

第五章

系统的启动和关闭

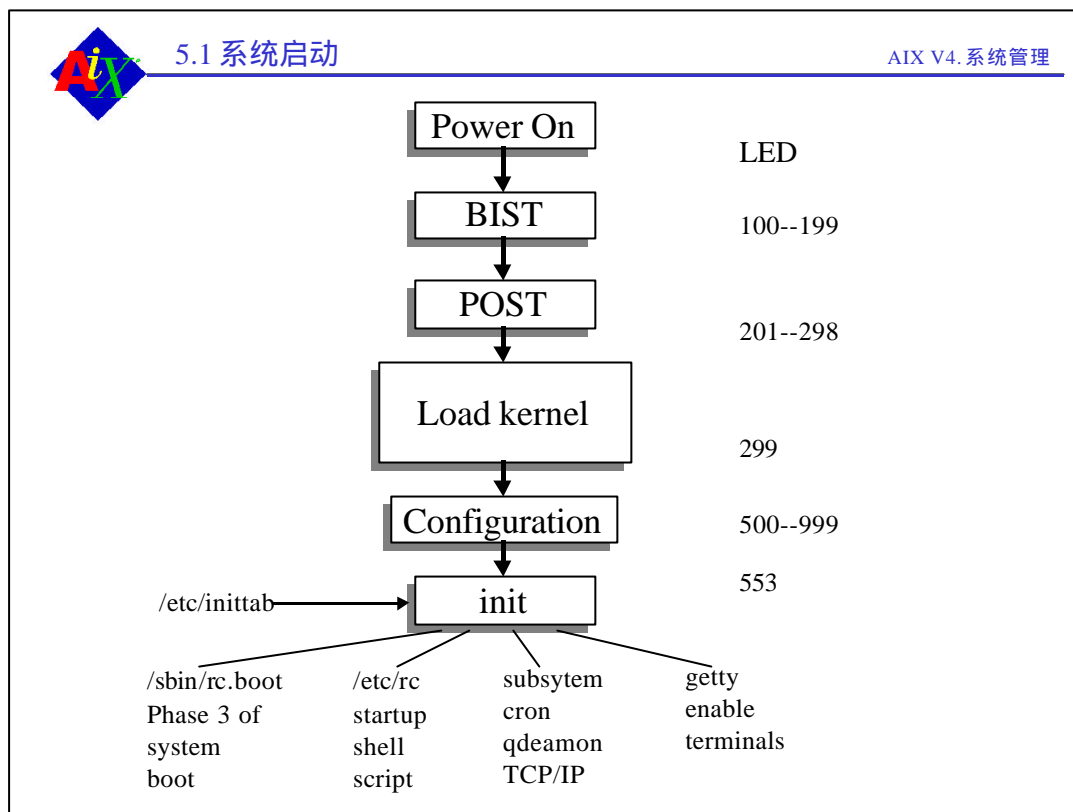


AIX V4. 系统管理

本章讲述RSI C/6000系统的启动，关闭和系统环境的管理。

学完本章，应该掌握的要点是：

- 了解系统的启动和关闭的方法
- 掌握系统开启和关闭的方法
- 了解/etc/inittab文件的功能及内容
- 掌握管理系统环境的方法



系统启动问题是我们可遇到的比较普遍的问题之一。从系统加电到进入多用户模式，出现等待登录界面，要依次进入下面如下几个阶段。

阶段1: 只读储存器的引导程序导入 (ROS IPL)

这个阶段包括上电自检 (post)，查找引导设备，并且把引导核心装载进内核。初始化硬件检测是要校验主要硬件是可用的，在微通道 (MAC) 系统中分为两个部分，首先是built-in self test (BIST) 部分，其次是power-on self test (POST) 部分。但是在PCI系统中这部分仅仅由POST控制。

在这之后，系统在RAM文件系统 (RAMFS) 中装载启动逻辑卷 (BLV) 并传递控制权给BLV。由于rootvg在这时是不可用的，所以BLV中有启动所需的所有信息，其中包括AIX核心，然后核心执行部分初始化工作，包括硬件和虚存的初始化。进程0转化为调度进程 (scheduler)，启动进程1，init。此时的init不是运行时的init，是RAM fs中的ssh的别名。稍后，该进程被运行时init进程所取代。Ssh (simple shell, 简易shell) 不使用共享库，包含引导系统的硬码序列。Ssh两次执行rc.boot程序，在阶段1和阶段2各一次，系统进入下个阶段。



5.1.1 PCI RISC/6000 系统启动模式

AIX V4. 系统管理

Normal Mode

- 程序和进程运行
- 终端可用
- 所有的文件可以被访问
- 通讯已启动
- 多用户方式

Standalone Mode

- 系统管理服务软盘
- 系统程序

查看和改变系统设置

查看和改变系统boot列表顺序

执行硬件诊断

执行系统功能

有两种启动方式，Normal方式和Standalone方式。Standalone方式调用系统管理服务软盘，它包含系统程序，使你可以看到有关系统的信息和执行的任务，例如运行的测试、设置口令，以及更新系统程序，这些程序在系统管理服务软盘或CD上。



5.1.2 调用单用户模式

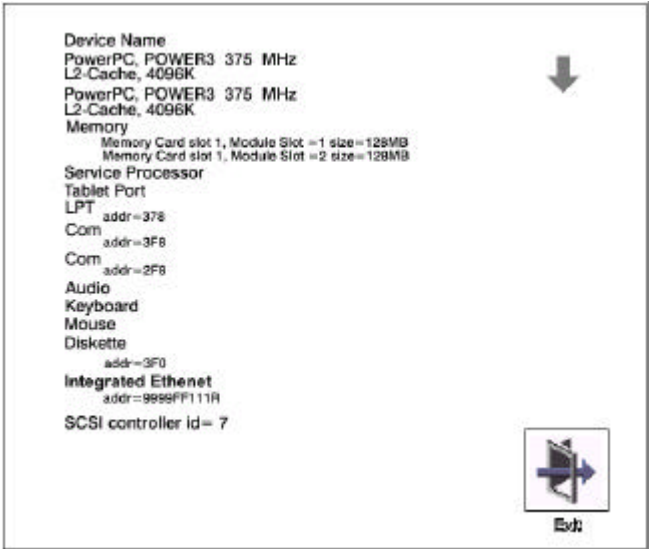
AIX V4. 系统管理

调用单用户模式：

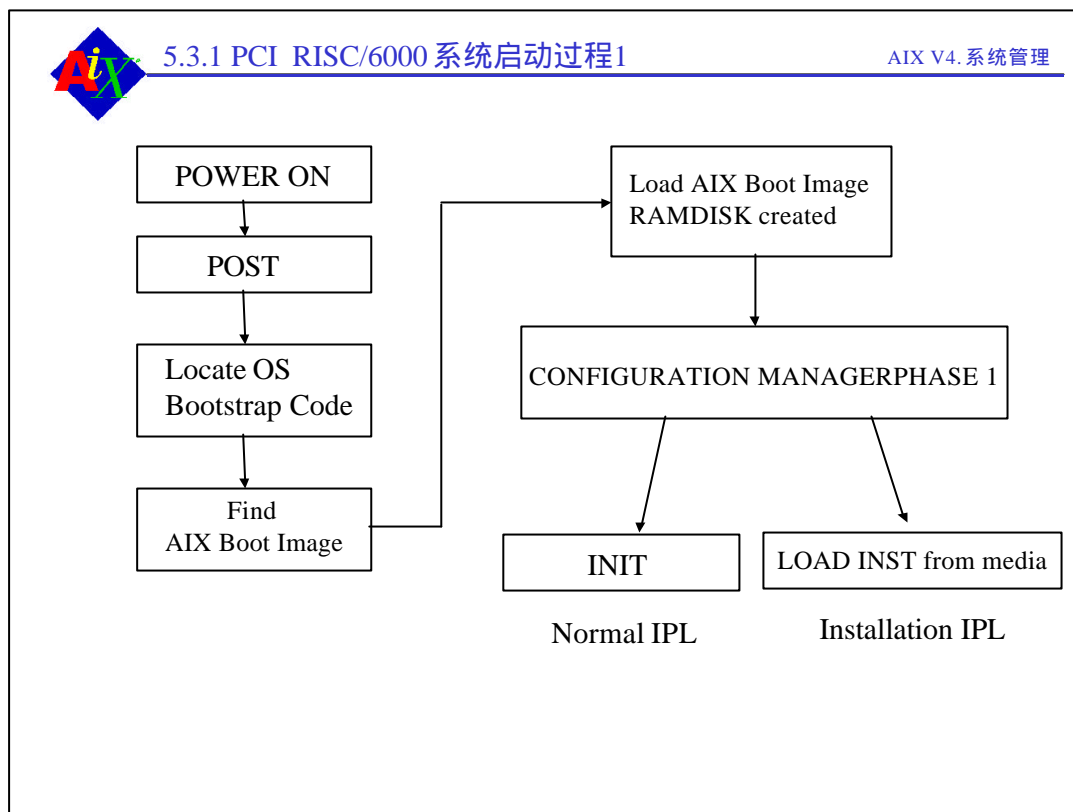
1. 插入系统软盘或CD到设备中
2. 加电或重新启动机器
3. 当屏幕第一个图标出现后，在最后一个图标出现之前按F5或5键



Config







Init进程执行rc.boot程序，rc.boot的阶段1执行基本设备的配置，包括以下步骤：

调用restbase程序为基本设备（即配置rootvg所必需的设备）在RAM文件系统中建立定制ODM(Object Database Manager对象数据管理器)设备库，建立库时要依据存储在引导映像末端的定制数据。

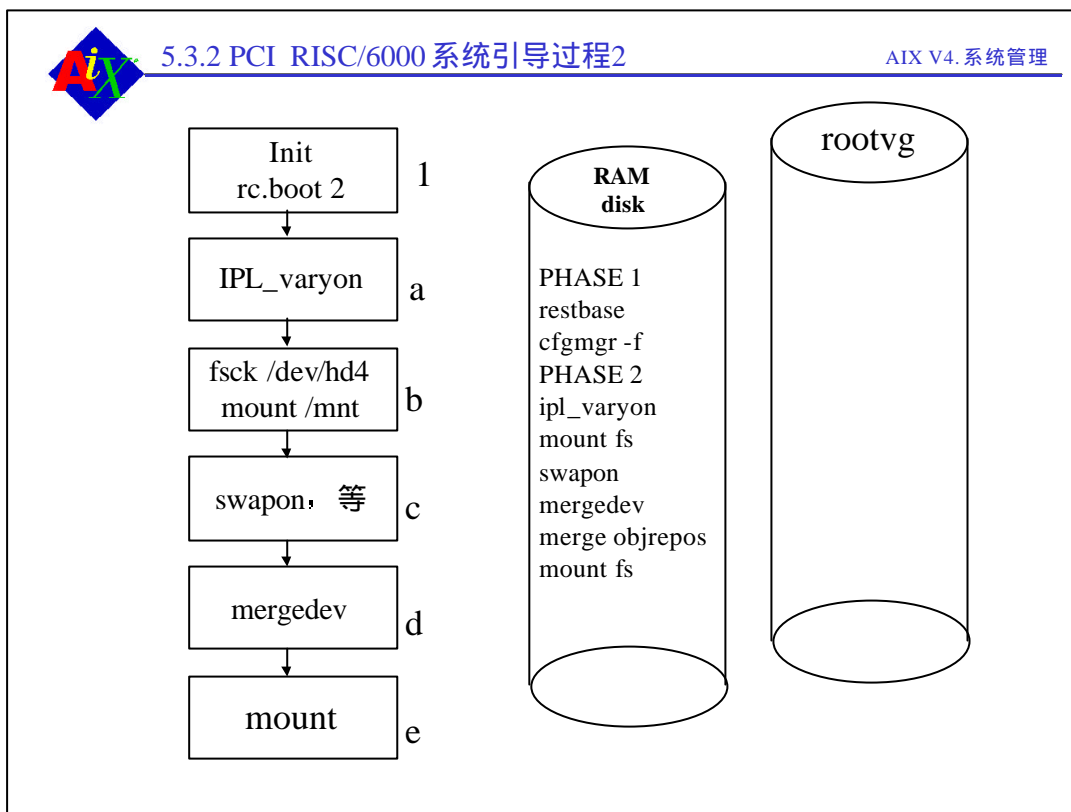
调用配置管理程序cfgmgr，由配置管理程序依据阶段1的配置规则（rule），来配置基本设备。配置管理程序是一个规则驱动的程序，能够在IPL阶段自动配置设备，或者在系统运行过程中按Config_Rules中的规则配置设备。

配置管理程序启动CPU板（sys），总线（bus），SCSI和逻辑卷管理程序（LVM）和根卷组（RVG）的配置方法程序（Configuration Method），在这一阶段，要配置系统总线和总线上连接的适配器。在预定义的配置数据库中所有的适配器都将配置到系统中。

各配置方法程序装入相应设备的驱动程序，创建特别文件，并在ODM数据库中建立定制数据。

在阶段1的最后，用bootinfo命令调用启动设备。

之后，LVM已经运行，系统能够访问磁盘。

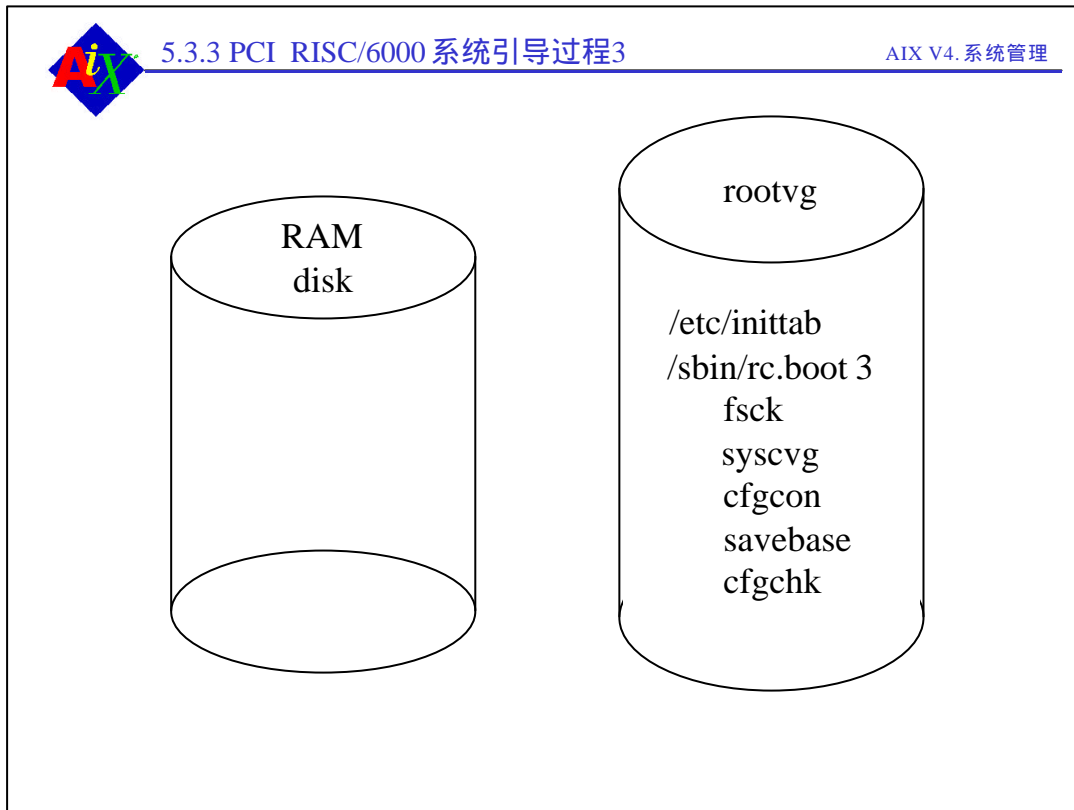


1. init(ssh)进程激活控制台的初始化，然后调用rc.boot 2:

- a. 调用ipl_varyon程序激活〔vary on〕根卷组rootvg。这里的卷组是一到多个磁盘组成的一个单位。
- b. 将硬盘上的文件系统临时安装到RAM文件系统〔RAMfs或RAMdisk〕的子目录上，执行fsck来检查根文件系统/dev/hd4，并且对根卷组jfs日志所记录的未完成事务作相应的处理。
- c. 运行swapon来启动换页进程，换页进程启动后，也即建立了虚存管理系统。
- d. 把LVM信息写入磁盘。调用mergedev，确保已配置设备的主/次设备号与磁盘上/dev目录下的特别文件相符，同时将RAM文件系统里的ODM定制数据写入到磁盘上的ODM中。
- e. 卸除临时安装的文件系统，将/dev/hd4安装到/，安装/usr,/var等。

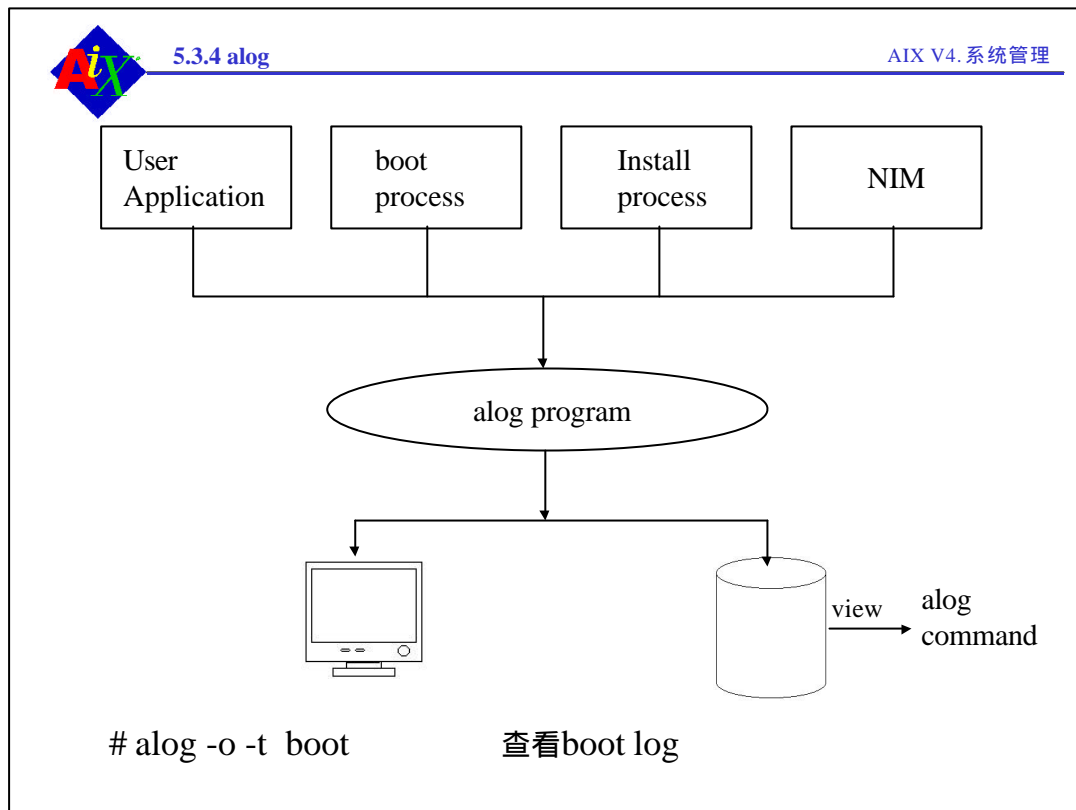
2. 执行newroot命令，从RAM文件系统切换到硬盘的根文件系统。在RAMdisk的init.程序在阶段2退出时，核心释放RAMdisk以及初始化阶段所占用的内存。

3. 运行时init进程处理/etc/inittab文件中的表项。



此时，RAMdisk已经释放，/etc/inittab文件启动了rc.boot程序中的阶段3:

- 如果安装了nfs，则启动portmapper和rpc.statd
- 安装/tmp文件系统
- 运行syscvlg使rootvg的镜像备份数据同步
- crgmgr-s配置所有连接的设备，定制所有用户指定的设备。此时，系统中的ODM库全部更新完毕。
- cfgcon配置系统控制台。
- savebase将基本的定制数据写回引导映象中，引导映象则被写回逻辑卷中。
- cleantty确保使控制台是唯一的终端设备
- 启动sync进程
- 启动error进程
- 运行cfgchk检查已知的配置冲突，检查缺失的设备
- init进程继续处理inittab文件中的内容
- 显示登录提示
- 引导过程结束



BOS提供了一个日志管理程序——alog 命令，应用程序和用户可以利用该命令来管理日志。alog命令读取标准输入，将输出写到标准输出上，同时也写到一个固定大小的文件上。该文件被用作循环日志文件，这意味着如果文件写满，新的记录将覆盖最旧的记录。

alog程序处理的日志文件可以从命令行指定，也可以在ODM库中维护的alog配置数据库中定义。系统支持的日志类型有boot, bosinst和nim。

用户在开机之后有时会离开控制台或忙其他事情，无暇留意控制台上的提示信息，在某些情况下，特别是系统不能正常引导时，这些信息却十分有用，可能正是解决问题的着手之处。

因此，系统的rc.boot程序和配置管理程序利用alog程序，在引导的过程中记录了许多重要的事情和信息。要查看引导信息可以用命令alog -o -t boot。如果系统不能引导，那么可从光盘引导进入维护方式，进一步查看系统引导日志的内容。



```

Window Edit Options Help
init:2:initdefault:
brc::sysinit:/sbin/rc.boot 3 >/dev/console 2>&1 # Phase 3 of system boot
powerfail::powerfail:/etc/rc.powerfail 2>&1 | alog -tboot > /dev/console #
Power Failure Detection
rc:2:wait:/etc/rc 2>&1 | alog -tboot > /dev/console # Multi-User checks
fbcheck:2:wait:/usr/sbin/fbcheck 2>&1 | alog -tboot > /dev/console # run /
etc/firstboot
srcmstr:2:respawn:/usr/sbin/srcmstr # System Resource Controller
rctcpip:2:wait:/etc/rc.tcpip > /dev/console 2>&1 # Start TCP/IP daemons
rcnfs:2:wait:/etc/rc.nfs > /dev/console 2>&1 # Start NFS Daemons
cron:2:respawn:/usr/sbin/cron
piobe:2:wait:/usr/lib/lpd/pio/etc/piobinit >/dev/null 2>&1 # pb cleanup
qdaemon:2:wait:/usr/bin/startsrc -sqdaemon
writesrv:2:wait:/usr/bin/startsrc -suritesrv
uprintfd:2:respawn:/usr/sbin/uprintfd
logsymp:2:once:/usr/lib/ras/logsymp # for system dumps
pmd:2:wait:/usr/bin/pmd > /dev/console 2>&1 # Start PM daemon
httpdlite:2:once:/usr/IMNSearch/httpdlite/httpdlite -r /etc/IMNSearch/http
dlite/httpdlite.conf & >/dev/console 2>&1
diagd:2:once:/usr/lpp/diagnostics/bin/diagd >/dev/console 2>&1
dt:2:wait:/etc/rc.dt
cons:0123456789:respawn:/usr/sbin/getty /dev/console
inittab (83%)

```

文件格式:

id:runlevel:action:command

从读/etc/inittab 文件使用命令

telinit q

init进程可以更改当前系统的运行级别（runlevel）。一般而言，在更改运行级时，init进程将会终止当前运行级的进程，并启动新运行级指定的进程。对运行的约定如下：

0--1	保留未用
2	多用户环境
3--9	可以根据用户需要定义
S,s,M,m	维护模式
a,b,c	当运行级别修改到a, b, c时，init进程不终止当前运行级的进程，而只启动新运行级指定的进程。

- **id:** 不多于14个字符，表示进程。终端使用它们的逻辑设备名作标示。
- **runlevel:** 不多于20个字符，缺省的运行进程师有效的。如果init改变了运行级别，一个SIGTERM信号将发送给所有的进程并且不定义新的运行级别。如果20秒内一个进程没有被禁止，将有一个SIGKILL信号发出。系统缺省的运行级别有两种：单用户方式和多用户方式。
- **action:** 处理进程怎样运行。
 - Respawn: 如果进程不存在，启动它。
 - wait: 启动进程，并且等待它结束。
 - once: 启动进程，如果进程停止，不再启动它。
 - sysinit: 在命令被运行前试图访问控制器。
- **command:** 命令行启动进程。

使用mkitab和chitab命令来改变/etc/inittab文件，而不是直接编辑它。



5.4 系统资源控制器

AIX V4. 系统管理

系统资源控制器SRC (System Resource Controller) 通过提供统一的管理接口, 使得管理员管理和控制子系统的工作大大简化。

- 通过统一的接口来启动, 停止和显示子系统的状态
- 记录子系统的非正常终止
- 可对子系统, 子系统组和子服务器程序进行跟踪
- 支持远程的控制操作
- 提供刷新子系统的功能

子系统 (subsystem) 是执行相关功能的一个或一组相关程序的集合。子系统组 (subsystem group) 是特定子系统的集合。将子系统合并成组可以让用户同时控制几个子系统。

子服务程序 (subserver) 是属于某个子系统的一个进程或daemon进程, 并且通过该子系统来控制。



- 启动子系统
startsrc -s qdaemon
- 停止之系统
stopsrc -s qdaemon
- 刷新子系统
refresh -s qdaemon
- 显示子系统状态
lssrc -a

停止子系统或子系统组，stopsrc命令可以用如下的选项：

- f 强制子系统尽快终止，释放所有资源，不等待任何应用程序的动作完成。
- c 取消停止动作，使得子系统资源稍后再释放。

上述两种选项均未用时，则开始一个正常的停止动作，要求子系统在所有应用程序的动作完成后停止当前的所有进程，并释放资源。子系统不能再接受新的请求了。

- s 用于对单个子系统进程启动，停止和列示状态的操作。
- g 用于对属于指定组的子系统进程启动，停止和列示状态的操作。

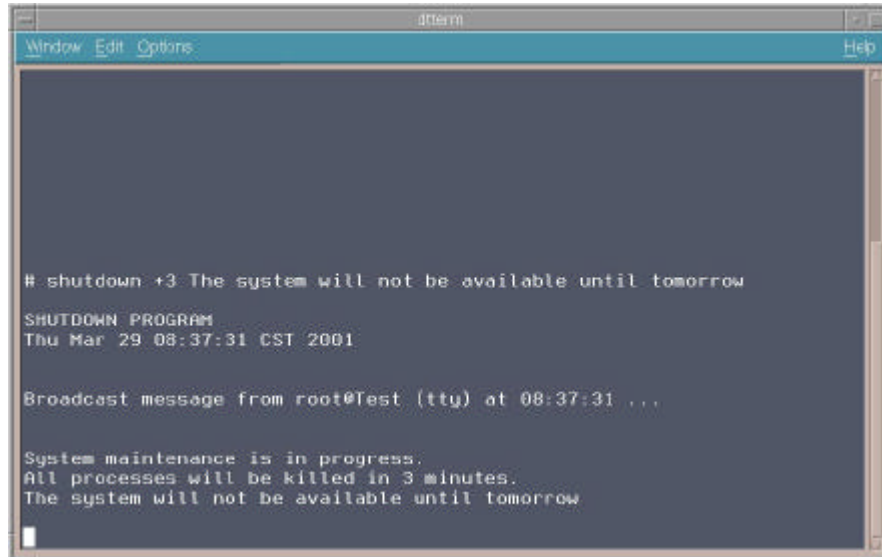


5.5 系统关闭

AIX V4. 系统管理

系统关闭命令:

```
# shutdown [-options] [+time message]
```



```
Window Edit Options Help

# shutdown +3 The system will not be available until tomorrow

SHUTDOWN PROGRAM
Thu Mar 29 08:37:31 CST 2001

Broadcast message from root@Test (tty) at 08:37:31 ...

System maintenance is in progress.
All processes will be killed in 3 minutes.
The system will not be available until tomorrow
```

如果使用时未加任何选项，系统将在所有活动的终端上显示一条消息（利用wall命令），一分钟后停止所有的活动的终端，终止系统的所有进程，同步硬盘数据，拆卸所有文件系统，然后终止进程。

-F选项可以快速关机（不显示广播的警告信息），-r选项指定系统停止后重新引导，-m选项使系统停止并进入维护方式。-k选项只显示关机信息，并不真正关闭系统。

A screenshot of an AIX terminal window titled 'csh@root'. The window has a menu bar with 'Window', 'Edit', 'Options', and 'Help'. The terminal displays the output of the 'shutdown' command. The text shown is: '# shutdown +3 The system will not be available until tomorrow', 'SHUTDOWN PROGRAM', 'Thu Mar 29 08:37:31 CST 2001', 'Broadcast message from root@Test (tty) at 08:37:31 ...', 'System maintenance is in progress.', 'All processes will be killed in 3 minutes.', 'The system will not be available until tomorrow', 'Broadcast message from root@Test (tty) at 08:39:31 ...', 'shutdown: PLEASE LOG OFF NOW !!!', 'All processes will be killed in 1 minute.'.

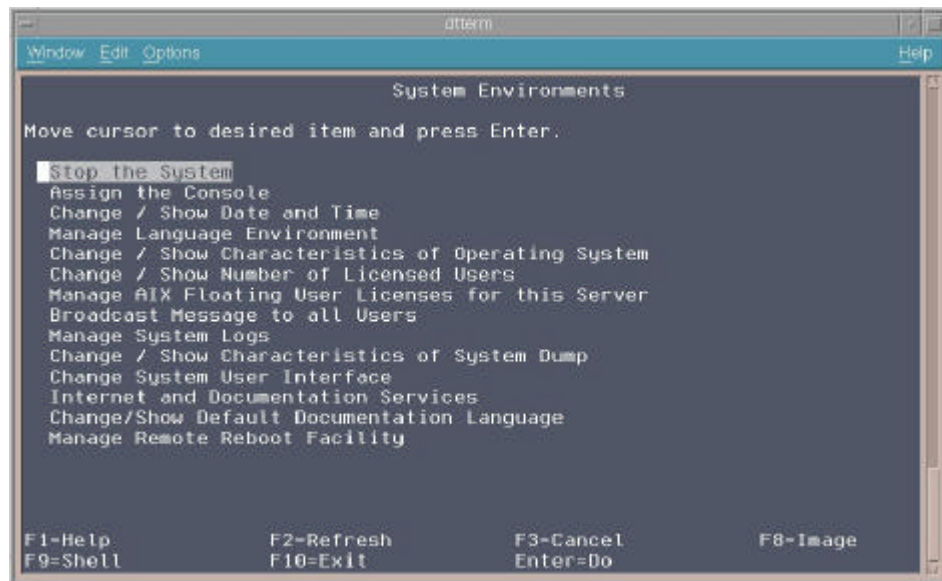
```
# shutdown +3 The system will not be available until tomorrow
SHUTDOWN PROGRAM
Thu Mar 29 08:37:31 CST 2001

Broadcast message from root@Test (tty) at 08:37:31 ...

System maintenance is in progress.
All processes will be killed in 3 minutes.
The system will not be available until tomorrow

Broadcast message from root@Test (tty) at 08:39:31 ...

shutdown: PLEASE LOG OFF NOW !!!
All processes will be killed in 1 minute.
```



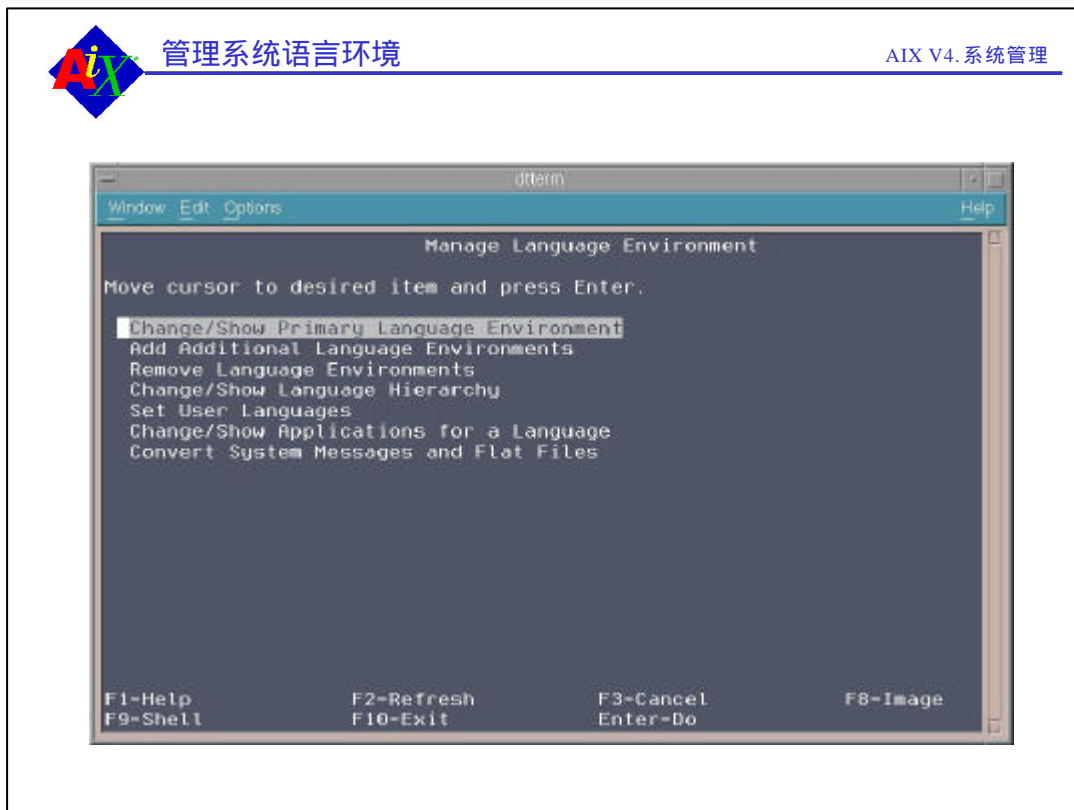
用命令data或SMIT菜单来修改系统的日期和时间，只有root有权修改系统的日期。

```
# data [mmddHHMM[.ssyy]]
```

可以用chtz命令或通过SMIT来修改系统的当前时区，实际上是修改了/etc/enviroment文件中的TZ变量。系统时钟按照CUT（GMT）工作。date和其他命令利用TZ变量将CUT时间转化成本地时间。

在/etc/enviroment文件中设置新的时区，可以用命令：

```
# chtz GMT0BST
```

在系统安装的过程中，根据用户指定的语种，系统安装了缺省的语言环境，并在/etc/enviroment设置了环境变量LANG的值。

语言环境的设置将会影响到操作方式，字符分类，大小写转换，数字和货币格式，日期和时间格式等等。

许多组合式地方语言是基于多个代码集的，在修改LANG环境时要确信所选择的语言代码集能够满足用户的需要，并且符合键盘和字型的选择。

修改系统的本地语言，可以用 chlang 命令。例如，chlang C 用于POSIX代码集，chlang zh_CN用于简体中文的EUC代码集。该命令将会修改/etc/enviroment 文件中LANG环境变量的缺省设置。用户必须退出系统然后再次登录，所修改的语言环境才会生效。

从一种代码集的ASCII文本文件或消息向另一代码集转换，要用到iconv命令，也可以通过SMIT来实现。



检查题：

1. 系统中的l进程是什么进程？该进程由那些文件来初始化的？
2. 用户是否只能从控制台执行shutdown命令？

