

第 6 章 进程管理

6.1 显示/修改进程属性

进程的属性主要包括进程的 ID 号、父进程的 ID 号、运行时间、执行的程序名和优先级等。

6.1.1 显示进程属性

要显示进程的属性可以使用 `ps` 命令，AIX 支持所有 AT&T 和 BSD 风格 `ps` 命令。要使用 BSD 风格，只是简单地把命令选项的减号去掉，例如：

```
# ps alx
上面命令的 AT&T 版本是：
# ps -elf
```

在 `ps` 命令输出中，PRI 和 NI 值表示进程的优先级。`nice` 是计算 `priorities` 值的一部分，`priorities` 值的范围是 0 ~ 127。值越低，进程越经常被安排执行，值越高意味着优先级越低。

6.1.2 修改进程优先级

要改变进程的优先级，可以使用 `nice` 命令。`nice` 命令后面可以跟 0 ~ 39，同样也是值越高意味着优先级越低。`nice` 命令句法有两种形式：`nice -Increment`（增量）和 `nice -n Increment`（增量）。当必须使用负值时，可以用后者。如果不指定一个值，`nice` 缺省为增长 10。下列命令为一个命令增加 10 的优先级。

```
# nice -10 CommandName(命令名)
```

下列命令为一个命令减少 10 的优先级

```
# nice CommandName
```

`renice` 命令也提供 `-n` 选项。`renice` 的句法为：

```
# renice Priority -p PID
```

如果没有另外的选项被指定，那么 `-p` 可以不要。

6.2 停止进程

AIX 使用 `kill` 命令来停止进程。AIX 有两种 `kill` 命令：`/usr/bin/kill` 和 `shell` 内建的命令 `kill`。这两种命令可以使用的信号量不同。信号量用来通知进程执行相应的动作。例如：`TERM` 信号是强制进程退出，而 `HUP` 是强制进程挂起。下面的命令显示了 `kill` 命令可以用的信号量。

```
# /usr/bin/kill -l
```

它可以使用如下的信号量:

NULL	HUP	INT	QUIT	ILL	TRAP	
IOT	EMT	FPE	KILL	BUS	SEGV	
SYS	PIPE	ALRM	TERM	URG	STOP	
TSTP	CONT	CHLD	TTIN	TTOU	IO	
XCPU	XFSZ	MSG	WINCH	PWR	USR1	
USR2	PROF	DANGER	VTALRM	MIGRATE	PRE	
GRANTR	ETRACT	SOUND	SAK			

```
# kill -l
```

它可以使用如下的信号量:

HUP	ALRM	MSG	SAK	INT	TERM	
WINCH	QUIT	URG	PWR	ILL	STOP	
USR1	TRAP	TSTP	USR2	ABRT	CONT	
PROF	EMT	CHLD	DANGER	FPE	TTIN	
VTALRM	KILL	TTOU	MIGRATE	BUS	IO	
PRE	SEGV	XCPU	GRANT	SYS	XFSZ	
RETRACT	PIPE	SOUND				

AIX 也有 killall 命令, 任何用户都能用来运行杀死除了发送的进程外所有进程。句法是:

```
# killall -Signal
```

6.3 性能检测

管理进程的目的在于提高系统运行的性能, 因此首先我们要检测目前系统运行性能, 分析系统性能的瓶颈。AIX 支持 top, sar, vmstat 和 iostat 等标准的 UNIX 命令。

top

采用动态刷新显示系统性能信息, 包括: 系统的顶级进程, 系统状态 (负担的平均值), 使用和释放的存储器数量, 以及在系统上激活的单个的进程状态。与 ps 命令相比, ps 命令仅给出系统某一时刻的“快照”, 而 top 是按一定的时间间隔显示系统信息和它们的变化。在多处理系统上, top 可以显示在每 CPU 的状态, 不过 AIX 上没有这功能。

sar

显示累积的系统活动信息, 包括 CPU 利用率, 缓冲区活动, 数据从设备读出和读入, 终端活动, 正在使用的特定系统调用数目, 交换数量, 队列长度, 还有另外的核心信息。

vmstat

量化地显示系统进程所使用的虚存数量; 也显示陷阱和 CPU 活动信息。

iostat

统计硬盘，终端，处理器的 I/O 情况。

6.4 系统资源控制器 SRC

AIX 有一个独立的管理守候进程的方法：系统资源控制器（System Resource Controller，SRC）。SRC 也是一个守候进程，叫 srcmstr，它由 init 在 /etc/inittab 启动。srcmstr 管理启动、停止、或刷新一个守候进程或一组守候进程的请求。使用 SRC 命令取代通过键入程序名字来启动守候进程，或取代使用 kill 命令停止一个守候进程。这样不必记住诸如：当启动一个守候进程时是否使用 "&" 符号，或当杀死一个守候进程时，应用什么信号量。SRC 也允许通过一个命令停止或者启动一组相关联系的守候进程。

AIX 按一定的层次组织守候进程，并且这个组织结构被设置为对象类 SRCsubsys 和 SRCsubsvr，存在 ODM 中。最底层的守候进程是 subservers。在一个刚被安装的系统上唯一的 subservers 是 inetd 的子系统：如 ftp, telnet, login, finger 等等。查看这些 subservers，可以使用 odmget 命令：

```
# odmget SRCsubsvr
```

```
SRCsubsvr:
```

```
sub_type = "ftp"
subsysname = "inetd"
sub_code = 21
```

```
SRCsubsvr:
```

```
sub_type = "uucp"
subsysname = "inetd"
sub_code = 540
```

```
SRCsubsvr:
```

```
sub_type = "telnet"
subsysname = "inetd"
sub_code = 23
```

```
SRCsubsvr:
```

```
sub_type = "shell"
subsysname = "inetd"
sub_code = 514
```

```
SRCsubsvr:
```

```
sub_type = "login"
subsysname = "inetd"
sub_code = 513
```

SRCsubsvr:
 sub_type = "exec"
 subsysname = "inetd"
 sub_code = 512

SRCsubsvr:
 sub_type = "finger"
 subsysname = "inetd"
 sub_code = 79

SRCsubsvr:
 sub_type = "tftp"
 subsysname = "inetd"
 sub_code = 69

SRCsubsvr:
 sub_type = "ntalk"
 subsysname = "inetd"
 sub_code = 517

SRCsubsvr:
 sub_type = "echo"
 subsysname = "inetd"
 sub_code = 7

SRCsubsvr:
 sub_type = "discard"
 subsysname = "inetd"
 sub_code = 9

SRCsubsvr:
 sub_type = "chargen"
 subsysname = "inetd"
 sub_code = 19

SRCsubsvr:
 sub_type = "daytime"
 subsysname = "inetd"
 sub_code = 13

SRCsubsvr:
 sub_type = "time"
 subsysname = "inetd"
 sub_code = 37

SRCsubsvr:

```
sub_type = "comsat"
subsysname = "inetd"
sub_code = 1512
```

SRCsubsvr:

```
sub_type = "bootps"
subsysname = "inetd"
sub_code = 67
```

SRCsubsvr:

```
sub_type = "systat"
subsysname = "inetd"
sub_code = 11
```

SRCsubsvr:

```
sub_type = "netstat"
subsysname = "inetd"
sub_code = 15
```

再上一层是 subsystem。从上面的命令显示,我们可以看到在每个 subserver 节都列出 inetd subsystem。要看所有的子系统的,可以使用 odmget SRCsubsys 命令:

```
# odmget SRCsubsys
```

SRCsubsys:

```
subsysname = "qdaemon"
synonym = ""
cmdargs = ""
path = "/usr/sbin/qdaemon"
uid = 0
auditid = 0
standin = "/dev/console"
standout = "/dev/console"
stderr = "/dev/console"
action = 1
multi = 0
contact = 2
svrkey = 0
svrmttype = 0
priority = 20
signorm = 30
sigforce = 15
display = 1
```

```
waittime = 20
grpname = "spooler"
```

SRCsubsys:

```
subsysname = "writesrv"
synonym = ""
cmdargs = ""
path = "/usr/sbin/writesrv"
uid = 0
auditid = 0
stdin = "/dev/console"
stdout = "/dev/console"
stderr = "/dev/console"
action = 1
multi = 0
contact = 2
svrkey = 0
svrmtpe = 0
priority = 20
signorm = 30
sigforce = 31
display = 1
waittime = 20
grpname = "spooler"
```

SRCsubsys:

```
subsysname = "lpd"
synonym = ""
cmdargs = ""
path = "/usr/sbin/lpd"
uid = 0
auditid = 0
stdin = "/dev/console"
stdout = "/dev/console"
stderr = "/dev/console"
action = 1
multi = 0
contact = 3
svrkey = 0
svrmtpe = 0
priority = 20
signorm = 0
sigforce = 0
display = 1
```

```
waittime = 20
grpname = "spooler"
```

SRCsubsys:

```
subsysname = "inetd"
synonym = ""
cmdargs = ""
path = "/usr/sbin/inetd"
uid = 0
auditid = 0
stdin = "/dev/console"
stdout = "/dev/console"
stderr = "/dev/console"
action = 2
multi = 0
contact = 3
svrkey = 0
svrmtpe = 0
priority = 20
signorm = 0
sigforce = 0
display = 1
waittime = 20
grpname = "tcpip"
```

相关的 subsystem 组成 subsystem group, 这是 SRC 的最高一层。subsystem group 在上面命令显示的 grpname 项中描述。这样在上面的输出显示中, lpd subsystem 是 spooler subsystem group 的一部分, 同样 inetd subsystem 是 tcpip subsystem group 的一部分。要看所有的 subsystem 和 subsystem group 更容易的方法是使用 lssrc -a 命令:

```
# lssrc -a
```

Subsystem	Group	PID	Status
syslogd	ras	5708	active
sendmail	mail	6196	active
portmap	portmap	6452	active
inetd	tcpip	6708	active
named	tcpip	7224	active
tftpd	tcpip	7482	active
snmpd	tcpip	7740	active
dpid2	tcpip	7998	active
nimesis	nim	8514	active
biod	nfs	9808	active
nfsd	nfs	10070	active

rpc.mountd	nfs	10328	active
rpc.statd	nfs	9062	active
rpc.lockd	nfs	10846	active
qdaemon	spooler	11356	active
writesrv	spooler	11618	active
pppcontrold	uucp	12388	active
llbd	ifornes	11874	active
glbd	ifornes	15226	active
i4llmd	iforls	15738	active
i4glbcd	ifornes	6002	active

也能使用下列命令在远程主机上列出所有的 subsystem 和 subsystem group:

```
# lssrc -h remote_host -a
```

最经常被使用的 SRC 命令是 startsrc、stopsrc 和 refresh, 每个命令都有如下的选项:

- s 把命令用于一个 subsystem, 通过 lssrc -a 命令列出使用的 subsystem 名字。
- g 把命令用于一个 subsystem group, 通过 lssrc -a 命令列出使用的 subsystem group 名字。

这些命令的名字即说明它们的作用: 启动一个 subserver、subsystem 或 subsystem group。

使用 startsrc 命令。

例如, 启动 rpc.mountd subsystem(它实际上是 rpc.mountd 守候进程)键入 :

```
# startsrc -s rpc.mountd
```

启动 nfs subsystem group:

```
# startsrc -g nfs
```

这个命令启动所有组成 nfs subsystem group 的 subsystem (守候进程): nfsd, biod, rpc.mountd, rpc.lockd, 和 rpc.statd.

停止一个 subsystem 或 subsystem group, 同样地使用 stopsrc 命令。

要停止并且重启守候进程, 或让守候进程再读一个配置文件, 例如, 要让 inetd 再读/etc/inetd.conf, 使用 refresh 命令为:

```
# refresh -s inetd
```

6.5 进程的定时启动

可以使用 cron 守候进程来定时启动进程, 该进程使用的 crontab 文件在

/var/spool/cron/crontabs 目录中，以用户名为文件名，比如 root 用户的 crontab 文件就为 root。AIX 支持 AT&T 风格 crontab 文件，每条包含下列格式：

分钟 (0 到 59)
小时 (0 到 23)
月的日子 (1 到 31)
一年 (1 到 12)
星期的日子 (0 到 6，整个星期六到星期天)
shell 命令

AIX 也可以直接调用 crontab 命令。crontab 命令有 -e 选项，该选项将装载 crontab 的内容到编辑环境。使用何种编辑器取决于 EDITOR 变量值。一旦从编辑环境保存并且退出，将改变新 crontab 文件并且很快地生效。

正常情况下，crontab spool 目录定义在 /var/spool/cron。另外 AIX 有一个从 /usr/spool 到 /var/spool 连接，这是操作系统为了保持与先前的版本兼容。