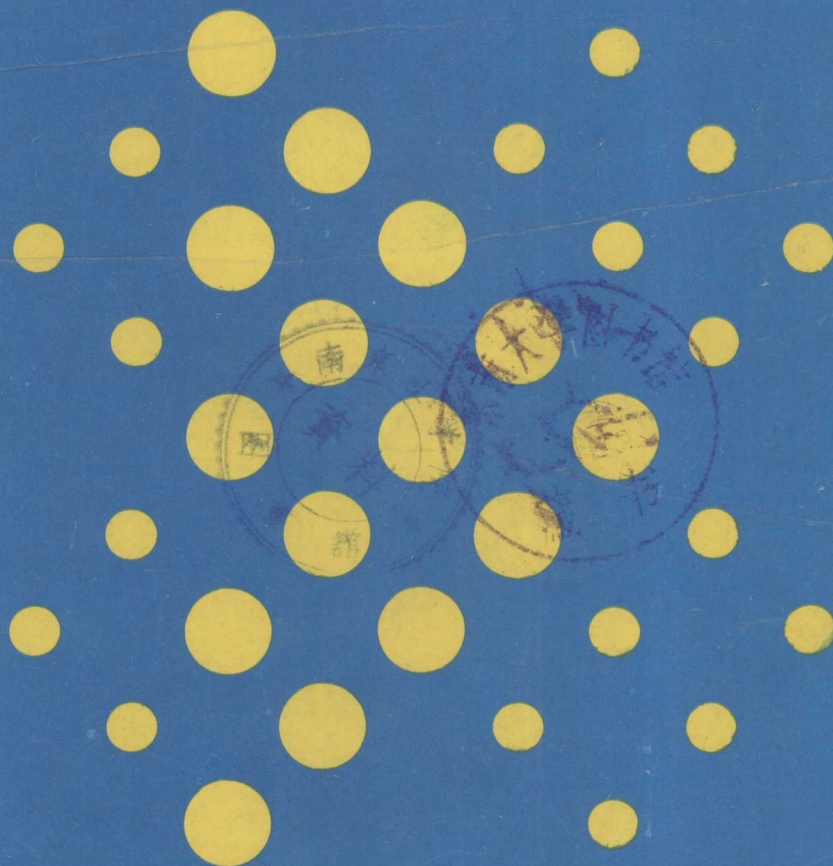


UNIX系统V

实用指南

第二分册



中国科学院计算机公司
香港港粤投资有限公司

1985·3·北京

第 二 分 册

UNIX 系统 V 用户指南及
基本实用程序手册

87018878-87

6721001

578

目 录

第一章	介绍.....	(1)
第二章	UNIX 操作系统概貌.....	(2)
第三章	基本过程.....	(6)
第四章	基础实用程序包.....	(21)
附录 A	基础实用程序包分页手册.....	(94)
附录 B	命令及实用程序包索引	(153)
索 引 (略)		



第一章 介绍

(一) 一般性介绍

(二) 关于所提供的文件

(一) 一般性介绍

本手册对 3B2-300 型计算机上的 UNIX 系统 V 进行描述。(UNIX 是 AT & T 公司 BELL 实验室的商标; 3B 是西部电子公司商标。) 主要叙述包括:

★ 提供的文件。

★ 操作系统的特征。

★ 基础实用程序包。

在本手册中的有关信息、按章节进行组织, 如下:

第一章是对本手册的简介。

第二章是对 UNIX 操作系统进行概括说明。

第三章是为初学的用户提供一个基本操作过程。其中主要包括如何访问系统及熟练地使用 UNIX 系统中目录和文件。

第四章是描述 3B2 计算机上的基础实用程序。

附录 A: 分页手册, 为在第四章中定义的一些实用程序 (命令)、基础实用程序包提供系统分页手册。

附录 B: 本附录对所有命令提供一个字母顺序索引, 并指出与每条命令相联系的实用程序。实用程序所建立的命令清单也在此附录中。

索引: 提供在本手册中的按主题词查找信息的方法。(此章略)

(二) 关于所提供的文件

本手册提供对 UNIX 操作系统 V 的简介。为使一个初学者变为使用本系统的熟练者, 本手册提供了足够的信息。当你阅读了第二、三章之后你将会知道, 如何访问系统 (login), 如何建立、修改、删除文件; 以及文件系统是怎样构成的。在此, 本文仅介绍了操作系统的基本特征及能力。基础实用程序包的命令更复杂的应用在“3B2-300 计算机 UNIX 系统 V 高级用户指南”中予以介绍。高级用户指南是本手册基本描述的进一步扩充。

除基础实用程序包在 3B2 计算机上提供了以外, 其它的实用程序也是可获得的。附录 B 中介绍了各种实用程序, 可供参考。关于这些实用程序包及其说明资料的可获得与否的信息, 可与西部电器公司你的销售代理人接触。

第二章 UNIX 操作系统概貌

(一) 一般性介绍

(二) 文件系统

★ 一般介绍

★ 目录

★ 普通文件

★ 特殊文件

★ 可装卸的文件系统

★ 安全性

(三) SHELL

(一) 一般性介绍

UNIX 操作系统是一个软件程序的汇集，它的设计是企图支持广泛的用户。故提供了如下的系统特征：

- 分级文件系统。
- 灵活，易于使用的命令语言。
- 有进行顺序、异步、后台操作能力。
- 强有力的上下文编辑。
- 文件预处理及正文处理性能灵活。
- 高级程序设计语言。

(二) 文件系统

(★) 一般介绍。

从用户角度看，UNIX 操作系统中最有意义的部份要算是文件系统。文件系统是目录和文件的统一的集合，且是按照树状分级结构进行排列的。该结构的最底部称为根（/）。从根部分枝出三种不同类型的文件：普通磁盘文件，目录和特殊文件。从根引出的其末端到达文件的各个分枝称为路径或路径名。如图2-1所示即是一个文件系统结构的例子。文件系统的主要特征包括：

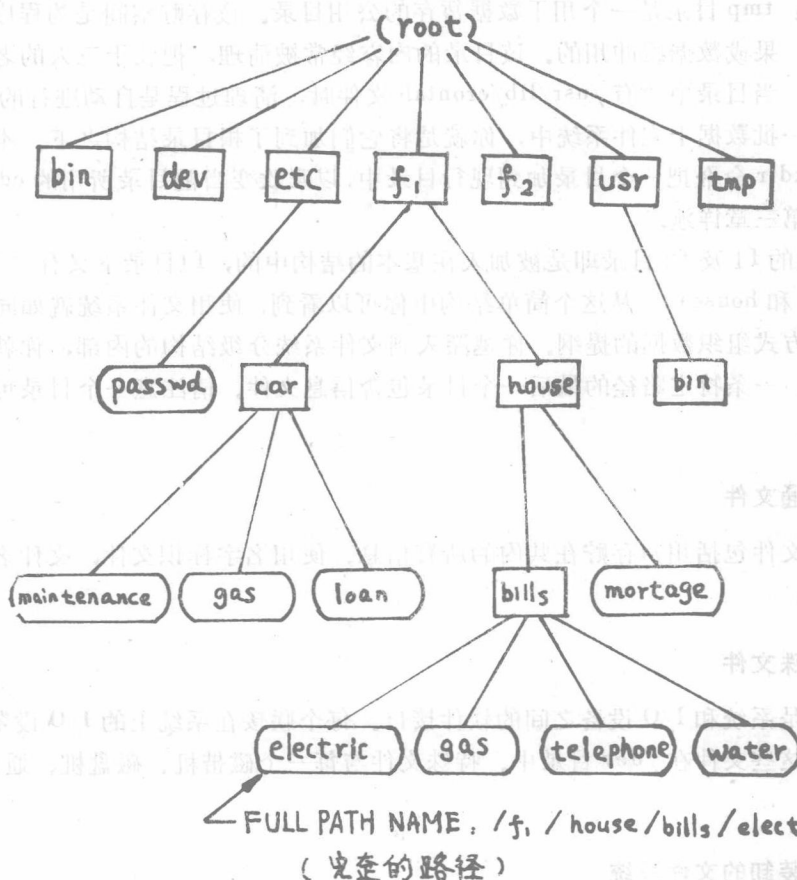
简单地用名字来标识目录及文件。因为你可以定义你自以为最有意义的名字。所以，标识这些目录及文件的方法，规定了有联系的数据存贮类型。

- 可安装及拆卸的（可装卸性）文件系统。

- 文件跨目录进行链接。
- 自动进行文件空间的分配与回收。(对用户透明)
- 按拥有者、同组人和其他人基本的标识符,构成的目录及文件的安全性。
- 建立、访问、移动及处理文件的能力。
- 终端、磁盘文件及主存之间统一的输入输出管理。

(★) 目录。

什么是目录呢?目录即是提供了文件名字与数据文件之间的一种映象。系统使用名字识别目录。一个目录的名字必须是小于等于14个字符的序列。通过指定一直通向某文件的各个目录名字,可指出一条通向该文件的完整的路径。其路径中每个名字间将用/符分隔开。一条完整的路径应从根部开始。见图2-1所示的通往 electric 文件的完整路径例子。



说明: : 目录
 : 文件

图2-1 典型文件系统结构

下面简述基本文件系统的结构。在文件系统中保留了一些目录供系统使用。一个为系统准备的基本目录称为根目录。根目录由/符号命名。所有其它的目录及文件均被组织在根目录之下。目录的结构类似一棵倒挂的树。树的底部为根，其长出的分枝即通向各种目录及文件。其根目录下的一些目录形成 UNIX 系统的基础。现列出如下：

/bin: bin 目录包括许多 UNIX 系统命令。

/dev: dev 目录包括一些特殊设备文件，用于完成与终端、磁盘机、磁带机及其它相类似设备接口的通讯。

/etc: etc 目录包括与系统注册及系统事务性管理相关联的那些系统文件和命令。

例如该目录中有通行字文件 (PASSWD)、同组人文件 (GROUP)、及系统缺省约定 (PROFILE)。

/usr: usr 目录包括与用户信息相关的那些目录和文件。例如该目录中有新闻目录 (NEWS) 及用户邮件目录 (MAIL)。

/tmp: tmp 目录是一个用于数据暂存的公用目录。该存储空间是为程序存取中间结果或数据缓冲用的。该目录的内容经常被清理，把长于三天的老文件删除。

当目录中含有 /usr/lib/crontab 文件时，清理过程是自动进行的。

当你加入一批数据于文件系统中，你就是将它们加到了根目录结构之下。本手册的后面将叙述使用 mkdir 命令把一个目录加到现行目录中，以及改变当前目录所用的 cd 命令。这些基本过程将在第三章详述。

图2-1所示的 f1 及 f2 目录即是被加入在基本的结构中的，f1 目录下又有二个以上的目录被加入 (car 和 house)。从这个简单结构中你可以看到，使用文件系统就如同使用一个按着一定的逻辑方式组织数据的提纲。你越深入到文件系统分级结构的内部，你就越能获得更为详尽的信息。一条特定路径的最后一个目录包含信息文件。请注意一个目录可以包含其它的目录和文件。

(★) 普通文件

一个普通文件包括用户存贮在其内的所有信息。使用名字标识文件，文件名的长度不得长于14个字符。

(★) 特殊文件

特殊文件是系统和 I/O 设备之间的软件接口。每个联接在系统上的 I/O 设备都通过特殊文件访问。这些文件在 /dev 目录中。特殊文件与每一个磁带机、磁盘机、通讯线路等设备相对应。

(★) 可装卸的文件系统

文件系统的根已被安置在一个设备上。整个文件系统层次不必与文件系统的根在同一设备上。文件系统层次的一部分安装在与根不同的设备上，这种文件系统层次叫做可装卸的文件系统，也可称为可安装的文件系统。

(★) 安全性

访问目录和文件是由一些许可标识位来控制的。任何一个目录及文件均与九个保护位及一个类型标识位相联系。图2-2表示了它们的含义。

项 目	含 义			
标识位位置	10	987	654	321
标识位名字	TYPE	OWNER	GROUP	OTHER
设置情况	d, b, c, p, -	rwX	rwX	rwX

图2-2 目录或文件许可位含义。

类型(TYPE)位标识:

(d): 目录名。

(b): 块特殊文件。

(c): 字符特殊文件。

(p): 管道特殊文件。(先进先出)

(-): 普通文件。

访问控制的三个级别提供为所有者(OWNER)、同组人(GROUP)和其它人(OTHER)。读(r)、写(w)和执行(x)许可标记分别控制所有者、同组人和其它人。读(r)、写(w)或执行(x)的存在即是一种特定许可的实现。这将在第三章中进一步描述。在所有者、同组人或其它人的许可标记中有连字符(-)即指明相应的许可被否定了。所有者访问许可权提供给那样的用户,该用户的用户标识符与目录或文件的所有者名字相匹配。同组人访问许可权提供给所有那样的用户,他们的组标识符与目录或文件的组名字相匹配。其他人访问许可权提供给所有那样的用户,他们的用户标识符不能与文件或目录的所有者名字或组名字相匹配。系统提供了显示许可标记和目录内容清单以及改变许可标记的命令。只有目录或文件的所有者才能改变访问许可标记。

(三) SHELL

shell 是用户与系统的程序接口。它解释用户从终端输入的命令。

请注意,在本手册中描述的 shell 是 UNIX 系统的 shell(sh),其他的 shell 也能提供用户与系统的接口。

shell 还有它自身的命令语言。用 shell 命令语言书写的程序称为 shell 正文。shell 建立了一些变量(条件)。在这些变量下命令及 shell 程序被执行。这些 shell 可变的定义是命令执行的环境,用户可以改变这个环境。

第三章 基本过程

(一) 一般性介绍.....	(6)
(二) 访问系统.....	(7)
★一般介绍.....	(7)
★进入注册.....	(7)
★退出注册.....	(8)
★设置系统终端选择.....	(8)
★使用打印机.....	(9)
★建立和改变命令执行环境.....	(10)
(三) 输入命令.....	(12)
★一般介绍.....	(12)
★命令行文法.....	(12)
★修改和删除命令行.....	(13)
★终止一个活动的命令.....	(13)
★命令输出转向.....	(13)
★执行后台进程.....	(14)
(四) 目录和文件操作方式.....	(15)
★一般介绍.....	(15)
★点(.)和点点(··)目录.....	(15)
★建立目录.....	(16)
★建立文件.....	(16)
★加入、修改和删除文件内容.....	(16)
★文件的删除.....	(16)
★目录的删除.....	(17)
★查找文件.....	(17)
★打印和显示文件内容.....	(18)
★读、写和执行许可标记.....	(18)

(一) 一般性介绍

本章为初学 UNIX 系统的用户阐述基本的操作过程。其主要内容有：

- 怎样访问系统。
- 如何输入和执行命令。
- 如何熟练地操纵目录和文件。

(二) 访问系统

★一般介绍。

声明：如果你是第一次访问该操作系统的用户，那么请用 setup 注册登记。为得到有关初次使用你的系统的信息，请参考“3B2计算机的所有者/操作员手册”。

访问 UNIX 操作系统被叫做 login。这种系统注册方式是 UNIX 所必须的特殊规程。反之，退出系统注册也是所需的特殊规程。对于每一个用户来说注册登记是与系统相联系的必要方式。在你进入系统之前，你必须分配一个注册标识并通知系统管理员。这一注册标识应包括如下信息。

- 注册名字。
- 用户标识符 (uid)。
- 用户组标识符 (gid)。
- 口令字。

上述这一信息将被纳入用/etc/passwd 命名的文件中。用户组标识符存放在用/etc/group 命名的文件中。

当你是第一次访问系统时，有一些基本操作需要逐步熟悉。这些操作是：

- 为你的终端设置系统选择。
- 进入或退出打印操作。
- 建立或改变你的命令执行环境。

上述的这些基本系统操作将在下面章节中进一步详述。

★进入注册

注册之前，你必须先打开你的终端电源并且正确地调整你的终端设备（设置制表符、波特速率等等）。当然你的终端应是已联接在系统上的。你还可能必须初始化联接你的终端到系统上去的通讯链路，为了你的终端与系统同步，可能需要按几下回车键（carriage return 注）或终止键（break），这取决于你的终端和通讯链路的类型。

注：符号<CR>在本手册中统一表示为回车，即（carriage return）。

另外，有关终端设备的特殊信息，请参考有关的终端设备手册。

好了，就算你终端已与系统建立起了初始化通讯，这时系统马上会提示出下面的信息之

一。

login:

or

Console Login:

你此时应将你的注册名键入，以回车符做为输入结束。注意，当你键入错误时你可使用特定的擦除符（#）和删除符（@）进行修改。擦除符（#）用于擦除字符，删除符（@）用于删除当前命令行。接着系统则要求你键入你的口令。

Password:

这时应送入你的口令，也以回车符作为结束。出自安全性的考虑，这时系统对你键入的

口令将不予以显示。作为系统管理功能的一部分，你可以做出不分配口令的选择。若不分配口令，那么当你建立了注册登记之后系统不向你索要口令。若你注册的终端是联接在主终端接口上，注册提示信息将是 **CONSOLE LOGIN**。还有，如果你以 **root** 注册（超级用户），系统提示符为井号（**#**）。

如果你已经正确地向系统注册了名字和通过了口令问答，系统将回答几个日期信息。正常系统提示是美元符（**\$**），除非你是超级用户或改变了提示符。若你送入的注册名或口令不正确，系统将回答：

login incorrect

login:

当你成功地完成了注册工作，（也就是接收到（**\$** 或（**#**））你 就被系统安置在你的注册目录中了。这个目录也叫做你的家目录。**cd** 命令（更换目录）参数缺省时，一般是你被安置在家目录中。**shell** 变元 **HOME** 定义了 **cd** 命令缺省值。**shell** 变元 **PS1** 规定了最初的 **shell** 提示（一般是美元符标记）。有关 **shell** 变元的更详细内容将在本章的末尾标题为“建立与改变命令执行环境”一节里叙述。

★退出注册

退出之前，你应当在 **shell** 命令态上。这个意思是说你接收到了初态 **shell** 提示（美元符标记）。退出注册的动作是键入一个 ASCII 码的 EOF 符号。在许多终端上 EOF 符是由 **CONTROL** 键及 **d** 键同时按下而产生的。这个动作称为 **control-d**。系统将重新回答一个 **login:** 提示，这表明 **control-d** 已被系统接收。这时终端就可以关掉了。

★设置系统终端选择

• 一般介绍

系统如何响应你的终端，取决于各种终端选择的设置。所有当前的系统选择通过命令行都能输出来看一看：

stty -a<CR>

涉及 **stty** 命令的更完整内容请参阅第四章及命令手册 **tty** 命令部分。

对于许多终端来讲，下述命令行的任一个即可建立一组完善的系统选择。

stty ek echo -lcase tabs<CR>

or

stty sane<CR>

• 擦除符号选择

标准的 UNIX 系统擦除符号是井号符（**#**）在用 **<CR>** 结束一个输入行之前，按照一个井字符擦除一个其他字符的约定使用井字符。关于使用擦除符的进一步描述可见本章后面输入命令一节中修改、删除命令行部分。

对一个终端而言系统选择的典型选取是回应和擦除。下面的命令行将建立这种能力。

stty echoe<CR>

如果你更喜欢使用 **BACK SPACE** 键来作为擦除符号，那么可以输入下述命令行。注意

音调符 (^) 只用于 stty 命令, 且定义跟随的字符为控制字符。

`stty erase ^h <CR>`

• 打印机和终端延迟动作选择

在输出某些字符序列时, 因打印机机械配置或终端各自的不同特性, 故难于保持与系统输出的同步。尤其是当输出的序列中包含有回车、行给进、水平移位、垂直移位、退格或格式给进都需要有相应的延迟以完成特定的动作。根据你所具有的打印机或终端的类型, 你可能需要设置一些或所有的这些延迟。确定这些延迟的 stty 命令选择见下述。

- 1, 回车延迟: `cr0`, `cr1`, `cr2` 和 `cr3`。
- 2, 行给进: `nl0` 和 `nl1`。
- 3, 水平移位延迟: `tab0`, `tab1`, `tab2`, `tab3`。
- 4, 垂直移位延迟: `vt0` 和 `vt1`。
- 5, 格式给进延迟: `ff0` 和 `ff1`。
- 6, 退格延迟: `bs0` 和 `bs1`。

一般情况下, 如果在你的打印机上输出内容发生变形, 那你就很可能需设置延迟或制表符。初态不加延迟。下述一个简单命令行为上面各种动作设置最长的可能的延迟。

`stty cr3 nl1 tab3 bs1 ff1 vt1 <CR>`

★使用打印机

• 一般介绍

驱动一台打印机的一种经常的设备配置是把打印机联接在你的 CRT 终端的外部接口上。其它的配接方式也是可能的。比如, 将打印机联接到 3 B 2 主机接口上做为一种系统设备也是一种可行的配接方法。作为一个系统设备, 打印作业能够放在被打印文件清单中, 在软件控制下打印作业依照先来先为之服务的原则提交给打印机去输出的。这种处理方式称为假脱机 (Spooling) 方式。打印作业以假脱机方式送给打印机, 用户终端这时即可自由地做其它事情。

注意, 这里关于使用打印机的讨论仅涉及联接到用户 CRT 终端上的打印机。打印机配置作为一个系统设备实现假脱机作业, 由相应的选择命令来管理, 不是基本实用程序包的内容。

• 在终端键盘上操作进入或退出打印机。

你必须能够用终端键盘使计算机能够产生你的终端屏幕信息的硬拷贝, 使一台联接在你的 CRT 终端上的打印机能进行打印的标准方式为键入 `CONTROL-r`。使它不进行打印的标准方式为键入 `CONTROL-t`。

特定专用键盘配置上有一特殊功能键 (或与之相类似的键) 叫打印专用键 (`PRINTER ON LINE`)。这类特殊功能键如同一个交替式活动开关使打印机能进行或不进行打印。打印机的控制通常独立于命令行。键的操作也不破坏命令行。确定如何正确地进入或退出打印机, 请参考关于你的终端及打印机手册。

• 设置打印机制表停

只要执行了 `tabs` 命令就在一打印机上设置了制表停。打印机必须能够设置制表停。下面的命令行在打印机上设置缺省的制表停（每次跳 8 个符号）

```
tabs<CR>
```

确定如何正确地在你的打印机上设置制表停，请参考关于你的终端及打印机的手册。

★建立或改变命令执行环境

• 一般介绍

当你在系统上注册时，缺省约定文件（`/etc/profile`）就定义了命令执行环境。缺省约定是 UNIX 操作系统的一部分，它能被更改，这是系统管理的功能。由此可见，缺省约定文件（`/etc/profile`）随系统变化而改变。图3-1所示即一典型的缺省约定文件。对于每一个用户来说均能通过附加一个文件名为 `.profile` 的文件到注册目录中，改变这个缺省约定文件。这个文件中的内容早在用户收到初始 shell 提示（`PS1` 的值）之前予以处理的。shell 提供的

```
# @(#)profile.sh 1.4
trap "" 1 2 3
export TZ LOGNAME
TZ = EST5EDT
case "$0" in
-sh | -rsh)
    trap: 1 2 3
    cat /etc/motd
    trap "" 1 2 3
    if mail -e
    then echo "you have mail"
    fi
    if [ $LOGNAME != root ]
    then
        news -n
    fi
    ; ;
-su)
    ;
    ; ;
esac
trap 1 2 3
```

图3-1 典型的3B2/etc/profile文件

```

# Set default of cd command.
HOME = /f1/fred
# Set command search path.
PATH = : $ HOME/bin:/bin:/usr/bin:
# Set primary shell prompt.
PS1 = huh?
# Set terminal type
echo "What terminal are you using?"
1 = dumb
2 = 450
3 = 300
4 = 5410
Enter number corresponding to your terminal type: \c"
read term
case $term in
    1) TERM=dumb;stty cr2 nl0 -tabs;;
    2) TERM=450;stty cr2 nl0,tabs;;
    3) TERM=300;stty cr2 nl0,tabs;;
    4) TERM=5410;stty cr0 nl0,tabs;;
    *) echo "$term unknown, TERM set to dumb";
       TERM=dumb;stty cr2 nl0 -tabs;;
esac
#Make environment variables global
export HOME PATH PS1 TERM
#Output system date and time
date

```

图3-2 profile文件实例

一些基本变量的定义即确定了某些命令是如何被执行的。这些 shell 变量有：

- **CDPATH**: 定义了搜索 cd 命令自变量的路径。
- **HOME**: 不带自变量执行 cd 命令, 按 HOME 的值去改变目录。HOME 的值是由 login 设置且不可改动。
- **IFS**: 定义内部域分隔符, 这些符号是 shell 初始化设置的, 包括空格、制表符和换行符。
- **LOGNAME**: 当你在系统上注册时设置你的用户名。这个变量用于需要用户注册名的 shell 程序中, 结果该程序即可为任何执行它的用户而工作。
- **MAIL**: 定义了一个文件(定整路径名), 在这个文件中你可以收到来自其它用户的邮件。
- **PATH**: 定义了为你的输入命令而查找的目录路径。典型的变量值包括 /bin 和 /usr/bin 路径。
- **PS1**: 定义了最初的 shell 提示。最初的 shell 提示是美元符 (\$)。如果你以 root 注

册, 则初态的 shell 提示为井符 (#)。

- **PS2**: 定义了第二次的 shell 提示。最初的二次提示被置为 (>) 符。二次提示的意思是命令的执行需要索取附加的输入信息。

- **TERM**: 为特定的程序 (如屏幕编辑) 定义你的终端类型。

- **TERMCAP**: 定义了一个文件的路径名, 它指出各种类型终端的能力。

- **TZ**: 在一特定的命令输出中, 时间区变之用于定义如何表示时间 (日期), 这些 shell 变量的定义以及任何其它命令及用户定义的参数在 . profile 文件中设置。

• 缺省约定文件

图3-1所示为一典型的/etc/profile 文件。该文件在所有用户注册时执行。

图3-2所示的为一 . profile 实例。在这个例中所用的路径名 结构与图2-1中所示一致。

你可以通过 ed 命令在你注册的目录中建立你的 . profile。在图3-2中行首带有井符的是注解行。这些行可被删除。如果你希望在你的 . profile 文件中含有注解行, 请记住如果井是擦除符应在井字符之前加上反斜线符 (\)。

(三) 输入命令

★一般介绍

在 UNIX 操作系统中执行和进入命令态即要用到 shell。在你接收到最初态的 shell 提示 (一般当你没有改变 PS1 值时提示为 \$ 符) 之后, 你就可以输入一个命令行了。这个最初的 shell 命令提示就说明你已在 shell 命令级上了。

命令行的格式称为文法。命令行的结束符即回车符也是文法的一部分。它表示命令行开始被执行。当一个命令行执行结束时, 就又会显示 shell 初态提示。在执行一命令行之前 (按下回车键以前) 命令行是可以被修改或删除的。终止一个正在执行处理的命令行的方法也被提供。这些及其它的一些考虑将在下面段落中进一步描述。

★命令行文法

shell 命令行的一般格式 (文法) 是由一个或多个字组成的符号串并用回车符结束。一个命令行均由三个主要部分组成。

(1) 命令名。

(2) 命令变量。

(3) 命令行结尾符 (回车)。

命令行的第一个字总是命令名。命令变量是命令执行所必须提供的信息, 它可由一些字组成。表示命令行格式的标准方法是用 [] 将选择变量括起来。(注意, 在本手册中对于命令行文法的其它约定, 将在第四章 “基本实用程序包” 中介绍。命令变量大多数就是文件名。并非所有的命令均需要变量。在本册中命令行尾的回车符用 <CR> 来表示)。

下面是一个命令行的例子。它打印以 letters 命名的一个文件内容。在这个例子中文件名 letters 和 -08 是变量, pr 是命令名。-08 变量告诉打印命令在打印时每行空 8 位 (从左起

缩进 8 位)。

```
<pr -08 letters<CR>
```

命令还能在一行上以成组方式进入，每个命令之间用分号(;)隔开。下面的例子是输出行目录的路径名并且在 CRT 上显示相应目录清单。

```
pwd;ls<CR>
```

上述例子中应注意它们均没有使用任何变量。响应以上命令行如下。

```
/f1/fred
```

```
letters
```

```
newfile
```

★修改和删除命令行

在命令行键入结束符之前，对该行是可以进行修改的。修改一个命令行有两种方法，那就是用擦除符和删除符，(在你未重新定义时分别使用井符和@符。)删除符(@)是删除当前命令行。擦除符(井)是仅擦除前面的一个字符，一系列的擦除符删除其前面的同样多个字符直到当前行的开始。例如系统读入 cksjff井井井井at file7井1即解释为：

```
cat file1.
```

注意：当你希望写入擦除符或删除符作为你的数据内容时，则必须键入反斜线(\)作为转义符。例如，在你输入的数据中希望写入一个(井)符则必须写成(\井)。

★终止一个活动的命令

大多数正在处理的命令或程序均能采用下述的任一方式予以停止。

- Delete (DEL)
- Break (BREAK)
- Rubout (RUB OUT)

当一个命令(进程)在后台被执行时，标识该进程的进程号被输出。终止后台进程要用到 kill 命令。被终止进程的进程号是 kill 命令的执行变量。仅进程的所有者(或一个超级用户)才能终止一活动的命令(进程)。注意所有你创建的活动的后台进程能用 kill 0 命令来终止。kill 0 命令不需要其它的变量。kill -9 [进程号]命令真正结束一个进程。

★命令输出转向

• 一般介绍

命令的标准输出被定向到你的终端。命令的标准输出也可转向到一个文件或另一个命令上。输出的每一种转向将在下面的段落中叙述。

• 输出转向到一个文件

命令的输出通过转向到一个文件，而能被保存(为将来存取而贮存起来)。提供了两种技术，即写到一个文件中或附到一个文件后面。这两种技术类似之处在于，这个文件如果以前不存在，那么就都去建立它。不同之处在于如果文件已存在，则写入是指重写(删和写)这个文件。而附加则是补充这些新的输出在这个文件的末尾。

例如下面的命令行即是两个文件的内容合并建立第三个文件。

```
cat file1 file2>file3<CR>
```

而下面的命令行与上面的例子产生相同的结果。

```
cat file2>>file1<CR>
```

区别在于并没有产生一个单独的文件，而只是将文件2附加到文件1尾部。记住这二者之区别是至关重要的。符号(>)是直接将其输出送到其后面被命名的文件中。命令>NAME将通过系统去建立这个被NAME命名的空文件而不关心该文件以前是否存在。如果这个文件是已存在的那么这个文件的内容被重写。符号(>>)是将输出附到后面被命名的文件尾部。>>NAME，如果文件name不存在就建立它，如果文件name已存在，该文件不被冲空。

• 输出命令定向到另一个命令

把一个命令的输出定向到另一个命令上，我们称为管道传送。通过在两个命令之间使用(|)符把一个命令的输出传输给另一个命令。管道的功能提供了把一系列孤立命令联合成更大的更复杂的功能的方法。管道技术不需要建立中间文件，而孤立命令的暂时存贮空间必须使用中间文件。

用于计算当前目录上文件和目录的数目时，可考虑用下述孤立命令。

```
ls>temp<CR>
```

```
wc -l temp<CR>
```

但在下面的命令行中，Ls命令的输出用管道传给了wc命令，进行目录(行)统计。这就排除了对临时文件的需求。

```
ls | wc -l<CR>
```

★执行后台进程

• 一般介绍

所谓进程是一个或多个命令的执行。后台处理可以在另一个进程正在运行时使你的终端用于其它任务。你可以开始执行一道命令(进程)并立即返回到shell命令级上，这就需要在命令行的末尾(在回车之前)加入一个符号(&)。例如，打印一个以SAMPLE命名的文件并且转向输出到一个以PAGE命名的文件。这时后台进程将由下面的命令行启动。

```
pr sample>page&<CR>
```

系统回应一个进程号随后即会有美元符提示。在后台进程活动时，记录下与命令执行相关联的进程号是非常重要的。每条命令均被标识了相应的进程号。一后台进程的进程号被用于确定进程何时结束。进程号也可用于终止一个特殊进程。

在后台运行时，如果被执行的命令输出数据给终端(标准输出)应当被转向到一个文件上。如果输出没有转到一个文件上，那么输出就将到终端并且与其它正显示的数据相合并。

• 确定后台进程的完结

当一个命令在后台执行且已完成执行时，系统并不给出提示。看命令是在执行或是已完成的唯一方法是询问进程状态，使用PS命令可报告分配给用户的所有进程的状态。特殊地，下述命令行报告了特定的用户所有的活动进程。