

最新UNIX系统V丛书（共五册）

系统管理员参考手册

(四)

H

北京中国科学院希望高级电脑技术公司

目 录

intro(1).....	(6)
300,300s(1).....	(8)
4014(1).....	(9)
450(1).....	(10)
ar(1).....	(11)
at, batch(1).....	(12)
awk(1).....	(14)
bc(1).....	(16)
bfs(1).....	(18)
banner(1).....	(20)
basename,dirname(1)	(21)
bdiff(1).....	(22)
cal(1).....	(22)
calendar(1).....	(23)
cat(1).....	(23)
cd(1).....	(24)
chmod(1).....	(24)
chown, chgrp(1).....	(27)
cmp(1).....	(27)
col(1).....	(28)
comm(1).....	(29)
cp,ln,mv(1).....	(29)
cpio(1).....	(30)
crontab(1).....	(32)
crypt(1).....	(33)
csplit(1).....	(34)
ct(1C)	(35)
cu(1C).....	(36)
cut(1).....	(39)
date(1).....	(40)
dc(1).....	(42)
dd(1M).....	(43)
deroff(1).....	(44)
df(1M).....	(45)

diff(1).....	(46)
diff3(1).....	(47)
dircmp(1).....	(48)
du (1M).....	(48)
echo(1).....	(49)
ed,red(1).....	(49)
edit(1).....	(57)
egrep(1).....	(60)
enable,disable(1).....	(61)
env(1).....	(61)
ex(1).....	(62)
expr(1).....	(65)
factor(1).....	(67)
fgrep(1).....	(67)
file(1).....	(68)
find(1).....	(69)
gdev,hpd,erase,hardcopy,tekset,td(1G).....	(70)
ged(1G).....	(71)
getopt(1).....	(76)
getopts,getoptcyt(1).....	(78)
glossary(1).....	(79)
graph(1G).....	(80)
graphics(1G).....	(81)
greek(1).....	(82)
grep(1).....	(84)
gutil(1G).....	(84)
help(1).....	(86)
helpadm(1N).....	(87)
hp(1).....	(88)
hpio(1).....	(89)
id(1N).....	(91)
ipcrm(1).....	(92)
ipcs(1).....	(92)
ismpx(1).....	(95)
join(1).....	(96)
jterm(1).....	(97)
jwin(1).....	(97)
kill(1).....	(98)
layers(1).....	(98)

line(1)	(100)
locate(1).....	(100)
login(1)	(101)
logname(1).....	(104)
lp, cancel(1)	(104)
lpstat(1).....	(105)
is(1).....	(106)
machid, pdp11, u3b, u3b2, u3b5, vax(1)	(109)
mail, rmail(1)	(109)
mailx(1)	(112)
makekey(1)	(128)
mesg(1)	(128)
mkdir(1)	(129)
newform(1).....	(129)
newgrp(1M).....	(131)
newS(1)	(132)
nice(1).....	(133)
nl(1).....	(133)
nohup(1).....	(134)
od(1)	(135)
pack, pcat, unpack(1).....	(136)
passwd(1)	(137)
paste(1).....	(138)
pg(1)	(139)
pr(1)	(141)
ps(1)	(144)
pwd(1).....	(147)
relogin(1M).....	(147)
rm, rmdir(1)	(148)
sag(1G)	(149)
sar(1)	(150)
sdiff(1)	(152)
sed(1)	(153)
setup(1)	(155)
sh, rsh(1)	(156)
shl(1)	(159)
sleep(1)	(168)
sort(1)	(168)
spell, hashmake, spellin, hashcheck(1)	(171)

spline(1G)	(172)
split(1)	(173)
starter(1)	(174)
stat(1G)	(174)
stty(1)	(177)
su(1N)	(181)
sum(1)	(182)
sync(1M)	(183)
sysadm(1)	(183)
tabs(1)	(190)
tail(1)	(193)
tar(1)	(193)
tee(1)	(195)
test(1)	(195)
time(1)	(197)
timex(1)	(197)
toc, dtoc, ttoc, vtoc(1G)	(198)
touch(1)	(199)
tplot(1G)	(200)
tput(1)	(200)
tr(1)	(203)
true, false(1)	(203)
ttty(1)	(204)
umask(1)	(204)
uname(1)	(206)
uniq(1)	(206)
units(1)	(207)
usage(1)	(207)
uucp, uulog, uuname(1C)	(208)
uustat(1C)	(210)
uuto, uupick(1C)	(212)
uux(1C)	(213)
vi(1)	(215)
wait(1)	(221)
wall(1)	(221)
wc(1)	(222)
who(1)	(222)
write(1)	(224)
xargs(1)	(225)

引言

本手册描述了在AT & T的3B2计算机上运行的基本软件的命令。

其他几本书中有另外一些有价值的信息：

■《用户指南》(P-H)。介绍了UNIX系统的概貌并对正文编辑器的使用、自动可重复作业和信息传递作了介绍。

■《程序员指南》(p-H)。概述了UNIX系统的程序设计环境并对各种程序设计工具作了介绍。

■《程序员参考手册》(P-H)。描述了各种命令、系统调用、子程序、库及程序员所需的文件格式。

■《系统管理员指南》(AT & T)。提供了管理任务的过程及对管理任务的解释。

■《系统管理员参考手册》(AT & T)。描述了系统管理员使用的命令。

由于每一条命令都是单个的实用程序，因此所有命令都以字母顺序出现在本手册中“命令”这一节中。这一节中提供的实用程序的种类有：

1. AT & T窗口实用程序
2. 基本的网络实用程序
3. 盒式磁带控制实用程序
4. 目录及文件管理实用程序
5. 编辑实用程序
6. 基本实用程序
7. 图形实用程序
8. 帮助实用程序
9. 行式打印机假脱机实用程序
10. 内部进程的通讯
11. 性能测试实用程序
12. 安全管理实用程序
13. 拼写实用程序
14. 终端过滤器实用程序
15. 终端信息实用程序
16. 用户环境实用程序

安全控制实用程序是特意为美国用户提供的。

第一部分 命令

第一部分中的条目描述的是由用户直接调用的程序或者是由命令语言过程调用的程序，这与子程序有所不同，子程序由用户的程序调用。命令通常都驻在目录/bin下（表示二进制程序），另外有些命令驻在目录/usr/bin下。这两个目录可由命令解释程序shell自动搜索。在3B2计算机上运行的UNIX系统还有一个叫作/usr/lbin的目录，这个目录中，可包含一些局部命令。

命令后跟的数字主要是为了方便对照。命令后面带着(1),(1C),(1G)，一般表示这条命令在本手册中。（第一部分中那些适合于程序员使用的命令放在《程序员参考手册》(P—H)中)。命令后面带着(1M),(7),(8)表示这条命令在《系统管理员手册》(AT&T)的某个章节中。命令后面如果跟着(2),(3),(4),(5)，表示这条命令在《程序员参考手册》(P—H)的某个章节中。

命令部分中的每一条目都以单一的命字出现在每一页的上方。所有条目都按字母顺序排列，只有intro(1)例外，它排在第一条。某些条目也许描述了几个命令，在这种情况下，该条目也只出现一次，并按其“主名”的顺序排列。主名即出现在每一页上方的名字。“次名”就直接列在相关的主命令下。

所有条目的描述都遵照下列格式（不是每个条目都完全包括这些内容的）：

■名字 给出主名（在有些情况下，还有次名）并简单说明用途。

■格式 总结所描述命令的用法。在这部分中，用到一些常用的说明约定：

· 黑体字印的字符串要按它们出现的那样键入。

斜体字印的字符串表示可替换的自变量原型及在手册其他地方可找到的命令名字。

· 方括号 [] 括起来的自变量表示该自变量是任选的。当一个自变量原型是“name”或“file”时，总是指的是文件的名字。

· 省略号...表明前面的自变量可以重复。

· 最后的一种约定由命令本身使用。以负号(一)、加号(+)和等号(二)开头的自变量通常被看成是某种类型的规定自变量，即使它出现在文件名可以出现的位置也是如此。因此，用一、+、=来作为文件名的开头是不明智的。

■说明 讨论怎样使用命令。

■例子 适当地给出一些用法的例子。

■文件 包括了该程序用到的文件的名字。

■退出码 讨论了命令终止时的值。这个值在shell的环境变量‘?’中可用。（见sh(1)）。

■注意 给出在一些特殊情况下有帮助的信息。

■参见 提供相关信息的线索。

■诊断 讨论可能出现的出错消息。自释用的消息不列出来。

■警告 讨论各条命令的限制或边界条件。

■缺陷 列出软件中已知的但还未改正的错误。通常，还给出一个很短的建议修补方法。

第一部分前面是一张“目录表”（列出主命令和次命令条目）。“目录表”中的每一行列出命令的摘要。

怎样开始

这里的讨论给出你开始使用UNIX系统所需的基本信息：怎样注册和注销，怎样通过你的终端进行通信及怎样运行一个程序。（有关系统更完整的介绍，见《用户指南》(P-H)）。

注册

你必须把UNIX系统与一个全双工的ASCⅡ终端相连。你还必须有一个合法的注册名，这可以从系统管理员这里得到（同时还可得到怎样访问你的UNIX系统的信息）。通常的终端的速度是每秒120,240,480和960个字符（1200,2400,4800和9600波特）。有些UNIX系统对不同的终端速度有不同的存取方法，而有些系统则是通过一个公共的存取方法来提供几种速度。在后一种情况下，有一种“优先”速度。如果你用设置成不同速度的终端来存取系统，你面临的将是一串无意义的字符（Login：错误的终端速度消息）。此时只要继续按“break”、“interrupt”或“attention”键，就会出现Login：消息。

大多数终端都有一个速度转换开关，必须把这个开关设置到一个合适的速度，并且半双工开关要置成全双工。当联接完成后，系统显示Login：，你键入你的注册名，然后按回车键，如果你有口令，系统就要求打入，但并不在终端打印出来，或“回应”。你注册成功后，“return”，“new-line”和“line-feed”键的意思就一样了。

要保证你键入的注册名都是小写字母。键入大写字母会使得UNIX系统认为你的终端只能产生大写字母，并把你后面的注册信息全当成大写。在你成功的注册之后，shell将在你的屏幕显示一个\$符号。

当你注册时，在出现提示符前可能先碰到有关日期的消息。见Login(1)，它详细讨论注册过程，和stty(1)，这条命令告诉你怎样向系统描述你的终端。profite(4)（在《程序员参考手册》(P-H)中）解释如何在每次你注册时自动完成这后一工作。

注销

有两种方法注销：

- 如果你是拨话进入的，你只要挂断电话就行。
- 你可以键入文件结束标记来注销（ASCⅡ中的EOT字符，键入时打“control-d”），shell将终止，屏幕上将再出现Login：消息。

怎样通过你的终端通信

当你向UNIX系统打入字符时，你打入的字符被收集起来，且暂时存放在一个地方。虽然字符能回应给你，但这些字符并没有传给程序，直到你打入了“return”（或“new-line”），这在上面的“注册”中已描述过。

UNIX系统终端的输入/输出是全双工的。它有完整的预读功能,这意味着你可以在任何时刻键入字符,即使当时正有程序印出东西给你。当然,如果你在终端输出时打入,那么你的输入字符将与输出字符混杂在一起。不管在什么情况下,你能打入的内容都会按正确的顺序保留和解释。对预读的总数有一个限制但它是很大的,不太可能被超出。

字符@将抹掉一行中在它之前打入的所有字符,实际上就是删除了整个一行。(@也叫作删行符)。字符#抹去最后打入的字符,连续使用#将后退抹去字符,但不会超这一行的开头。要打入@和#本身,就要在这两个字符前加上\ (这样,要删除一个\,你要用两个#) 这些缺省的删字和删行符可以改变。见stty(1)。

control-S (在ASC II 中,称字符DC3) 的键入可以通过同时按下control键和字母键S进行,用来暂停输出。对CRT终端,这个键是有用的,它可以防止输出在还未看清之前就消失。打入Control-q可恢复输出(control-q也叫DC1)。这样,如果你打入了命令cat yourfile, yourfile中的内容在你看清之前,在屏幕上很快滚过,你就可以打入Control-S来暂时冻结输出一会,打入control-q后,将使输出恢复其正常的速度。control-S和control-q在这样用时,不会传给任何其他程序。

ASC II 的DEL字符(或“rubout”)也不传递给程序,但它要产生一个中断信号,就象“break”“interrupt”或“attention”的信号一样。这个信号通常导致你正运行的程序终止。其典型的用法是用来中止你不需要的长篇输出。对于程序,可以设计成统统忽略这些信号,或者设计成在信号产生时能识别这些信号并采取一定的行动(而不是终止程序的运行)。如编辑程序ed(1)就能捕捉到中断,并停止其正在做的,而不是终止编译编辑程序运行。因此,一个中断可用来终止编辑程序的输出而不丢失正在编辑的文件。

除了适应终端速度外,UNIX系统对你是否有一个带换行功能的终端或者对是否要用“回车”或“行馈给”来模拟也尽量做到灵活。在后一种情况下,所有“回车”符的输入都改成输入“行馈给”符,并且“回车”和“行馈给”在终端上回应。如果你进入了错误的方式, stty(1)命令会帮助你。

制表符在UNIX系统的源程序中可以随便使用。如果你的终端没有制表功能,你可以在输出时把制表符改成空格,在输入也回应空格。Stty(1)可以再一次用来设置或再设置这种方式。系统假定制表符定为八个字符位置。命令tab(1)在可能的情况下,将在你的终端上设置制表符。

怎样运行程序

当你成功地注册进入UNIX系统后,一个叫作shell的程序就与你的终端通信。shell读入你打进的每一行,把行分解成命令名和自变量,然后执行命令。一条命令就是一个可执行程序。通常shell总是在你的当前目录下查找具有给定名字的程序,如果命令不在该目录下,则shell再在系统目录下查找,如/bin和/usr/bin。系统提供的命令并没有什么特别的地方,只是它们都放在shell可以找到的目录之下。你也可以把命令放在你自己的目录下,命令shell在你的目录下查找命令。有关shell的环境变量\$PATH的讨论,见手册中条目sh(1)下的子标题“参数替换”。

输入给shell的行的第一个字是命令名。命令和自变量间通过空格符或制表符分割。

当一个程序结束运行后, shell!将重新取得控制并返回一个提示符,表示它已准备好接

受另外一条命令。shell还有许多其它功能，这将在sh(1)中详细描述。

当前目录

UNIX系统有一个按目录层次结构组织的文件系统。当接受了你的注册名后，系统管理员就为你创建一个目录（通常这个目录与你的注册名相同，称作你的注册目录或主目录）。你注册进入时，这个目录就成为你的当前目录或工作目录，你打入的任何文件名，都缺省地认为是在这个目录下。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。进入或修改其它目录和文件的许可，要由相应的文件所有者或系统管理员决定，或是允许或是不允许。要改变当前目录，用命令cd(1)。

路径名

为了引用不在当前目录下的文件，你必须使用路径名。

完整的路径名以“/”打头，这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（后跟一个/），直至到达最后的文件或目录名（如，/usr/ae/filex引用目录ae下的文件filex，而ae本身是usr下的子目录，usr又是根目录的子目录）。用Pwd(1)可以打印出你正使用的目录的完整路径名。有关路径名的形式定义见《程序员参考手册》（P—H）中的intro(2)。

如果你的当前目录中有子目录，则各文件的路径名以相应的子目录开头（不带前缀/）。路径名可以用于任何需要文件名的地方。

影响文件内容的重要命令有cp(1), mv（见cp(1)），和rm(1)，这三条命令分别复制，移动和删除文件。要查看文件或目录的状态，用ls(1)。用mkdir来创建一个目录，用rmdir（见rm(1)）来删除一个目录。

正文输入及显示

几乎所有的正文都是通过编辑程序输入的。UNIX系统常用的编辑程序有ed(1)和vi(1)。

在终端上打印正文常用命令cat(1), pr(1)和pg(1)。cat(1)命令在终端上显示ASCII正文文件的内容，不作任何处理。pr(1)命令为正文编页，提供标题，并且具有多列输出的能力。pg(1)命令连续地显示正文，每次显示的内容不超过你的终端屏幕。

写一个程序

一旦你用编辑程序把你的程序正文输入到文件中后，你就可以把文件交给合适的语言处理程序了。语言处理程序只接受遵守文件命名约定的文件：所有C语言程序的后缀名必须是.C，FORTRAN程序的后缀名必须是.f。语言处理程序的输出将放在当前目录下的名字为a.out的文件中，除非你用了任选项把它保存到另外的文件中。（用mv(1)重新命名a.out）。如果程序是用汇编语言写的，你或许需装入和它相连的子程序库。（见《程序员参考手册》（P—H）中的ld(1)）。

当你完成这个过程而没产生任何诊断消息，你可以在\$提示符下把程序名传给shell，从而运行这个程序。你的程序也可以象系统程序那样从命令行接受参数，见《程序员参考手册》（P—H）中的exec(2)。有关写程序和运行程序的更详细的信息，见程《序员指南》（P—H）。

与其他人通信

有些命令提供了内部用户通信功能。即使你不想用它们，但最好还是知道点这些命令，因为其他用户可能想与你取得联系。`mail(1)`和`mailx(1)`将把消息留下，并在另一个用户下次注册后，通知这个用户，并在整个过程中，每隔一定间隔就发出通知。要与另一个当前已注册的用户通信，用`write(1)`。本手册中的这些条目还有如何应答这两条命令的建议，如果你是目标的话，这些信息会有用的。

有关与其他人通信的细节见《用户指南》（P-H）中的第八章。

INTRO(1)

名字

`intro`——介绍命令和应用程序。

说明

本节按字母顺序描述了在AT & T 3B2计算机上可用的命令。在标题上按用途作了些区分。

下面的这些实用程序包与计算机一起提供：

AT & T窗口实用程序

基本的网络实用程序

盒式磁带控制实用程序

目录和文件管理实用程序

编辑实用程序

核心实用程序

图形实用程序

帮助实用程序

内部进程的通讯

行式打印机假脱机实用程序

性能测试实用程序

安全管理实用程序

拼写实用程序

终端过滤器实用程序

用户环境实用程序

下列实用程序包可供选购：

网络支持实用程序

远程文件共享实用程序

手册中每一页的命令语法

除非另外指出，在手册每页的格式这一节中描述的命令根据下面给出的语法接受任选项和其他自变量，并且必须按下面给出的说明加以解释。

`name [-option...] [cmdarg...]`

这里

() 把不是一定要的任选项或命令自变量括起来。

... 表明任选项或命令自变量可出现多次。

name 可执行文件的名字

option (总是以“—”开头)

noargletter...或,

argletter optarg[,...]

noargletter 单个的字母,表示任选项不带任选项自变量。注意在“—”后可以有不止一个的noargletter任选项组合在一起。(后面的规则5)

argletter 单个的字母,表示一个任选项需带任选项自变量。

optarg 任选项自变量(字符串),与前置的argletter相配。必须用逗号隔开,或者用空格分开并用引号括起来(后面的规则8)。

cmdarg 路径名(或其他命令自变量),不以“—”开头,或用“—”本身来指明标准输入。

命令语法标准:规则

下面这些命令的语法规则不是所有当前命令都要遵守的,但所有新命令都将遵守这些规则。所有的shell过程都要用getopt(1)来分析定位参数及检查任选项的合法性。它支持下面的规则3至规则10。其他规则的强制性由命令本身确定。

1. 命令名字(上面的name)的长度2—9个字符。
2. 命令名中只能有小写字母和数字。
3. 任选项(上面的option)的名字只能是一个字符。
4. 所有的任选项都必须前置“—”
5. 不带自变量的任选项可以在一个“—”后组合。
6. 跟在任选项后的第一个任选项自变量(上面的optarg)必须前置空格符。
7. 任选项的自变量不能任选。
8. 一个任选项后跟的多个任选项自变量必须用逗号分开,或者用空格分开并用引号括起来。(如—OXXX,Z,YY或—O“XXX Z YY”)。
9. 命令行中的所有任选项都必须前置一个操作对象(上面的Cmdarg)
10. “——”可用来指明任选项的结束。
11. 任选项间的相对次序无所谓。
12. 操作对象的相对次序(上面的cmdarg),会影响它们的有效性,这由它们所在的命令决定。
- 13 “——”前加空格或“——”后加空格只能用来表示标准输入。

参见

getopts(1)。

exit(2),wait(2),getopt(3C),这几条命令在《程序员参考手册》中。

本手册前面的“怎样开始”。

诊断

一旦命令执行终止,将返回两个字节的状况信息。其中一个由系统提供,指出终止的原因,另一个由程序提供(在“正常”终止的情况下)[见wait(2)和exit(2)]。前一个字节在正常结束时,其值为0;后一字节在成功执行后,其值通常也是0,一个非零的值表明遇

到了麻烦,如错误的参考,错误的或者是不可访问的数据。通常把后一字节叫做“出口,码”“出口状态”,或“返回码”,它仅在有特殊约定时才给予解释。

警告

有些命令在处理只带空字符的文件时会产生意想不到的结果。这些命令通常把正文输入行作为字符串看待,因此一旦在一行中遇到了空字符(字符串结束符)就会产生混乱。

300(1) (终端过滤器实用程序)

名字

300,300S——用来处理DASI300和300S终端的特殊功能。

格式

300[+12][-n][-dt,1,c]

300S[+12][-n][-dt,1,c]

说明

300命令支持DASI300(GSI300或DTC300)终端的特别功能,并对它的使用进行优化。300S对DASI300S(GSI300S或DTC300S)终端执行同样的功能。它把半行正向,半行反向和整行反向的动作转变成正确的垂直动作。在下面对300命令的讨论中,要注意除非你的系统中有“文件工作台”软件,否则,对一些命令的引用,(如nroff,neqn,eqn等)将得不到正确结果。该命令还能试着画出希腊字母和其他特殊字符,同时它也允许方便地使用每英寸12个字符的正文。它能节省约5%至7%的打印时间。300命令还可用来清楚地打印方程,打印序列是这样的:

```
neqn file...(nroff) 300
```

警告

如果你的终端有一个PLOT开关,那在使用300以前要保证该开关是开着的。

300命令的动作可通过任选标志自变量来改变,以便能处理每英寸12字符的正文,分数行的行距,消息和延迟。

+12 允许用每英寸12个字符,六行的正文。DASI300终端通常只允许两种类型的组合:10个字符,6行/英寸,或12个字符,8行/英寸。要得到每英寸12个字符,6行的这种组合,用户必须把PITCH开关转至12,并且用+12任选项。

-n 控制半行的宽度。半行宽度缺省时等于4个垂直图形增量。由于每个增量等于一英寸的 $1/48$,这样每行10个字符的需要8个增量,而每行12个字符的只需6个增量,n的第一个值复盖了缺省值,这样就可按个人的喜好来安排下标和上标的位置了。例如,nroff中的半行可以用-2任选项使它做起来就如同四分之一行那样。用户也可对每英寸12个字符,8行的模式,用-3任选项来得到合适的半行,当然这要先把PITCH置到12。

-dt,i,c 控制延迟因子。缺省值是-d3,90,30。DASI300终端在碰到很长的行,带有许多制表符或碰到一个没有空格或与之等价的字符的长字符串,有时会产生一些特殊的输出。在一行中对每t个制表符,及对每c个不带空格或制表符的连续字符串,将插入一个空字符。如果一行的长度大于1字节,则 $1 + \text{总长度}/20$ 个空字符就加在该行的最后。用隐含的缺省值,就可在表的最后省去这些项。给t(c)赋上零,将导致每个制表符(字符)产生两个空字节。前一种设置可用于

C程序，而后一种则可用于象/etc/passwd的文件。由于终端的动作根据印出和装入系统的特殊字符不同而有变化，因此用户需要用这些值进行试验，以得到正确的输出。一d任选项的存在仅是作为在其他情况下不能正确打印的少数几种情形的一种最后手段。例如，文件/etc/passwd可以用-d3,30,5，而对呈锯齿形的多层C程序来说，用值-d0,1则很好。

注意延迟控制与正在用的回车与行馈的互相影响很大，建议在大多数情况下使用stty(1)的方式n10 cr2或n10 cr3。

在需要用手工加纸或在文件的中间改变字形时，300命令可以与nroff的一S标志或-ra请求一起使用。在这种情况下你就必须用行馈键来得到响应，而不是用回本键了。

在多数情况下，下面的序列是等价的：

nroff-T300 files...和nroff files... |300

nroff-T300-12 files...和nroff files... |300+12

因些除非需要特殊的延迟成任选项，可以避免用300命令。在某些情况下，300的附加的移动优化会产生排列得更好的输出。

negn对300支持的希腊字符和其他字符进行命名和输出，在greek(5)中说明。

参见

450(1), mesg(1), graph(1G), stty(1), tabs(1), tplot(1G)。

eqn(1), nroff(1), tbl(1)，这几个命令在“文件工作台软件”2.0的技术讨论与参考手册中》

《程序员参考手册》中的greek(5)

缺陷

有些特殊字符在第一列不能正确打出，因为打印头不能再从第一列往左移。

如果你的输出中有希腊字母或反向行馈符，用磨擦供纸来代替格式供纸。虽然后者对大多数图形都足够好，但在碰到一个或多个反向行馈时，会使正文的第一行出错，且在改变方向和变形希腊字符时，会出现错误。

4014(1) (终端过滤器实用程序)

名字

4014——为Tektronix 4014终端分页。

格式

4014 [-t] [-n] [-cN] [-pl] [file]

说明

4014的输出是用于Tektronix 4014终端的。为了与屏幕相适应，4014每屏排列66行，并把屏幕分成N栏，在单栏的情况下，自动加入八个空行的页位移（缺省的）。需要时能接受制表符。空格符和回退符，并把它们画出来。Teletype Model 37的半行和逆行序列能被解释并画出。在每一页的最后，4014等待从键盘上输入换行符（空行），才开始继续处理下一页。在这种等待状态下，命令lcmd将把cmd送给shell。

命令行的任选项是：

-t 页间不等待（在把输出倒向文件时有用）

- `-n` 在光标的当前位置上印出并且不清除屏幕。
- `-cN` 把屏幕分成N栏并在最后一栏后等。
- `-pL` 把页长置为L; L接受度量因子i (英寸) 和l (行)。缺省值是行。

参见

`pr(1)`, `tc(1)`

“文件工作台软件”版本2.0的《技术讨论和参考手册》中的`troff(1)`。

450(1) (终端过滤器实用程序)

名字

450——处理DASI 450终端的特殊功能。

格式

450

说明

450命令支持DASI 450终端的特殊功能, 并且能优化使用。任何功能相同的终端, 如Diablo 1620或Xerax 1700也同样支持。它把半行正向半行反向和整行反向的移动转换成正确的垂直移动。它也试图以与`300(1)`同样的方式来画希腊字母和其他特殊符号。值得注意的是, 除非你的系统有“文件工作台软件”, 否则有些命令(如`eqn`, `nroff`, `tbl`等)将不能运行。用450命令, 按下列顺序可整齐地打印等式:

```
neqn file... |nroff |450
```

警告

在用450命令以前, 要保证你的终端上的PLOT开关是接通的。SPACING开关也必须放在合适的位置(或是每英寸10个字符或是每英寸12个字符)。不管是哪种情况, 垂直方向的间距都是6行/英寸, 除非你用适当的转义序列动态地将其改成8行/英寸。

在需要手工插入纸或在文件中间改变字形的时候, 可以把450与`nroff`的`-s`标志和`rd`请求一起用。在这种情况下, 要得到响应就要打行馈键, 而不是回车键了。

在许多情况下(不是全部), 可以不用450而用下面这种形式:

```
nroff -T450 files...
```

或

```
nroff -T450 -12 files
```

这样就可避免使用450, 除非需要一些特殊的延迟或任选项。不过在某些情况下, 450的附加的移动优化能产生排列得更好的输出。

`neqn`对450支持的希腊字母和其他特殊字符进行命名和输出, 这在`greek(5)`中说明。

参见

`300(1)`, `mesg(1)`, `stty(1)`, `tabs(1)`, `graph(1G)`, `tplot(1G)`。

“文件工作台软件”版本2.0的《技术讨论与参考手册》中的`eqn(1)`, `nroff(1)`, `tbl(1)`。
《程序员参考手册》中的`greek(5)`

缺陷

有些特殊字符在第一列不能印出, 因为打印头不能从第一列往左移了。

如果你的输出中有希腊字母或者反向行馈符, 用摩擦供纸来代替格式供纸。虽然后者对大多数图形都足够好, 但在碰到一个或多个反向行馈时, 会使正文的第一行出错, 在改变方

向和变形希腊字符时,会出现错误。

AR(1) (目录和文件管理实用程序)

名字

ar——可移植档案文件和库文件的维护。

格式

ar key [posname] afile [name] ...

说明

ar命令维护组合在一个档案文件中的一组文件。它的主要用途是创建和维护由连接编辑程序使用的库文件。当然它也可以用于任何相同的目的。**ar**用的神奇的字符串和文件头中,包括有可打印的ASCII字符。如果档案文件是由可打印文件组成,则整个档案文件都是可打印的。

当**ar**建立档案文件时,它创建一个文件头,其格式在所有机器上都是可移植的。可移植的档案文件格式和其结构在**ar(4)**中详细描述。这个文件符号表〔在**ar(4)**中描述,由连接编辑程序**ld(1)**使用〕以一种有效的方式对目标文件库产生多重影响。如在档案文件中至少有一个目标文件,则档案文件符号表就只能由**ar**创建并维护。档案文件符号表放在一个特别命名的文件中这个档案文件总是档案文件中的第一个文件,用户不能对其进行引用或访问。当用**ar(1)**命令来创建和维护档案文件的内容时,就重建符号表。下面描述的-s任选项将强制重建符号表。

与命令任选项不同, **key**作为**ar**命令行的一部分是必须要有的。**key** (可以由一打头)是下列字符中的一个: **drqtmx**。**key**的自变量,由下列字符集中的一个或多个字符组成: **vuaibcls**。**posname**是档案文件中的一个成员文件的名字,在定位档案文件中的其它文件时,用作基准点。**name**是档案文件文件中的组成。**key**中字符的意义如下:

d 从档案文件中删除命名的文件。

r 替换档案文件中的命名文件。如果任选字符**u**与**r**一起使用,则只有那些修改日期比档案文件晚的文件被替换。如果用了字符集**abi**中的一个任选定位符,则自变量**posname**一定要同时出现,并指明新的文件放在**posname**之后(用**a**)还是之前(用**b**或**i**)。其他情况下,新文件放在最后。

q 迅速地将命名文件加在档案文件的最后,这时任选定位字符非法。命令并不检查新加入的成员是否已在档案文件中。这个任选项对一点一点逐步建立一个大的档案文件是有用的,可以省去大量的工作量。

t 打印出档案文件的目录表。如果未给出名字,则档案文件中的所有文件都将列出。如果指定了名字,则只列出这些名字的文件。

p 打印出档案文件中的文件。

m 把指定名字的文件移到档案文件的最后。如果有定位字符出现,则**posname**的自变量也必须同时出现,并且就象在**r**中那样指出把这些文件移到何处。

x 抽取指定名字的文件。如果未给出名字,则档案文件中的所有文件都将抽取出来,但不管在什么情况下,**x**都不会改变档案文件。

key的自变量的意义如下:

v 详尽地一个文件一个文件地描述由一个老档案文件和其组成文件产生新档案文件的过程。当与**t**一起用时,它给出有关这些文件的所有信息的长格式清单。当与**x**一起使用时,

就在每一个文件前加上一个文件名字。

c 阻止在创建`afile`时缺省产生的消息。

l 把中间文件放在局部目录（当前目录）下，而不是放在缺省的中间目录`TMPDIR`下。

s 强制重新产生档案文件符号表，即使`ar`并未由修改档案文件内容的命令调用。这条命令在`strip(1)`命令作用于档案文件之后，可用来恢复档案文件符号表。

文件

`$TMPDIR/*` 中间文件

`$TMPDIR`就是通常的`/usr/tmp`，但可通过设置环境变量`TMPDIR`来重新定义。（见`tmpnam(3S)`中的`tmpnam()`）

参见

`ld(1)`, `lorder(1)`, `strip(1)`, `tmpnam(3S)`, `a.out(4)`, `ar(4)`

注意

如果同一个文件在自变量表中出现两次，则它可能两次加入档案文件。

AT(1) (用户环境实用程序)

名字

`at`, `batch`——在以后的一段时间里执行命令。

格式

`at time [date] [+increment]`

`at -r job...`

`at -! [job] ...`

`batch`

说明

`at`和`batch`从标准输入中读入在以后的时间里执行的命令。

当系统装入级许可后，在`batch`作业队列执行时，`at`允许你指定命令什么时候执行。`at`可与下列任选项一起使用：

-r 去除以前用`at`调度的作业。

-l 向调用用户报告所有作业的调度情况。

标准输出与标准出错输出邮给用户，除非这些输出被定向到其他地方。当命令执行时，`shell`环境变量，当前目录及`umask`和`ulimit`被保存起来，而打开文件的描述符、陷阱和优先级则丢失。

如果用户的名字在文件`/usr/lib/cron/at.allow`中，则用户就被允许使用`at`。如果该文件不存在，就检查文件`/usr/lib/corn/at.deny`，以决定是否应禁止用户访问`at`。如果两个文件都不存在，则只有根才有权递交作业。如果`at.deny`是空的，则谁都可使用`at`。`allow/deny`文件的每一行包括一个用户名字。这两个文件只能由超级用户修改。

`time`可以规定为1, 2, 3或4位数字。一位或两位数字看成是小时，四位数字则是小时和分。也可用两个数字之间加冒号来指定时间，意思是小时：分。可以加上前缀`am`或`pm`，否则就理解成24小时的时钟。特殊时间象`noon`, `midnight`, `now`和`next`也能识别。

任选的`date`可以用一个月名后跟天数来规定（可能的话还有前面带逗号的年数），或者