



UNIX 系统 V/386 第 4 版
用户参考手册

*UNIX® SYSTEM V/386
RELEASE 4*

User's Reference Manual



UNIX Software Operation

電子工業出版社

目 录

1. 命令

<code>intro(1)</code>	命令和应用程序介绍	(1)
<code>acctcom(1)</code>	搜索并显示进程记帐文件	(4)
<code>ar(1)</code>	维护可移植的档案或库	(6)
<code>at, batch(1)</code>	在以后某个时刻执行命令	(8)
<code>atq(1)</code>	显示已排队并等待在指定时刻运行的作业	(12)
<code>atrm(1)</code>	删除由 <code>at</code> 或 <code>batch</code> 产生的 <code>spool</code> 作业	(12)
<code>awk(1)</code>	模式扫描和处理语言	(13)
<code>banner(1)</code>	显示大尺寸字符	(16)
<code>basename, dirname(1)</code>	给出路径名的若干部分	(17)
<code>bc(1)</code>	任意精度的算术运算语言	(18)
<code>bdiff(1)</code>	大型 <code>diff</code>	(21)
<code>bfs(1)</code>	大文件扫描程序	(22)
<code>cal(1)</code>	显示日历	(25)
<code>calendar(1)</code>	提醒服务	(26)
<code>cat(1)</code>	串接并显示文件	(27)
<code>cd(1)</code>	改变工作目录	(28)
<code>chgrp(1)</code>	改变文件的组属关系	(29)
<code>chmod(1)</code>	改变文件方式	(30)
<code>chown(1)</code>	改变文件属主	(33)
<code>clear(1)</code>	清除终端屏幕	(34)
<code>cmp(1)</code>	比较两个文件	(34)
<code>col(1)</code>	倒向换行过滤程序	(35)
<code>comm(1)</code>	选择或舍去两个已排序文件的公共行	(36)
<code>compress, uncompress, zcat(1)</code>	压缩、扩展或显示经过扩展的文件	(37)
<code>cp(1)</code>	拷贝文件	(39)
<code>cpio(1)</code>	档案文件的拷入和拷出	(40)
<code>crontab(1)</code>	用户 <code>crontab</code> 文件	(45)
<code>crypt(1)</code>	加密/解密	(47)
<code>csh(1)</code>	具有类似C句法的shell命令解释程序	(48)
<code>csplit(1)</code>	分裂文件	(74)

ct (1C)	为远程终端衍生注册	(76)
ctags (1)	创建一个用于vi的标记文件	(77)
cu (1C)	呼叫另一个UNIX系统	(79)
cut (1)	剪下文件每一行中所选中的字段	(84)
date (1)	显示和设置日期	(86)
dc (1)	台式计算器	(89)
dd (1M)	转换并拷贝文件	(91)
deroff (1)	删除nroff/troff, tbl和eqn结构	(93)
df (1M)	报告空闲的磁盘块数和文件数	(94)
diff (1)	不同文件的比较程序	(96)
diff3 (1)	按三路比较有差别的文件	(99)
dircmp (1)	目录比较	(100)
du (1M)	总计盘的使用情况	(101)
echo (1)	回应实参	(102)
ed, red (1)	正文编辑程序	(103)
edit (1)	正文编辑程序(供偶而上机的用户使用的ex变种)	(114)
egrep (1)	用完全正则表达式在文件中搜索某个模式	(117)
enable,disable (1)	启用/禁用LP打印机	(119)
env (1)	设置命令执行的环境	(120)
ex (1)	正文编辑程序	(120)
expr (1)	把实参作为一种表达式来求值	(126)
extr (1)	从源文件中摘取字符串	(128)
face (1)	框式存取命令环境界面(FACE)的可执行程序	(132)
factor (1)	把数分解成其质数因子	(132)
fgrep (1)	在文件中搜索字符串	(133)
file (1)	确定文件类型	(134)
find (1)	查找文件	(135)
finger (1)	显示本地和远程用户的信息	(138)
fmli (1)	调用FMLI	(140)
fmt (1)	简单的正文格式化程序	(142)
fmtmsg (1)	在stderr或系统主控台上显示消息	(143)
fold (1)	长行折叠	(147)
ftp (1)	文件传输程序	(147)
gcore (1)	获取正在运行的进程的内存映象	(157)
gencat (1)	生成一个格式化的消息分类	(158)
getopt (1)	分析命令的任选项	(160)

getopts, getoptcv (1)	分析命令的任选项	(162)
gettxt (1)	在信息数据库中检索正文串	(164)
grep (1)	从文件中搜索模式	(166)
groups (1)	显示用户的组属关系	(167)
head (1)	显示文件的前面几行	(167)
iconv (1)	代码集转换公用程序	(168)
id (1M)	显示用户名与用户ID以及组名与组ID	(170)
ipcrm (1)	删除消息队列、信号灯集或共享内存ID	(170)
ipcs (1)	报告进程间通信设施的状态	(171)
ismpx (1)	返回窗口型终端的状态	(175)
join (1)	关系数据库算符	(175)
jterm (1)	恢复窗口型终端的层	(177)
jwin (1)	显示层的大小	(177)
kill (1)	按默认情况终止一个进程	(178)
ksh, rksh (1)	Korn Shell , 一种标准的/受限的命令和编程语言	(179)
last (1)	显示出最近用户或终端注册	(210)
layers (1)	开窗口终端的分层多路转接器	(211)
line (1)	读一行	(214)
listusers (1)	列出用户注册信息	(215)
ln (1)	连结文件	(215)
login (1)	注册	(217)
logname (1)	获取注册名	(219)
lp, cancel (1)	向 LP 打印服务程序提出发送/取消的请求	(219)
lpstat (1)	显示有关 LP 打印服务状态的信息	(225)
ls (1)	列出目录的内容	(227)
machid: pdp11, u3b, u3b2, u3b5, u3b15, vax, u370 (1)	取处理机类型真值	(231)
mail, rmail (1)	读邮件或给用户发邮件	(233)
mailalias (1)	翻译邮件别名	(240)
mailx (1)	交互式消息处理系统	(241)
makekey (1)	生成密钥	(257)
mesg (1)	允许/禁止发消息	(258)
mkdir (1)	建立目录	(259)
mkmsgs (1)	创建给 gettxt使用的消息文件	(260)
more, page (1)	浏览或翻阅正文文件	(262)
mv (1)	移动文件	(265)

nawk(1)	模式扫描和处理语言	(267)
newform(1)	改变正文文件的格式	(273)
newgrp(1M)	注册到新组中	(275)
news(1)	显示新闻条款	(276)
nice(1)	以低优先级运行命令	(277)
nl(1)	行编号过滤程序	(278)
nohup(1)	运行命令不受挂起和退出的影响	(280)
notify(1)	通知用户有新邮件到来	(281)
od(1)	八进制卸出	(282)
pack, pcat, unpack(1)	压缩和展开文件	(284)
passwd(1)	改变注册口令和口令属性	(286)
paste(1)	合并若干文件的相同行或一个文件中连续相同的后几行 ..	(289)
pg(1)	CRT的文件细读过滤程序	(290)
pr(1)	显示文件	(294)
printf(1)	显示格式化的输出	(296)
priocntl(1)	进程调度程序控制	(298)
ps(1)	报告进程状态	(304)
pwd(1)	工作目录名	(308)
rcp(1)	远程文件拷贝	(309)
relogin(1M)	重新命名注册登记项来表示当前层	(310)
rlogin(1)	远程注册	(311)
rm, rmdir(1)	删除文件或目录	(313)
rsh(1)	远程shell	(315)
ruptime(1)	显示本地机的状态	(317)
rwho(1)	显示谁在本地机器上注册	(318)
sag(1)	系统活动态势图	(319)
sar(1)	系统活动报告生成程序	(321)
script(1)	生成终端会晤期的打印脚本	(325)
sdiff(1)	显示文件的对照差别	(325)
sed(1)	流编辑程序	(327)
sh, jsh, rsh(1)	标准shell, 作业控制和受限命令解释程序	(330)
shl(1)	shell层管理程序	(347)
sleep(1)	挂起一段时间以后再执行	(349)
sort(1)	排序和/或合并文件	(350)
spell, hashmake, spellin, hashcheck(1)	查找拼写错误	(353)

<code>split(1)</code>	把文件分成若干段	(355)
<code>srchtxt(1)</code>	显示消息数据库中的内容或从中搜索正文串	(356)
<code>strchg, strconf(1)</code>	改变或查询流配置	(358)
<code>strings(1)</code>	在目标文件或二进制文件中查找可显示的串	(361)
<code>stty(1)</code>	设置终端任选项	(361)
<code>su(1M)</code>	使用户成为超级用户或另一个用户	(370)
<code>sum(1)</code>	显示检查和并计算文件的块数	(372)
<code>sync(1M)</code>	更新专用块	(372)
<code>tabs(1)</code>	设置终端制表符	(373)
<code>tail(1)</code>	交付文件的最后部分	(376)
<code>talk(1)</code>	与另一个用户对话	(377)
<code>tar(1)</code>	磁带文件存档程序	(378)
<code>tee(1)</code>	在管道上建立“三通”	(381)
<code>telnet(1)</code>	使用TELNET通信协议的远程系统的用户界面	(381)
<code>test(1)</code>	条件计算命令	(386)
<code>tftp(1)</code>	普通的文件传输程序	(389)
<code>time(1)</code>	命令计时	(391)
<code>timex(1)</code>	命令计时, 报告进程记帐数据和系统活动情况	(391)
<code>touch(1)</code>	更新文件的访问时间和修改时间	(393)
<code>tput(1)</code>	初始化终端或查询terminfo数据库	(393)
<code>tr(1)</code>	翻译字符	(397)
<code>true, false(1)</code>	提供真值	(398)
<code>truss(1)</code>	跟踪系统调用和信号	(399)
<code>tty(1)</code>	取终端名	(403)
<code>umask(1)</code>	设置“文件创建方式”屏蔽码	(404)
<code>uname(1)</code>	显示当前UNIX系统的名字	(404)
<code>uniq(1)</code>	报告一个文件中的重复行	(405)
<code>units(1)</code>	转换程序	(406)
<code>uucp, uulog, uuname(1C)</code>	UNIX系统与UNIX系统之间的拷贝	(407)
<code>uuencode, uudecode(1C)</code>	加密二进制文件或解密它的ASCII表示	(410)
<code>uuglist(1C)</code>	显示本地UNIX系统上可用的服务级清单	(411)
<code>uustat(1C)</code>	UUCP状态查询和作业控制	(412)
<code>uuto, uupick(1C)</code>	公共的UNIX系统到UNIX系统的文件拷贝	(415)
<code>uux(1C)</code>	UNIX系统到UNIX系统的命令执行	(417)

vacation(1)	对发来的邮件消息自动应答	(420)
vi(1)	基于 ex 的面向屏幕(直观)显示的编辑程序	(421)
wait(1)	等待进程完成	(431)
wc(1)	词计数	(432)
who(1)	谁在系统中	(432)
whois(1)	Internet用户名目录服务	(435)
write(1)	写给其他用户	(436)
xargs(1)	构造实参表并执行命令	(438)

intro (1)

名称

intro——命令和应用程序介绍

说明

本节(本手册)按字母顺序描述了在AT&T 386计算机上所用的命令。在标题中按用途作了某些区分。

下列公用程序软件包与计算机一起提供:

AT&T 窗口公用程序(AT&T Windowing Utilities)

基本网络公用程序(Basic Networking Utilities)

盒式磁带控制器公用程序(Cartridge Tape Controller Utilities)

目录和文件管理公用程序(Directory and File Management Utilities)

编辑公用程序(Editing Utilities)

基本公用程序(Essential Utilities)

加框的存取命令环境公用程序(Framed Access Command Environment Utilities)

进程间通信(IPC)公用程序(Inter-Process Communication Utilities)

作业记帐公用程序(Job Accounting Utilities)

行式打印机 SPOOL 公用程序(Line Printer Spooling Utilities)

保安性管理公用程序(Security Administration Utilities)

拼写公用程序(Spell Utilities)

系统性能分析公用程序(System Performance Analysis Utilities)

终端信息公用程序(Terminal Information Utilities)

传送控制协议(Transmission Control Protocol)

用户环境公用程序 (User Environment Utilities)

下列公用程序软件包可以买到:

网络支持公用程序 (Networking Support Utilities)

远程文件共享公用程序 (Remote File Sharing Utilities)

手册页命令句法

在手册页的**格式**段中描述的命令根据下面给出的句法接受任选项和其它实参，并且必须按下面给出的说明加以解释，除非另加注记。

name [*-option*] [*cmdarg* ...]

这里:

[] 把不一定需要的任选项 (*option*) 或命令实参 (*cmdarg*) 括起来。

... 表明任选项或命令实参可以出现多次。

name 可执行文件的名称。

option (总是以“-”开头)。
noargletter... 或,
argletter optarg ['...']

noargletter 单个字母，表示一个不带任选项实参的任选项。注意，在“-”后面可以有多个 *noargletter* 任选项组合在一起 (参见后面的规则 5)。

argletter 单个字母，表示一个需要带任选项实参的任选项。

optarg 任选项实参 (字符串)，与其前面的 *argletter* 一致。注意，跟在 *argletter* 后面的一组 *optarg* 必须用逗号分开，或者用空白类符分开并用引号引括 (参见后面的规则 8)。

cmdarg 不以“-”开头的路径名 (或其它命令实参)，或是“-”本身表示标准输入。

命令句法标准: 规则

并非所有当前命令都遵守下面这些命令句法规则，但所有新命令都将遵守这些规则。所有的 shell 过程都要用 **getopts**(1) 来分析位置参数并核实任选项的合法性，它支持下面的规则 3 至规则 10。其它规则的实施与否应由命令本身确定。

1. 命令名称 (上述的 *name*) 的长度必须是 2~9 个字符。

2. 命令名中只能有小写字母和数字。
3. 任选项(上述的 *option*)的名字只能是一个字符。
4. 所有的任选项前都必须置“-”。
5. 不带实参的任选项可以在一个“-”后组合在一起。
6. 跟在任选项后的第一个任选项实参(上述的 *optarg*)的前面必须有空白类符。
7. 任选项实参不是可有可无的。
8. 跟在一个任选项后面的一组任选项实参必须用逗号分开, 或者用空白类符分开并用引号引括(如, `-o xxx,z,yy` 或 `-o "xxx z yy"`)。
9. 命令行中的所有任选项都必须都放在操作对象(上述 *cmdarg*)之前。
10. “—”可用来指示任选项结束。
11. 任选项之间的相对次序无关紧要。
12. 操作对象(上述的 *cmdarg*)的相对次序会影响它们的含意, 这由它们所在的命令决定。
13. 在“-”前后加空白类符只能用来表示标准输入。

参见

`getopts(1)`。

《程序员参考手册》中的`exit(2)`, `wait(2)`, `getopt(3C)`。

本手册前面的**如何起步**一节。

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(在“正常”终止的情况下)由程序提供[参见 `wait(2)` 和 `exit(2)`]。前一个字节在正常结束时, 其值为 0; 后一个字节在成功执行后, 其值通常也是 0, 而一个非零的值指示遇到了麻烦, 如: 错误的参数, 错误的或者是不可访问的数据。通常把后一个字节叫做“出口码”、“出口状态”或“返回码”, 它仅在涉及特殊约定的场合才给予解释。

告诫

有些命令在处理含有空符(`null`)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符(字符串结束符)就会产生混乱。

acctcom(1)

名称

acctcom——搜索并显示进程记帐文件

格式

acctcom [*options*] [*file ...*]

说明

acctcom 按照**acct(4)**中描述的格式读取*file*文件、标准输入或**/var/adm/pacct**，并将选出的记录写到标准输出上。每个记录均代表一个进程的执行。在标准输出上显示：**COMMAND NAME**、**USER**、**TTYNAME**、**START TIME**、**END TIME**、**REAL (SEC)**、**CPU (SEC)**、**MEAN SIZE(K)**，并且有时还显示**F(fork/exec** 标记，1 为没有 **exec** 的 **fork)**、**STAT**(系统出口状态)、**HOG FACTOR**、**KCORE MIN**、**CPU FACTOR**、**CHARS TRNSFD** 和 **BLOCKS READ**(读写的全部块数)。

如果命令在超级用户特权下执行，那么在此命令名前前置一个**#**号。如果某个进程没有与一个已知终端相联，那么在 **TTYNAME** 字段中将显示一个**?** 号。

如果没有指定文件*file*，并且标准输入与某个终端或**/dev/null**相关联(如同在shell中使用**&**时的情形一样)，那么 **acctcom** 将从**/var/adm/pacct** 中读取，否则从标准输入读取。

如果给出了实参文件名*file*，则按它们各自的顺序读取这些文件。每个文件通常都是向前读取的，即按进程完成的时间顺序读。通常，文件**/var/adm/pacct** 为被考查的当前文件；一个繁忙的系统可能需要若干个这样的文件，其中除当前文件外，其余所有的文件都可在**/var/adm/pacctincr** 中找到。

任选项(*options*):

- a** 显示有关选定进程的某些平均统计数字。这些统计数字将在输出记录后面显示。
- b** 反向读取，首先显示最后的命令。当读标准输入时，此任选项无效。
- f** 在输出中显示 **fork/exec** 标志及系统出口状态栏。本任选项的数字输出

以八进制表示。

- h** 不显示平均存储大小，而显示进程执行期间消耗的全部有效的CPU时间比值。这一“霸占(hog)因子”用公式: (总 CPU 时间)/(流逝的时间)来计算。
- i** 在输出中显示含有I/O计数的栏目。
- k** 不显示所占内存的大小，而显示在核心态下花费的总时间(kcore=minutes)。
- m** 显示平均内存大小(默认值)。
- r** 显示CPU因子(用户时间/(系统时间+用户时间))
- t** 分别显示系统和用户CPU时间。
- v** 在输出中去掉栏标题。
- l line** 只显示属于终端/dev/term/line的进程。
- u user** 只显示属于 user 的进程，该user可用：用户ID、可转换为用户ID的注册名、一个#号(仅用于指出在超级用户特权下执行的进程)或一个?号(仅用于指出与未知用户 ID 相关联的进程)来指定。
- g group** 只显示属于 group 组的进程。group可由组ID或组名指定。
- s time** 选择在 time 或 time 之后存在的进程，time的格式为 hr[:min[:sec]]。
- e time** 选择在 time 或 time 之前存在的进程。
- S time** 选择在 time 或 time 之后启动的进程。
- E time** 选择在 time 或 time 之前结束的进程。当对 -S 和 -E 使用同一个time时，只显示在 time 时已存在的进程。
- n pattern**
显示与pattern匹配的命令。除 + 号表示出现一次或多次外，pattern 可以是与 regcmp(3G)中的正则表达式一样的正则表达式。
- q** 不显示任何输出记录，只显示-a任选项中的平均统计数字。
- o ofile** 按输入数据格式将选择的进程记录拷贝到ofile中，禁止在标准输出上显示。
- H factor** 只显示因子超过factor的进程，该factor即为上面-h任选项中解释的“霸占因子”。

-O *sec* 只显示CPU系统时间超过 *sec* 秒的进程。

-C *sec* 只显示总CPU时间(系统时间+用户时间)超过*sec*秒的进程。

-I *chars* 只显示传输的字符数多于*chars*所给出的截止数的进程。

文件

/etc/passwd
/var/adm/pacctincr
/etc/group

参见

ps(1),su(1)。

《程序员参考手册》中的 acct(2), regcmp(3G)。

《系统管理员参考手册》中的 acct(1M), acctcms(1M), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M), acct(4), utmp(4)。

注释

acctcom只报告已结束的进程，对活动进程则应使用ps(1)。

如果 *time* 超过了现在的时间，则解释为前一天的这一时间。

ar(1)

名称

ar——维护可移植的档案或库

格式

ar [-V] -key [*arg*] [*posname*] *afile* [*name* ...]

说明

ar命令维护已被组合成单一档案文件的一组文件。它主要用于创建和更新库文件。当然它也可以用于任何类似的目的。ar所使用的魔数字符串和文件头由可显示的ASCII字符组成。如果一个档案由一些可显示文件组成，则整个档案也是可显示的。

当ar创建一个档案时，它以一种在所有机器上都是可移植的格式创建文件头。可移植的档案格式和结构在 ar(4)中有详细描述。档案符号表 [在 ar(4)中描述] 由连接

编辑程序 `ld` 使用，以一种有效的方式对目标文件库实现多次扫描。仅当档案中至少有一个目标文件时，才由 `ar` 创建并维护档案符号表。档案符号表放在一个特别命名的文件中，它永远是档案中的第一个文件，用户不能对其进行引用或访问。当用 `ar` 命令来创建和更新档案的内容时，符号表被重建。下面描述的 `s` 任选项将强制重建符号表。

任选项 `-v` 导致 `ar` 在标准出错输出中显示它的版本号。

与命令任选项不同，`key` 是 `ar` 命令行中不可缺少的一部分，它可以是下列字符：`drqtpmx` 中的一个。而 `key` 的实参则由下列字符集：`vuaibcls` 中的一个或多个字符组成。`posname` 是档案中一个成员的名字，在给档案中的其它文件定位时，用其它作参考点。`afile` 是 (`ar` 要处理的) 档案文件。而一或多个 `name` 则是档案文件的组成文件。下面是 `key` 取值字符的含意：

- `d` 从档案文件中删除指名的文件。
- `r` 替换档案文件中的指名文件。如果任选的字符 `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。如果使用了字符集 `abi` 中的一个任选的定位符，则实参 `posname` 必须同时出现，并指明新的文件应该放在 `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` 迅速地将指名文件附加到档案文件的最后。此时，任选的定位符无效。命令并不核实新加入的成员是否已在档案文件中。这个任选项对一件一件地逐步建立一个大的档案文件有用，可以避免工作量平方效应。
- `t` 显示档案文件的目录表。如果未给出名字，则列出档案中的所有文件。反之，则只列出那些给出名字的文件。
- `p` 显示档案文件中指名的文件。
- `m` 把指名的文件移到档案文件的最后。如果有定位符出现，则实参 `posname` 也必须同时出现，并且要象在 `r` 中那样指出把这些文件移到什么地方。
- `x` 抽取指名的文件。如果未给出名字，则抽取档案中的所有文件。不管在什么情况下，`x` 都不会改变档案文件。

其它用作 `key` 的实参含意如下：

- `v` 对于由老档案文件和若干组成文件构成一个新档案文件的过程，一个文件一个文件地给出详细描述。当与 `t` 一起使用时，它给出有关这些文件的所

有信息的详细清单。当与 **x** 一起使用时，则在每一个抽取的文件前显示文件名。

- c** 在创建 *afile* 时，阻止按默认原则产生的消息。
- l** 这个任选项已过时，它可被识别但被忽略。它在下一版本中将被删除。
- s** 即使 **ar(1)** 并未调用修改档案内容的命令，也要强制重新产生档案符号表。这条命令可用来在 **strip(1)** 命令作用于档案文件之后恢复档案符号表。

参见

ld(1), **lorder(1)**, **strip(1)**, **a.out(4)**, **ar(4)**。

注释

如果同一个文件在实参表中出现两次，那么它可能两次加入档案文件。

由于档案(处理)程序不再使用临时文件，因此 **-l** 任选项已过时并将在下一版本中删除。

作为约定，档案以字符 **.a** 为后缀。

at(1)

名称

at, **batch**——在以后某个时刻执行命令

格式

```
at [-f script] [-m] time [date] [+increment]  
at -l [job ... ]  
at -r job ...  
batch
```

说明

at 和 **batch** 从标准输入中读入打算在以后某一时刻执行的命令。**at** 允许你指定命令在什么时候执行，而用 **batch** 排了队的作业则在系统的负荷水平允许时被执行。**at** 可与下列任选项一起使用：

- f *script*** 从名为 *script* 的文件中读取要执行的命令。
- l [*job*]** 报告为发出本命令的用户已安排的所有作业，或仅报告指定的 *jobs*。

-m 在作业完成后，即使它不产生输出，也向用户发一邮件以指出作业已结束。只有当作业尚未产生邮件消息时，才发送邮件。

-r job 删除以前用 **at** 安排的指定 *jobs*。

除非改向到其他地方，否则标准输出与标准出错输出将邮递给用户。当命令执行时，保留 shell 环境变量、当前目录、umask 和 ulimit，而打开的文件描述符、捕获和优先级则丢失。

如果用户的名字出现在文件 `/usr/sbin/cron.d/at.allow` 中，则允许该用户使用 **at**。如果该文件不存在，则核查文件 `/usr/sbin/cron.d/at.deny` 以决定是否应禁止该用户访问 **at**。如果两个文件都不存在，则只有根才有权递交作业。如果只有 **at.deny** 存在并且为空，则允许所有用户使用 **at**。allow/deny 文件的每一行包含一个用户名。这些文件只能由特权用户修改。

如果设置了环境变量 **DATEMSK**，那么它将指向一个样板文件，**at** 使用该文件以决定有效的 *time* 和 *date* 值来替换下面描述的值。关于使用 **DATEMSK** 的更多的信息，请参见此说明部分的最后一段。

time 可规定如下：*h*, *hh*, *hhmm*, *h:m*, *h:mm*, *hh:m*, *hh:mm* (在这里，*h* 表示小时，*m* 表示分钟)。除非附加 **am** 或 **pm**，否则 *time* 将理解为 24 小时制时间。若在 *time* 后附加 **zulu**，则表示格林尼治平均时 (GMT)。*time* 也可以取值为：**noon**、**midnight** 和 **now**。**at now** 将回应出错消息 **too late**，一般使用 **now** 应伴随增量实参 *increment*，如：**at now + 1 minute**。

任选的 *date* 可以指定为一个其后跟随有天数的月名 (也可能还有前面带逗号的年号)，或者一个星期的某一日 (月名或星期中的某日可以是完整拼出的，也可以缩写为三个字符)。两个特殊的“日子”：**today** 和 **tomorrow** 也能识别。如果没给出 *date*，且给出的小时数大于当前小时数，则假定为 **today**，否则就假定为 **tomorrow**。如果给出的月份小于当前的月份 (并且未给出年份)，则假定是下一年。

任选的 *increment* 只是一个数字，它以下列单词之一为后缀：**minutes**、**hours**、**days**、**weeks**、**months** 或 **years** (单数形式也可接受)。修饰符 **next** 可以放在 *increment* 之前，它意味着：“+1”。

因此，有效的命令包括：

```
at 0815am Jan 24
```

```
at 8:15am Jan 24
at now + 1 day
at now next day
at 5 pm Friday
```

at 和 **batch** 把作业号和安排时间写到标准出错输出上。

at -r 删除以前由 **at** 或 **batch** 安排的作业。其作业号就是由 **at** 或 **batch** 命令返回的数字。你也可通过键入 **at -l** 来得到作业号。除非是特权用户，否则只能删掉你自己的作业。

如果设置了环境变量 **DATEMSK**，则 **at** 将它的值看成样本文件的全路径名，该样本文件包含格式串。这些串由字段描述字和正文字符组成并通过适当设置环境变量 **LANG** 或 **LC_TIME**，用不同的语言提供一组内容丰富的可允许的日期格式(参见 **environ(5)**)。(参见 **getdate(3C)** 中关于可允许的字段描述字的清单，该清单为 **calendar(1)** 允许的描述字的一个子集，全部描述字在本手册 **date(1)** 中列出。)上面描述的关于实参 *time* 和 *date* 的格式、特殊名字 **noon**、**midnight**、**now**、**next**、**today**、**tomorrow** 以及实参 *increment*，在 **DATEMSK** 被设置后将不被识别。

示例

at 和 **batch** 命令从标准输入中读入将在以后执行的命令。**sh(1)** 向我们提供了另外一些指定标准输入的不同方法。在你的命令中，将标准输出改向也许是有用的。

下面这个命令序列可用在终端上：

```
batch
sort filename > outfile
<control-D> (按住“control”再键入“d”)
```

下面这个命令序列演示了如何把标准出错输出改向到管道。这在 **shell** 过程中是有用的(输出改向规格说明的次序很重要)：

```
batch <<!
sort filename 2>&1 >outfile | mail loginid
!
```

为了让一个作业重新安排自身，可在 **shell** 过程中调用 **at**，并在 **shell** 文件中包括类似如下的这段代码：