

如何启动和停止一个 UNIX 系统

马 力 (西昌保险公司)

UNIX 有别于 DOS 的最基本特点是 UNIX 是一个多用户,多任务的分时操作系统。启动一个 UNIX 操作系统不仅仅是接通电源,还必须装入操作系统,检查文件系统,选择系统操作方式等步骤使系统初始化,以便执行相应操作;同时,关闭一个 UNIX 系统,也绝对不能随手断电。电源故障,硬件出错,人为干扰而非正常关闭系统,容易造成系统崩溃,甚至磁盘损坏。因而 UNIX 系统提供了一组专门工具,以供关机时生成有序的事件序列,保证系统的一致性和完整性。

UNIX 系统的启动和停止,成了系统管理员的日常重要工作。

一、启动系统

1. 启动过程

首先系统上电,运行自检程序(POST),进行内存测试和显示系统硬件情况,随后 UNIX 引导程序被 INT 19H 读入内存并获得控制权,通过 BIOS 例程读入核心文件 /UNIX 并运行核心代码,核心代码将进行核心级初始化和用户级初始化两项工作。读入主引导块,装入并执行在活动部分上的初始化引导程序后,系统将显示如下引导信息:

```
SCO UNIX SYSTEM V / 386
BOOT
:
```

这时可键入回车,默认 hd(40)unix 引导,加载操作系统。作为正常内核装入,系统将显示:

```
loading kernel hd(40)unix.text
.....
loading kernel hd(40)unix.data
.....
loading kernel hd(40)unix.bss
.....
```

随后还会显示出一个系统配置报告,该报告将操作系统识别出的硬件设备名、地址、中断向量、DMA 通讯和设备说明,以及存储和文件系统状态信息情况显示在

屏幕上。

由于 UNIX 系统对文件系统的正常性依赖极大,因而在操作系统加载完以后,将检查根文件系统是否“干净”,若文件系统受到破坏,系统提示:

```
fsstat:root file system needs checking
OK to check the root file system (ldev / root) (y / n)?
```

键入 Y,调用 fsck(ADM)对文件系统进行清理。fsck 的执行由 /etc/bcheckrc 脚本检查上次关机过程是否已写入文件系统的正常标志。若未正确设置该标志,自动执行 fsck,以便修复被破坏的文件系统。fsck 程序将进行块及其大小检查,并作路经名,连接情况,关联记数和空闲表检查。

系统进入 init 运行级,要求用户选择系统操作方式。

```
INIT:SINGLE USER MODE
type CONTROL-2 to continue with normal startup,
(or give the root password for system maintenance);
```

这时可按提示进入多用户方式或系统维护方式。进入多用户态后,要求输入当前时间,随后调用 tcbck(ADM),检查安全数据基。若文件丢失,系统将提示从后援中恢复这些文件。

至此,系统启动结束,提示:

```
Welcoma to SCO UNIX System V / 386 Release 3.2 scasysv!
login:
```

用户可登录注册。

2. 启动中的几个主要文件

(1) /BOOT 文件:BOOT 是用于装载并执行独立的 UNIX 程序的交互式程序,主要用于装载和执行 UNIX 操作系统核心。在引导过程中,/BOOT 将读 /etc/default/boot,显示引导过程信息,装入并执行内核(/UNIX)。

(2) /etc/default/boot 文件:该文件是一个文本文件,规定了系统启动程序的参数设置。下面是一个 /etc/default/boot 文件样本。

```
DEFBOOTSTR = HD(40)UNIX
AUTOBOOT = YES
FSCKFIX = YES
MULTIUSER = YES
PANICBOOT = NO
MAPKEY = YES
SERIAL8 = YES
```

其中:

DEFBOOTSR = HD(40)UNIX 使启动程序默认从硬盘上装载操作系统核心。

AUTOBOOT = YES 默认内核自动引导,默许启动时间为 60 秒,可用 TIMEOUT 任选参数设置等待时间。

FISCKFIX = YES 引导过程中,FISCK 自动清理根文件系统。

MULTIUSER = YES 只要非根文件系统清理,进入多用户态。

PANICBOOT = NO 系统崩溃时,停留在崩溃状态,等待系统管理员手工启动。

MAPKEY = YES 保证对控制台键盘实现键盘映象。

SERIRL8 = YES 使系统控制台附加在串行口上,并送 8 位信息。

(3) /etc/init: /etc/init 程序在核心初始化的最后阶段开始执行,且其进程 id(pid)为 1。该进程按一定的规则,启动 /etc/inittab 文件中所列的进程,同时引导系统进入所规定的运行级别,首先 init 读 /etc/inittab 中的 inittdefault 项,当所有进程都创建成功后,init 进入循环等待。

/etc/init 的主要作用是:打开主控台 /dev/console,检查文件系统的一致性,执行 /etc/rc 进行必要的设置,在多用户态下启动 getty 程序。

UNIX 系统能提供给用户多种不同的操作方式,通常称为 init 状态。init 状态属于一种软件配置,在这种配置下允许相应进程运行。

init 状态及功能如下:

0——断电,系统关闭,该状态用于改变硬件,移动机器。

1——单用户方式。可消除所有非控制台有关的进程,属于系统管理状态。

2——多用户方式,文件系统被安装,多用户服务也被启动。

3——采用 RFS 的多用户方式(远程文件系统)。

4——(未定义)

5——关机并再引导,运行驻留在 non-Volatile RAM 中的程序。

6——关机到 ROM 引导,然后重新启动到在 inittab

文件中的初始缺省项中定义的状态。

s——单用户方式。

S——远程控制台单用户方式。

a,b,c——虚拟运行级。

在一个运行的 UNIX 系统中,起支撑作用的是进程 0,进程 1,在引导过程中,核心通过分配一个进程数据区结构来创建进程 0,进程 0 创建并被初始化后,系统通过构造进程 0 的副本来创建进程 1。在用户级的初始化中,/etc/init 在单用户下启动 shell,而在多用户方式下,运行 /etc/brc,/etc/bchk 等外壳程序,并启动 /etc/rc。d,调用 /etc/getty,为每个终端生成一个 getty 子进程,以便在读入用户注册名后完成用户的注册过程。

在进入更高编号系统状态启动相应进程和装配文件系统,会有更多的服务可用。相反,进入更低编号的系统态,服务则相应减少,在整个系统启动和运行时,使用 init 命令,可使系统在运行级之间进行自由切换。

在 UNIX 系统中,运行级分别对应于一个文件 /etc/rcn 和目录 //etc/rcn.d/*。(n:0-6)。例如由其它运行级切换到运行级 2(init 2),该命令启动 /etc/rc2 和 /etc/rc2.d/* 下的相应进程,rc2 设置环境参数,将 SAF 初始化,并运行 /etc/rc2.d 中的程序 ./etc/rc2.d 目录下的进程名称须以大写字母 S 或 K 开头,而后是两位指明它相对于目录中其它进程的执行顺序的数字,最后再加上说明这个进程功能的名字作结尾。我们可以按照这些脚本的顺序和目录的内容跟踪 UNIX 系统切换状态时经历的完整操作序列。

(4) /etc/inittab:该文件也是一个文本文件。/etc/inittab 列出了在系统中每次从一态转入另一态时必需要做的任务。通常在引导系统时,init 启动的进程完成、收到掉电信号以及运行 /etc/init 或 /etc/telinit 时,init 读取 inittab 文件。init 从 /etc/inittab 中取得指示,其内容控制了所有的 init 状态,控制了已消亡的进程再生。

二、关闭系统

由于 UNIX 是一个复杂的系统,在正常运行下,有很多临时动态的数据存在,如果直接关机将造成数据丢失,进而造成系统的不一致性和不完整性。因而停止一个

UNIX 系统运行的工作并不仅仅是关掉电源。关机过程的最后一步才是物理上的关机。

在关闭系统前,必须将系统缓冲区清理干净,关掉所有打开的文件,仔细撤消系统和用户进程,更新系统文件和日志,拆卸掉所有非根文件系统。可以采用 /etc/shutdown 脚本或使用 haltsys 以及 reboot,init 命令来关闭一个系统。

shutdown 要求使用者为超级用户,且必须在根(/)目录下,shutdown 是一个用 shell 编写的程序,驻留在 /etc 目录下。该命令执行时首先发出警告,通知用户离开系统,进而开始取消进程,卸去文件系统并把系统转为单用户状态,关闭所有的 gettys,不再允许用户登录,卸掉硬盘,修改超级块并停止处理器,整个系统功能下降。在通知用户系统将要关机后,用 who 检查系统中不再有用户时,即可键入 shutdown。屏幕显示:

```
THE SYSTEM IS BEING SHUT DOWN NOW!!
```

```
Log off now or risk your files being damaged.
```

此时用户应关闭全部文件,保护会话过程,作注销工作。出现:

```
** Safe to Power off **
```

```
. or .
```

```
** Press Any Key to Reboot **
```

shutdown 命令使用格式:

```
shutdown[-y -g <time> -i[0156SS]-f"message"su]
```

其中,-y 选项表示关闭时,对进一步认可无需提示;-g<time>表示从发出 shutdown 命令起到停止处理器所用时间;-i 指明了 init 级;-f"message"可送一信息给用户;选项 su 表示是否进入单用户,取值范围是 yes 或 no,默认为 no。

作为 haltsys(ADM)命令属于立即关闭系统命令,它比 shutdown 命令来得更快,如果一个用户正在编辑一篇文章还没有保存,haltsys 命令并不能帮助用户将文章保存起来,因而通常该命令用于紧急情况或维护态下使用。

reboot 和 haltsys 程序的唯一区别是它在正常停机之后不用按任意键,马上自动引导,重新启动系统,相应于 init 6,仅在紧急情况和单用户态下使用。reboot 程序实质上是和 haltsys 程序链接在一起。

参考文献:

[1]卢显良主编,《UNIX 系统管理》,清华大学出版社,1993。

社,1993。

[2][美]Stephen Coffin 著 戴建鹏等译,《UNIX 使用大全》,电子工业出版社,1991。

利用 CMOS 的含义 实现微机的解锁

曹国钧 (国家医药管理局重庆设计院)

286 及其以上微机的正常启动都要靠 CMOS 电路的正常设置,该 CMOS RAM 中有 64 个字节(00-3FH)RAM 存放着实时时钟与系统配置(如:软盘,硬盘类型参数、内存配置和口令字等),一旦在 SETUP 配置过程中不小心设置了口令,在下次启动机器时就无法进入 SETUP 或启动微机系统了。

在微机的 CMOS 参数设置中,有一项叫 Password Checking Option(密码检查项),它在 AMIBIOS 配置的微机中有三种选择项(在 AST 或 COMPAQ 微机中仅有 Always 一项):

- (1) Disable:微机无密码设置;
- (2) Setup:进入 SETUP 程序需要密码;
- (3) Always:微机正常启动和进入 SETUP 程序均需要密码的支持。

下面讨论第二和第三种情况的解锁。

对于第二种情况,即仅是在不知道密码的情况下无法进入 SETUP 程序,但可正常启动微机系统,这种情况可直接用软件的方法进行解决。

CMOS RAM 在微机系统 I/O 的端口地址为 70H 和 71H,其中 70H 为地址索引端口,71H 为数据端口,利用这两个口地址,可以将微机 CMOS 中的内容读出来,下面就是一个读 CMOS 内容的汇编程序((*)行改为 OUT AL,71H,则为写 CMOS 的程序),CMOS 内容放在 DS:200H 处,并将 CMOS 信息存在 CMOS 文件中,其重要参数的意义如下:

```
-U 100 11F
0E0A:0100 MOV BX,0200
0E0A:0103 MOV DX,0000
```