



UNIX 系统 V/386 第 4 版
程序员参考手册

*UNIX® SYSTEM V/386
RELEASE 4*

Programmer's Reference Manual



UNIX Software Operation

電子工業出版社

目 录

1. 命令

intro(1)	编程命令介绍	(1)
admin(1)	创建和管理 SCCS 文件	(2)
ar(1)	可移植的档案文件和库文件的维护程序	(6)
as(1)	汇编程序	(8)
cb(1)	C 程序整理程序	(9)
cc(1)	C 语言编译程序	(10)
cdc(1)	改变SCCS delta的注释	(15)
cflow(1)	生成C程序流程图	(17)
cof2elf(1)	COFF到ELF目标文件的转换	(19)
comb(1)	组合SCCS delta	(20)
convert(1)	把档案文件转换成通用格式	(21)
cscope(1)	交互式地检查C程序	(22)
ctrace(1)	C程序调试程序	(28)
cxref(1)	生成C程序的交叉引用	(32)
delta(1)	针对 SCCS文件建立delta	(35)
dis(1)	目标代码反汇编程序	(37)
dump(1)	卸出目标文件的选定部分	(38)
get(1)	获得SCCS文件的版本	(40)
help(1)	寻求对给定消息编号或SCCS命令的帮助	(46)
install(1)	安装命令	(46)
ld(1)	目标文件的连接编辑程序	(48)
ldd(1)	列出动态依赖文件	(51)
lex(1)	生成完成简单词法分析任务的程序	(52)
lint(1)	C程序的检查程序	(55)
lorder(1)	找出一个目标库中的顺序关系	(58)
lprof(1)	显示逐行执行的统计造型数据	(59)
m4(1)	宏处理程序	(62)
make(1)	维护、更新和重新生成程序组	(65)

mcs(1)	对目标文件的注释节进行处理	(71)
nm(1)	显示目标文件的名字表	(73)
prof(1)	显示造型文件的数据	(74)
prs(1)	显示SCCS文件	(77)
regcmp(1)	编译正则表达式	(81)
rmDEL(1)	从SCCS文件中删除delta	(82)
sact(1)	显示当前SCCS文件的编辑活动情况	(83)
sccsdiff(1)	比较SCCS文件的两个版本	(84)
sdb(1)	符号调试程序	(85)
size(1)	显示目标文件各段的字节数	(91)
strip(1)	从目标文件中删除符号表、调试信息及行号信息等	(92)
tsort(1)	拓扑排序	(93)
unset(1)	取消对SCCS文件实施的前一个get命令	(94)
val(1)	检验SCCS文件的有效性	(95)
vc(1)	版本控制	(96)
what(1)	显示标识串	(99)
yacc(1)	另一个编译程序的编译程序	(100)

2. 系统调用

intro(2)	系统调用及出错编号介绍	(102)
access(2)	确定文件的可访问性	(119)
acct(2)	启用或禁止对进程的记账	(121)
adjtime(2)	修正时间以便与系统时钟同步	(122)
alarm(2)	设置进程闹钟	(123)
brk, sbrk(2)	改变数据段空间分配	(123)
chdir, fchdir(2)	改变当前目录	(124)
chmod, fchmod(2)	改变文件方式	(126)
chown, lchown, fchown(2)	改变文件的文件属主和用户组	(128)
chroot(2)	改变根目录	(130)
close(2)	关闭文件描述字	(131)
creat(2)	创建新文件或重写已存在的文件	(132)
dup(2)	复制打开的文件描述字	(134)
exec: execl, execv, execl, execve, execlp, execvp(2)	执行文件	(135)

exit, _exit (2)	终止进程	(138)
fcntl (2)	文件控制	(140)
fork (2)	创建一个新进程	(144)
fpathconf, pathconf (2)	取得可配置的路径名变量	(145)
fsync (2)	使文件的内存状态与物理媒体上的状态一致	(147)
getcontext, setcontext (2)	获取和设置当前用户文境	(148)
getdents (2)	读目录项并把它放在独立于文件系统的格式中	(149)
getgroups, setgroups (2)	获取或设置补充组访问清单ID	(150)
getmsg (2)	从流上取下一个消息	(151)
getpid, getpgid, getppid, getpgrp (2)	获取进程、进程组和父进程的ID	(154)
getrlimit, setrlimit (2)	控制最大的系统资源消耗量	(154)
getsid (2)	获取会晤期ID	(156)
getuid, geteuid, getgid, getegid (2)	获取实际用户ID、有效用户ID、实际用户组ID和 有效用户组ID	(157)
ioctl (2)	控制设备	(157)
kill (2)	向进程或进程组发送信号	(159)
link (2)	连接文件	(161)
lseek (2)	移动读/写文件指针	(162)
memcntl (2)	存储管理控制	(163)
mincore (2)	确定页在内存中的驻留情况	(166)
mkdir (2)	创建目录	(166)
mknod (2)	创建目录、特别文件或普通文件	(168)
mmap (2)	映射内存中的页	(170)
mount (2)	安装文件系统	(172)
mprotect (2)	设置内存映射保护	(174)
msgctl (2)	消息控制操作	(175)
msgget (2)	获取消息队列	(176)
msgop: msgsnd, msgrcv (2)	消息操作	(178)
munmap (2)	去除内存页的映射	(180)
nice (2)	改变分时进程的优先级	(181)
open (2)	打开文件进行读或写	(182)
pause (2)	暂停一个进程直到收到信号为止	(185)
pipe (2)	创建进程间通道	(186)
plock (2)	内存中进程、正文或数据的加锁或解锁	(187)

poll(2)	输入/输出的多路复用	(188)
prctl(2)	进程调度控制	(190)
prctlset(2)	通用的进程调度控制	(198)
profil(2)	执行时间统计	(200)
ptrace(2)	进程跟踪	(201)
putmsg(2)	在流上发送消息	(203)
read(2)	读文件	(205)
readlink(2)	读符号连接的值	(209)
rename(2)	改变文件的名字	(210)
rmdir(2)	删除目录	(212)
semctl(2)	信号灯控制操作	(213)
semget(2)	获取一组信号灯	(215)
semop(2)	信号灯操作	(216)
setpgid(2)	设置进程组ID	(219)
setpgrp(2)	设置进程组ID	(220)
setsid(2)	设置会话期ID	(220)
setuid, setgid(2)	设置用户ID和组ID	(221)
shmctl(2)	共享内存控制操作	(222)
shmget(2)	获取共享内存段标识符	(224)
shmop: shmat, shmdt(2)	共享内存操作	(225)
sigaction(2)	详细的信号管理	(227)
sigaltstack(2)	设置或获取信号替代栈的上下文	(229)
signal, sigset, sighold, sigrelse, sigignore, sigpause(2)	简化的信号管理	(230)
sigpending(2)	检查被阻塞和被挂起的信号	(232)
sigprocmask(2)	修改或检查信号屏蔽	(233)
sigsend, sigsendset(2)	向进程或进程组发送信号	(233)
sigsuspend(2)	安装信号屏蔽并挂起进程直至收到信号	(235)
stat, lstat, fstat(2)	获取文件状态	(236)
statvfs, fstatvfs(2)	获得文件系统的信息	(239)
stime(2)	设置时间	(241)
swapctl(2)	管理对换空间	(242)
symlink(2)	建立与文件的符号连接	(244)
sync(2)	更新专用块	(245)
sysi86(2)	针对具体机器的功能	(246)
sysfs(2)	获得文件系统类型的信息	(249)

3. 函数

5

ctermid (3S)	产生终端文件名	(290)
ctime, localtime, gmtime, asctime, tzset (3C)	把日期和时间转换成字符串	(290)
ctype: isdigit, isxdigit, islower, isupper, isalpha, isalnum, isspace, iscntrl, ispunct, isprint, isgraph, isascii (3C)	字符处理	(293)
cuserid (3S)	获得用户的字符注册名字	(295)
difftime (3C)	计算两个日历时间之差	(295)
directory: opendir, readdir, telldir, seekdir, rewinddir, closedir (3C)	目录操作	(296)
div, ldiv (3C)	计算商和余数	(298)
drand48, erand48, lrand48, nrand48, mrand48, jrand48, srand48, seed48, lcong48 (3C)	产生均匀分布的伪随机数	(299)
dup2 (3C)	复制打开的文件描述字	(301)
ecvt, fcvt, gcvt (3C)	把浮点数转换成字符串	(301)
end, etext, edata (3C)	程序中的最后位置	(302)
fclose, fflush (3S)	关闭或刷清流	(303)
ferror, feof, clearerr, fileno (3S)	流状态查询	(304)
ffs (3C)	找第一个设置位	(304)
fmtmsg (3C)	在stderr或系统控制台上显示消息	(305)
fopen, freopen, fdopen (3S)	打开一个流	(309)
fpgetround, fpsetround, fpgetmask, fpsetmask, fpgetsticky, fpsetsticky (3C)	IEEE浮点数环境控制	(311)
fread, fwrite (3S)	二进制输入/输出	(312)
frexp, ldexp, logb, modf, modff, nextafter, scalb (3C)	操纵浮点数的各个部分	(313)
fseek, rewind, ftell (3S)	重新定位流中的文件指针	(315)
fsetpos, fgetpos (3C)	重新定位流中的文件指针	(316)
ftw, nftw (3C)	遍历文件树	(317)
getc, getchar, fgetc, getw (3S)	从流上获取字符或字	(319)
getcwd (3C)	获取当前工作目录的路径名	(320)
getdate (3C)	转换用户格式的日期和时间	(321)
getenv (3C)	返回环境名的值	(325)
getgrent, getgrgid, getgrnam, setgrent, endgrent, fgetgrent (3C)	获得组文件表项	(325)
getitimer, setitimer (3C)	获取/设置间隔计时器的值	(327)
getlogin (3C)	获得注册名	(328)

getmntent, getmntany (3C)	获取mnttab文件表项	(329)
getopt (3C)	从实参向量中获取选项字母	(330)
getpass (3C)	读一个口令	(333)
getpw (3C)	从UID中获得名字	(333)
getpwent, getpwuid, getpwnam, setpwent, endpwent, fgetpwent (3C)	操纵口令文件表项	(334)
gets, fgets (3S)	从流中取一个字符串	(335)
getspent, getspnam, setspent, endspent, fgetspent, lckpwwdf, ulckpwwdf (3C)	操纵影子口令文件表项	(336)
getsubopt (3C)	分析串中的子选项	(338)
gettimeofday, settimeofday (3C)	获得或设置日期和时间	(342)
gettext (3C)	检索正文串	(343)
getut: getutent, getutid, getutline, pututline, setutent, endutent, utmpname (3C)	存取utmp文件项	(344)
getutx: getutxent, getutxid, getutxline, pututxline, setutxent, endutxent, utmpxname, getutmp, getutmpx, updwtmpt, updwtmptx (3C)	存取utmpx文件项	(346)
getvfsent, getvfsfile, getvfsspec, getvfssany (3C)	获得vfstab文件项	(349)
hsearch, hcreate, hdestroy (3C)	管理散列搜索表	(351)
initgroups (3C)	初始化补充组存取表	(353)
insque, remque (3C)	向队列中插入或从中删除元素	(354)
isnan, isnand, isnanf, finite, fpclass, unordered (3C)	确定浮点数的类型	(355)
l3tol, ltol3 (3C)	在三字节整数与长整数之间转换	(356)
localeconv (3C)	获得数字的格式化信息	(356)
lockf (3C)	对文件的记录加锁	(360)
lsearch, lfind (3C)	线性搜索与更新	(362)
makecontext, swapcontext (3C)	操纵用户文境	(364)
makedev, major, minor (3C)	管理设备号	(365)
malloc, free, realloc, calloc, memalign, valloc (3C)	内存分配程序	(366)
mbchar: mbtowc, mblen, wctomb (3C)	多字节字符处理	(367)
mbstring: mbstowcs, wcstombs (3C)	多字节串函数	(369)

memory: memccpy, memchr, memcmp, memcpy, memmove, memset (3C)	内存操作	(369)
mkfifo (ec)	创建新的FIFO	(370)
mktemp (3C)	产生唯一的文件名	(371)
mktime (3C)	把tm结构转换成日历时间	(372)
mlock, munlock (3C)	对内存页加锁(或解锁)	(374)
mlockall, munlockall (3C)	对地址空间加锁或解锁	(375)
monitor (3C)	准备执行造型文件	(375)
msync (3C)	使内存与物理存储媒体同步	(377)
nl_langinfo (3C)	有关语种的信息	(378)
offsetof (3C)	结构成员的偏移量	(379)
perror (3C)	显示系统出错消息	(380)
popen, pclose (3S)	启动去往/来自进程的管道	(380)
printf, fprintf, sprintf (3S)	显示格式化的输出	(382)
psignal, psiginfo (3C)	系统信号消息	(386)
putc, putchar, fputc, putw (3S)	把字符或字写到流上	(386)
puts, fputs (3S)	把串写到流上	(387)
putenv (3C)	改变或增加环境中的值	(388)
putpwent (3C)	写口令文件登记项	(389)
putspent (3C)	写影子口令文件登记项	(390)
qsort (3C)	快速排序	(390)
raise (3C)	向程序发送信号	(391)
rand, srand (3C)	简单的随机数发生器	(392)
realpath (3C)	返回真实的文件名	(392)
remove (3C)	删除文件	(393)
scanf, fscanf, sscanf (3S)	转换格式化的输入	(394)
setbuf, setvbuf (3S)	指定流缓冲	(397)
setjmp, longjmp (3C)	非局部的无条件转移	(399)
setlocale (3C)	修改和查询程序的地域	(400)
sigsetjmp, siglongjmp (3C)	带信号状态的非局部无条件转移	(402)
sigemptyset, sigfillset, sigaddset, sigdelset, sigismember (3C)	对信号集进行操纵	(403)
sleep (3C)	当前执行进程挂起一段时间	(404)
ssignal, gsignal (3C)	软件信号	(405)
stdio (3S)	标准的带输入/输出缓冲的软件包	(406)
stdipc: ftok (3C)	标准的进程间通信软件包	(408)

strcoll(3C)	字符串核对	(409)
strerror(3C)	取得出错信息串	(409)
strftime, cftime, asctime, (3C)	把日期与时间转换成字符串	(410)
string: strcat, strdup, strncat, strcmp, strncmp, strcpy, strncpy, strlen, strchr, strrchr, strpbrk, strspn, strcspn, strtok, strstr(3C)	字符串操作	(412)
strtod, atof(3C)	把字符串转换成双精度浮点数	(414)
strtol, strtoul, atol, atoi(3C)	把字符串转换成整数	(415)
strxfrm(3C)	字符串变换	(417)
swab(3C)	对换字节	(417)
sysconf(3C)	获得可配置的系统变量	(418)
system(3S)	发出shell命令	(419)
tcsetpgrp(3C)	设置终端前台进程组id	(420)
tmpfile(3S)	创建临时文件	(421)
tmpnam, tmpnam(3S)	为临时文件起名	(421)
truncate, ftruncate(3C)	把文件设置成指定的长度	(423)
tsearch, tfind, tdelete, twalk(3C)	管理二叉搜索树	(424)
ttyname, isatty(3C)	寻找终端名字	(427)
ttyslot(3C)	在当前用户的utmp文件中找一个空位	(428)
ungetc(3S)	把字符推回输入流中	(429)
vprintf, vfprintf, vsprintf(3S)	显示可变实参表的格式化的输出	(429)
elf(3E)	目标文件访问库	(431)
elf_begin(3E)	构造文件描述字	(435)
elf_cntl(3E)	控制文件描述字	(439)
elf_end(3E)	结束对目标文件的使用	(440)
elf_errmsg, elf_errno(3E)	出错处理	(441)
elf_fill(3E)	设置填充字节	(442)
elf_flagdata, elf_flagehdr, elf_flagelf, elf_flagphdr, elf_flagscn, elf_flagshdr(3E)	操纵标志	(443)
elf_fsize: elf32_fsize(3E)	返回目标文件类型的大小	(444)
elf_getarhdr(3E)	检索档案成员头	(445)
elf_getarsym(3E)	检索档案符号表	(446)
elf_getbase(3E)	获得目标文件的基位移	(447)
elf_getdata, elf_newdata, elf_rawdata(3E)	取得节数据	(447)

elf_getehdr: elf32_getehdr, elf32_newehdr (3E)	检索依赖于类的目标文件头	(451)
elf_getident (3E)	检索文件标识数据	(452)
elf_getphdr: elf32_getphdr, elf32_newphdr (3E)	检索依赖于类的程序头表	(453)
elf_getscn, elf_ndxscn, elf_newscn, elf_nextscn (3E)	获得节的信息	(454)
elf_getshdr: elf32_getshdr (3E)	检索依赖于类的节头	(456)
elf_hash (3E)	计算散列值	(457)
elf_kind (3E)	确定文件类型	(457)
elf_next (3E)	顺序档案成员访问	(458)
elf_rand (3E)	随机档案成员访问	(459)
elf_rawfile (3E)	检索未经解释的文件内容	(460)
elf_strptr (3E)	建立字符串指针	(461)
elf_update (3E)	更新ELF描述字	(462)
elf_version (3E)	协调ELF库与应用程序的版本	(465)
elf_xlate: elf32_xlatetof, elf32_xlatetom (3E)	依赖于类的数据转换	(466)
nlist (3E)	从名字表中获得登记项	(468)
basename (3G)	返回路径名中的最后一个元素	(469)
bgets (3G)	从流中读入直至下一个定界符	(469)
bufsplit (3G)	将缓冲区分解成字段	(470)
copylist (3G)	把文件拷贝到内存中	(471)
dirname (3G)	报告文件路径名中父目录的名字	(472)
gmatch (3G)	shell全局模式匹配	(473)
isencrypt (3G)	确定字符缓冲区是否已加密	(474)
mkdirp, rmdirp (3G)	创建、删除路径中的目录	(475)
p2open, p2close (3G)	打开或关闭命令的管道	(476)
pathfind (3G)	在指定的目录中搜索命名的文件	(478)
regcmp, regex (3G)	编译并执行正则表达式	(479)
regexpr, compile, step, advance (3G)	正则表达式编译与匹配例程	(481)
str: strfind, strrspn, strtrns (3G)	串操纵	(484)
strccpy: streadd, strcadd, strecpy (3G)	拷贝字符串, 压缩或扩展转义代码	(485)
intro (3M)	数学库介绍	(486)

bessel: j0, j1, jn, y0, y1, yn (3M)	贝塞尔函数	(487)
erf, erfc (3M)	误差函数及误差函数的补	(488)
exp, expf, cbrt, log, logf, log10, log10f, pow, powf, sqrt, sqrtf (3M)	指数函数、对数函数、幂函数、平方根函数	(488)
floor, floorf, ceil, ceilf, copysign, fmod, fmodf, fabs, fabsf, rint, remainder (3M)	下整数函数, 上整数函数, 余数函数, 绝对值函数	(490)
gamma, lgamma (3M)	对数伽玛函数	(491)
hypot (3M)	欧几里德距离函数	(492)
matherr (3M)	出错处理函数	(492)
sinh, sinhlf, cosh, coshlf, tanh, tanhlf, asinh, acosh, atanh (3M)	双曲函数	(496)
trig: sin, sinf, cos, cosf, tan, tanf, asin, asinf, acos, acosf, atan, atanf, atan2, atan2f (3M)	三角函数	(497)
assert (3X)	验证程序断言	(499)
crypt (3X)	口令及文件加密函数	(499)
dlclose (3X)	关闭共享目标	(501)
dlderror (3X)	获得诊断信息	(502)
dlopen (3X)	打开共享目标	(503)
dlsym (3X)	取得共享目标中符号的地址	(505)
libwindows (3X)	窗口终端函数库	(506)
maillock (3X)	管理用户邮箱的锁文件	(508)
malloc, free, realloc, calloc, mallopt, mallinfo (3X)	内存分配程序	(509)
sputl, sgetl (3X)	以独立于机器的方式访问长整数数据	(512)

4. 文件格式

intro (4)	文件格式介绍	(513)
a.out (4)	ELF(可执行和连接格式) 文件	(513)
ar (4)	档案文件格式	(515)
core (4)	内存映像文件	(517)
limits (4)	针对具体实现的常量的前导文件	(519)
sccsfile (4)	SCCS文件的格式	(521)
strftime (4)	针对具体语言的串	(524)
timezone (4)	设置默认的系统时区	(525)

utmp, wtmp (4)	utmp和wtmp登记项的格式	(525)
utmpx, wtmpx (4)	utmpx和wtmpx登记项的格式	(527)

5. 杂项

intro (5)	杂项介绍	(530)
ascii (5)	ASCII字符集映象	(530)
environ (5)	用户环境	(531)
fcntl (5)	文件控制选项	(535)
jagent (5)	窗口终端的宿主机控制	(537)
langinfo (5)	语言信息常量	(538)
layers (5)	在layers (1)下宿主机与窗口终端的协议	(540)
math (5)	数学函数及常量	(543)
nl_types (5)	本国语言数据类型	(544)
prof (5)	函数中的造型统计	(544)
regexp: compile, step, advance (5)	正则表达式编译和匹配例程	(546)
siginfo (5)	信号产生信息	(550)
signal (5)	基本信号	(552)
stat (5)	由stat系统调用返回的数据	(556)
stdarg (5)	处理可变的实参表	(557)
types (5)	基本的系统数据类型	(559)
ucontext (5)	用户上下文	(560)
values (5)	与机器有关的值	(561)
varargs (5)	处理可变实参表	(562)
wstat (5)	等待状态	(564)
xtproto (5)	xt驱动程序使用的多路通道协议	(565)

1. 命令

intro(1)

名称

intro——编程命令介绍

说明

这一节按字母顺序介绍编程命令。除非另外注明,否则命令按照下面的语法接收选项和其他实参:

name [*option*(*s*)] [*cmdarg*(*s*)]

这里,

name 是一个可执行文件的名字。

option 是 `-noargletter(s)` 或 `-argletter <>optarg`, 这里:

noargletter 是单个字母,表示一个不需要实参的选项

argletter 是单个字母,表示一个需要实参的选项

< > 可有可无的白空类符

optarg 是一个选项实参(字符串),与前面的 *argletter* 匹配。

cmdarg 本身是 '-', 指明标准输入,或不以 '-' 开始的路径名(或其他命令实参)。

在本手册中,很多地方引用 *TMPDIR*、*BINDIR*、*INCDIR* 和 *LIBDIR*, 这些代表的是目录的名字,必要时在每一条命令上指定它们的值。例如, *TMPDIR* 可能指的是 `/var/tmp`。它们不是环境变量,不能被赋值(但确实有一个叫 *TMPDIR* 的环境变量,其值是可以设定的。见 `tmpnam(3S)`)。也有不少地方引用 *LIBPATH*, 这是连接编辑器和其它工具的默认搜索路径。

参见

`exit(2)`, `wait(2)`, `getopt(3C)`。

《用户参考手册》中的 `getopts(1)`。

诊断

每条命令终止时都返回两个字节的的信息。其中一个是由系统提供的,它给出命令终止的原因;另一个(在“正常”终止时)是由程序提供的,见 `wait(2)` 和 `exit(2)`。前一个字节为 0 表示正常终止;后一字节通常以 0 表示执行成功,如果不是 0,则表示出了问题,比如实参错,数据错或不可访问。该消息有几种叫法:“出口码(exit code)”,“出口状态(exit status)”或“返回码(return code)”。它仅当特殊约定时才给予解释。

告诫

当处理包含空符的文件时,有些命令会产生预想不到的结果。这些命令经常会把文本输入行当做是字符串,从而和字符串里的空符(字符串终结符)相混淆。

admin(1)

名称

admin——创建和管理 SCCS 文件

格式

```
admin [-n] [-i [name]] [-rrel] [-t [name]] [-fflag [flag-val]] [-dflag [flag-val]]
[-a login] [-e login] [-m [mrlist]] [-Y [comment]] [-h] [-z] files
```

说明

admin用来创建新的SCCS文件和修改已存在的SCCS文件的参数。**admin**的实参可以以任意次序出现,它们由以“-”开始的关键字母实参和命名文件组成(注意,SCCS文件名必须以字符 **s.** 开始)。假如指定的文件不存在,要创建这个文件,按指定的关键字母实参初始化其参数。对于那些未被关键字母实参初始化的参数,系统赋给它一个默认值。如果指定的文件存在,则改变那些对应于指定的关键字母实参的参数,而其它的参数不变。

如果指定了一个目录,那么**admin**把该目录中的每一个文件当作是一个命名文件,而对非 SCCS 文件(路径名的最后一个部分不是以 **s.** 开始)和不可读的文件,**admin** 不作提示就把它们忽略掉。假如给出的名字是“-”,则从标准输入读;每一个标准输入行被当作一个 SCCS 文件的名字来处理。对于非 SCCS 文件和不可读文件,系统不给提示就把它们忽略掉。

关键字母实参如下所述。由于实参独立地作用到每一个命名的文件,因此,每个实参的解释就像只有一个命名的文件要处理时那样。

-n	这个关键字母表示要创建一个新的SCCS文件。
-i [name]	<i>name</i> 是一个文件名,新的SCCS文件的正文将从该文件中获得。该正文构成了文件的第一次 delta (参阅 -r 关键字母中有关 delta 的编号方式)。如果使用了关键字母 i ,但省略了文件名,则通过标准输入获得正文,直到遇到文件尾为止。如果省略了这个关键字母,那么把 SCCS 文件创建成空文件。在提供 i

关键字的 **admin** 命令下,只能创建一个 SCCS 文件。使用单个的 **admin** 命令来创建两个或更多的 SCCS 文件,必须把它们创建成空文件(没有 **-i** 关键字)。注意, **-i** 关键字蕴含着 **-n** 关键字。

-rrel 这是插入了初始 **delta** 的版本。这个关键字仅当使用了 **-i** 关键字时才可以使用。如果没有使用 **-r** 关键字,那么初始 **delta** 插入到版本 1 中。初始 **delta** 的级别总是 1(默认情况下,初始 **delta** 被命名为 1.1 版)。

-t[name] *name* 是一个文件的名称,从这个文件中获取这个 SCCS 文件的描述性正文。如果使用了 **-t** 关键字,同时 **admin** 命令创建了一个新 SCCS 文件(还使用了 **-n** 和(或) **-i** 关键字),那么必须提供描述性正文文件名。在 SCCS 文件存在的情况下:

(1) 没有文件名的 **-t** 关键字会删除当前在 SCCS 文件中的描述性正文(如果有的话)。

(2) 有文件名的 **-t** 关键字使命名的文件中的正文(如果有的话)代替当前在 SCCS 文件中的描述性正文(如果有的话)。

-f flag 这个关键字指定了一个将要放在 SCCS 文件中的标志及其值。在单个的 **admin** 命令行中可以有几个 **-f** 关键字。允许的标志以及它们的值如下:

b	允许在 get 命令上使用 -b 关键字来创建分支 delta 。
cceil	最大数(即上整数):它比0大,但小于或等于9999,可以用 get 命令检索以便编辑它。未规定的 c 标志默认为9999。
ffloor	最小数(即下整数)。它比0大但小于或等于9999,可以用 get 命令检索以便编辑它。未规定的 f 标志默认为1。
dSID	get 命令使用的默认 delta 编号(SID)。
i[str]	它使由 get 或 delta 命令发出的“没有标识关键字(ge6)”的消息,被当作是致命的错误来处理。当没有该标志时,这个消息只是一个警告。假如在被搜索或被储存的正文中没有找到 SCCS 标识关键字(参阅 get(1)),就发出这个消息。如果给出了值,关键字必须与给出的字符串完全匹配。这个字符串必须包含一

- 个关键字,同时没有嵌入新行。
- j** 允许并发的`get`命令在SCCS文件同样的SID版本上编辑。该标志允许对 SCCS 文件的同样的版本进行多重的并发修改。
- l***list* 不再产生`delta`的版本清单(使用`get -e`可防止这些版本被修改)。*list* 有下列语法:
- `<list> : : = <range>|<list>,<range>`
`<range> : : = RELEASE NUMBER|a`
- list*中的字符等价于为命名的SCCS文件规定了所有版本。
- n** 当在一个新的版本中产生一个`delta`时,要跳过一些版本(如果有的话), (例如,在 `delta2.7` 后产生 `delta5.1` 时,版本 3 和 4 被跳过)。n 使得 `delta` 命令在每个被跳过的版本中创建一个“空”`delta`。这些空 `delta` 用来做为“锚点”,以便以后的分支 `delta` 可以由此创建。没有这个标志会使得在 SCCS 文件中不存在被跳过的版本,从而在将来无法由此创建分支 `delta`。
- q***text* 可由用户定义的正文,它可代替`get`命令搜索的SCCS文件正文中出现的所有`%Q%`关键字。
- m***mod* 这是SCCS文件的模块名称,用来代替`get`搜索的SCCS文件正文中出现的所有`%M%`关键字。如果没有指定 `m` 标志,那么其值就为去掉 SCCS 文件的头 `s.`后的名称。
- t***type* 这是SCCS文件的模块类型,用来代替`get`搜索的SCCS文件正文中出现的所有`%Y%`关键字。
- v** [*pgm*] 使`delta`命令产生修改请求(MR)编号的提示信息,作为创建一个变化的原因。任选的值说明一个 MR 编号合法性检查程序(参阅 `delta(1)`)的名称。这个程序将把模块名、类型标志的值(见上文 `ttype`)和 `mrlist` 作为其实参。如果创建 SCCS 文件时设置了这个标志,那么也必须使用 `m` 关键字母,而不管其值是否为空。
- d***flag* 从SCCS文件中删除指定的标志。仅当处理已存在的SCCS文件时,才可以指定`-d`关键字母。在单个 `admin` 命令中可以提供几个`-d`关键字母。允许的标志名称,见`-f`关键字母。