



北京希望电脑公司 UNIX SYSTEM V 4.0 技术丛书之五 (上、下)

高级系统管理

下

海洋出版社

北京希望电脑公司 UNIX SYSTEM V 4.0 技术丛书之五

高级系统管理

(上、下)

刘有军 廖彬山 谢小兵 编译
甘登岱 徐拥军 魏彬 校

下

海洋出版社
1991.5

内容摘要

UNIX SYSTEM V 4.0 版是 UNIX SYSTEM V 操作系统的最新版本。新版本由十五册书组成, 内容极为丰富, 包括基本操作系统、大量的开发工具、网络功能和图形用户接口等, 是系统分析人员和应用开发人员难得的参考书。本书是这套丛书中的一册。欲购本套丛书的用户可直接与北京 8721 信箱资料部联系, 联系电话 2562329, 邮码 100080。

北京希望电脑公司 UNIX SYSTEM V 4.0 技术丛书之五

高级系统管理

(下册)

编译 刘有军 廖彬山 谢小兵
审校 甘登岱 徐拥军 魏彬
责任编辑 闫世尊 刘莉蕾 钱晓彬

海洋出版社出版 (北京市复兴门外大街 1 号)

双青印刷厂印刷

开本: 787 X 1092 1/16 印张: 22 字数: 542 千字

1991 年 5 月第一版 1991 年 5 月第一次印刷

印数: 1-3000 册

ISBN 7-5027-2054-5 / TP.27

定价 ¥: 15.00 元

目 录

前言	(1)
0.1 前言	(1)
0.2 本书组织方法	(1)
0.3 手册的使用方法	(2)
第十章 进程调度	(4)
10.1 引言	(4)
10.2 进程调度概述	(5)
10.3 配置调度程序	(7)
10.4 利用 dispadmin 改变调度程序参数	(13)
第十一章 恢复服务	(17)
11.1 引言	(17)
11.2 恢复操作概述	(18)
11.3 利用恢复服务	(20)
11.4 系统恢复	(28)
11.5 恢复服务的快速查阅	(35)
第十二章 安全性	(39)
12.1 引言	(39)
12.2 安全性管理概述	(39)
12.3 使系统安全的几点建议	(40)
12.4 登录和口令	(40)
12.5 登录记录	(48)
12.6 特殊管理登录和系统登录	(49)
12.7 口令恢复	(50)
12.8 文件保护	(51)
12.9 用户设置标识和集体设置标识(Set-UID 和 Set-GID)	(53)
12.10 安全性方法的快速查阅	(56)
第十三章 访问服务	(58)
13.1 引言	(58)
13.2 访问设备服务概述	(60)
13.3 端口监视程序管理	(65)
13.4 服务管理	(73)
13.5 端口监视程序 ttymon	(78)
13.6 终端线路设置	(90)
13.7 监听程序	(97)
第十四章 软件管理	(107)
14.1 简介	(107)

14.2	软件管理综述	(108)
14.3	安装你的软件的建议	(112)
14.4	设置缺省安装	(114)
14.5	存贮与程序包的交互动作	(118)
14.6	安装软件包	(119)
14.7	从远程机器安装软件:使用 RFS 的一个例子	(123)
14.8	检查安装的准确性	(123)
14.9	显示所安装包的有关信息	(124)
14.10	删除包	(127)
14.11	软件管理的快速参考	(128)
第十五章	存贮设备管理	(133)
15.1	简介	(133)
15.2	存贮设备管理综述	(134)
15.3	管理存贮设备的建议	(140)
15.4	维护设备和介质	(141)
15.5	管理设备属性	(152)
15.6	管理设备组	(159)
15.7	管理设备登记	(162)
15.8	设备管理快速索引	(163)
第十六章	系统设置	(172)
16.1	介绍	(172)
16.2	系统设置综述	(174)
16.3	设置控制台终端	(175)
16.4	启动计算机	(176)
16.5	建立一个软键码并指定一个固件口令	(176)
16.6	系统设置步骤	(178)
16.7	在初始设置后改变系统参数	(179)
16.8	系统设置的快速索引	(184)
第十七章	用户和组管理	(187)
17.1	介绍	(187)
17.2	用户和组管理概述	(188)
17.3	用户和组管理的建议	(188)
17.4	控制对系统和数据的访问	(189)
17.5	批设置工作环境:系统和用户配置	(201)
17.6	用户通信	(208)
17.7	用户和组管理的快速参考	(212)
附录 A	设备名和缺省分区	(214)
A.1	介绍	(214)
A.2	设备名	(214)

A.3 硬盘缺省分区	(215)
A.4 磁带分区	(218)
A.5 软盘分区	(219)
附录 B 目录和文件	(220)
B.1 概述	(220)
B.2 目录和文件的调整	(220)
B.3 root 文件系统下的目录	(226)
B.4 / etc 目录下的目录	(227)
B.5 / etc 目录下的文件	(230)
B.6 / usr 下的目录	(235)
B.7 / usr 下的文件	(236)
B.8 / var 下的目录	(237)
B.9 / var 下的文件	(239)
附录 C 使用 sysadm	(242)
C.1 简介	(242)
C.2 遍历菜单窗口	(242)
C.3 帧操作工具	(246)
C.4 实例:增加一个新用户的注册	(252)
C.5 界面操作小结	(256)
C.6 系统管理菜单	(264)
附录 D 修改 sysadm 的风格	(272)
D.1 sysadm 风格修改概述	(272)
D.2 编写自己的求助信息	(274)
D.3 创建或修改一个菜单项	(281)
D.4 创建或修改一个任务项	(284)
D.5 删除一个菜单项或任务项	(286)
附录 E 出错信息	(288)
E.1 介绍	(288)
E.2 UNIX 系统的 NOTICE 信息	(288)
E.3 UNIX 系统的 WARNING 信息	(290)
E.4 UNIX 系统的 PANIC 出错信息	(295)
E.5 UNIX 系统调用出错信息	(300)
E.6 诊断监测程序出错信息	(311)
E.7 已装备的设备表出错信息	(313)
E.8 固件出错信息	(314)
E.9 启动和配置出错信息	(315)
E.10 PUMP 出错信息	(323)
E.11 基本网络工具(BNU)出错信息	(324)
附录 F 邮件子系统管理	(328)

F.1 管理邮件子系统	(328)
词汇表	(335)

前言

0.1 前言

如果你现在是一名运行 UNIX 系统版本 4.0 的计算机的系统管理员, 则本书将会对你大有裨益。你或许是一个小公司的经理, 可能要亲自操作你的机器; 当然, 也可能, 你是一位大型组织的管理人员, 在那里, 有许多用户共享网络计算机。在任何一种情况下, 本书均可帮助你安装和维护系统上的各种资源, 以满足用户的需要。

在 UNIX 系统版本 4.0 所介绍的新功能中, 有许多新的用于管理的软件工具, 其中包括一个新版的系统管理菜单。这些工具将帮助你安装机器和软件、建立最大满足用户需要的资源和环境、帮助你做好日常维护并提供紧急故障排除服务。

0.2 本书组织方法

本书为你提供有关系统管理方面的所有信息。每章讲述一种功能, 如文件系统管理, 安全和备份服务。此外, 附录、词汇及索引的提供使得本手册更易于理解和使用。

0.2.1 安排

本手册各章以字母顺序排列, 其顺序和系统管理主菜单中的功能(Topics)顺序一致(键入 Sysasm 命令后, 该系统管理主菜单将出现在屏幕上)。

0.2.2 每章组织

每章讲述一个功能, 并给出执行该功能的指导。对某些功能而言, UNIX 系统 V 版本 4.0 为你提供一用户一友好菜单界面, 这就使得你在做管理任务时, 可以不使用 UNIX 系统 shell 命令。该界面通过 Sysadm 命令访问。对于该界面提供的功能, 在相应章节的开始部分提供一调用适当菜单的说明。由于菜单(附有其他屏幕信息)是自解释的, 因此, 各章中各不包括如何使用菜单的详细说明。

此外, 本书附录 C, “怎样使用 Sysadm 界面”, 提供了一如何使用菜单示例, 并且定义了菜单系统的所有部分。再者, 在你通过 system 命令使用菜单时, 本界面还提供了用户可以访问的联机“help message”。

每一章均提供了访问相应 Sysadm 菜单的说明, 并给出了可用以代替菜单选项的 shell 命令一览表。

0.2.3 符号约定

本节讲述本书所采用的符号约定。

* 计算机输入输出文字(如用户键入的命令或系统产生的屏幕信息)均以 monospace 字型显示, 如下面例子:

\$ ls -l report.oct17
-r--r--r-- 1 jim doc 3239 May 26 11:21 report.oct17

* 可替代的正文部分(即可用指定值来替换的正文部分)以斜体出现, 如下例

\$ cat filename(斜体)

斜体文字 *filename* 將被你所指定的文件名替換。

* 屏幕上的注释，即非计算机输的正文文字均以斜体出现，并在排版时被缩进，如下例：

command interaction

Press RETURN to continue.

* 提示用户输入的命令后都有 RETURN 键(如 shell 命令或菜单选择之后)。这个 RETURN 键对所有的 UNIX 系统命令或菜单选择都是隐含的。

有时，系统会明显地指出要打入 RETURN 键，例如，在一个交互式的过程中，用户不愿打入任何正文，这时，可以打入 RETURN 键省去正文输入，如：

Type any key to continue: RETURN

* 控制字符用字符串“CTRL-字符”表示，如“CTRL-d”。要输入一个字符所对应的控制字符，可先按下 CTRL，然后敲这个字符。“CTRL-字符”中指定的字符有大小写之分，要按手册中指定的大小写格式敲入控制字符。比如，“CTRL-d”与“CTRL-D”是有区别的。

* 本手册例子中的系统提示符为 ATT UNIX 系统 V 版本 4.0 的缺省值:

☐对一般的用户，缺省系统提示符为“\$”。

☐对于在根目录注册的用户，其缺省系统提示符为'#'。

0.3 手册的使用方法

这本书旨在帮助用户管理好运行 UNIX 系统 V 版本 4.0 的计算机。特别地，它可以

帮助用户了解管理工作的内容，帮助用户在一台计算机上安装、配置、维护 UNIX 系统 V 版本 4.0。

用好这本书，必须要具有一些有关 UNIX 的最基本的知识，如 UNIX 的目录结构和 SHELL。还必须具有一些最基本的硬件知识，如怎样开机，怎样安装外部设备(如调制解调器、终端和打印机)。为了安装和管理计算机，可参考安装手册和有外部设备的随机资料。可以查一下《产品资料索引》，从中找到一些指导性资料。

0.3.1 对于没有经验的系统管理员

对于没有经验的 UNIX 系统管理员，可以把这本手册做为教课书。首先要阅读第一章“概述”，这一章讲系统管理员的职责，并说明怎样组织和开展管理工作，做好每一部分工作需要阅读哪些章节等。

这样，就可以根据需要有目的地去阅读一些章节。所有的系统管理员都要做书中所描述的许多管理工作，但有些管理功能不必涉及。这要依赖于软硬件和用户。总而言之，要根据需要查阅这本书的某些章节。

0.3.2 对于有经验的管理员

对于富有经验的管理员，如果是把这本书只作为参考手册，想查出解决问题的详细的系统命令，这时可查阅有关内容的目录(或索引)。如果要查系统某一功能可参阅相应章节的第一页，也可查附录 C“使用 sysadm 界面”，附录中列出了一些菜单和包含在系统管理菜单界面中的功能。

0.3.3 菜单使用指南

如果想通过菜单界面来完成某一具体具体功能，就要知道怎样进入菜单，可查阅相应于这一功能的章节的第一页。进入菜单后，可参考附录 C“怎样使用 sysadm 界面”来操作。

0.3.4 如果你不使用菜单

如果不打算使用菜单界面，这时可通过阅读某一章来查出完成某一管理功能所需要的命令，这些命令是可以执行的 Shell 命令。

第十章 进程调度

10.1 引言

UNIX 系统调度程序决定进程何时进行, 调度程序根据配置参数, 进程行为 and 用户请求保持进程优先级, 并利用优先级给 CPU 分配进程。

系统 V4.0 版本可使用户控制进程运行的顺序以及各进程使用 CPU 的时间。

缺省状态下, V4.0 版本调度程序的分时规则同以前版本。分时规则随时调整进程优先级以为交互式进程提供良好的响应时间, 为占用大量 CPU 时间的进程提供良好的计算能力。

操作系统 V4.0 版本的调度程序既提供分时规则, 也提供实时规则。实时调度可使用户对每一进程设置固定的优先级。优先级最高的进程一能运行即获得 CPU, 即使系统进程也能运行。可编写一段应用程序用以使实时进程从系统获得确保的响应时间。

对大多数 UNIX 环境而言, 系统设定的调度程序已经足够, 不需要实时进程: 系统管理员不必更改设置的参数, 用户也无需改变其进程的优先级。当然, 如果一程序对时间有严格要求时, 实时进程有时是满足这种要求的唯一方法。

注意:

不小心地使用实时进程不能会对分时进程的性能产生极坏的影响。

本章是写给系统内装有操作系统 V4.0 版本调度程序的管理员的。管理员需理解调度程序至少有两个原因:

1. 调度程序对系统的性能有着难以超越的影响。系统设定的调度程序在有代表性的工作环境中运行良好, 但管理员应该理解调度程序工作的原理以明确是否应重新设置以更好地满足各自需要。

2. 一个实时程序的故障或恶意的实时用户可能会将其他进程(包括内核处理)锁住。用户是不允许产生实时进程的, 只有诚实的用户才会遵守这种规定。管理员应该知道, 这种新的方式可能会给调度程序带来麻烦, 也应该为预防事故和滥用这种功能做准备。

要了解调度程序的编程信息, 参见《程序员指南: 系统服务和编程支持工具》一书。控制进程管理的基本用户指令是 `priocntl(1)`, 该命令在 User 的《用户参考手册》中介绍。控制进程管理的基本功能调用是 `priocntl(2)`, 该命令在《程序员参考手册》中介绍。

本章其余部分安排如下:

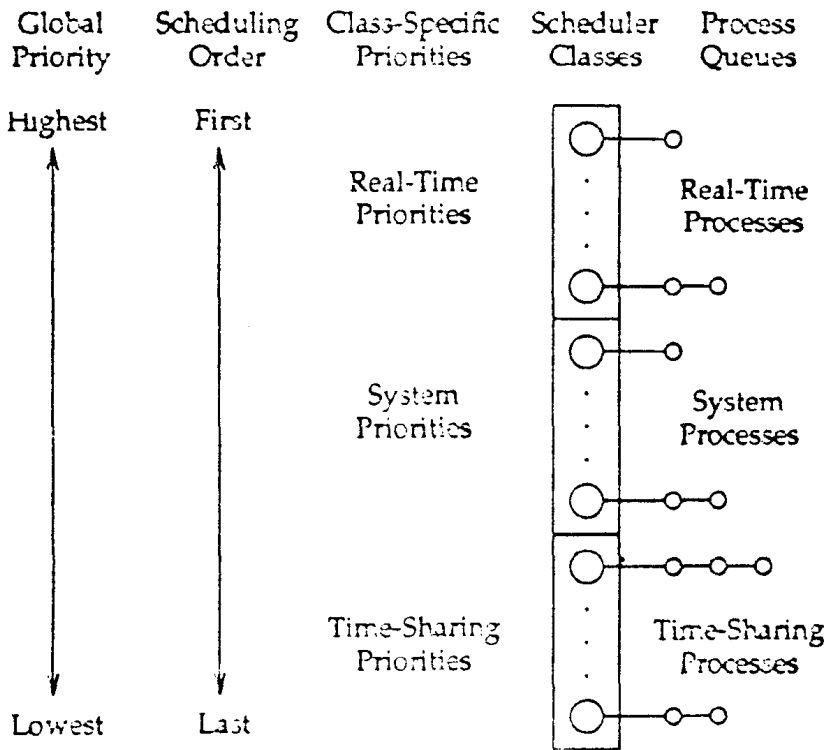
1. “进程调度程序概述”部分介绍调度程序做的工作以及程序怎样利用可调参数, 控制调度程序, 以及两个调度程序参数表: `ts_dptbl(4)`(用于分时的参数)和 `rt_dptbl(4)`(用于实时的参数)。

3. “利用 `dispmix` 改变调度程序参数”部分介绍怎样在运行的系统上显示或改变调度程序参数。利用 `dispadmin` 改变的参数重新启动时不再保留。要对调度程序设置做永久的改变, 必须在 `master.d` 目录下改变调度程序的参数表。

10.2 进程调度概述

下表说明系统 V 和版本 4 进程调度程序的工作原理:

Figure 10-1: The System V Release 4 Process Scheduler



产生一进程后，进程就获得其调度程度参数，包括调度程序类和在类中的优先级。只有在用户发出请求时进程地改变其类。系统在用户请求和进程的调度程序为别的基础上管理进程的优先级。

在系统规定的设置中，初始化进程属于分时类。由于进程继承其调度程序参数，在缺省状态下所有用户以分时进程登录。

调度程序将各类优先级转化为整体优先级。进程的整体优先级决定其何时执行——调度程序总是执行具有最高优先级的可执行进程。调度程序将进程分配给 CPU 以后，进程就开始执行直到用完其时间片，“睡眠”，或被更高优先级进程抢占。有同样优先级的进程轮流执行。

管理员在设置表中指定缺省时间片，用户亦可给实时进程分配各进程的时间片。

在 `ps(1)` 命令下选择 `_cl` 可以显示一进程的整体优先级，利用 `priocntl(1)` 和 `dispadm(1M)` 命令可显示各类优先级的设置信息。

缺省时，所有实时进程比任何内核进程的优先级高，内核进程的优先级又比分时进程的优先级高。

注意：

只要有可执行的实时进程，内核进程和分时进程就不能执行。

下面介绍三种缺省类的调度规则。

10.2.1 分时类

分时规则的目标是为交互式进程提供好的响应时间和为依赖 CPU 的进程提供良好的计算能力。调度程序频繁改变 CPU 的任务以提供良好的响应时间，但也不能太过频繁，以免在切换任务上浪费时间。时间片的数量级一般在几百毫秒。

分时规则随时改变优先级定义不同长度的时间片。调度程序提高仅用少许 CPU 就睡眠（“进程睡眠”是指比如进程启动 I/O 操作如终端读入或磁盘读入）的进程，频繁的休眠是交互式任务的特点，如编辑或运行简单的 shell 指令。另一方面，分时规则降低占用大量 CPU 时间，而不休眠的进程的优先级。

缺省的分时规定给予优先级低的进程较大的时间片。优先级较低的进程一般较长占用 CPU。其他进程抢先占有 CPU，但优先级较低的进程占用 CPU 后，它占用的时间更长。在一时间段内，若优先级较高的进程能够运行，它将抢占运行进程。

调度程序利用分时参数表 `ts_dptbl` 中可设置的参数来管理分时进程，该表包含分时类特有的信息。

10.2.2 系统类

系统类采用固定优先级规则运行内核进程如服务程序和内务处理进程（如分页守护程序）。系统类仅用于内核，用户从系统类在不能增加，也不能删除进程。系统类的优先级是由内核编码设定，一旦设定，系统进程的优先级便不能改变。（以内核工作方式运行的进程不属于系统类）。

10.2.3 实时类

实时类采用优先级固定的调度规则，以便关键进程能以预定的次序运行。实时优先级只有在用户要求时才会改变。优先级固定的规则同分时规则恰恰相反，在分时规则中，系统改变优先级以提供良好的交互式响应时间。

获得特许的用户可使用 `priocntl` 命令或 `pricoctl` 系统调用来分配实时优先级。

调度程序利用实时参数表 `rt_dptbl` 中可以设定的参数来管理实时进程，该表包含了与实时类特有的信息。

10.3 配置调度程序

系统规定的设置包括分时调度程序类和实时类。分时类随典型 UNIX 系统的工作量而调整。该工作量包含大量的交互式进程。实时类随应用程序的需要而调整。

对于传统的分时应用，例如软件开发，办公室应用，文件生成，实时进程似无必要。另外实时进程也许并不称心如意。首先，实时进程占用不能标记页码的存储器：实时进程的 u 块永远无法标记页码。其次，实时进程带来造成性能问题的新方法：高优先级的实时进程可能锁住所有其他进程。在只需要分时处理的环境中，你也许想从设置中删去实时调度程序类，参见后文“改变调度程序设置”。

另一方面，若计算机上运行的应用程序对进程运行的先后顺序有严格要求，则实时调度程序是满足这种要求的唯一可用的方法。

注意：

实时进程可能会对分时进程的性能有很大的负作用。

本节描述了控制调度程序设置的参数和表格，介绍了重新设置调度程序的方法。首先做设你的工作量相对于系统资源(例如 CPU 能力，基本存储量，输入/输出能力。假设你的工作量需要硬件的大量计算和太多的输入输出，重新设置调度程序不会有太大帮助。欲了解更多情况参见“性能管理”。

10.3.1 缺省的整体优先级

下表给出了各调度程序类的调度顺序和整体优先级。

Scheduling Order	Global Priority	Scheduler Class
first	159	Real-Time
	.	
	100	
	99	System
	.	
	60	
last	59	Time-Sharing
	.	
	0	

系统设立后，系统从下节要描述的各调参数表构造上述信息。尽管不强求你设置调节程序类以产生象系统规定的连续的不重复的整体优先级，我们建议你这样做以求简单。另外，建议你将实时整体优先级设定的比其他类要高。这些要求简化了调度程序的设置，它们应能适应对调节程序的其他要求。

内核进程例如交换和分页守护程序在系统调节程序类内执行。内核进程必须用户进程竞争 CPU 时间，在系统规定的设置中所有实时进程的优先级比系统进程的优先级高。因此，编写实时应用程序时要小心以确保内核获得它需要的处理时间。另外，重新设置调度程序时，要确保系统类获得相对于分时类足够的优先级，从而给内核进程以它们所需的 CPU 时间。

10.3.2 可调参数

本节介绍控制调节程序设置的可调参数。这些参数在 `/etc/master.d` 目录下的文件中确定。

下述参数值在内核文件中确定。

- ☐ `MAXCLSYSPRI` 是系统类进程的最大整体优先级。内核开始系统进程时，以 `MAXCLSYSPRI` 为参考点分配进程的优先级。

注意、

- ☐ `MAXCLSYSPRI` 必须不小于 39，因为内核假定自己至少拥有 `MAXCLSYSPRI` 以下范围的优先级。如果你要求 `MAXCLSYSPRI` 低于 39，它将会被改成 39。最重要的系统进程的整体优先级应达到或接近 39，最不重要的系统进程的整体优先级应达到或接近 (`MAXCLSYSPRI-39`)。系统规定的 `MAXCLSYSPRI` 值为 99，这使所有系统进程的优先级比用户进程的优先级都高。
- ☐ `INITCLASS` 是分配给 `init` 进程的调度程序类。该调度程序类由 `init` 的产物获得，这些产物一般包含用户的登录 shells。缺省状态下，`INITCLASS` 为 `TS`，就是所有登录 shells 在缺省状态下是分时进程。
- ☐ `SYS__NAME` 是系统调度程序类的字符串名。`SYS__NAME` 的缺省值为 `SYS`。

下述参数在 `ts` 文件中确定，这些参数控制分时规则。

- ☐ `TSMAXUPRI` 确定了用户可利用 `priocntl` 系统调用修正分时进程优先级的范围：有效范围是从 `-TSMAXUPRI` 到 `+TSMAXUPRI`。`TSMAXUPRI` 的缺省值是 20。（设置 20 仿效了老版本的特点，不常用的调度程序界面 `nice` 和 `setpriority` 仍然有效）。
`TSMAXUPRI` 的值独立于所设定的整体分时优先级的数目，尽管我们建议至少设定 40 个分时优先级，下节介绍 `ts__dptbl` 将予以解释。在缺省时，有 60 个分时优先级，可用户只能在 -20 到 +20 之间修正其优先级。系统可根据进程行为使用剩余的优先级。
- ☐ `NAMETS` 确定分时调度程序类的字符串名。该名字通过 `priocntl` 系统调用返回并

被分配给可调参数 INITCLASS 以确定用户进程在缺省时的调度程序类。

NAMETS 的缺省值是 TS。

下面参数在 rt 文件中确定，该参数控制实时规则：

- ☐ NAMERT 确定实时调度程序类的字符串名字，缺省值为 RT。

10.3.3 实时参数表 rt_dptbl

调度程序使用 rt_dptbl(4)(实时调度程序参数表)管理实时进程。rt_dptbl 的系统规定版本同系统一起发送，管理员可对它进行调整以满足自己的要求。rt_dptbl 在 master.d 目录下 rt 文件中确定。如果系统文件含

INCLUDE:RT

行，rt_dptbl 就作为系统配置的一部分并构成内核。

你可根据系统上应用程序的情况来修正 rt_dptbl 的大小和值。下面是一个简单 rt_dptbl 的一部分。

rt__qlbpri	rt__qntm
100,	100,
101,	80,
102,	60,
103	40,
104,	20,
105,	10,

- ☐ rt__qlbpri 列包括整体优先级(决定进程何时运行的优先级)，数越高运行越早。
- ☐ rt__qntm 列包括缺省时间片(时间片与 rt__qlbpri 列的优先级有关)，这是拥有对应优先级的进程所能使用 CPU 的最长时间)。时间片由时钟频率确定。

表中最高设定的最高优先级是 105，因此优先级为 105 的进程总是比其他进程优先运行。如果进程不睡眠，优先级为 105 的进程在调度程序寻找另一进程之前要运行 10 个时钟频率(因为 105 是最高的优先级，优先级为 105 的进程用完其时间片后只有在有其他优先级为 105 的进程时，才被抢占运行)。优先级为 104 的进程运行 20 个时钟频率，等等。表中给出的最低优先级为 100，优先级为 100 的进程要运行 100 个时钟频率。

系统规定的实时优先级是 rt_dptbl 中设定的最低优先级。该优先级用于分配给要被改变为实时进程，又没有给定优先级的进程。如果 INITCLASS 设定为 RT，则该优先级被分配给 init 及其子进程。

尽管 rt_dptbl 包含实时优先级的缺省时间片，经过批准的用户可独立设定实时优先级和时间片。用户可为其实时进程设定任意时间片，包括无限时间片。系统假设实时进程志愿放弃 CPU 以便其他工作能够完成。

10.3.4 分时参数表 ts_dptbl

调度程序利用 ts_dptbl(4)(分时调度程序参数表)来管理分时进程。系统规定的 ts_dptbl 版本同系统一起发送，管理人员可以修正该表以满足自己的需要。要保存 ts_dptbl 的缺省版本的备份。由 master.d 目录下 ts 文件确定。该表做为系统配置的一部分自动构成内核。

你可以根据自己的需要修正 ts_dptbl 的值，但只有经验丰富的管理员才进行这样的修正。缺省值在很大范围的环境下有着长时间的良好性能，修正这些值一般帮助不大，不适当的值还可能对系统性能有很大的负作用。

如果你决定修正 ts_dptbl，我们建议至少要包括 40 个分时整体优先级。如此大的范围使调度程序有足够的空间根据进程对 CPU 的使用来区分进程，这样调度程序才能为交互式进程提供良好响应。系统规定的设置有 60 个分时优先级。下面是一个简单 ts_dptbl 的一部分：

glbpri	qntm	tqexp	slprt	mxwt	lwt
0,	100,	0,	1	5	1
1,	90,	0,	2	5	2
2,	80,	1,	3	5	3
3,	70,	1,	4	5	4
4,	60,	2,	5	5	5
5,	50,	2,	6	5	6
6,	40,	3,	7	5	7
7,	30,	3,	8	5	8
8,	20,	4,	9	5	9
9,	10,	4,	9	5	9

- ☐ glbpri 列含整体优先级(确定进程何时运行的优先级)。数目大者先运行。在表中整体优先级最高值为 9，最低值为 0。
- ☐ qntm 列含与 glbpri 列中优先级相联系的时间片，该时间片是拥有对应优先级的进程在其他进程之间使用 CPU 的最长时间。时间片由时钟频率确定。
在上表中，时间片从 10 个时钟频率到 100 个时钟频率。
- ☐ tqexp 列确定进程在睡眠前已用完时间片时的新的优先级。若优先级为 glbpri 列中一值的某进程没有睡眠就用它其时间片，调度程序就利用 tqexp 列修正其优先级，以做为返回 ts_dptbl 表的索引，新优先级在 ts_dptbl 中为整体优先级。(在缺省状态下，ts_dptbl 中各项的索引恰好同该项的整体优先级一致。当然这种一致并非必须)。

通常应当降低那些能用完时间片的分时进程的优先级，因为在现有优先级下，它们占用 CPU 时间太长了。较长占用 CPU 时间的进程的优先级应当降低以照顾使用少许 CPU 时间即睡眠的进程，这样的进程往往是交互式进程。