，这里仅列出较常用的 cd，lcd，mkdir，put，mput，get，mget，binary，ascii，prompt，help 与 quit 的使用方式。

ascii 将传输模式设为 ascii 模式。通常用于传送文字档。

binary 将传输模式设为 binary 模式，通常用于传送执行档，压缩档与影像档等。

cd remote-directory 将 remote host 上的工作目录改变。

lcd [ directory ] 更改 local host 的工作目录。

ls [ remote-directory ] [ local-file ] 列出 remote host 上的档案。

get remote-file [ local-file ] 取得远方的档案。

mget remote-files 可使用通用字元一次取得多个档案。

put local-file [ remote-file] 将 local host 的档案送到 remote host。

mput local-files 可使用通用字元一次将多个档案放到 remote host 上。

help [ command ] 线上辅助指令。

mkdir directory-name 在 remote host 造一个目录。

prompt 更改交谈模式，若为 on 则在 mput 与 mget 时每作一个档案之传输时均会询问。

quit/bye 离开 ftp。

利用 ftp，我们便可以在不同的机器上将所需要的资料做转移，某些特别的机器更存放大量的资料以供各地的使用者抓取，本校较著名的 ftp server 有 NCTUCCOA 与系上的 ftp.csie.nctu.edu.tw。这些 ftp server 均有提供一个 user 称为 anonymous，一般的“外来客”可以利用这个 username 取得该 server 的公共资料。不过 anonymous 在询问 password 时是要求使用 anonymous 的使用者输入其 email address，以往有许多台湾的使用者在使用国外的 ftp server 时并没有按照人家的要求输入其 email address，而仅是随便打一些字串，引起许多 internet user 和管理者的不满，对台湾的使用者的风评变得很差，因此遵循各 ftp server 的使用规则也是一件相当重要的事。

f. 关于通讯用的指令：

#### 1. write

这个指令是提供使用者传送讯息给另一个使用者，使用方式：

```
write username [tty]
```

#### 2. talk/ytalk/cytalk/ctalk

UNIX 专用的交谈程式。会将萤幕分隔开为你的区域和交谈对象的区域，  
同时也可和不同机器的使用者交谈。使用方式：

```
talk username[@host] [tty]
```

#### 3. mesg

选择是否接受他人的 messege，若为 messege no 则他人的 messege 将无法  
传送给你，同时他也无法干扰你的工作。使用方法：

```
mesg [-n|-y]
```

#### 4. mail/elm

在网路上的 email 程式，可经由此程式将信件 mail 给他人。使用方式：

```
mail [username]
```

```
mail -f mailboxfile
```

如有信件，则直接键入 mail 可以读取你的 mail。

elm 提供较 mail 更为方便的介面，而且可做线上的 alias。你可以进入 elm  
使用上下左右键来选读取的信件，并可按 h 取得线上的 help 文件。

使用方式：

```
elm [usermae]
```

```
elm -f mailboxfile
```

#### g. 编译器 ( Compiler ):

Compiler 的用处在于将你所撰写的程式翻译成一个可执行档案。在资工系常  
用的程式语言是 C，pascal，FORTRAN 等。你可以先写好一个 C 或 Pascal  
或 FORTRAN 的原始程式档，再用这些 compiler 将其翻成可执行档。你可以  
用这个方法制造你自己的特殊指令。

#### 1. cc/gcc (C Compiler)

```
/usr/bin/cc
```

```
/usr/local/bin/gcc
```

语法：cc [ -o execfile ] source  
gcc [ -o execfile ] source

execfile 是你所希望的执行档的名称，如果没有加上 -o 选项编译出来的可执行档会以 a.out 做为档名。source 为一个以 .c 做为结尾的 C 程式档。请参阅 cc(1) 的说明。

2. pc (Pascal Compiler)  
/usr/local/bin/pc

语法：pc [ -o execfile ] source

execfile 是你所希望的执行档的名称，如果没有加上 -o 选项编译出来的可执行档会以 a.out 做为档名。source 为一个以 .p 做为结尾的 Pascal 程式档。请参阅 /net/home5/lang/man 中 pc(1) 的说明。

3. f77 (Fortran Compiler)  
/net/home5/lang/f77

语法：f77 [ -o execfile ] source

execfile 是你所希望的执行档的名称，如果没有加上 -o 选项编译出来的可执行档会以 a.out 做为档名。source 为一个以 .p 做为结尾的 FORTRAN 程式档。

h. 有关列印的指令：

以下为印表所会用到的指令，在本系的印表机有 lp1，lp2（点矩阵印表机），lw，sp，ps，compaq（雷射印表机），供使用者使用。

1. lpr  
lpr 为用来将一个档案印至列表机的指令。

用法：  
lpr -P[ printer ] file1 file2 file3 ....

或  
lpr -P[ printer ] < file1

例子：  
lpr -Plp1 hello.c hello.lst hello.map  
lpr -Plp1 < hello.c

前者以参数输入所要印出的档案内容，后者列印标准输入档案（standard input）

的内容，因已将 `hello.c` 转向到标准输入，故会印出 `hello.c` 的档案内容。

## 2. `lpq`

`lpq` 是用来观察 `printer queue` 上的 `Jobs`。

用法：

```
lpq -P[ printer ]
```

## 3. `lprm`

`lprm` 是用来取消列印要求的指令。通常我们有时会印错，或是误送非文字档资料至 `printer`，此时就必须利用 `lprm` 取消列印 `request`，以免造成资源的浪费。

用法：

```
lprm -P[ printer ] [ Jobs id | username ]
```

`lprm` 用来清除 `printer queue` 中的 `Jobs`，如果你使用 `Job Id` 作为参数，则它将此 `Job` 自 `printer queue` 清除，如果你用 `username` 作为参数，则它将此 `queue` 中所有 `Owner` 为此 `username` 的 `Jobs` 清除。

i. 更改个人使用资料：

### 1. `passwd`

`passwd` 是用来更改你的使用密码，用法为：

```
passwd [ username ]
```

在使用 `passwd` 的时候，它会先问你的旧密码，然後询问两次要更改的密码，确定无误後才将你的密码改掉。

### 2. `chsh`

`chsh` 是提供使用者更换 `login shell` 的指令，你可经由此更换自己使用的 `shell`。

## GNU

GNU 是 GNU's Not Unix 的缩写。

它的目标是在 `UNIX` 系统基础上，建立一个可自由发布和可移植的完善的 `Unix` 类操作系统。因开始实施 `GNU` 项目时，当时还没有多少高质量的自由软件可供项目使用。所以，为 `GNU` 项目做出贡献的人们先从系统的应用软件和工具入手。它的系统核心是 `HURD`，但是由于 `LINUX` 的提前出现，让许多人认为，`HURD` 没有必要再开发下去，用 `LINUX` 作 `GNU` 的核心就很理想了。

## FSF

FSF 是 Free Software Foundation 的缩写，即自由软件基金会

十几年前，世界级的顶尖程序员 `Richard Stallman` 提出了一个伟大的信念：计算机系统应该对用户开放，软件应该自由使用。在他的倡导下，掀起了开发自由软件的运动热潮。1984 年，`Richard Stallman` 的努力得到了麻省理工学院的正式支持。后来他就成立了自由软件基金会这个倡导自由软件的组织。

该组织的主要项目是 GNU，一个自由发布和可移植的类 UNIX 系统。

## CPL

GPL 是 General Public License 的缩写，即通用公共许可证

这是为了能够规范自由软件，由自由软件基金会（FSF）提出，并制定成文的一种不同于传统的软件许可证。它的核心思想就是保证任何人可以共享和修改自由软件，有权取得、修得、修改和重新发布自由软件的源代码。这使得自由软件总的费用相当低廉，而且在使用 Internet 的情况下是免费的。

该条款还规定自由软件的衍生作品必须以 GPL 作为它重新发布的许可证。这一规定保证了自由软件及其衍生作品继续保持自由状态。也就是说防止了不法分子将自由软件更改后注册自己的版权。

而且通用公共许可证还容许销售自由软件。也就是说，自由软件本身是免费的，但销售公司可以从销售中收取一定的服务费用。这为公司介入自由软件事业敞开了大门。同时公司的介入弥补了自由软件的不足，对推动自由软件应用起了很大的作用。就象 Red Hat 公司在发布 LINUX 的过程中不断壮大，同时也大大促进了 LINUX 的发展。

正是这个条款的合理制订，使得自由软件事业得以全面发展。可以说，自由软件之所以有今天的成就，CPL 功不可没。

## (A)

帐户名称 (Account Name) — 等同于登录标识、用户标识或用户名。是指派给 UNIX/Linux 系统上用户的名称。可以在系统上对多个用户设置唯一的帐户名称，每个用户具有不同的访问(权限)级别。在安装完 Linux 之后，帐户名称由超级用户 (Superuser) 或 root 操作员指派。

AfterStep — 用户界面(窗口管理器)之一，AfterStep 使得 Linux 的外观很象 NeXTSTEP，而且还有些增强功能。要获取更多关于 AfterStep 的信息，请访问 [www.afterstep.org](http://www.afterstep.org)。(另请参阅“Enlightenment”、“GNOME”、“KDE”和“X Window 系统”。)

Awk (Aho、Weinberger 和 Kernighan) — 一种编程语言，因其模式匹配语法而特别有用，通常用于数据检索和数据转换。一个 GNU 版本称为 Gawk。

APM (高级电源管理，Advanced Power Management) — 一种工业标准，它允许系统处理器和各个组件进入省电模式，包括挂起、睡眠和关机。APM 软件对于移动设备尤为重要，因为它节省了电池电量。

附加符号 (Append Symbol) — 两个键盘字符 > (也就是 >>)。通常用它将命令的输出发送到文本文件，将数据附加到文件的尾部，而不是替换现有的内容。例如，`ls -a >> output.txt` 将当前目录列表发送到名为 output.txt 的文件，并将其添加到该文件的尾部。重复执行该命令会不断地将新数据添加到文件尾部。(另请参阅“管道符号”和“重定向符号”。)

归档文件 (Archive) — 含有多个文件的单个大型文件，通常对其进行压缩以节省存储空间。经常创建归档文件以方便计算机之间的传送。流行的归档格式包括 ARJ、TAR、ZIP 和 ZOO。它们都可以用来创建这样的归档文件。

ARJ — 流行的文件压缩 / 归档工具，可以用于 UNIX/Linux、DOS/Windows 和其它操作系统。用这种方式压缩的文件的扩展名通常是 .arj 或 .ar。

后台进程 ( Background Process ) — 运行时无须用户输入的程序。可以在诸如 UNIX/Linux 之类的多任务操作系统上运行多个后台进程，而用户则与前台进程交互（例如，数据输入）。有些后台进程（例如守护程序）从来都不需要用户输入。其它一些进程只是在用户忙于目前运行于前台的程序时才临时处于后台。

Bash ( Bourne Again SHell ) — Bourne Shell 的增强版。（另请参阅“Korn Shell”。）

BDF 字体 — 用于 X Window 系统的各种位图字体。（另请参阅“PostScript 字体”和“TrueType 字体”。）

Bean — JavaBeans 体系结构的组件。

Beowulf — 由相对廉价的计算机（包括 PC）组成的网络，这些计算机可能使用不同的处理器和硬件体系结构，由 Linux 和特殊的系统级软件将它们联合成一个海量并行计算系统。最终结果是一个能够以较低成本获得超级计算机计算能力的系统。这类系统对于诸如气象建模之类的计算密集型任务而言是非常理想的，因为可以将计算分配到几十、几百甚至几千个同时运行的独立处理器上。有时也称为“Beowulf 级超级计算机”或者超级群集计算机。更多关于 Beowulf 技术的信息，请阅读以下文章：[www.linuxworld.com/linuxworld/lw-2000-04/lw-04-parallel.html](http://www.linuxworld.com/linuxworld/lw-2000-04/lw-04-parallel.html) 和 [www.ibm.com/press/prnews.nsf/Searchvw/3f4e88b102477aa5852568460067a52a](http://www.ibm.com/press/prnews.nsf/Searchvw/3f4e88b102477aa5852568460067a52a)。

Bin — 一个含有可执行程序的目录，这些程序主要是二进制文件。

二进制文件 ( Binary ) — 已被编译成可执行程序的源代码。在 UNIX/Linux 世界里，有些软件仅作为源代码分发；有些软件包既包含源代码又包含二进制文件；还有一些软件包则只以二进制文件格式分发。

引导盘 ( Boot Disk ) — 一张软盘，其中含有操作系统（如 Linux）引导（启动）计算机并从命令行运行一些基本程序所需的足够内容。如果因某种原因导致系统表现为无法引导，那么引导盘是必需的。引导盘还用于对硬盘进行分区和格式化、恢复主引导记录 ( Master Boot Record ) 或者复制特定文件等。

Bot — 机器人 ( Robot ) 的简称。它是一个程序，旨在几乎没有人工干预下在因特网上搜索信息。

Bourne Shell — 一种流行的命令行 shell，它具有的优点比 DOS 命令提示符多很多。（另请参阅“Bash”和“Korn Shell”。）

BSD ( Berkeley 软件分发版，Berkeley Software Distribution ) UNIX — 加州大学伯克利分校开发的 UNIX。（另请参阅“FreeBSD”。）

Bzip2 — 一种较新的 UNIX/Linux 文件压缩程序，它比 Gzip 提供更大的压缩比。

(C)

CGI (公共网关接口, Common Gateway Interface) — 在 Web 服务器上, 用来在脚本和 / 或应用程序之间传输数据, 然后将该数据返回给 Web 页面或浏览器。CGI 脚本经常是使用 Perl 语言创建的, 它能够生成动态 Web 内容 (包括电子商业购物篮、讨论组、调查表单以及实时新闻等)。

CHS (柱面 / 磁头 / 扇区, Cylinder/Head/Sector) — FDISK 在分区期间所需的磁盘信息。

客户机 (Client) — 向服务器请求服务 (例如, 电子邮件) 的机器。

CLU (命令行实用程序, Command Line Utility) — 从命令行会话或 shell 运行的程序, 如 Tar 或 Mkdir。

群集 (Cluster) — 由运行 Linux 的工作站 (PC 或其它机器) 组成的网络。(另请参阅 “Beowulf”。)

COLA — 因特网新闻组 comp.os.linux.announce 的简称, 该新闻组宣布一些与 Linux 相关的参考资料。

命令行界面 (Command Line Interface) (CLI) — 全屏或窗口化的文本方式会话, 在该会话中用户通过输入命令来执行程序, 这些命令可以带参数, 也可以不带参数。CLI 显示来自操作系统或程序的输出文本, 并为用户输入提供命令提示符。

命令提示符 (Command Prompt) — DOS/Windows 和 OS/2 术语, 是命令行界面的一部分, 用户在该界面中输入命令。(另请参阅 “Shell 提示符”。)

编译 (Compile) — 将编程源代码转换成可执行程序。

编译型语言 (Compiled Language) — 一种语言, 它需要编译器程序将编程源代码转换成可执行的机器语言二进制程序。一经编译, 就可从程序的二进制形式多次运行程序, 而无需再次编译。编译型语言 / 程序运行往往比解释型语言或伪代码语言快, 但却需要编译器 (可能很昂贵), 而且用编译型语言编程常常会比用解释型语言和伪代码语言编程难。编译型语言的例子有 C 和 C++、COBOL 以及 FORTRAN。

编译器 (Compiler) — 用于将编程源代码转换成可执行程序的程序。

控制台应用程序 (Console Application) — 不需要 (即便是提供了) 图形用户界面就能运行的命令程序。

Cron — Linux 守护程序, 它在指定时间或按指定间隔执行规定的任务。



## (D)

守护程序 (Daemon) —操作系统的后台进程，通常具有 root 安全级别许可权。守护程序通常隐藏在后台，直至被某个事件（例如特定的时间或日期、时间间隔、收到电子邮件等）触发后它才会进入活动状态。

桌面 (Desktop) —操作系统用户界面，旨在表示一个在上面放东西的办公桌。操作系统的桌面并不使用有形的电话、电灯、收 / 发箱等，而是使用程序及数据图标、窗口、任务栏和类似的东西。Linux 可以使用许多不同的桌面环境，包括 KDE、GNOME 和 X11，它们可以由用户安装。（另请参阅“GUI”、“窗口管理器”和“X Window 系统”。）

设备驱动程序 (DeviceDriver) —一种程序，它充当操作系统与设备（端口、驱动器、监视器、打印机等）之间的媒介，它向操作系统说明该设备具备哪些能力，同时将操作系统命令转换成该设备可以理解的指令。

分发版 (Distribution) —将 Linux 内核（核心）连同各种用户界面、实用程序、驱动程序及其它软件打包成可交付给用户的软件包。分发版通常是以免费下载或廉价 CD-ROM 软件包的形式提供给用户的。流行的分发版包括 Caldera OpenLinux、CoreLinux、Debian、Red Hat、Slackware、SuSE、TurboLinux 及其它软件。

Dpkg (Debian 软件包管理器，Debian PackageManager) —可从因特网下载的打包及安装工具，它包含在 Debian Linux 中，但与其它分发版兼容。它生成具有 .DEB 扩展名的文件。与 RPM 类似。

## (E)

Emacs (用 MACroS 进行编辑，Editing with MACroS) — 一个流行的文本编辑器。

Enlightenment — 用户界面（窗口管理器）之一。更多关于 AfterStep 的信息，请访问 [www.afterstep.org](http://www.afterstep.org)。（另请参阅“AfterStep”、“GNOME”、“KDE”和“X Window 系统”。）

## (F)

文件系统 (FileSystem) —一组程序，它们告诉操作系统如何访问及解释存储在磁盘或磁带驱动器或者其它存储媒介上的内容。常见的文件系统包括：FAT 和 FAT-32 (DOS/Windows)、HPFS (OS/2)、NFS、NTFS (Windows NT/2000) 以及其它文件系统。

过滤器 (Filter) —一种程序，它（从文件、程序输出或命令行输入）读取数据作为输入，根据一组预定义条件处理输入（如按字母顺序排序），然后输出处理过的数据。一些常见的过滤器包括 Awk、Grep、Sed 和 Sort。

Finger — UNIX/Linux 命令，它提供登录用户的有关信息。



前台进程 (ForegroundProcess) — 在多任务操作系统 (诸如 UNIX/Linux) 中, 前台进程是用户当前与之交互的程序 (例如, 数据输入)。随着用户在程序之间切换, 会导致这些程序在不同的时刻处于前台。在层叠的窗口环境中, 前台进程是最前面的窗口。

FreeBSD (免费伯克利软件分发版, Free Berkeley Software Distribution) — 类似于 Linux, 因为它包含许多 GNU 程序, 并且它运行的许多软件包与 Linux 所运行的相同。但一些内核功能的实现却不尽相同。(另请参阅“BSD UNIX”。)

FTP (文件传送协议, File Transfer Protocol) — 与其它计算机 (常常是软件资源库) 来回传送文件的方法。

(G)

GCC (GNU C 编译器, GNU C Compiler) — 由 GPL 管理的一个高质量 C 编译器。

GIMP (GNU 图像操作程序, GNU Image Manipulation Program) — 一种用于 Linux 的、流行的图像编辑器 / 绘图程序。

GNOME (GNU 网络对象模型环境, GNU Network Object Model Environment) — 一种用于 Linux 的用户界面 (窗口管理器), 它是用 Gtk 构建的。更多关于 GNOME 的信息, 请访问 [www.gnome.org](http://www.gnome.org)。(另请参阅“AfterStep”、“Enlightenment”、“KDE”和“X Window 系统”。)

GNU (GNU 不是 Unix, GNU is Not Unix) 项目 — 麻省理工学院 (MIT) 自由软件基金会 (Free Software Foundation, FSF) 为开发和促进替代专有 UNIX 实现的产品所进行的项目。GNU 软件使用 GPL 许可证。

GNU/Linux — 等同于 Linux。之所以这么称呼, 是因为 Linux 分发版中包含的许多组件都是 GNU 工具。

GPL (GNU 通用公共许可证, GNU General Public License) — 一种公用及再分发许可证。请访问 [www.linuxdoc.org/LDP/gs/app-gpl/node1.html](http://www.linuxdoc.org/LDP/gs/app-gpl/node1.html) 来查阅 GPL 协议的副本。

Grep (全局正则表达式及打印, Global Regular Expression and Print) — 一个工具, 它在文件中搜索文本字符串, 然后输出任何含该模式的行。

Gtk/Gtk+ (GIMP 工具箱, GIMP ToolKit) — 一个功能强大而且快捷的开放源码图形库, 用于 UNIX/Linux 上的 XWindow 系统, 程序员可以用来创建按钮、菜单及其它图形对象。(另请参阅“GNOME”、“Motif”和“Qt”。)

GUI (图形用户界面, Graphical User Interface) — 图标、窗口及屏幕上其它图形图像的集合, 它们提供了用户与操作系统交互的方法。(另请参阅“桌面”和“窗口管理器”。)

Gzip ( GNU zip ) — UNIX/Linux 最初的文件压缩程序。最新的版本生成扩展名为 .gz 的文件。( 扩展名 .z 或 .Z 表明是较老版本的 Gzip。 ) 压缩是为了使文件紧凑以节省存储空间并缩短传送时间。( 当与 Tar 结合使用时 , 生成文件的扩展名可能为 .tgz、.tar.gz 或 .tar.Z。 )

## ( H )

Home 目录 — 用户登录之后所在的目录。

HTML ( 超文本标记语言 , Hyper Text Markup Language ) — 用于设计 Web 页面的标准标记语言。标记 “ tag ” 或格式化命令允许 Web 页面设计人员确定突出显示、定位图形及创建超链接等等。

HTTP ( 超文本传输协议 , Hyper Text Transport Protocol ) — 一组创建的准则 , 用于请求和发送基于 HTML 的 Web 页面。

## ( I )

Init — 操作系统装入后立即运行的第一个进程。它以单用户方式启动系统或生成 shell 来读取启动文件 , 并打开指定用于登录的端口。

解释型语言 ( Interpreted Language ) — 与编译型程序不同 , 每次运行解释型程序时都要由解释器程序实时地将源代码转换成二进制形式 , 而编译型程序由编译器一次性将源代码转换成可执行代码 , 随后从其二进制形式运行。解释型语言 ( 以及用它们编写的程序 ) 往往要比编译型语言及伪代码语言 / 程序慢 , 并且通常只有有限的底层操作系统功能访问权限或直接访问硬件的权限。但从另一角度来说 , 它们无需编译器 ( 可能非常昂贵 ) , 并且经常包含在操作系统中 , 通常比编译型语言更容易编程。解释型语言的例子有 BASIC、Perl、Python 和 REXX/Object REXX。

## ( J )

Java — Sun Microsystems 开发的、独立于操作系统的面向对象编程语言。Java 通常用于 Web 服务器。Java 应用程序和 applet 有时以下载的形式提供给用户 , 以便在他们的系统上运行。Java 编程语言可以编制应用程序或较小的 Java “ applet ”。Java 是 C++ 语言稍加简化的版本 , 通常是进行解释而不是编译。( 另请参阅 “ JIT 编译器 ”。 )

Java Applet — 嵌入在 Web 页面内的小型 Java 程序 , 它在浏览器内运行 , 而不是作为独立的应用程序运行。Applet 不能够访问本地计算机上的一些资源 , 如文件和串行设备 ( 调制解调器、打印机等 ) , 通常也不能通过网络与其它计算机通信。

JavaBeans — Java 语言的组件体系结构。JavaBeans 组件称为 Bean。

JavaScript — 跨平台万维网脚本编制语言 , 似乎与 Java 有关。它可以用作服务器端脚本编制语言、由服务器解析的 HTML 中的嵌入语言以及浏览器中的嵌入语言。

JDK( Java 开发工具箱 ,Java Development Kit ) — 由 Sun、IBM 或其它公司开发的 Java 编程工具箱，可以用于 UNIX/Linux 及其它操作系统。

JFS ( 日志文件系统，Journaled/Journaling File System ) — 包含内置备份 / 恢复能力的文件系统。对索引的更改在生效之前先写入日志文件，这样如果索引遭到破坏 ( 例如，在写索引期间断电 )，那么就可以从日志重建索引，包括对索引的更改。

Jini — 读作 “ Genie ”。Sun 开发的软件，工作起来有些象 “ 即插即用 ”，因为它允许硬件设备在被连接时向操作系统通报自己 ( 并提供关于该设备的详细信息 )，而无须系统重新引导。它也向系统所连接的网络通报它自己，从而方便地向网络提供设备共享。

JIT ( Just-In-Time ) 编译器 — Java 语言编译器，它允许实时地将 Java 解释型程序自动编译成本机机器语言，以使程序执行的速度更快。有些 JVM 包含 JIT 编译器。

日志记录 ( Journaling ) — 也就是 “ 日志记录 ( logging )”。将信息写入日志文件，可以作为一种跟踪更改的方法。

JVM ( Java 虚拟机，Java Virtual Machine ) — Java 运行时环境，是运行 Java 程序所必需的，包含 Java 解释器。不同的操作系统 ( Linux、OS/2、Windows 98 等等 ) 需要不同的 JVM，但任一 JVM 都可以运行某个 Java 程序的同一版本。

( O )

面向对象 ( Object-Oriented ) — 一种软件开发方法，它为程序员提供标准可重用的软件模块 ( 组件 )，而无需开发人员每次都编写定制编程代码。使用标准组件缩短了开发时间 ( 因为其他程序员已经编写并测试了这些组件 )，并且通过使用相同的组件确保了程序具有标准的外观。

OO — 参阅 “ 面向对象 ”。

开放源码 ( Open Source ) — 一个稍显模糊的术语，是指同源代码一起发布的软件。提供源代码这一事实并不一定意味着用户可以修改和重新分发源代码。这个术语有时可以和 “ 免费软件 ” 互换使用，尽管它们的意思并不总是相同。( 另请参阅 “ 公共域 ” 和 “ 共享软件 ”。)

OSS ( 开放声音系统，Open Sound System ) — 用于在 UNIX/Linux 下访问声卡及其它音频设备的设备驱动程序。它是从 Linux 声音驱动程序 ( Linux Sound Driver ) 发展而来的，支持大多数流行的音频芯片及适配器。

OSS ( 开放源码软件，Open Source Software ) — 参阅 “ 开放源码 ”。

所有者 ( Owner ) — 对文件具有访问特权的用户；通常是创建该文件的用户。

( P )

伪代码 ( P-code , Pseudo-code ) 语言 — 一种解释型语言。伪代码语言有点象个 “ 混血儿 ” , 执行方式介于编译型语言和解释型语言之间。和解释型语言一样,伪代码编程语言无需编译,在执行时自动转换成二进制形式。然而,和编译型语言不同的是,这种可执行的二进制文件是以伪代码的形式而不是机器语言的形式存储的。此外,不同于解释型语言的是,不必在每次运行该程序的时候,都必须将其转换成二进制。在第一次转换成伪代码后,以后每次执行都可使用该伪代码版本。伪代码语言 ( 以及用它编写的程序 ) 往往比编译型语言及程序慢,但比解释型语言快,它们通常具有访问一些低层操作系统功能的权限,但却不能直接访问硬件。它们不需要有时很昂贵的编译器,通常与操作系统包含在一起,有些伪代码语言比编译型语言更容易编程。伪代码语言的例子有 Java、Python 和 REXX/Object REXX。

PAM ( 可插入的认证模块, Pluggable Authentication Modules ) — 用于系统安全性的可替换的用户认证模块,它允许在不知道将使用何种认证方案的情况下进行编程。这允许将来用其它模块来替换某个模块,却无需重写软件。

面板 ( Panel ) — Linux 中对应于 Windows 任务栏的名称。

分区 ( Partition ) — 磁盘驱动器的一个连续部分,它被操作系统当作物理驱动器。这样,可以为一个磁盘驱动器赋予几个驱动器符号。

PCF 字体 — X Window 系统使用的各种位图字体。

PD — 参阅 “ 公共域 ”。

PDF ( 可移植文档格式, Portable Document Format ) 文件 — 用 Adobe Acrobat 或其它能够生成该格式输出的程序创建的二进制文件。用于生成独立于操作系统的文档,这种文档可以使用 Acrobat Reader 或其它程序 ( 包括配备有 Acrobat Reader 插件的 Web 浏览器 ) 来查看。

Perl ( 实用摘录与报告语言, Practical Extraction and Report Language ) — 一种常用的脚本编制 / 编程语言。经常用在 UNIX/Linux Web 服务器上生成 CGI 脚本。

权限 ( Permission ) — 读写文件和目录及执行程序的权限。超级用户或 root 操作员可以逐个文件、逐个目录地,或者按照帐户名称 ( 用户标识 ) 赋予各种权限级别。

PGP ( 高质量保密程序, Pretty Good Privacy ) — 用于 UNIX/Linux 及其它操作系统的高安全性公钥数据加密程序。

管道符号 ( Piping Symbol ) — 键盘字符 | ( 典型 101 键键盘的 Enter 键上面反斜杠的上档字符 )。经常用来将某个命令或程序的输出提供给另一个命令或程序。例如, history | grep mcopy ( 用 history 命令 ) 将 .bash\_history 文件的内容发送到 grep 程序,以搜索字符串 “ mcopy ”。( 另请参阅 “ 附加符号 ” 和 “ 重定向符号 ”。 )

移植 ( Port/Ported/Porting ) — 一个过程,即获取为某个操作系统平台编写的程序,并对其

进行修改使之能在另一 OS 上运行，并且具有类似的功能。通常很少或者干脆就不尝试定制程序以利用新操作系统的特有功能，这与为某个特定操作系统优化应用程序不同。

可移植 (Portable) — 描述一类软件的术语，这类软件旨在只需少量修改和重新编译就可在多个操作系统上使用。

POSIX( UNIX 可移植操作系统接口 ,Portable Operating System Interface for uniX ) — 一组编程接口标准，它们规定如何编写应用程序源代码以便应用程序可在操作系统之间移植。POSIX 基于 UNIX，它是 The Open Group 的 X/Open 规范的基础。

PostScript — Adobe Systems 开发的页描述语言，它告诉打印机如何在打印页上显示文本或图形。

PostScript 字体 — 可以用于 OS/2、MS Windows 和 X Window 系统的大量字体。这类字体文件的扩展名包括 .afm、.pfa 和 .pfb。有时称为 Adobe Type 1 字体或 ATM( Adobe Type Manager ) 字体。PostScript 字体通常需要与 PostScript 兼容的打印机。(另请参阅“BDF 字体”和“TrueType 字体”。)

进程 (Process) — 正在执行的程序。(另请参阅“多任务”和“多线程”。)

公共域 (Public Domain) — 可供任何人以任何目的使用和修改的软件，甚至可以将其并入商业软件的分发。公共域软件不保留版权，作者也不保留任何权利。(另请参阅“开放源码”和“共享软件”。)

公钥加密 (Public Key Encryption) — 一种包括两个单独密钥 (公钥和私钥) 的数据加密方法。使用公钥加密的数据只能用私钥解密，反之亦然。一般而言，公钥是公开的，可以用来加密发送给私钥持有者的数据，私钥用来对数据进行签名。

Python — 一种面向对象伪代码编程语言。

(Q)

Qt — 一种功能强大且快捷的开放源码图形库，用于 UNIX/Linux 上的 X Window 系统。它被程序员用来创建按钮、菜单和其它图形对象。(另请参阅“Gtk/Gtk+”和“KDE”。)

队列 (Queue) — (也时被错误地拼成 Que。) — 一系列等待执行的任务，如“打印队列”。

(R)

RAID (独立 / 廉价磁盘 / 设备冗余阵列，Redundant Array of Independent/Inexpensive Disks/Devices) — 一种提供数据冗余、改善性能和 / 或从磁盘崩溃中迅速恢复数据的方法，它是通过在多个磁盘驱动器上分布或复制数据来实现这一点的。常用的 RAID 类型包括 RAID 0 (数据条带化)，RAID 1 (磁盘镜像) 和 RAID 5 (具有分布式奇偶校验的条带化)。RAID 配置通常需要 SCSI 磁盘驱动器 (而不是 IDE/EIDE)，可能要求磁盘相同 (相同的容量、品牌

等等)。操作系统将 RAID 阵列看作单个设备。

RC 文件 — 含有程序(应用程序甚至操作系统)启动指令的脚本文件。这一文件在操作系统启动时会自动执行,它含有要运行的指令(命令或其它脚本)列表。

RCS(修订控制系统, Revision Control System) — 一组程序,它们控制组环境下文件的共享访问并跟踪文本文件的变化。常用于维护源代码模块的编码工作。

Rdev — 用于获取关于某个 Linux 系统信息的实用程序。用来查询和设置图像 root 设备、视频方式以及交换设备和 RAM 磁盘。

重定向符号(Redirection Symbol) — 键盘字符 >。经常用于将命令的输出发送到文本文件。例如,ls -a > output.txt 将当前目录列表发送到名为 output.txt 的文件。重复该命令将会用新数据替换该文件的内容。(另请参阅“附加符号”和“管道符号”。)

RFS(远程文件共享, Remote File Sharing) — 一个程序,它让用户访问其它计算机上的文件,就好像文件在用户的系统上一样。

Root 操作员 — 具有执行所有系统级任务权限的用户标识。(也称作超级用户。)

Root 窗口 — 运行 Linux 桌面的底层会话。

RPM(RPM 软件包管理器, RPM Package Manager) — 一种用于因特网下载包的打包及安装工具,它包含在某些 Linux 分发版中。它生成具有 .RPM 扩展名的文件。与 Dpkg 类似。

(S)

脚本(Script) — 一组存储在文件中的命令。用于进行自动重复的执行。(另请参阅“RC 文件”。)

会话(Session) — 用户在登录到注销期间与操作系统之间的完整交互过程。

共享软件(Shareware) — 一类商业软件,它以“先试用再买”的形式提供。如果客户在经过较短的试用期后继续使用该产品,那么他们就被要求支付规定数额(通常微乎其微)的费用。(另请参阅“开放源码”和“公共域”。)

Shell — 含有操作系统命令行界面的文本方式窗口。

Shell 提示符 — shell 的用户输入区域。尽管在 DOS shell 中命令提示符指定为大于号(>),但在 Linux 中,命令提示符通常是百分号(%)、美元符号(\$)或其它特殊符号,这取决于所使用的 shell。(另请参阅“命令提示符”。)

Shell 脚本 — 设计成在 shell 启动时自动运行的脚本。



SHTTP (安全超文本传输协议, Secure Hyper Text Transport Protocol) — HTTP 的一种安全的加密版本,用于金融交易以及其它借助因特网发送的保密信息。斜杠 (/) — 文件路径名中使用的符号,而不是 DOS/Windows 和 OS/2 操作系统中使用的反斜杠 (\)。

源代码 (Source Code) — 程序员输入的、原始状态的编程命令。有些编程语言允许命令实时地由程序解释器执行。其它语言则要求必须先将命令编译成可执行程序 (二进制) 后才能使用这些命令。在 UNIX/Linux 世界中,有些软件仅以源代码形式分发;另一些软件包则同时包含源代码和二进制代码;还有一些则仅以二进制格式分发。

假脱机 (Spool) (外围设备联机并发操作, Simultaneous Peripheral Operation On-Line) — 将数据发送给一个程序,该程序将该数据信息放入队列以备将来使用 (例如,打印假脱机程序)。

SQL (结构化查询语言 (Structured Query Language)) — 用于操作关系数据库中的记录和字段 (行和列) 的语言。有时被错误地读成 “sequel”。

隐写术 (Steganography) — 将一段信息隐藏在另一段信息中的做法。一个示例是在数字化照片中放置不可见的数字水印。

字符串 (String) — 字符序列,如 “搜索字符串”。

超级用户 (Superuser) — 通常与 root 操作员同义。

交换 (Swap) — 暂时将数据 (程序和 / 或数据文件) 从随机存取存储器移到磁盘存储器 (换出), 或反方向移动 (换入), 以允许处理比物理内存所能容纳的更多的程序和数据。也称为虚拟内存。

交换空间 (Swap Space) — 被交换数据在磁盘上暂时存放的地方。Linux 用专用磁盘分区而不是特定交换文件来作为交换空间。

符号链接 (Symbolic link) — 程序或文件的别名或快捷方式。

同步 (Sync) — 将所有暂挂的输入 / 输出强制写回磁盘驱动器。

系统日志 (Syslog) — UNIX/Linux 系统日志记录程序,其中存储了所有系统消息或错误。

(T)

标记 (Tag) — 诸如 HTML 之类的标记语言中的命令,它以某种方式 (如粗体、居中或使用某种字体) 显示信息。

Tar (磁带归档, Tape ARchive) — 包含在 UNIX/Linux 中的一个文件打包工具,用于将一组文件组装成一个组合的文件以便归档起来更容易。它最初设计用于磁带备份,但现在它也可用于其它存储介质。独自运行时,生成具有 .tar 扩展名的文件。当与 Gzip 结合用于数据压缩时,生成的文件扩展名可能是 .tgz、.tar.gz 或 .tar.Z。

Tar 包 (Tarball) — 用 Tar 实用程序创建的文件, 含有一个或多个归档的, 也有可能进行了压缩的文件。

TeX — 基于宏的流行的文本格式化程序。它是包括 LaTeX 和 TeX 在内的其它此类格式化程序的基础。

文本编辑器 (Text Editor) — 用于编辑文本文件的程序。类似于字处理程序, 但没有大多数 / 全部格式化功能 (例如设置页边距、斜体和字体等等)。经常用于书写或编辑脚本、程序和 ASCII 文本文件 (如 README.1ST)。

文本格式化程序 (Text Formatter) — 准备文本文档以供打印的程序, 允许用户执行许多布局功能, 如页边距、页眉、页脚、缩排、分页和对齐。

TFTP (小型文件传送协议, Trivial File Transfer Protocol) — FTP 的简化版本, 它没有 FTP 的认证以及其它许多基本特性。

线程 (Thread) — 一小段程序, 其行为就象是较大程序的一个独立子集, 也称为“进程”。多线程程序能够比单个程序或单线程程序运行得快得多, 因为它可以并行 (而不是串行 (顺序)) 地执行几个甚至多个不同的任务。而且, 单个应用程序内的多个线程可以共享资源, 并且相互之间可以来回传递数据。

分时 (Time-sharing) — 一种允许多个用户分享处理器的方法, 它以时间为基础给每个用户分配一部分处理器资源, 按照这些时间段轮流运行每个用户的进程。(另请参阅“多任务”。)

Torvalds, Linus — 1991 年 Linux 内核的最初创建者, 是 Linux 版权的拥有者, 目前仍然是 Linux 开发项目的协调人。

Touch — 一个命令, 更改文件的日期 / 时间戳记, 但不影响其内容。

TrueType 字体 — 与 PostScript 字体不一样, 它们旨在成为与打印机无关的各种字体。可用于 Apple Macintosh 和 Windows。不常用于 UNIX/Linux。(另请参阅“BDF 字体”和“PostScript 字体”。)

Tux — 虚构的 Linux 企鹅吉祥物的名字。