汉化UNIX系统 V/386

用户及系统管理参考手册

（3.20版）

苏州电子计算机厂

汉化UNIX系统 V/386

用户及系统管理参考手册

（3.20版）

下

编　　译　盛　啸　许晓荣

诸　华　周巧生

沈志方　叶慧平

许建伟

校　　订　葛健豪

责任编辑　张盛森

苏州电子计算机厂

一九九一年二月

前　言

本手册描述了UNIX系统的特征。它所提供的既不是UNIX系统的概观，也不是系统执行过程的细节，而是叙述其在计算机上运行的基本软件的命令。

本手册分为三个部分：

- **1—系统命令、系统维护命令以及应用程序；**
- **7—特殊的文件；**
- **索引包。**

诸如name（1M）、name（7）之类形式的命令以及后面带着（1）、（1C）、（1G）之类的命令均表示该命令在本手册中。命令后跟着的数字主要是为了方便对照（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》中）。涉及到诸如name（N）形式的条目，N是一个数字〔(2)、(3)、(4)或(5)〕，数字后可能还跟上一个字母，指的是《程序员参考手册》第N章节的name条目。

命令部分的所有条目均按字母次序排列，只有intro（1）例外，它作为第一条目。某些条目也许描述了几个例行程序、命令等，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；“次名”直接列在相关的主命令下。

1部分　所包含的命令和程序是：

- **用于管理UNIX系统的命令和程序；**
- **由用户直接调用或由命令语言过程调用的命令和程序。**

这与子程序有所不同，子程序是由用户的程序来调用的。

命令通常留驻在目录/bin（表示二进制程序）下，也有些留驻在目录/usr/bin下。这两个目录可由称之为shell的命令解释程序来进行自动搜索。shell将搜索在用户。profile中的路径，用户应确认已在.profile中建立了该路径。在计算机上运行的UNIX系统还有一个称为/usr/lbin的目录，该目录包含了一些局部命令。

该部分分下列几个子类：

- **1—通用命令；**
- **1C—通信命令；**
- **1G—图形命令；**
- **1M—维护命令；**

7部分　讨论了涉及输入/输出设备的系统文件的特性。该部分中的文件名通常就是硬件的设备名，而不是特殊文件本身的名字。

索引包部分所出现的实用程序包是：

1. **基本系统；**
2. **编辑软件包；**
3. **远程终端包。**

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）：

■名字 给出了主名（在有些情况下还有次名），并且简单陈述了其用途。

■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。

□斜体字印的字符串通常表示可替换的自变量原型及在手册的其它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。当自变量原型是“name”或“file”时，它总是指文件名。

□省略号…表示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(-)、加号(+)和等号(=)开头的自变量通常被看成是某种类型的标志自变量，即使它出现在文件名可以出现的位置也是如此。因此，以－、＋、＝来作为文件名的开头是不明智的。

■说明 讨论如何来使用这些命令。

■例子 适当地给出一些使用举例。

■文件 包含了由该程序访问的文件的名字。

■退出码 讨论了命令中止时的值，该值适用于shell环境变量'?'（见sh(1)）。

■参见 提供了相关信息的线索。

■诊断 讨论了可能出现的错误信息，自释用的信息不列出来。

■警告 讨论了各条命令的限制和边界条件。

■缺陷 列出了软件中已知的但还未改正的错误。必要时还给出一个简短的建议修补方法。

（由于本手册第1节第7两部分均为命令和硬件设备名，有近300条条目；索引包部分的实用程序包又以三种内容散在各相关的命令条目中，如果列出全目录，需占不少篇幅且不说，还相当繁琐，也难免重复，故我们对本手册只给出前两部分总目录，而细目只根据各编译者的任务，按字母顺序列出一部分。这可能会给读者在阅读和查找时带来一定的困难和不便，好在本“诸言”介绍详尽，如果能细读定有所补。敬请谅解——编辑者。）

怎样开始

这里的讨论提供了用户开始使用UNIX系统所需的基本信息：包括如何注册和注销，如何通过终端进行通信以及如何运行一个程序(有关系统更完善的介绍，参见《用户指南》)。

注册

用户必须把UNIX系统与一控制台或全双工ASCII 终端相连，还必须有一个合法的注册名，该注册名可以从系统管理员处获得（同时还可得到如何访问你的UNIX系统方面的信息）。通常的终端速率是每秒120，240，480和960个字符（即1200，2400，4800和9600波特）。有些UNIX 系统对于不同的终端速度有不同的访问方法，而有些系统则是通过一个公共的访问方法来提供几种速度。在后一种情况下，有一种“优先”速度。若用户用设置成不同速度的终端来访问系统，则该用户会面临一串无意义的字符的出现（login：错误终端速度下产生的信息）。此时，只要按“break”，“interrupt”或“attention”键，就会出现login：信息。

大多数终端都有一个用来选择合适速度的速度选择开关和用来置成全双方式的半双工／

全双工转换开关。当联接完成后，系统显示**login:**，用户键入注册名，然后按回车键，若用户有个口令，则系统会向用户询问口令，但系统不把口令回送到终端上。用户注册成功后，“return”，”new-line”和“line-feed”键的意义是相同的。

在注册时打入的注册名应为小写形式，否则UNIX系统会认为该终端只能产生大写字母，并把后面的注册信息全当成是大写形式。

用户注册时，在接收到提示符前可能先碰到有关日期的信息。有关更详细的信息见login(1)，它详尽地讨论了注册过程和stty(1)，这条命令告诉用户如何来描述用户终端。profile(4)（在《程序员参考手册》中）解释如何在每次注册时自动也完成这个后续工作。

退出

有二种注销方法：

- 若用户是拨话进入的，则只要挂断电话就行。
- 用户可向shell打入文件结束标记来注销（ASCII中的EOT字符，键入时打“CTRL-D”），shell将中止，屏幕上将再次出现**login:**信息。

如何通过终端进行通信

当用户向UNIX系统输入字符时，用户所键入的字符被收集起来并暂存，虽然字符能回送给用户，但这些字符并没有被传给程序，直至用户键入“return”（或“new-line”）时才被传送。

UNIX系统终端的输入/输出是全双工的。它具有完整的预读功能。这意味着你可以在任何时刻键入字符，即使当时有一程序正在向你显示信息。当然，如果在终端输出时打入字符，那么输入的字符将与输出字符混杂在一起。在任何情况下，打入的内容都会按正确的程序被保留和解释。预读的总数是有限制的，但是它很大，因此，不大可能被超出。

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

CTRL-S（ASCII DC3符）的键入可以通过同时按下控制（controt）键和字母键S来完成，用来暂停输出。对于CRT终端，这个键是有用的，它可以防止输出信息在还未看清之前就消失。打入CTRL-Q可恢复输出（DC1符）。这样，若你打入了命令**cat yourfile**中的内容在你看清之前，屏幕上会出现很快滚过现象，你可以打入CTRL-S来暂时冻结输出；打入CTRL-Q后，将使输出恢复其正常的速度。CTRL-S和CTRL-Q在这种方式下使用时，不会传给任何其它程序。

ASCII的DEL（或“rubout”）字符也不传递给程序，但它要产生一个中断信号，就象“break”、“interrupt”、“attention”信号一样。这个信号常会引起正在运行的程序终止。其典型的用法是用来终止你不需要的长篇输出。对于程序，可以设计成统统忽略这个信号，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）。例如编辑程序ed(1)序就能捕捉到中断并暂停正在做的工作，而不是终止编辑程序的运行。因此，一个中断可用来停止一个编辑程序的输出而不会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”和“换行（line-feed）”组来模拟也尽量做到灵活。在后一种情况中，所有“回车”符的输入都改成输入“换行”符（标准的行结束符），并且“回车”和“换行”组被送到终端上。

制表符（tab）在UNIX系统的源程序中可以自由地使用。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符，在输入时也回送空格符。还有，stty(1)可以设置或再置这种方式。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，可以在你的终端上设置制表标记位置。

如何运行程序

当用户成功地注册进入UNIX系统后，一个称为shell的程序处于与该用户通信状态，shell读入用户打入的每一行并把它的分解成命令名和它的自变量，然后执行命令。一条命令是一个可执行程序。通常shell总是首先以该给定的名字查看用户的当前目录，若不在该目录中，则查找系统目录（例如/bin或/usr/bin）。系统提供的命令除了它们被存放在shell可以找到的目录中以外没有什么特别。用户也可以把命令放在自己的目录下并引导shell在用户目录中寻找命令，有关shell的环境变量$PATH的讨论，见本手册条目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；命令和它的自变量用空格符或制表符隔开。

当一个程序终止时，shell通常将重新获得控制并返回一个提示符，表示它已为接受另一条命令准备好了。

当前目录

UNIX系统有一个按目录层次结构组成的文件系统。当接受了你的注册名后，系统管理员就为你建立一个目录（通常该目录名与你的注册名相同，称作为注册目录或主目录）。当你注册时，该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中。由于你是该目录的拥有者，因此你具有全部的读、写、删、改的权限。进入或修改其它目录和文件的许可，要由相应的文件拥有者或系统管理员决定。要改变当前目录，使用cd(1)命令。

路径名

为了引用不在当前目录下的文件或目录，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。在斜线符之后，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）。使用pwd(1)可以打印出你正使用的目录的完整路径名、有关路径名的正式定义见《程序员参考手册》中的intro(2)。

若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。

影响文件的重要命令有cp(1)，mv（见cp(1)）和rm(1)，这三条命令分别用来复制、移动（即更名）和删除文件、要查文件或目录的状态，使用ls(1)，创建目录使用mkdir(1)，而删除目录则使用rmdir（见rm(1)）。

文本输入和显示

几乎所有的文本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)。在终端上打印本文的常用命令有cat(1)，pr(1)和pg(1)。cat(1)命令在终端上显示ASCII文本文件的内容，不作任何处理。pr(1)命令为文本编页、提供标题，并且具有多列输出的能力。pg(1)命令连续地显示文本，每次显示的内容不超过终端屏幕。

与其它用户通信

有些命令提供了内部用户通信功能。即使你不相使用它们，最好也要有所了解，因为其它用户可能想与你联系。mail(1)和mailx(1)将把信息留下，并在另一个用户下次注册时通知这个用户或者在整个过程中，每隔一定时间间隔就发出通知、要与另一个当前已注册的用户通信，则使用write(1)，若你是他们的目标，则本手册中的对应条目还有如何来对这二条命令响应的建议。

有关与其它用户通信的细节详见《操作/系统管理指南》第八章。

本手册全部编译稿，承蒙周裕安、葛健豪同志作了审改和校订，在此谨表谢意。

目　录

第一部分

1—系统、维护命令及应用程序

INTRO(1)

名字

intro——介绍命令和应用程序

说明

本节按字母次序描述适用于你的计算机的命令（包括系统维护命令）。有关本节中的命令应与列在《程序员参考手册》第1，2，3，4，5章节中的命令一道使用。诸如name(1)，name(2)，name(3)，name(4)，name(5)之类形式的命令指的是《程序员参考手册》中的条目，而诸如name(1)，name(1M)，name(1C)，name(1G)，name(7)或name(8)之类形式的命令则指的是本手册中的条目。用法上的某些区别在标题中指出。

下列应用程序包与计算机一起提供：

- 基本系统实用程序
- 编辑程序包
- 扩展终端接口支持软件包
- 安全管理软件包
- 2K字节文件系统实用程序包
- 网络支持实用程序包
- 远程文件共享实用程序包

手册中的命令句法

除非另有说明，在手册中每页的“格式”这一节中描述的命令根据下面的句法来接受任选项和其它自变量，并且应按下面所作的说明来加以解释。

name〔-option…〕〔cmdarg…〕

这里

〔〕　把不是一定要的任选项或命令自变量括起来。

…　表示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。注意在一个“—”符后可以有不止一个的noargletter任选项组合在一起（下文的规则5）。

argletter　单个的字母，表示一个任选项需带任选项自变量。

optarg　与前置的argletter相配的任选项自变量（字符串）。二者必须用逗号隔

开，或者用空格隔开并用引号括起来（下文的规则 8 ）。

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

命令句法标准：规则

这些命令句法规则不是所有当前命令都要遵循的。但所有的新命令都将遵循这些规则。所有的 shell 过程都要用 getopts(1) 来分析定位参数并检查任选项的合法性。它支持下面的规则 3 至规则10，其它规则的执行由命令本身来进行。

1．命令的名字（上面的name）为2～9个字符长。

2．命令名必须是小写字母和数字。

3．任选项名（上面的option）为 1 字符长。

4．所有的任选项都必须前置“—”。

5．不带自变量的任选项可在一个“—”后组合起来。

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．任选项自变量不能任选。

8．一个任选项后跟的任选项自变量组必须用逗号隔开，或者用空格分开并用引号括起来（例如：-0 xxx，z，yy或-0“xxx z yy”）。

9．命令行中的所有任选项必须前置一个操作对象（上面的cmdarg）。

10．“--”用来指明任选项的结束。

11．任选项间的相对次序没有关系。

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“—”后加空格仅用来表示标准输入。

参见

getopts(1)，exit(2)。

《程序员参考手册》中的wait(2)，getopt(3c)。

本手册前面的“怎样开始”章节。

诊断

一旦命令执行终止，将返回两个字节的状态信息，其中一个由系统提供，它指出了终止的原因，另一个由程序提供（参见wait( )和exit(2)）。前一个字节在正常结束时，其值为 0；后一字节在执行成功时，其值通常也为 0，一个非零的值表示遇到了诸如出现错误的参数，坏的或不可访问的数据之类的麻烦。通常把后一个字节称为“退出码”、“退出状态”或“返回码”，它仅在有专门的约定时才给予解释。

缺陷

遗憾的是，不是所有的命令都遵守前述的句法的。

警告

有些命令在处理含有空白字符的文件时会产生意想不到的结果，通常会把文本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。

名字

300，300s——用来处理DASI 300和300s型终端的特殊功能。

格式

300〔+12〕〔-n〕〔-dt,l,c〕

300s〔+12〕〔-n〕〔-dt,l,c〕

说明

命令300支持DASI 300（GSI 300或DTC 300）终端的特殊功能，并对它的使用进行了优化。300s对DASI 300s（GSI 300s或DTC 300s）型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。该命令还能绘制希腊字母和其它特殊字符，同时也允许方便地使用每英寸12个字符的文本。它能节省约5％至7％的打印时间（编译者注：原文为5％至70％）。命令300还可用来清楚地打印方程。打印序列是这样的：

neqn file… 〔nroff〕300

告诫信息：若用户终端带有PLOT开关，则在命令300使用前，确认它处于ON状态。

命令300的动作可以通过任选标志自变量来改变，以便能处理每英寸12个字符的文本、分数线、信息和延迟。

+12 允许使用每英寸12个字符，每英寸6行的文本。DASI 300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，并且加+12任项项。

-n 控制半行的宽度。缺省时，半行宽度等于4个垂直图形增量。由于每个增量等于1/48英寸，故每行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的第一位数字可使缺省值无效，因此用户可按个人的喜好来安排下标和上标的位置。例如，nroff中的半行可以用-2任选项使它做起来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，当然首先要把PITCH置到12的位置上。

-dt,l,c 控制延迟因子。缺省时为-d3，90，30。DASI 300终端在碰到很长的行、带有许多制表符或非空白、不同的字符长串时有些会产生奇异的输出。在一行中对每隔t个制表符或每隔c个非空白、非制表符的连续字符串，将插入一个空白（延迟）字符。若一行的长度大于l个字节，则在该行结尾插入1+（总长度）/20个空白符。使用隐含的缺省值，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。前者用于C语言程序，而后者则用于诸如/etc/passwd之类的文件。由于终端的动作根据印出和装入到系统的特殊字符的不同而有所变化。所以，用户需要用这些值进行试验，以得到正确的输出。-d任选项的存在仅是作为在其它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来打印，而对呈锯齿形的多层C语言程序来说，用值-d0，1则很好。

注意，延迟控制与正在用的回车与换行之间的相互影响很大，建议在大多数情况下使用stty(1)的n10 cr2或n10 cr3方式。

在需要用手工输纸或在文件的中间改变字形时，命令300可以与nroff的-s 标志或 .rd 请求一起使用。在这种情况下你就必须用“换行”键来得到响应，而不是用回车键。

在许多（不是所有）情况下，下面的序列是等效的：

nroff-T300 files…和nroff files…|300

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

因此，除非需要特殊的延迟或任选项，可以避免使用命令300。然而在某些情况下，300附加的移动优化会产生排列得更好的输出。

参见

450(1)，mesg(1)，graph(1G)，stty(1)，tab(1)，tplot(1G)

缺陷

有些特殊字符在第一列不能正确打出，这是因为打印头不能再从第一列往左移的缘故。

若输出中有希腊字母或反向换行符，则使用摩擦供纸来代替纸页牵引。虽然采用纸页牵引方式对于大多数图形都足够好，但在碰到一个或多个反向换行时，会使文本的第一行出错，且在改变方向和变形希腊字母时，会出现错误。

注意

troff(1)，nroff(1)和eqn(1)不属于这个UNIX系统版本的内容。

4014(1) （扩展终端接口支持程序）

名字

4014——为TEKTRONIX 4014终端分页。

格式

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

说明

命令4014的输出是用于TEKTRONIX 4014终端的。为了与屏幕相适应，4014把输出分为66行/屏并且把屏幕分成N栏，在单栏的情况下（缺省情况），自动加入一个8个空格的页位移。必要时不能把制表符、空格符和退格符接受并标绘出来。37型TELETYPE（电传打字机）的半行和逆行序列能被解释并标绘出来。在每一页的页尾，4014等待来自键盘由用户输入的换行符，然后继续处理下一页。在这个等待状态下，命令!cmd将把cmd送给shell。

该命令行的任选项为：

-t　页间不等待（把输出定向到一文件中时有用）。

-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待。

-pL　设置页长为L；L接受比例量因子i（英寸）和l（行），缺省时为行。

参见

pr(1)

450(1) （扩展终端接口支持程序）

名字

450——处理DASI 450终端的特殊功能。

格式

450

说明

命令450支持DASI 450或诸如Diablo 1620或Xerox 1700之类的与DASI 450终端功能一致的终端的特殊功能并且能优化使用。它把半行正向、反向和整行反向的移动转换成正确的垂直移动，还试图以与300(1)相同的方式来标绘希腊字母和其它特殊符号。有必要指出，除非系统含有“文件工作台”软件，否则有些命令（例如，eqn，nroff，tbl等）将无法工作。命令450还可用来整齐地打印等式。打印序列如下：

neqn file…|nroff|450

告诫信息：在命令450使用前，要保证终端的PLOT开关处于ON位置，SPACING开关应置在合适的位置上（或者是10个字符/英寸或者是12个字符/英寸）。在这二种情况下，垂直方向的间距都是6行/英寸，除非用户适当的转义序列动态地将其改成8行/英寸。

在需要手工输纸或在文件中间改变字形的时候，可以把命令450与nroff的-s标志和.rd请求结合使用。在这种情况下，用户必须使用“换行”键来得到响应，而不是回车键。

在许多（不是所有）的情况中，命令450使用下列序列来替代更为有利：

nroff-T450 files…

nroff-T450-12 files…

因此，除非需要特殊的延迟或任选项，可以避免使用命令450。然而在某些情况下，450的附加的移动优化会产生排列得更好的输出。

参见

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

缺陷

有些特殊字符在第一列不能正确打出，这是因为打印头不能再从第一列往左移的缘故。

若输出中有希腊字母或反向换行符，则使用摩擦供纸来代替纸页牵引。虽然采用纸页牵引方式对于大多数图形都足够好，但在碰到一个或多个反向换行时，会使文本的第一行出错，且在改变方向和变形希腊字母时，会出现错误。

注意

troff(1)，nroff(1)和egn(1)不属于该UNIX系统版本的内容。

ACCEPT（1M） （基本系统）

名字

accept，reject——允许或禁止LP申请

格式

/usr/lib/accept destinations

/usr/lib/reject [-r [reason]] destinations

说明

命令 accept 允许lp(1)为命名的 destinations 而接受申请。destinalions 可以是一台行式打印机(LP)，或者是一组打印机。使用lpstat(1)来查找destinations的状态。

命令reject禁止lp(1)为命名的destinations 而接受申请。destinations或者一台行式打印机(LP)，或者是一组打印机。使用lpstat(1)可以查看 destinations 的状态，下面的任选项用于reject命令：

-r〔 reason〕　与禁止lp接受申请有关的reason，这个reason适用于所有的打印机，并且在下一个-r任选项之前都有效。当用户直接对命名的destinations提出申请或者由 lpstat(1)来申请时，reason 由lp 打出来。若 -r 任选项不存在或者给出的是不带 reason 的 -r 任选项，则使用缺省的 reason。

文件

/usr/spool/lp/*

参见

enable(1)，lp(1)，lpadmin(1M)，lpsched(1M)，lpstat(1)。

ACCT(1M)　（基本系统）

名字

acct：acctdisk，acctdug，accton，accwtmp——记帐和各种记帐命令的概观

格式

/usr/lib/acct/acctdisk

/usr/lhb/acct/acctdusg〔 -ufile〕〔 -pfile〕

/usr/lib/acct/accton〔 file〕

/usr/lib/acct/acctwtmp "reason"

说明

记帐软件被构造成一组工具（该组工具由 C 语言和 shell 过程组成），它可用来建立记帐系统。在安装系统时，记帐系统最初处于"off"（关闭）状态。acctsh（1M）描述了建立在 C 语言顶部的 shell 过程组。

联机时间记帐由各种不同的程序来处理，这些程序把记录写到 /etc/utmp 中，如同在 utmp(4)中描述的一样。描述在 acctsh（1M）中的程序把该文件转换成会话（使用终端）时间和管理记录，它们最后由acctmerg（1M）来汇总。

进程记帐由UNIX系统核心来执行。当一个进程终止时，每个进程有一个记录被写到一个文件中(通常为 /usr/adm/pacct)。在accprc(1M)中的程序对这类数据进行汇总，用来管理；acctcms(1M)被用来汇总命令的使用情况。当前的进程数据可以使用acctcom(1)来确定。

进程记帐和联机时间记帐（或如acct(4) 所描述的格式的任何记帐记录）可通过 acctmerg 来合并并且汇总成总的记帐记录（参见 acct(4) 中的 tacct 格式）。prtacct（参见 acctsh(1M)）被用来对任一或所有记帐记录进行格式化。

acctdisk 读取含有用户 ID、注册名和盘区数目的行，并且把它们转换成可与其它记帐记录合并的总的记帐记录。

acctdusg从标准输入（通常从find/－print）处读入且计算从注册起的磁盘资源的占用

情况（包括间接盘区）。若给出任选项-u，则acctdusg管理的由那些文件名组成的记录都不被放到file中（寻找试图逃避磁盘用户的潜在的原始资料）。若给出任选项-p，则file是口令文件的名字，若口令文件为/etc/passwd，则不需要这个任选项（更详细的内容见diskusg(1M)）。

accton单独地关闭进程记帐功能。若给出file，则它必须是一个存在的文件的名字，该文件由核心来向它附加进程记帐记录（参见acct(2)和acct(4)）。

acctwtmp把一个utmp(4)记录写到标准输出上，该记录包含有当前的时间和一串用来描述reason的字符串。ACCOUNTING的记录类型被指定（参见utmp(4)）。reason须为不超过11个字符的字符串，该字符串包括字母、数字、$符或空格符。例如：下面是建议的分别用于重新自举和关闭过程的字符串：

```
acctwtmp uname >>/etc/wtmp
acctwtmp "file save" >>/etc/wtmp
```

文件

/etc/passwd	用于注册名与用户ID的转换
/usr/lib/acct	容有本手册子类1M中列出的所有记帐命令
/usr/adm/pacct	当前进程的记帐文件
/etc/wtmp	注册/注销经历文件

参见

acctcms(1M), acctcom(1), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), diskusg(1M), fwtmp(1M), runacct(1M)。《程序员参考手册》中的acct(2), acct(4), utmp(4)。

ACCTCMS(1M) （基本系统）

名字

acctcms——来自每个进程的记帐记录的命令汇总

格式

/usr/lib/acct/acctcms [options] files

说明

acctcms读取一个或多个file，这些文件通常采用acct(4)所描述的格式。它为执行同一命名的命令的进程附加所有的记录，对它们进行分类，并且把它们写到标准输出上，这些记录通常采用内部汇总格式。该命令的任选项（options）为：

- -a 以ASCII形式而不是以内部汇总格式打印输出。该输出包括命令名、执行的时间、总的kcore分钟数、总的cpu分钟数、总的实际分钟数，平均存储空间大小（以K为单位）、每个援引的cpu分钟数、hog factor（因子）、字符传送以及盘区读写，如同acctcom(1)一样。输出通常通过kcore分钟数来进行分类。
- -c 通过总的cpu时间来分类，而不是通过总的Kcore分钟数来分类。
- -j 连接仅在"* * * other"下调用的所有命令。
- -n 由命令援引的数目来分类。
- -s 从此以后遇到的任何文件名都是采用内部汇总格式方式的。

-t　把所有的记录处理为总的记帐记录。缺省的内部汇总格式把每个字段划分为基本时间和非基本时间两部分。这个任选项组合这两个时间并把总和放到一个单个的字段中且提供了与旧形式（即：UNIX系统V）acctcms内部汇总格式记录的向上兼容能力。

下列任选项只能与-a任选项一起使用：

-p　输出一份仅有基本时间的命令汇总表。

-o　输出一份仅有非基本时间的命令汇总表。

当-p和-c一起使用时，则产生结合基本的和非基本时间的报告。除了执行的时间数、cpu分钟数和实际分钟数被划分为基本的和非基本的时间外，所有其它的输出汇总均使用这两个时间的总和。

执行日常命令记帐和维护运行累计的典型序列为：

```
acctcms file…>today
cp total previoustotal
acctcms -s today previoustotal>total
acctcms -a -s today
```

参见

acct(1M), acctcom(1), acctcon(1M), acctmerg(1M), acctprc(1M), acctsh(1M), fwtmp(1M), runacct(1M)。

《程序员参考手册》中的acct(2), acct(4), utmp(4)。

诊断

若-t被用于新的形式的内部汇总格式文件或用于旧的形式内部汇总格式文件上，则会产生意想不到的输出结果。

ACCTCOM(1)　（基本系统）

名字

acctcom——检索并打印进程记帐文件。

格式

acctcom〔〔options〕〔file〕〕

说明

acctcom以acct(4)描述的形式读file、标准输入或/usr/adm/pacct，并且把选择的记录写到标准输出上。每个记录表示一个进程的执行情况。其输出示出了COMMAND NAME, USER, TTYNAME, START TIME, END TIME, REAL（SEC）, CPU（SEC）, MEAN SIZE(K)以及任选的F（fork/exec标志：1=fork）, STAT（系统退出状态）, HOG FACTOR, KCORE MIN, CPU FACTOR, CHARS TRNSFD和BLOCKS READ（盘区读写总计）。

若命令是以超级用户特权方式执行的，则在该命令名前附上一个“#”符；若一个进程与一已知的终端无关，则在TTYNAME字段打印一个“?”符。

若没有指定files，并且标准输入与一终端或/dev/null文件有关，则读/usr/adm/pacct文件，反之，则读标准输入。

若给出 file 自变量，则它们被按序读入。各个文件通常被正向读入，即以进程完成时间的先后次序来读入。文件 /usr/adm/pacct 通常被确定为当前文件，除当前文件外，一个繁忙的系统可能需要多个这样的文件，它们可以在 /usr/adm/pacct？中找到。这些任选项为：

-a	示出关于选择的进程的一些常规统计资料。这些统计资料在输出记录后将被输出。
-b	反向读入，首先示出最近的命令。该任选项对于读标准输入时无效。
-f	在输出的列上打印fork/exec标志和系统退出状态。
-h	不示出平均的存储区域大小，而是示出进程在执行期间共占用的有效的cpu时间的时间部分，“hog factor”被计算为：（总的 cpu 时间）/（消逝的时间）。
-i	在输出中打印含有I/O计数的列。
-k	示出总的Kcore分钟数而不是存储区域的大小。
-m	示出平均存储区域大小（缺省值）。
-r	示出cpu因子（用户时间/（系统时间+用户时间））。
-t	分别示出系统和用户的cpu时间。
-v	从输出处排除列标题。
-l line	仅示出属于终端/dev/line的进程。
-u user	仅示出属于user的进程，该 user 可以由用户 ID 或可被转换成用户 ID 的注册名来指定。一个#符表示那些在超级用户特权级上执行的进程，而？符则指与未知的用户ID有关的那些进程。
-g group	仅示出属于 group 的进程，该 group 可通过用户组 ID 或用户组名来指定。
-s time	选择在 time 及以后存在的进程，time 以hr〔:min〔:sec〕〕这种格式给出。
-e time	选择在time及以前存在的进程。
-S time	选择在time及以后启动的进程。
-E time	选择在time及以前结束的进程。-S和-E使用相同的 time 示出在time时存在的进程。
-n pattern	仅示出与 pattern匹配的命令。除了“+”意味着一个或多个重复外，pattern可以是如ed(1)中所述的一个正则表达式。
-g	不打印任何输出记录。当它与-a 任选项连用时仅打印常规的统计资料。
-o ofile	以数据输入格式复制选择的进程记录到ofile中；抑制标准输出打印。
-H factor	仅示出超过factor值的进程，该处的 factor 与上面所述的-h 任何项的“hog factor”相同。
-O sec	仅示出cpu系统时间超过sec秒的进程。
-C sec	仅示出总的cpu时间（系统时间+用户时间）超过sec秒的进程。
-I chars	仅示出传送的字符数超过由chars规定的截断个数的进程。