

OS/2应用程序接口技术(API) —程序员手册

杨宋标 编译



中国科学院希望高级电脑技术公司

10210
Y166

A0241020-J

OS/2应用程序接口技术(API)

——程序员手册

杨宋标 编 译

中国科学院希望高级电脑技术公司

版权所有
不许翻印
违者必究

★北京市新闻出版局

准印证号：3195-90195

★订购单位：北京8721信箱资料部

★邮 码：100080

★电 话：2562329

★传 真：01—2561057

★乘 车：320、302、332、路车

到海淀黄庄下车

★办公地点：希望公司大楼一楼往里走

102房间

前 言

OS/2是具有单用户多任务功能的新型操作系统。它由美国MICROSOFT公司和IBM公司合作开发，被作为IBM-PC/AT，IBM-PS/2系列的操作系统而采用，此外还相继移植到了IBM公司以外的其他个人机上，被称为新一代的操作系统。OS/2的普及将推动应用程序的开发，OS/2和MS-DOS不同，它采用了以CALL为基础的接口，因而使用C之类的高级语言来开发应用程序变得更容易了。

本书以MS OS/2为基础按功能类别汇总了开发应用程序所必要的函数，并作了系统的解说，可作为程序员手册。MS OS/2 SDK (SOFTWARE DEVELOPMENT KIT—软件开发工具包) 最适合于在OS/2环境下进行应用程序的开发。本书是以MICROSOFT公司所欣赏的MS OS/2 1.03版为基础编辑而成的，通俗易懂。本书对除表示管理(Presentation Manager) 以外的217个函数，以函数的目的，调用方法，参数等方面的机能进行了说明。此外在第一部中具体地记述了函数使用实例，因而能在实践中灵活运用。

本书的目的是以通俗易懂的方法解说MS OS/2具有的丰富功能，但愿能对OS/2的学习者和应用程序的开发者有所帮助。

目 录

第一部分 API简介

第一章 何谓API	(1)
1.1 什么是API.....	(1)
1.2 API的优点.....	(2)
1.3 API的分类.....	(2)
1.4 有关 FAPI.....	(3)
1.5 API的使用方法, 注意点.....	(3)
1.6 API的功能类别概要.....	(4)
第二章 函数使用实例	(15)
2.1 应用程序的开发.....	(15)
2.2 段 (Segment) 的分配及释放程序.....	(15)
2.3 子处理启动程序.....	(18)
2.4 进程启动程序.....	(20)
2.5 通道的建立程序.....	(23)

第二部分 函数参考

参考的使用方法	(28)
第一章 动态链接函数	(29)
1.1 DosFreeModule	(29)
1.2 DosGetModHandle.....	(29)
1.3 DosGetModName	(30)
1.4 DosGetProcAddr	(31)
1.5 DosLoadModule	(31)
第二章 任务分配函数	(32)
2.1 DosCreateThread	(32)
2.2 DosCwait	(32)
2.3 DosEnterCritSec	(34)
2.4 DosExecPgm(F).....	(35)
2.5 DosExit (F).....	(37)
2.6 DosExitCritSec	(38)
2.7 DosExitList.....	(38)
2.8 DosGetInfoSeg	(39)

2.9 DosGetPid	(40)
2.10 DosGetPrty	(41)
2.11 DosKillProcess.....	(42)
2.12 DosPTrace	(42)
2.13 DosResumeThread	(44)
2.14 DosSelectSession.....	(45)
2.15 DosSetPrty.....	(45)
2.16 DosSetSession	(46)
2.17 DosStartSession	(47)
2.18 DosStopSession.....	(49)
2.19 DosSuspendThread	(50)
第三章 处理间的通信函数	(50)
3.1 DosCloseQueue	(50)
3.2 DosCloseSem	(51)
3.3 DosCreateQueue.....	(51)
3.4 DosCreateSem.....	(52)
3.5 DosFlagProcess	(53)
3.6 DosMakePipe	(54)
3.7 DosMuxSemWait	(55)
3.8 DosOpenQueue	(56)
3.9 DosOpenSem.....	(57)
3.10 DosPeekQueue	(57)
3.11 DosPurgeQueue.....	(59)
3.12 DosQueryQueue	(59)
3.13 DosReadQueue	(59)
3.14 DosSemClear.....	(61)
3.15 DosSemRequest.....	(62)
3.16 DosSemSet	(62)
3.17 DosSemSetWait.....	(63)
3.18 DosSemWait	(64)
3.19 DosWriteQueue	(64)
第四章 存储管理函数	(66)
4.1 DosAllocHuge(F)	(66)
4.2 DosAllocSeg(F).....	(67)
4.3 DosAllocShrSeg	(68)
4.4 DosCreateCSAlias	(69)
4.5 DosFreeSeg(F).....	(69)
4.6 DosGetHugeShift(F).....	(70)
4.7 DosGetSeg.....	(70)

4.8	DosGetShrSeg	(71)
4.9	DosGiveSeg	(71)
4.10	DosLockSeg	(72)
4.11	DosMemAvail	(73)
4.12	DosReallocHuge	(73)
4.13	DosReallocSeg(F)	(74)
4.14	DosSubAlloc(F)	(74)
4.15	DosSubFree(F)	(75)
4.16	DosSubSet(F)	(76)
4.17	DosUnlockSeg	(76)
第五章	设备I/O函数	(77)
5.1	DosBeep (F)	(77)
5.2	DosCLIAccess (F)	(77)
5.3	DosDevConfig (F)	(78)
5.4	DosDevIOCtl(F)	(78)
5.5	DosPortAccess	(79)
5.6	KbdCharIn(F)	(80)
5.7	KbdClose	(82)
5.8	KbdDeRegister	(83)
5.9	KbdFlushBuffer (F)	(83)
5.10	KbdFreeFocus	(83)
5.11	KbdGetFocus	(84)
5.12	KbdGetStatus(F)	(84)
5.13	KbdOpen	(86)
5.14	KbdPeek (F)	(86)
5.15	KbdRegister	(89)
5.16	KbdSetStatus (F)	(90)
5.17	KbdStringIn(F)	(92)
5.18	KbdSynch	(93)
5.19	KbdXlate	(93)
5.20	VioDeRegister	(95)
5.21	VioEndPopUp	(96)
5.22	VioGetAnsi	(97)
5.23	VioGetBuf (F)	(97)
5.24	VioGetConfig(F)	(98)
5.25	VioGetCurPos(F)	(99)
5.26	VioGetCurType (F)	(100)
5.27	VioGetFont	(101)
5.28	VioGetMode (F)	(102)

5.29	VioGetPhysBuf(F).....	(104)
5.30	VioGetState	(104)
5.31	VioModeUndo	(106)
5.32	VioModeWait	(107)
5.33	VioPopUp	(108)
5.34	VioPrtSc.....	(109)
5.35	VioPrtScToggle	(109)
5.36	VioReadCellStr (F).....	(110)
5.37	VioReadCharStr (F).....	(111)
5.38	VioRegister	(111)
5.39	VioSavRedrawUndo	(115)
5.40	VioSavRedrawWait	(116)
5.41	VioScrLock (F).....	(117)
5.42	VioScrollDn (F).....	(118)
5.43	VioScrollLf (F).....	(119)
5.44	VioScrollRf (F).....	(120)
5.45	VioScrollUp (F).....	(121)
5.46	VioScrUnlock (F).....	(122)
5.47	VioSetAnsi	(122)
5.48	VioSetCurPos(F).....	(122)
5.49	VioSetCurType(F).....	(123)
5.50	VioSetFont	(124)
5.51	VioSetMode (F)	(125)
5.52	VioSetState	(128)
5.53	VioShowBuf (F)	(129)
5.54	VioWrtCellStr (F).....	(130)
5.55	VioWrtCharStr(F).....	(131)
5.56	VioWrtCharStrAtt(F).....	(131)
5.57	VioWrtNAttr(F).....	(132)
5.58	VoiWrtNCell(F).....	(133)
5.59	VioWrtNChar (F).....	(133)
5.60	VioWrtTTY (F).....	(134)
第六章	设备控制器函数	(136)
6.1	DosMonClose.....	(136)
6.2	DosMonOpen	(136)
6.3	DosMonRead	(137)
6.4	DosMonReg.....	(138)
6.5	DosMonWrite.....	(139)
第七章	计时器函数.....	(139)

7.1	DosGetDateTime (F)	(139)
7.2	DosSetDateTime (F)	(141)
7.3	DosSleep(F)	(142)
7.4	DosTimerAsync	(142)
7.5	DosTimerStart	(143)
7.6	DosTimerStop	(144)
第八章 FAPI程序执行控制函数		(144)
8.1	DosGetMachineMode	(144)
第九章 文件I/O函数		(145)
9.1	DosBufReset (F)	(145)
9.2	DosChdir (F)	(145)
9.3	DosChgFilePtr(F)	(146)
9.4	DosClose (F)	(147)
9.5	DosDelete (F)	(147)
9.6	DosDupHandle (F)	(147)
9.7	DosFileLocks(F)	(148)
9.8	DosFindClose(F)	(149)
9.9	DosFindFirst(F)	(150)
9.10	DosFinNext(F)	(153)
9.11	DosMkdir(F)	(155)
9.12	DosMove (F)	(155)
9.13	DosNewSize	(156)
9.14	DosOpen (F)	(156)
9.15	DosPhysicalDisk	(160)
9.16	DosQCurDir (F)	(162)
9.17	DosQCurDisk(F)	(162)
9.18	DosQFHandState (F)	(163)
9.19	DosQFileInfo (F)	(165)
9.20	DosQFileMode (F)	(166)
9.21	DosQFSInfo (F)	(167)
9.22	DosQHandType	(169)
9.23	DosQVerify (F)	(169)
9.24	DosRead (F)	(170)
9.25	DosReadAsync	(170)
9.26	DosRmdir(F)	(171)
9.27	DosScanEnv	(172)
9.28	DosSearchPath	(172)
9.29	DosSelectDisk (F)	(174)
9.30	DosSetFHandState (F)	(174)

9.31	DosSetFileInFo (F)	(176)
9.32	DosSetFileMode(F)	(177)
9.33	DosSetFSInfo (F)	(178)
9.34	DosSetMaxFH	(179)
9.35	DosSetVerify (F)	(179)
9.36	DosWrite... (F)	(180)
9.37	DosWriteAsync	(181)
第十章	错误处理函数	(183)
10.1	DosErrClass (F)	(183)
10.2	DosError (F)	(184)
10.3	DosSetVec(F)	(185)
10.4	DosSystemService	(186)
第十一章	信息函数	(190)
11.1	DosGetMessage	(190)
11.2	DosInsMessage (F)	(191)
11.3	DosPutMessage (F)	(191)
第十二章	程序启动函数	(192)
12.1	DosGetEnv (F)	(192)
12.2	DosGetVersion(F)	(193)
第十三章	信号函数	(193)
13.1	DosHoldSignal (F)	(193)
13.2	DosSendSignal	(194)
13.3	DosSetSigHandler (F)	(194)
第十四章	支持各国语言的程序设计接口	(196)
14.1	DosCaseMap(F)	(196)
14.2	DosGetCollate(F)	(197)
14.3	DosGetCp	(198)
14.4	DosGetCtryInfo (F)	(199)
14.5	DosGetDBCSEv (F)	(201)
14.6	DosSetCp	(202)
14.7	KbdGetCp	(203)
14.8	KbdSetCp	(203)
14.9	KbdSetCustXt	(204)
14.10	VioGetCp	(205)
14.11	VioSetCp	(205)
第十五章	鼠标器 API的函数	(206)
15.1	MouClose	(206)
15.2	MouDeRegister	(206)
15.3	MouDrawPtr	(206)

15.4	MouFlushQue	(207)
15.5	MouGetDevStatus	(207)
15.6	MouGetEventMask.....	(208)
15.7	MouGetHotKey	(209)
15.8	MouGetNumButtons	(209)
15.9	MouGetNumMickeyes	(210)
15.10	MouGetNumQueEI	(210)
15.11	MouGetPtrPos	(211)
15.12	MouGetPtrShape	(212)
15.13	MouGetScaleFact.....	(213)
15.14	MouInitReal	(214)
15.15	MouOpen	(214)
15.16	MouReadEventQue	(215)
15.17	MouRegister	(216)
15.18	MouRemovePtr.....	(218)
15.19	MouSetDevStatus	(218)
15.20	MouSetEventMask	(219)
15.21	MouSetHotKey	(220)
15.22	MouSetPtrPos	(221)
15.23	MouSetPtrShape	(222)
15.24	MouSetScaleFact	(223)
15.25	MouSynch.....	(224)
附录A	MS—DOS函数与OS/2函数的对照	(225)
附录B	错误码一览表	(226)

第一部分 API简介

第一章 何谓API

1.1 什么是API

API (Application·Program·Interface) 是指在程序中使用各种各样OS/2的系统资源的所有函数调用。OS/2这样的多任务系统,其内存,文件,键盘,鼠标或显示器等均由操作系统管理。通过对各种资源的统一管理,调整各程序之间的关系,从而可提高系统的运行和维护效率。

应用程序在使用系统各资源时,向OS/2提出使用要求。这时API就充当应用程序和OS/2之间的接口。OS/2系统的结构如图1—1所示。

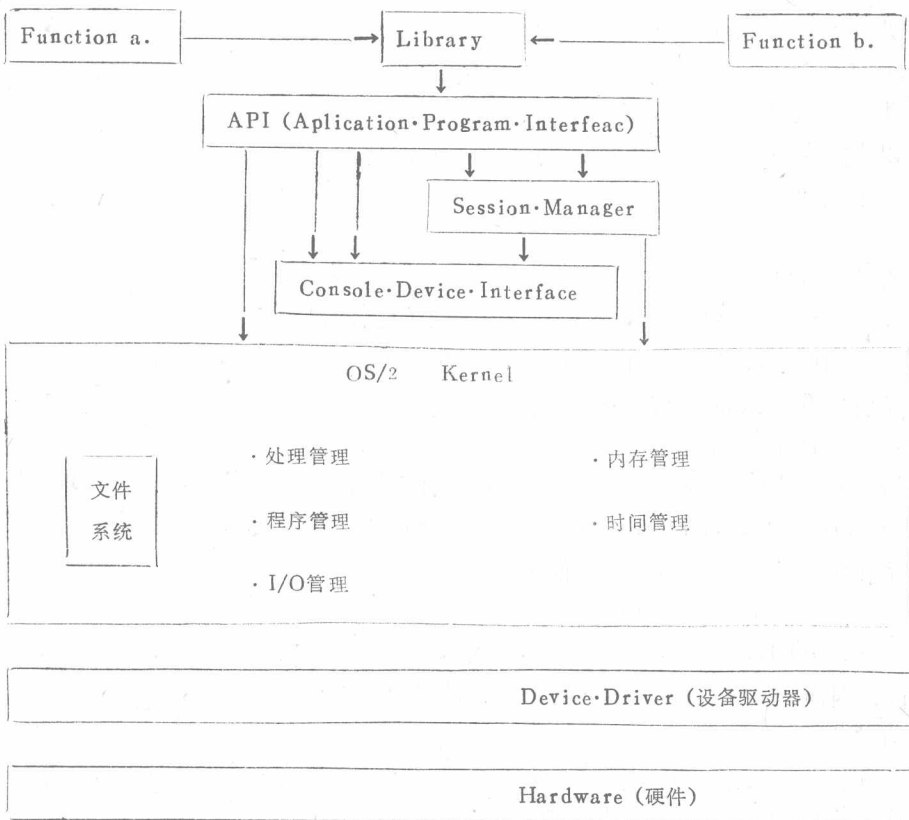


图1—1 系统结构

1.2 API的优点

API与MS—DOS3.X的版本(以下就称DOS3.X)相比函数调用的方法大大地改变了。DOS3.X的函数调用是通过INTxxH命令进行的,这种方法是把函数号和参数设置于寄存器然后发INTxxH命令来执行,执行的结果由进位标志设置,错误码由AX寄存器返回。

API的函数调用使用了堆栈,利用堆栈能接受许多参数。此外以CALL—RETURN的形式进行函数调用,通过这一方法就能从C, PASCAL之类的高级语言直接调用函数了。另外API利用了动态链接功能,动态链接功能是OS/2值得一提的特长之一。本来我们进行的链接被称为静态链接。要结合的程序在链接时才被组装起来。在静态链接中要结合的例行程序由所有参照该例行程序的程序组装成一个文件。而以动态形式作成的模块,是在程序装入或执行时才组装成,动态链接的例行程序一装入,参照的程序就和这例行程序结合。因而即使参照同一例行程序的执行程序有几个,但被参照的例行程序只有一个,这种方法缩小了程序文件,提高了内存的使用效率。

在DOS3.X中,各函数是根据函数号来区别的,而在OS/2中,键盘、鼠标等系统资源能被称为Kbd、Mou等名称有意义的函数来管理,所以进行函数扩充时功能区分就变得容易了。此外,由于以CALL进行函数调用因而函数个数的限制也就没有了。

API的优点归结如下:

- 高级语言能直接调用
- 能传递多个参数
- 以动态形式链接
- 能以有意义的名字指定函数
- 没有函数数目的限制

1.3 有关API的分类

OS/2执行DOS3.X程序的实方式和多任务用的执行程序的保护方式两种。API有保护方式专用的API和两种方式都能用的API,甚至还有实方式专用的API。

以下是各种API的特征和注意点

〔保护方式用的API〕

保护方式用的API能利用OS/2所具有的所有功能。要进行处理控制,处理间的通信,就要使用保护方式用的API。但由于这些OS/2的功能在DOS3.X正不能执行,因而使用该API的程序在实方式和DOS3.X上不能运行。

〔两用方式的API (FAPI) 〕

两用方式的API叫做Family API (FAPI)。FAPI能在实方式,保护方式或DOS3.X上运行,因而FAPI支持的函数所受的限制和DOS3.X一样。

〔实方式用的API〕

实方式用的API是把用DOS3.X的INTxxH命令执行的函数调用用实方式来实现。用它编成的程序能在OS/2的实方式和DOS3.X上运行。

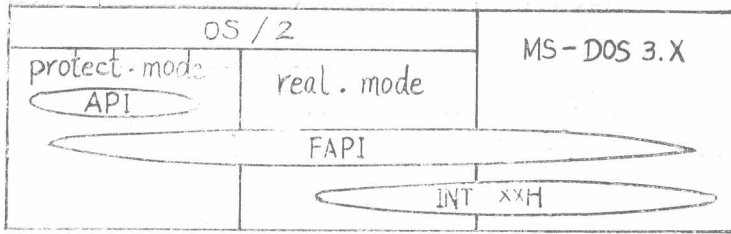


图1-2 API的分类

1.4 有关FAPI

FAPI是能在OS/2的保护方式、实方式以及DOS3.X上使用的可调用函数包。即：只使用FAPI的程序能以OS/2的实方式/保护方式两种方式运行，它们被称为具有两性方式的程序。

使用FAPI两性方式的程序内部，有从FAPI向INTxxH转换的仿真器。FAPI仿真器的装入是由OS/2的BIND实用程序进行的。

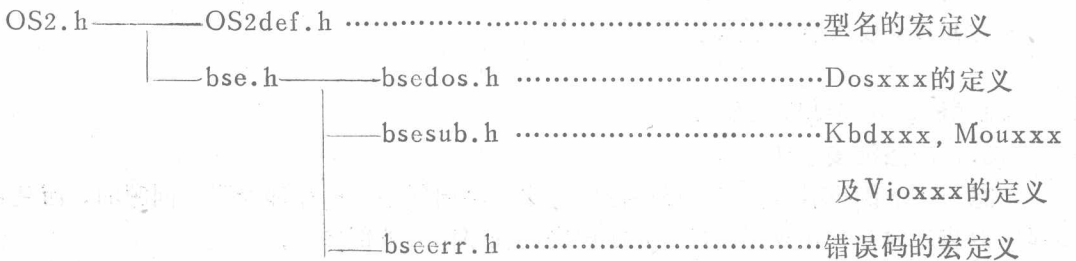
仅用FAPI编成的程序，虽然能在两种方式上运行，但也有缺点。第一，由于要备有FAPI仿真器，因而程序文件变得大了。第二，在以实方式运行的情况下，由于要边执行边转换函数，因而程序的运行速度就变得慢了。

在使用FAPI编制两性方式的程序时，有必要考虑以上的问题。

1.5 API的使用方法，注意点

要使用API必须把说明该函数的HeadFile取到源文件中来。OS/2 1.03版有两个Head-File，即：bsedos.h和bseub.h来说明所有的Dosxxx和Kbdxxx，Mouxxx或Vioxxx函数。

HeadFile的取入是指在Dosxxx的情况下，把“#include <bsedos.h>”、在Kbdxxx，Mouxxx或Vioxxx的情况下，把“#include <bseub.h>”加到源程序中去。在OS/2 1.03版中对于函数调用必要的HeadFile有如下几种。



API利用堆栈以CALL—RETURN的形式来传递参数。其堆栈的形式和C语言等通常使用的方式不同，是以倒向的方式堆放的。

图1—3 显示了堆栈的结构和框架

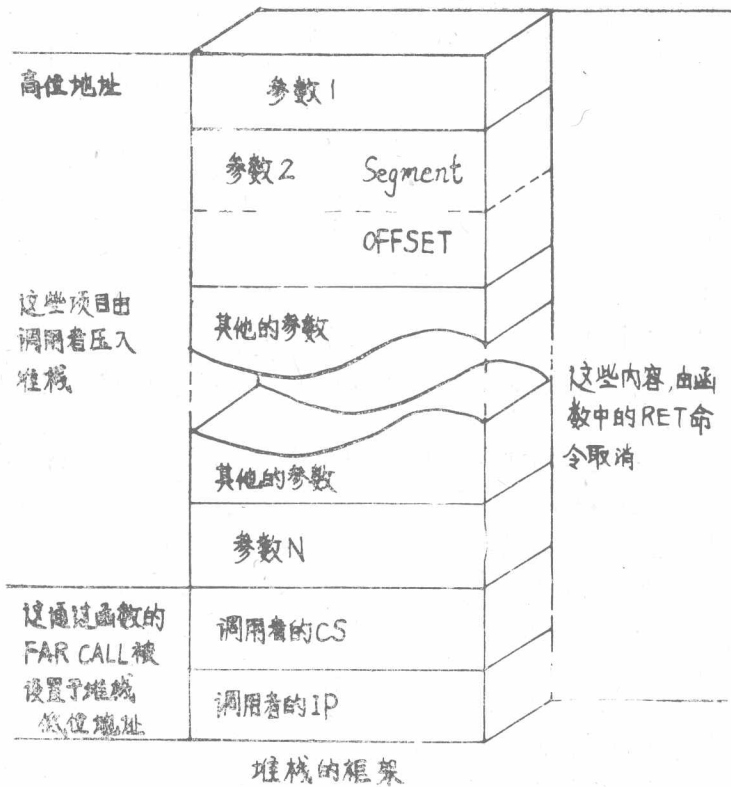


图1—3 堆栈的框架

使用API编制应用程序时，必须注意以下几点：

- 所有的参数通过堆栈传递。
- 返回码由AX寄存器返回给调用者。
- 参数的地址用Selector:Offset (32位) 的形式来表示。
- 所有的函数从堆栈删除参数。

1.6 API的功能类别概要

1.6.1 动态链接函数

动态链接函数并不是在应用程序链接时才解决向程序库的外部参照的问题的，而是在应用程序被装入内存时（或，执行时）才解决向程序库的外部参照。

动态链接有装入链接和执行链接两种。装入链接是在应用程序被装入时解决向程序库的参照，而执行链接和它不同，在执行链接的情况下，源程序中使用函数调用。应用程序通过调用把DLL (Dynamic·Link·Library) 装入内存。

在OS/2中备有作为执行链接用的动态链接函数。动态链接函数有以下一些。

表1—1 动态链接函数

函数名	说 明	参照Page
DosFreeModule	从内存释放动态链接模块	61
DosGetModHandle	返回动态链接模块的句柄	62
DosGetModName	返回动层链接模块名	63
DosGetAddr	返回动态动态链接的过程地址	64
DosLoadModule	把动态链接模块装入内存	64

注意：如果函数名后有*则说明是Family API函数。

1.6.2 任务分配函数

多任务处理有指在同一个计算机系统上同时运行几个程序的功能。在OS/2中，把CPU的处理时间细分，然而根据任务的优先顺序进行分配从而进行多任务处理。

在OS/2中有能实现这样多任务处理的任务分配函数。任务分配函数有以下一些。

表1—2 任务分配函数

函数名	说 明	照参Page
DosCreateThread	产生非同步类型的执行进程	67
DosCWait	等待子处理的结束	68
DosEnterCritSec	禁止当前处理的其他进程的执行	71
DosExecPgm *	其他的程序作为子处理执行	72
DosExit *	退出程序	75
DosExitCritSec	允许当前处理中其他进程的执行	77
DosExitList	保护以处理结束为目的程序清单	78
DosGetInfoSeg	返回全局/局部段的地址	79
DosGetPid	返回处理，进程，父处理的ID	82
DosGetPrty	返回处理的优先级	83
DosKillProcess	返回处理的结束码	84
DosPTrace	给程序的调试提供接口	85
DosResumeThread	再启动进程	88
DosSelctSession	把会话 (Session) 设置于前台	89
DotSetPrty	设定处理的优先级	90
DosSetSession	设定子会话的状态	92
DosSartSession	起动车话 (Session)	94
DosStopSession	终止会话	97
DosSuspendThread	阻塞进程的执行	99

1.6.3 处理间的通信函数

OS/2由于是多任务处理的，故它同时能执行几个应用程序。象这样的操作系统必须具有处理间的通信（使用资源的通知，应用程序之间的数据传送）手段。

在OS/2中具有以下一些作为通信手段的功能：

- 信号量
- 公用存贮区（存贮管理函数）
- 通道
- 队列
- 信号（信号函数）

在OS/2中它们的产生，削除，打开，关闭等是由处理间的通信函数进行的。处理间的通信函数有以下一些：

表1—3 处理间的通信函数

函数名	说明	参照Page
DosCloseQueue	关闭等待队列	101
DosCloseSem	关闭系统信号量	102
DosCreateQueue	建立等待队列并打开它	103
DosCreateSem	产生系统信号量	105
DosFlagProcess	设置处理的外部事件标志	107
DosMakePipe	建立通道	109
DosMuxSem Wait	根据信号量阻塞当前进程	110
DosOpenQueue	打开等待队列	112
DosOpenSem	打开既存的系统信号量	113
DosPeckQueue	从等待队列读出元素	114
DosPurgeQueue	清除等待队列	116
DosQueryQueue	返回等待队列的长度	117
DotReadQueue	从等待队列读出元素并删除	118
DosSemClear	清除信号量	120
DosSemRequest	获得信号量的所有权	121
DosSemSet	无条件地设置信号量	123
DosSemSet Wait	设置信号量，直到信号量的下次清除为止一直阻塞当前进程	124
DosSem Wait	一直阻塞当前进程，直到信号量被清除	125
DosWriteQueue	向等待队列追加元素	126