

UNIX 分时 系统程序员手册

(第一卷)

《计算机研究与发展》编辑部

译

UNIX 系统目前已成为国际上最著名的分时系统之一。由于它简洁、通用和使用方便,因此科学研究、教学和商业等各个领域得到了广泛的应用。目前它已在16位小型机和16位微型机上占统治地位,并在8位微型机、32位超级小型机和几种中大型计算机上开始普及。

在近几年工作中,我们接触了若干种小型机和微型机上运行的UNIX及其变种,主要有UNIX(V7)、ONIX、XENIX、VENIX、UniPlex、ZEUS、UNIFLEX、CROMIX等。这些系统基本上都是美国贝尔实验室UNIX系统第七版或从第七版演变来的。

UNIX系统第七版主要包括有UNIX操作系统(UNIX系统核心)、C程序设计语言、Fortran77、shell命令解释程序及有关调试、正文编辑、资料准备、格式加工等一系列系统软件和应用软件。有关这些软件的设计和实现原理、使用方法等基本上包括在由贝尔实验室发行的《UNIX程序员手册》(第七版)中。由于工作的需要,由中国科学院计算机技术研究所《计算机研究与发展》编辑部组织几个单位合作翻译了该手册。

《手册》原文分两卷三册。第一卷包括:命令和应用程序、系统调用、子程序、特别文件(设备)、文件格式、游戏、杂类和系统维护等八个部分,主要介绍了UNIX的使用,是UNIX系统用户(一般用户和系统开发者)不可缺少的参考材料。第二卷包括38篇论文,这些资料涉及到UNIX系统总体介绍、正文编辑、资料准备、程序设计、支持工具和语言以及它的实现原理和维护方法。资料的编撰者都是这些子系统的设计和实现者。所以每篇论文都比较简炼而又系统地介绍了这些子系统的概貌、设计原理、实现过程和特点,以及和整个UNIX系统的关系。因此,这是学习、研究UNIX系统的科技、教学、生产人员及大学高年级学生的必读资料。对于一般系统使用者来说,这些资料也给他们使用和进一步开发这个系统提供了背景和基础。就目前我国进口的所有UNIX系统及其变种来说,所提供的资料中,上述材料以“参考手册”、“用户手册”、“实用手册”、“管理员手册”等各种名目进行了重新编排,除了与各个具体系统有关的部分略有不同外,基本上都超不出上述《手册》所提供的內容。因此,这份材料无论对标准的UNIX还是UNIX变种都是适用的。

两卷全部译文约100万字,第一卷30万字,该卷由孙玉方、刘日升、孔宪文、吴斗发、田勇等同志译,刘日升、孙玉方等同志校;第二卷约70万字,由孙玉方、刘日升、陈华璞、倪惜珍、丁茂顺、肖连华、杨均、肖廷廷、同得一、李建国、王旭、王铁生、王欣欣、于晓迪、姚杰、杭诚方、章桂明、夏祥生、董立群等同志译,刘日升、孙玉方、徐联芳、方裕等同志校,最后由计算机研究与发展编辑部总审校定稿。

由于译文量大,又涉及整个UNIX系统的各个方面,所以尽管我们花了很多力量对译文进行审校、统一,但疏漏之处在所难免,恳切地希望读者给予批评指正。

最后,对仲萃豪同志的指导、杨美清同志及有关领导同志们的支持表示感谢。

目 录

第一卷引论.....(1)

1. 命令和应用程序.....(7)

intro	命令介绍(7)
ac	注册记帐(8)
adb	调试程序(8)
ar	档案文件和库文件维护程序(13)
arll	与PDP-11兼容的档案和库维护程序(14)
arcv	把档案文件转换成新的格式(15)
as	汇编程序(16)
at	在以后的一个时间执行命令(16)
awk	模式扫描和处理语言(17)
basename	删去文件名的前后缀(19)
bc	任意精度的算术运算语言(20)
cal	打印日历(21)
calendar	提醒服务(21)
call	打电话(22)
cat	串连和打印(22)
cb	C程序整理程序(22)
cc	C编译程序(22)
cd	改换工作目录(24)
chmod	改变属性(24)
chown	改变文件主人或用户组(25)
ciri	清除i节点(25)
cmp	比较两个文件(26)
col	过滤程序反向行馈给(26)
comm	选出或删除两个已排序文件的公共行(27)
cp	复制(27)
cron	时钟系统进程(28)
crypt	加密/解密(28)
cu	调用UNIX(29)
date	打印并设置日期(30)
dc	台式计算器(31)
dcheck	文件系统目录一致性检查(32)
dd	转换并复制文件(33)
deroff	删除nroff, troff, tbl以及eqn构造(34)
df	空闲的盘空间(35)
diff	不同文件的比较程序(35)
diff3	三路不同文件的比较(36)
du	总计盘的使用情况(37)
dump	增量文件系统转储(37)
dummdir	打印出转储磁带上的文件名(38)
echo	回应自变量(39)

ed	 文本编辑(39)
eqn	 数学公式排版(44)
expr	 把自变量当成表达式(46)
f77	 Fortran77 编译程序(47)
file	 确定文件类型(49)
find	 找文件(49)
graph	 画图(50)
grep	 在文件中查找一种模式(51)
icheck	 文件系统存储一致性检查(53)
iostat	 报告 I/O 的统计数字(54)
join	 关系数据库的操作符(54)
kill	 以极端方式终止一个进程(55)
ld	 连接编辑程序(55)
learn	 UNIX的计算机辅助教学(57)
lex	 词法分析程序的生成程序(58)
lint	 C 程序的检验程序(58)
ln	 建立一个链(60)
login	 注册(60)
look	 在一排序文件中找一些行(61)
lorder	 找出一目标库的次序关系(61)
lpr	 行式打印机的假脱机程序(62)
ls	 列出目录内容(62)
m4	 宏处理程序(63)
mail	 在用户之间发送或接收信件(65)
make	 维护程序组合(66)
man	 打印这个手册的几节(68)
mesg	 允许或拒绝消息(69)
mkdir	 建立目录(69)
mkfs	 构造一个文件系统(69)
mknod	 建立特别文件(70)
mount	 安装和拆卸文件系统(71)
mv	 移动或重新命名文件(71)
ncheck	 根据 i 节点号产生名字(72)
newgrp	 向一新用户组注册(72)
nice	 以低优先权运行(73)
nm	 打印名字清单(73)
nohup	 运行一个不受挂断信号影响的命令(74)
number	 把阿拉伯数字转换为英文(74)
od	 八进制转储(74)
passwd	 改变注册口令(75)
plot	 图形过滤程序(75)
pr	 打印文件(76)
prof	 显示直方图(profile)数据(77)
ps	 进程状态(78)
pstat	 打印系统实情(79)
ptx	 置换索引(81)

pwd		工作目录名字(82)
quot		概述文件系统所有权(83)
ratfor		正规的Fortran变种(83)
rc		Ratfor编译程序(84)
refer		在资料中寻找并插入文献引用(85)
restor		增量文件系统恢复(86)
rev		颠倒文件的行(88)
rm		删除(解除联结)文件(88)
sa		系统记帐(88)
sdb		C符号调试程序(89)
sed		字符流编辑程序(92)
sh		命令语言(94)
size		目标文件的大小(100)
sleep		停止执行一段时间(100)
sort		排序或合并文件(101)
spell		查找拼写错误(102)
spline		插值平滑的曲线(103)
split		把文件分成各段(104)
strip		消除符号和浮动位(104)
struct		结构 Fortran程序(104)
stty		设立终端任选(105)
su		暂时替换用户id(107)
sum		合计并计算文件的块数(107)
sync		更新专用块(107)
tabs		设置终端制表符(108)
tail		传递文件的最后部分(108)
tar		磁带存档程序(108)
tbl		关于nroff或troff的格式表(110)
tc		照相排版仿真程序(111)
tee		管道组装(112)
test		条件命令(112)
time		命令计时(113)
tk		Tektronix 4014的分页程序(113)
touch		更新该文件的最近修改日期(113)
tp		管理磁带档案(113)
tr		字符翻译(114)
troff		文本格式化与排版(115)
true		提供真值(116)
tsort		拓扑排序(116)
tty		取终端名(116)
uniq		报告文件中重复的行(116)
units		变换程序(117)
uucp		unix到unix的复制(117)
uudiff		机器之间比较目录(118)
uulog		uucp用户记录询问(119)
uux		unix到unix的命令执行(119)

wait	等待进程完成(120)
wall	写给所有用户(120)
wc	字计数(120)
who	谁在系统中(120)
write	写给其它用户(121)
xsend	保密信件(121)
yacc	又一个编译程序的编译程序(122)
2. 系统调用	(122)
intro	系统调用和错误编号介绍(122)
access	确定文件的可存取性(125)
acct	开始或停止记帐(125)
alarm	在一个指定的时间后发信号(125)
brk	改变内存分配(126)
chdir	改变缺省目录(126)
chmod	改变文件的属性(127)
chown	改变文件主和用户组(127)
close	关闭一个文件(127)
creat	创建一个新文件(128)
dup	复制一个打开文件的描述字(128)
exec	执行一个文件(129)
exit	终止一个进程(131)
fork	产生一个新进程(131)
getpid	取得进程标识(131)
getuid	取得用户和用户组标识(131)
indir	间接系统调用(132)
ioctl	控制设备(132)
kill	发信号给一个进程(132)
link	联结一个文件(133)
lseek	移动读/写指针(134)
mknod	建立一个目录或特别文件(134)
mount	安装或拆卸文件系统(135)
mpx	创建和处理多重文件(135)
nice	置程序优先数(138)
open	用读或写方式打开(138)
pause	暂停直到收到信号为止(138)
pipe	创建一个进程间的通道(139)
profil	执行时间直方图(139)
ptrace	进程跟踪(140)
read	读一个文件(141)
setuid	置用户 ID 和组 ID(141)
signal	捕捉或忽略信号(141)
stat	得到文件状态(143)
stime	设置时间(144)
stty	设置和取出终端属性(144)
sync	更新专用块(146)

time		得到日期和时间(147)
times		得到进程时间(147)
umask		设置文件创建属性屏蔽(148)
unlink		删除目录登记项(148)
utime		设置文件时间(148)
wait		等待进程结束(148)
write		写一个文件(149)
3. 子程序		(149)
intro		库函数介绍(149)
abort		发生一个故障(150)
abs		整数的绝对值(150)
assert		程序验证(150)
atof		把 ASCII 码 转换为数字(151)
crypt		DES 编 密 码(151)
ctime		把日期和时间变 换 成 ASCII 码(151)
ctype		字符分类(152)
dbm		数据库子程序(153)
ecvt		输出转换(154)
end		程序中的最后存储单元(154)
exp		指数, 对数, 方幂, 平方根(155)
fclose		关闭或冲刷字符流(155)
ferror		字符流状态询问(155)
floor		绝对值, 低限, 高限函数(156)
fopen		打开一个字符流(156)
fread		有缓冲的二进制输入/输出(156)
frexp		分离尾数和指数(157)
fseek		重新定位字符流(157)
gamma		伽马 函 数 的 对 数(157)
getarg		Fortran 的 命 令 参 数(158)
getc		从字符流中取得字符或字(158)
getenv		环境名的值(159)
getgrent		取一组文件项(159)
getlogin		取注册名(159)
getpass		读一个口令(160)
getpw		从UID中取名字(160)
getpwent		取得口令文件项(160)
gets		从一个字符流中取字符串(161)
hypot		欧几里德距离(161)
j0		贝塞尔函数(161)
l3tol		进行 3 个字节整数和长整数之间的转换(162)
malloc		主存分配程序(162)
mktemp		产生一个唯一的文件名(162)
monitor		准备执行直方图统计(163)
nlist		从名字表中得到登记项(164)
perror		系统错误消息(164)
plot		图形接口(164)

popen	启动输入到一个进程或从一个进程输出(165)
printf	格式化的输出转换(165)
putc	把字符或字放到一个字符流上(166)
puts	把一个字符串放到一个字符流上(167)
qsort	快速排序(167)
rand	随机数生成程序(168)
scanf	格式化的输入转换(168)
setbuf	把缓冲赋与一个字符流(169)
setjmp	非局部转移(170)
sin	三角函数(170)
sinh	双曲函数(171)
sleep	暂停一段时间执行(171)
stdio	标准缓冲输入/输出程序包(171)
string	字符串运算(172)
swab	对换字节(172)
system	发出一个 shell 命令(172)
ttyname	找出一个终端的名字(173)
ungetc	把字符推回到输入流中(173)

4. 特别文件(173)

cat	照排机接口(173)
dc	DC-11通信接口(174)
dh	DH-11通信多路转换器(174)
dn	DN-11 ACU 接口(174)
dp	DP-11 201 数据-电话接口(174)
dz	DZ-11通信多路转换器(174)
hp	RH-11/RP04, RP05, RP06, RM03 活动磁头磁盘(175)
hs	RH-11/RS03-RS04 固定磁头 磁盘文件(175)
ht	RH-11/TE-16 磁 带接口(176)
kl	KL-11 DL-11 异步接口(176)
lp	行式打印机(176)
mem	主存储器(177)
null	数据接收器(177)
pc	PC-11 纸带读入机/穿孔机(177)
rf	RF11/RS11 固定磁头磁盘 文件(178)
rk	RK-11/RK03或RK05磁盘(178)
rp	RP-11/RP03活动磁头磁盘(178)
tc	TC-11/TU56 DEC磁带(179)
tm	TM-11/TU-10 磁带 接 口(179)
tty	通用终端接口(180)
vp	多功能打印机-绘图机(183)

5. 文件格式(184)

a.out	汇编程序和连接编辑程序的输出(184)
acct	执行帐户文件(186)
ar	档案(库)文件格式(187)
core	内存映象文件格式(187)
dir	目录格式(187)

dump	增量转储格式(188)
environ	用户环境(189)
filsys	文件系统卷的格式(190)
group	用户组文件(192)
mpxio	多路传输 i/o(192)
mtab	文件系统安装表(193)
passwd	口令文件(194)
plot	绘图接口(194)
tp	DEC/磁带格式(195)
ttys	终端初值数据(195)
types	基本系统数据类型(195)
utmp	注册记录(196)
wtmp	用户注册历史(196)
6. 游戏	(197)
arithmetic	提供数字论据的练习(197)
backgammon	游戏(197)
bcd	转化为古老的媒介(197)
ching	八卦书及其它格言(197)
hangman	猜单词(198)
maze	产生迷宫问题(198)
quiz	测验你的智力(198)
wump	追捕 wumpus 的游戏(199)
7. 宏程序包和语言约定	(199)
ascii	ASCII 符号集图表(199)
eqnchar	对 eqn 特殊符号的定义(199)
greek	扩充的 TTY-37 铅字库的图形(201)
hier	文件系统的层次结构(201)
man	排版手册的宏(204)
ms	宏格式编排底稿(205)
term	约定的名字(207)
8. 系统维护	(208)
bproc	UNIX/32V 启动(208)
crash	系统出故障时做的事情(209)
cron	系统时钟进程(212)
fget.demon	系统进程 (daemon) fget(212)
getty	设置终端方式(212)
init	进程控制初始化(212)
lpd	行印机系统进程(213)
makekey	产生加密关键字(214)
sticky	具有持久性正文的可执行文件(215)
sysgen	用发行磁带生成 UNIX/32V 系统(215)
update	周期性地更换专用块(217)

第一卷 引 论

本卷描述了UNIX/32V系统普遍适用的特性。它并不打算提供有关UNIX/32V操作系统，它的设施，或它的实现的展望或指导信息。在第二卷中包含了有关这些课题的各种资料。尤其是，作为一种综述请看Ritchie和Thompson编写的“UNIX分时系统”，作为一种辅导材料请看Kernighan的“UNIX初学者入门”。

在其介绍的范围内，本卷力图及时、完整和简洁。后两点目标是矛盾的，显而易见的地方常常因为要简略而不加说明。这就意味着每一个程序是怎样就说是怎样，而不是说它应该怎样。可以料想到的，这意味着许多节不久就会过时。

本卷共分八节：

1. 命令
2. 系统调用
3. 子程序
4. 特别文件
5. 文件格式和约定
6. 游戏
7. 宏程序包和语言约定
8. 维护

命令是规定由用户直接引用的程序，这与子程序不同，子程序是规定由用户程序调用的。命令一般驻留在目录/bin（表示二进制〔binary〕程序）中。为了节省/bin的空间，某些程序也驻留在/usr/bin中。由命令解释程序自动检索这些目录。

系统调用是进入UNIX管理程序的入口。在UNIX/32V中，系统调用接口与C语言过程调用等同；将在第二节中描述等价的C过程。

有一批子程序可以使用，它们将在第三节中介绍。容纳这些子程序的基本程序库在intro(3)中介绍。这些功能按C语言形式描述，但绝大多数同样也可用于Fortran。

第四节的特别文件讨论了真正牵涉I/O设备的每一个系统‘文件’的特征，在本节中的名

前言：VAX-11上的UNIX/32V操作系统提供了与PDP-11上的UNIX系统大体相同的功能。

我们对先于我们在上面开展工作的许多人表示感谢，尤其是要感谢G.K. Swanson, W.M. Cardoza, D.K. Sharma和J.F. Jarvis在为VAX-11/780实现这个系统时给予的帮助。

T.B.London, J.F.Reiser

第七版前言：虽然第七版不再出自于他们之手，但是Ken Thompson和Dennis Ritchie仍然是UNIX分时系统之父和导师。这里介绍的许多改进都带有他们的影响。其它许多人也对UNIX取得的进一步成功作出了贡献，在这些人中我们特别要感谢A.V.Aho, S.R.Bourne, L.L.Cherry, G.L.Chesson, S.I.Feldman, C.B.Haley, R.C.Haight, S.C.Johnson, M.E.Lesk, T.L.Lyon, L.E.McMahon, R.Morris, R.Muha, D.A.Nowitz, L.Wehr和P.J.Weinberger。我们还特别赞赏T.A.Dolotta, A.G.Fraser, J.F.Maranzano和J.R.Mashey的有效的忠告和批评，我们同样不会忘记已故Joseph F.Ossanna的重要工作。

(B.W.Kernighan, M.D.McIlroy)

字指的是硬件的DEC设备名而不是特别文件本身的名字。

第五节的文件格式和约定提供了特定类型文件的结构方面的资料，例如，给出了装配程序和汇编程序的输出形式。仅仅由一个命令使用的文件，例如汇编程序的中间文件不加以考虑。

游戏归入第六节，使它们不能损害第一节中比较固定的信息。

第七节是为了以各种各样专门化的语言进行书写所必需的各种信息的集合：字符编码、排版用宏程序包等等。

第八节维护部分讨论了规定不是由普通用户使用的那些过程。这些过程常常涉及第一节中命令的用法，在第一节中通过用1M作为标记，力图把具有维护特色的命令单独列出来。

每一节由一些占一页左右的独立的项组成。该项的名字在该页左上角，附带有节号，有时还有一个子段的特征字母，例如，图形方面的是1G，而数学程序库是3M。在每节中的项是按字母排序的。每一项的开始页号是1；象这样的资料进行连续编号是做不到的，这份资料以许多不同的形式被重新出版。

所有的项基于一个共同的格式，并不是所有的子段都要出现。

名字子段列出该项涉及的命令以及子程序的精确名字，并且为此目的给出一个非常简短的说明。

格式概括了将被说明的程序的用法。使用了少量约定，尤其是在命令子段中：

黑体字看成是文字，完全象它们出现的那样印出。

方括号[]括住自变量，就表示该自变量是任选的。当一个自变量作为‘名字’给出时，它总是指文件名。

省略号‘...’用于表明各种自变量样本可以重复。

最后的一种约定由命令本身使用。以一个负号‘-’打头的自变量常常意味着是某类规定任选的自变量，即使它出现在文件名可以出现的位置上也是如此。因此，以‘-’打头来作为文件名是不明智的。

说明子段详细地讨论眼前的题目。

文件子段给出了构成该程序的文件的名称。

参见子段给出了有关信息的线索。

诊断子段讨论了可能产生的诊断指示。确定为不言自喻的消息不列出来。

缺陷子段给出了公认的缺陷，有时还包括毛病，偶而，也描述了建议的解决方法。

在第二节中**汇编语言**子段含有PDP-11汇编语言系统接口。

本卷的开头是目录表，按节组织，在每节内部按字母顺序排列。还有从目录表中派生出来的置换索引。在每一个索引项中，它引用的书面记录的标题后面是括在园括号中的相应的节号。这很重要，因为在各节之间有很多重复的名字，这主要是由于只应用特定系统调用的命令所引起的。

怎样开始

该节粗略地说明你在UNIX/32V上起动所需要的基本信息：怎样注册和销注，怎样通过你的终端进行通信，以及怎样运行一个程序。参见第二卷中的“UNIX初学者入门”以便得到对于这个系统的更完整的介绍。

注册

你必须从一台合适的终端上调用UNIX。UNIX终端分成如下几种类型：TTY 43、GE Terminet 300, DASI 300S和450, 以及极大多数诸如Datamedia 1520或HP 2640那样的视频终端。你还必须有一个合法的用户名, 它连同电话号码一起可以从系统管理员那里获得。同一个电话号码为在所有标准速度上运行的终端服务。在建立了数据连接之后, 注册过程就取决于你所用的终端的类型了。

300-波特终端:

这样的终端包括GE Terminet 300, 以及绝大多数以普通调制解调器运行的显示终端。一般来说, 这些终端有一个速度开关, 它应该打在‘300’处(或者‘30’处, 如果用于每秒30个字符的话), 还有一个半双工/全双工开关, 它应该置成全双工。(这个开关常常需要改变, 因为许多其它系统需要半双工。)在建立连接时, 系统给出‘login:’, 你就打入你的用户名, 后面是一个‘返回(return)’键。如果你有口令, 那么系统就要求打入, 但是终端上的打印机不输出, 这样口令就不会出现。在你注册之后, ‘返回’、‘换行(new line)’或‘行馈给(linefeed)’键将给出完全一样的结果。

1200-和1500-波特终端:

如果有半双工/全双工开关, 那么置成全双工位置。当你已建立了数据连接之后, 系统会印出少量无用字符(以不适当的速度印出‘login:’消息)。按下‘break’(或‘interrupt’)键, 这是一个与速度无关的信号, 告诉UNIX正在使用不同速度的终端。然后, 系统这次以另一种速度印出‘login:’。继续按‘break’键, 直到清晰地出现‘login:’, 然后用你的用户名给以回答。对于TTY 37终端, 以及任何有‘换行’功能(它由回车和行馈给组合而成)的终端, 用‘new line’键来终止你打入的每一行; 否则, 用‘return’键来终止你打入的每一行。

硬连线终端:

硬连线终端通常以正确的速度开始, 可以一直到9600波特; 否则, 要应用前面的程序处理。

对于所有这些终端, 重要的是, 你要尽可能地用小写字体打入你的名字; 如果你打成大写字母, 那么UNIX会认为你的终端不能生成小写字母, 并且将把所有以后的大写字母转换成小写。

你已经成功地注册进系统的标志是shell程序将印出一个‘\$’给你(shell将在以后‘怎样运行一个程序’标题下加以描述)。

为了得到更多的信息, 参阅stty(1), 它告诉你怎样调整终端的工作状态; 也可以参阅getty(8), 它更详尽地讨论注册顺序; 参阅tty(4), 它讨论了终端I/O。

注销

有三种途径注销:

只要直接挂断电话。

给shell打入一个文件尾标记(EOT字符, control-d)就可注销。shell将终止并且再次出现‘login:’消息。

你可以给出login(1)命令作为另一个用户直接注册进入。

怎样通过你的终端进行通信

当你打入字符时，系统中的一个程序收集你的字符并且把它们保留在一个秘密的地方。正如上面在注册中描述的，直到你打入了一个‘return’（或‘newline’）键，这些字符才送给一个程序。

UNIX终端I/O是全双工的。它有完整的预读功能，这意味着你可以在任何时刻打入字符，即使当时有程序正印出东西给你。当然，如果你在终端输出时打入，那么其输出将与你输入的字符混杂在一起。不过，你所打入任何内容都会按正确的顺序保留和解释。预读的总数有一个限度，但是它是很大的并且不太会被超出，除非系统遇到了什么麻烦。当预读限度超过时，系统抛弃所有已保留的字符。

在打入的输入字符中字符‘@’将抹掉该行中前面的所有字符，这样就可以修正在一行上的错误。同样，字符‘#’抹去最后打入的字符。连续地使用‘#’将后退着抹去字符，但不会超越该行的头。如果在‘@’和‘#’前面置于‘\’，则可以把它们传送一个程序（这样，为了抹去‘\’你需要两个‘#’）。这些约定可以通过stty(1)命令加以改变。

就象ASCII‘删除(delete)’（或者‘擦去(rubout)’）字符那样，‘break’，或‘interrupt’键会引起一个中断信号，它们不传送给程序。一般来说，该信号会导致你正在运行的程序终止。其典型地用于停止一个你不想要的长长的打印输出。不过，程序可以作这样的安排：或者统统地忽略这些信号，或者当它发生时通知程序（而不是终止程序）。例如，编辑程序捕捉了中断并且停止它正在做的工作，而不是终止这个工作，以便中断可以用于暂停一个编辑程序的打印输出而不丢失正被编辑的文件。

通过打入ASCII的FS字符产生quit信号（FS出现在不同终端多地的许多地方，最通常地是作为control-\或者control-|）。它不但会使运行程序终止而且还用被终止进程的内存映像产生一个文件。quit（停止）对于调试是很有用的。

除了适应终端的速度外，有关是否带有换行功能的终端或者是否必须用回车及行馈给来模拟UNIX都力图做到灵活。在后一种情况下，所有输入的回车被翻译成换行字符（标准的行定界符），并且一个回车符和一个行馈给都回应给该终端。如果你进入了错误方式，stty(1)命令会搭救你。

在UNIX源程序中制表符自由地使用。如果你的终端没有配备制表功能，那么你可以准备让它们转换成输出时的空格，并且在输入时作为空格回应。该系统假定制表符设置成八个列。此外，stty(1)命令将设置或清除这种方式。而命令tabs(1)将在许多终端上自动地设置成制表停。

怎样运行程序；shell

当你已经成功地注册后，称为shell的一个程序收听你的终端。shell读入打进的行，把它们分裂成命令名和若干自变量，并且执行该命令。一个命令就是一个可执行的程序。shell首先在你的当前目录（参见下文）中寻找具有给定名字的程序，如果这里没有，则在系统目录中寻找。除了它们被保留在一个shell可以找到的目录中外，有关系统提供的命令并没有什么特别之处。

命令名字总是输入行的第一个字；它及它的自变量互相之间用空格隔开。

当一个程序终止时，通常shell将重新获得控制并且印出一个‘\$’给你，表示它已准备好接受另一条命令。

shell还有许多其它的功能，详见sh(1)。

当前目录

UNIX有一个按目录层次结构组织的文件系统。当系统管理员给你一个用户名时，他也为你建立了一个目录（通常用和你的用户名相同的名字）。当你注册时，你打入的任何文件名由缺省规则约定都在这个目录中。由于你是这个目录的所有者，因此你具有全部的读、写、修改或销毁其内容的许可。对于其它目录和文件你所具有的许可由它们的所有者允许给予你或拒绝给予你。其实，没有几个UNIX用户能保护他们的文件免遭其他用户的破坏，更不用说仔细阅读了。

为了改变当前目录（而不是你在注册时赋予的一组许可），可使用`cd(1)`。

路径名

为了引用不在当前目录中的文件，你必须使用路径名。完整的路径名以‘/’打头，‘/’是整个文件系统的根目录的名字。在斜线之后是包含下一个子目录的目录名（后面跟一个‘/’），直至到达最后的文件名。例如，`/usr/lem/filex`引用目录`lem`中的文件`filex`，而`lem`本身是`usr`的子目录，`usr`又直接发源于根目录。

如果你的当前目录有子目录，那么文件的路径名在那里以不带前缀的‘/’的子目录的名字开始。

路径名可以用于任何需要文件名的地方。

修改文件内容的重要命令是`cp(1)`、`mv(1)`和`rm(1)`，它们分别复制、移动（即重新命名）和删除文件。为了找出文件或目录的状态要用`ls(1)`。为了构造目录，参见`mkdir(1)`；而为了消除它们，参见`rmdir`（在`rm(1)`中）。

关于文件系统的更详细的讨论，请参见Ken Thompson和Dennis Ritchie的‘UNIX分时系统’。即使你不打算在这一级使用这个系统，粗略地浏览本手册的第二节也可能有用，这一节讨论系统调用。

书写一个程序

为了把一个源程序的正文记入UNIX文件，请使用编辑程序`ed(1)`。在UNIX中由C编译程序`cc(1)`和Fortran编译程序`f77(1)`提供了两种基本的语言。在通过编辑程序记入程序正文并且写到一个文件上之后，你可以把该文件作为一个自变量交给合适的语言处理程序。该语言处理程序的输出将保留在当前目录中名为‘a.out’的一个文件中（如果输出很重要，就使用`mv`命令把输出移到一个最近一段时间很少受影响的名字下）。

当你最终经过这个完整的过程而没有产生什么诊断消息时，可以把它的名字提交给shell，运行这个结果程序，以响应‘\$’提示。

你的程序可以象系统程序那样从命令行上接收参数，参见`exec(2)`。

文本处理

几乎所有的正文都是通过编辑程序`ed(1)`打入的。在一个终端上极频繁地用于编写正文的命令有`cat`、`pr`、`roff`和`nroff`，它们都在第一节中说明。

`cat`命令直接在终端上输出ASCII文本，根本不作处理。`pr`命令为文本编页，提供标题并且有用作多列输出的设施。`nroff`是一种精致的文本格式编排程序，使用时不加保护，所以需要深思熟虑，但是对于普通资料，它已经被驯服了，参见`ms(7)`。`roff`是一种较简单的文本格式编排程序，不需要太深思熟虑。

`troff`为Graphic Systems照排机预先准备资料，它很象`nroff`，并且经常对同样的源程

序文本加工。它用于产生这本手册。

状态询问

各种存在的命令向你提供各种有用的信息。who(1) 打印当前注册的用户清单。date

(1) 打印当前时间和日期。ls(1) 将列出你目录中的文件或给出有关特殊文件概况信息。

意外的事情

某些命令提供用户间的通信。即使你并未打算利用它们，但是有关这些最好要学会，因为其它某些人可以用它对准你呢。

为了与当前注册的另外的用户通信，要使用write(1)；mail(1)将留下一种消息，当另一个用户随后注册时，会通知他消息的存在。在本手册的书面记录也建议如果你是一个通信目标，那么怎样响应这两个命令。

当你注册时，在第一个‘\$’之前可能有当天的消息映入你的眼帘。

对第六版的修改

下面是一组重要的且基本上是不兼容的改变，它们将影响从PDP-11上的第六版转过来的老用户。这里并不打算列出所有新的设施，乃至所有次要的但容易弄坏的改变，而仅仅是最本质的改变，没有这些改变就几乎不可能做任何事情。

寻址文件

文件中的字节地址现在是长整数（32位）。普通的seek已由lseek(2)所替代。每一个包含有seek的程序都必须修改。stat和fstat(2)也类似地受到影响，因为文件的长度现在以32而不是24位度量。

汇编程序

该语言已不再使用。误用将遭到严厉的惩罚。

stty和gtty

这两个系统调用已经被彻底地修改了，参见ioctl(2)和tty(4)。

档案文件

由ar(1)产生的文件的格式已经修改过了。为了转换成新的风格，请用arcv(1)。

C程序设计语言，lint

初始化的语法在每一个初值前要有等号=，用花括号{ }围住复合的初始值；数组和结构现在老实地初始化。诸如+=和-=那样的运算符现在以相反次序书写成：+=和-=。这样就消除了在诸如x=-2，y=*p以及a=/*b那样的结构中的二义性可能。你肯定还要了解

长整数

类型定义

强制 (cast) (用于类型转换)

联合 (用于更老实的存储共享)

#include <文件名> (它在标准的地方检索)

程序lint(1)给单个C程序或者期望一起装配的一组C程序核实已废弃的语法并且进行强类型检验。它与转换工作无关。

Fortran

老的fc由f77替代,这是Fortran77的真正的编译程序,与C兼容。在本语言中有明显的改变,参见在第二卷中的‘A Portable Fortran 77 Compiler’。

字符流编辑程序

程序sed(1)适用于转换到新系统中遇到的这类大量重复的编辑工作,这很值得学会。

标准I/O

老的fopen、getc、putc复合体以及老的-lp程序包都已不能用了,甚至getchar也已作了修改。所有这些都已由清晰且高效的stdio(3)程序包所替代。要知道的第一件事是getchar

(3)在文件尾返回的整数EOF(-1)(它已经不可能是一个字节的值),518字节的缓冲区已过时,还有一种确定的FILE数据类型。

Make

程序make(1)根据为每个软件给出的“makefile”方法,以一种规则的方式处理软件的重新编译和装配。它重新构成的仅仅和输入文件的修改日期所必须指出的一样多。“makefile”将在你构造新的系统时给予你指导。

shell, chdir

F.L.Bauer曾经说过,Algol68是必须由每个计算机科学家攀登的埃佛勒斯(珠穆朗玛峰)。对于UNIX用户来说,shell也是如此。在终端上超出引用简单命令范围以外的事情都是不同的。甚至连chdir现在都已拼成了cd。你将需要长期而艰苦地研究sh(1)。

调试

adb(1)是远比调试程序db强得多的替代物。第一次,用户尤其是要关心在adb命令中/和?的区别,注意确保他所要求的x的值是真正的x,而不是等于某个自动的x的栈位移的某个绝对单元。你可以老是用‘真实的’名字_x阻止C的外部变量。

dsw

这是一个不为人所知,但已经由rm -ri取代的不可缺少的设施。

引导过程

不用说,都是不同的。参见本卷的第八节以及第二卷中的‘启动UNIX’。

1. 命令和应用程序

INTRO(1) (UNIX/32V)

名字

intro—命令介绍

说明

本节按字母顺序说明了可以公用的命令。在标题上有意地作了一些区分:

- (1) 通常使用的命令
- (1C) 与其它系统通信的命令
- (1G) 主要用于绘图及计算机辅助设计的命令
- (1M) 主要用于系统维护的命令

页下脚(译文中在标题右边一译者注)上的‘local’字表示该命令不准备作为全面推广,字‘UNIX/32V’表示该命令的部分或全部解释仅适用于数字设备公司(DEC)

VAX-11的UNIX/32V系统。

参见

第(6)节的计算机游戏。

前言中的“怎样开始”。

诊断

每条命令结束都返回两个字节的状况信息。其中一个是由系统提供的，它给出了命令结束的原因；另一个（在“正常”结束时）是由程序提供的，见 `wait` 和 `exit(2)`。前一字节为0表示正常结束；后一字节通常以0表示命令执行成功，如果不为0，则表示出了问题，比如错误的参数，不好的或者不可访问的数据，或其它使当前任务不能执行下去的原因。这后一位有几种称呼：‘出口码’（`exit code`），‘出口状态’（`exit status`）及‘返回码’（`return code`）。它仅在有特别约定时才给予解释。

AC(1M)

名字

`ac`—注册记帐

格式

`ac [-w wtmp] [-p] [-d] [people] .....`

说明

`ac`为在当前`wtmp`文件存在期间注册的每个用户打印输出连接时间。还产生总的时间。`-w`用来指出替换的`wtmp`文件。`-p`打印出每人的总时间；如没有这一任选，则仅打印出总的时间。`-d`从零点开始每昼夜打印一次。任何`people`都可以限定仅针对指定的注册名打印。如果没有指定任何`wtmp`文件，则用`/usr/adm/wtmp`。

记账文件`/usr/adm/wtmp`是由`init`和`login`维护的。但这两个程序都不创建这个文件，所以，如果该文件不存在，则不进行连接时间记帐。为了开始记帐，就应该创建长度为0的记账文件。另一方面，如果该文件老不受干扰，那么它将无止境地增大。所以要定期收集所需要的所有信息，把文件截短。

文件

`/usr/adm/wtmp`

参见

`init(8)`，`login(1)`，`umtp(5)`

ADB(1) (UNIX/32V)

名字

`adb`—调试程序

格式

`adb [-w] [objfil [corfil]]`

说明

`adb`是个通用的调试程序，它可以用来检测文件并且为UNIX程序的执行提供一个受控的环境。