

2000系列
软件资料

VAX P.S.I X.25程序设计员指南

《小型微型计算机系统》编辑部

2000系列资料出版中心

目 录

第一章 X.25和X.29的程序设计

1.1 通过PSDN的通信	1
1.2 X.25和X.29程序设计	1

第二章 X.25程序设计的方针

2.1 X.25程序设计综述	2
2.1.1 系统服务的使用	2
2.1.1.1 状态和报错	3
2.1.2 编程步骤	3
2.1.2.1 编写执行外出呼叫的程序	3
2.1.2.2 编写处理外来呼叫的程序	4
2.2 数据结构	5
2.2.1 网络连接块	5
2.2.2 邮箱	5
2.3 VAXPSI 怎样处理外来呼叫	6
2.3.1 VAX PSI SERVER 数据基	6
2.3.2 VAX PSI怎样处理一个外来呼叫	7
2.3.3 用网络进程处理外来呼叫	8
2.4 MACRO编程	8
2.4.1 参数列表	8
2.5 高级语言编程	9
2.5.1 描述符	10

第三章 系统服务的使用

3.1 系统服务的使用介绍	11
3.2 建立和清除通信	11
3.2.1 分配一条传输通道	11
3.2.2 建立一条虚电路	12
3.2.2.1 使用缺省网络项	14
3.2.2.2 要求的系统资源	15
3.2.3 清除一条虚电路	15
3.2.4 去配一条通道	16
3.3 处理外来呼叫	16
3.3.1 分配一条用来接收数据的通道	16
3.3.2 建立一条虚电路的外来请求	16
3.3.2.1 接受请求	16
3.3.2.2 拒绝请求	17

3.3.2.3	18
3.4 传送数据	19
3.5 接收数据	20
3.6 中断的传送及其接纳认可	21
3.7 虚电路的复位和复位认可	22
3.8 重起动的接纳认可	22
3.9 宣布进程为网络进程	23
3.10 永久虚电路的使用	23
第四章 系统服务	
4.1 介绍	25
4.2 语法约定	25
4.3 系统服务的定义	25
4.3.1 公共的 \$QIO 参数	26
4.3.2 公共的 \$QIO 返回状态码	26
4.4 系统服务	27
4.4.1 \$ASSIGN—分配一条通道	27
4.4.2 \$CANCEL—清除一条通道上的虚呼叫	28
4.4.3 \$DASSGN—去配通道	29
4.4.4 \$QIO (IO\$_ACCESS) —建立一条虚电路	30
4.4.5 \$QIO (IO\$_ACCESS; IO\$_M_ABORT) —拒绝一个建立一条虚电路的请求	32
4.4.6 \$QIO (IO\$_ACCESS; IO\$_M_ACCEPT) —接受一个建立一条虚电路的请求	34
4.4.7 \$QIO (IO\$_ACCESS; IO\$_M_REDIRECT) —重新定向一个建立一条虚电路的请求	36
4.4.8 \$QIO (IO\$_ACPCONTROL) —宣布进程作为网络进程	38
4.4.9 \$QIO (IO\$_DEACCESS) —清除一条虚电路	41
4.4.10 \$QIO (IO\$_NETCONTROL) —中断的接纳认可	42
4.4.11 \$QIO (IO\$_NETCONTROL) —发送一个中断	43
4.4.12 \$QIO (IO\$_NETCONTROL) —复位一条虚电路或认可接纳一个复位	44
4.4.13 \$QIO (IO\$_NETCONTROL) —接纳重新起动的认可	45
4.4.14 \$QID (IO\$_READVBLK) —接收数据	46
4.4.15 \$QIO (IO\$_WRITEVBLK) —发送数据	47
4.5 系统服务完成时返回的状态码	48
4.5.1 测试返回状态码	49
4.5.2 特定的返回事件	50
4.5.2.1 资源等待方式	50
4.5.2.2 系统服务失败的异常方式	50

4.5.3 获得其它符号码的值	51
附录A、网络连接块 (NCB)	
A.1 NCB 的说明	52
A.2 NCB 的格式	52
A.3 NCB 项的功能	52
A.4 NCB 项的说明	54
A.5 NCB 举例	58
附录B、邮箱报文	60
附录C、系统服务调用的概述	62
附录D、返回状态值和第二状态值	
D.1 返回状态值	63
D.2 第二状态值	66
D.2.1 IO\$_ _ACCESS和IO\$_ _DEACCESS操作	66
D.2.2 IO\$_ _READVBLK操作	68
附录E 程序设计举例	
E.1 外来呼叫	69
E.1.1 处理外来呼叫的C语言程序	69
E.1.2 处理外来呼叫的MACRO程序	73
E.2 外出呼叫	78
E.2.1 进行外出呼叫的C语言程序	78
E.2.2 进行外出呼叫的MACRO程序	83

《VAS P.S.I. X.25程序设计员指南》

序号: AA—N801E—TE

操作系统及版本: VAX/VMS V5.0

软件版本: VAX PSI V4.2

译者: 徐守祥

校对: 刘明烈

第二章 X.25 程序设计的方针

有关通信中系统服务的使用,数据结构(见2.2节)的使用,和如何处理外来的呼叫(见2.3节),这一章给出概括性的指导。

2.4 节是对VAX MACRO 程序员的指导,2.5 节是对高级语言程序员的指导。结合这一章和第三、四章所给的内容弄懂附录E中的程序例子是十分有用的。

2.1 X.25程序设计综述

2.1.1 系统服务的使用

X.25 程序使用VMS系统服务通过PSDN进行通信。程序通过系统服务来:

- 分配和去配连到PSDN的逻辑通道(对于点对点链接则是连到另一机器)。
- 建立和清除通过PSDN运载数据的虚电路。
- 发送和接收数据。
- 发出控制和同步请求。

VAX/VMS所支持的每个程序设计语言都有一个调用系统服务的机构。其细节请看相应程序设计语言的用户指南。

使用\$QIO系统服务来建立一条虚电路。\$QIO系统服务也用于对外来建立一条虚电路请求的接受、拒绝、或重新定向。

在建立一条虚电路以前,使用\$QIO系统服务:

- 确定进程将处理哪些呼叫。
- 当虚电路建好以后,使用\$QIO系统服务:
- 发送和接收数据报文。
- 通过这条虚电路发出控制和同步请求。

通过在调用中指定相应的功能码,通知\$QIO来完成一个具体的功能。表2—1列出了这些功能码及其用途。

表2—1 QIO系统服务的功能码

功 能 码	用 途
IO\$_ACCESS	建立一条虚电路(见3.2.2节)
IO\$_ACCESS!IO\$_ABORT	拒绝一个建立虚电路的请求(见3.3.2.2)。
IO\$_ACCESS!IO\$_ACCEPT	接受一个建立虚电路的请求(见3.3.2.1)。
IO\$_ACCESS!IO\$_REDIRECT	重新定向一个建立虚电路的请求(见3.3.2.3)。
IO\$_ACPCONTROL	确定希望收的外来呼叫(见3.9)。
IO\$_DEACCESS	清除一条虚电路(见3.2.3)
IO\$_NETCONTROL	传送中断和接纳中断的认可(见3.6), 传送复位请求和接纳复位请求的认可(见3.7), 和接纳重起动的认可(见3.8)。
IO\$_READVBLK	接收通过虚电路由远程DTE传来的数据(见3.5)
IO\$_WRITEVBLK	通过虚电路传送数据(见3.4)。

当QIO服务完成后,控制返回到调用程序并且给出一个指示操作结果的状态码。有关

QIO服务的进一步细节，参见《VMS系统服务手册》。

2.1.1.1 状态和报错

当一个系统服务完成后，它返回一状态值。`$ASSIGN`，`$DASSGN`，和`$QIO`系统服务将返回的状态信息放在寄存器0中（`R0`）。对所有的QIO系统服务调用，成功的返回状态`SS$_NORMAL`仅表明成功地排队了该请求。

当一个系统服务完成后，状态值也放于I/O状态块（IOSB）的第一个字中。IOSB的位置是在给出系统服务参数时给定的（见第四章相关系统服务的定义）。IOSB中返回状态的成功表明该操作的成功完成。提供接收这种返回信息的机构（空间等）是由用户程序负责的。

例如，当使用QIO（`IO$_ACCESS`）建立一条虚电路时，`R0`中的返回状态`SS$_NORMAL`表明已成功地排队了该QIO操作，但在IOSB中的返回状态`SS$_NORMAL`则说明远程DTE已经接受了该呼叫。注意，在`R0`中是成功状态而IOSB中却是另外一种状态，这种情况是可能的。例如，当建立一条虚电路时，`R0`中含有`SS$_NORMAL`，表明成功地排队了该操作，而IOSB含有`SS$_CLEARED`，这说明远程DTE拒绝该呼叫。

程序中应该检查该状态，以了解该服务操作是否成功地完成。若未成功地完成，程序中则必须采取补救的措施。

4.5到4.5.3更详细地说明了这些返回状态码。

2.1.2 编程步骤

2.1.2.1和2.1.2.2 给出了进行外出呼叫和处理外来呼叫的编程中所涉及的大体步骤。

对于程序设计内容的具体细节，参见第三、四章。

2.1.2.1 编写执行外出呼叫的程序

执行外出呼叫的程序由下列步骤组成：

1、包含VAX PSI 库

在所写的任何程序中都要包含VAX PSI库。

- 对于MACRO，使用下列命令：

```
.LIBRARY "SYS$LIBRARY:PSILIB"
```

并且通过在程序中指定下列行来说明VAX PSI专用的符号：

```
$PSIDEF
```

- 对于高级语言，写一个含有下列行的MACRO模块：

```
.LIBRARY "SYS$LIBRARY:PSILIB"
```

```
$PSIDEF < = > , < :: >
```

汇编这个文件，并将其产生的目标文件与用高级语言所写的程序相连接。

2. 说明网络连接块

网络连接块（NCB）用来向VAX PSI软件传递信息。在程序中的适当地方定义这个数据结构，及其内容（例如，在MACRO程序的开头）。NCB尤其是存放所要求的网络功能选项的地方（其细节见2.2节）

3. 分配一条通道

程序中必须使用`$ASSIGN`系统服务分配一条到`NWA0`；设备的通道（有关分配通道的进一步内容见3.2.1节）。

4. 建立一条虚电路

使用刚分配完的通道，建立一条到远程DTE的虚电路。QIO (IO\$_ACCESS) 系统服务建立这条虚电路（有关建立虚电路的进一步内容见3.2.2）。

5. 传送数据

现在，可以使用QIO (IO\$_ACCESS) 系统服务，通过这条虚电路向远程的 DTE发送数据（有关传送数据的进一步内容见3.4节）。

6. 清除这条虚电路

当发送完数据后，使用QIO (IO\$_DEACCESS) 系统服务，清除这条虚电路（有关清除虚电路的进一步内容见3.2.3节）。

7. 去配这条通道

最后，程序中必须用\$DASSGN 系统服务，去配它所使用过的这条通道（更多的内容见3.2.4节）。

2.1.2.2 编写处理外来呼叫的程序

一个处理外来呼叫的程序由下列步骤组成：

1. 包含VAX PSI库

所写的任何程序都要包含VAX PSI库。

- 对MACRO，使用下列命令：

```
.LIBRARY "SYS$LIBRARY:PSILIB"
```

并且通过在程序中确定下列行来说明VAX PSI专用的符号。

```
$PSIDEF
```

- 对于高级语言，编写一个含有下列行的MACRO模块：

```
.LBRARY "SYS$LIBRARY:PSILIB"
```

```
$PSIDEF < == >, < :: >
```

汇编这个文件，并将其产生的目标文件与用高级语言所写的程序相连接。

2. 分配一条通道

使用\$ASSIGN系统服务，分配一个到网络邮箱 (SYS\$NET) 的通道，以便由VAX PSI软件接收外来的信息。同时，分配一条到设备NWA0: 的通道（有关分配通道的进一步内容见3.2.1节）。

3. 检查外来的呼叫

VAX PSI利用一个NCB将有关外来呼叫的信息发送给程序。使用 QIO (IO\$_READVBLK) 系统服务读这些数据（细节见3.5节）。NCB 是在邮箱的信息字段中的。由它当中可以发现该呼叫来自何方，它要求什么功能，和我们确定是否接受该呼叫的其它细节。

注意：处理外来的呼叫有许多不同的方法。或者启动一个接收程序，将呼叫交给它，或者将呼叫交到一个已存在的进程。3.3节给出了这些选择的细节。

4. 接收数据

当已经接受该呼叫后，也可以使用 QIO (IO\$_READBLK) 系统服务读外来的数据。

5. 去配该通道

最后，程序中必须用\$DASSGN 系统服务，去配它所使用过的这条通道（更多的内容见3.2.4节）。

注意，用户程序要负责RESETS和INTERRUPTS的认可。由于在任一时刻都可能收到RESETS，所以必须保证我们的应用能做到这一点。有关RESETS接纳认可的进一步内容，参见3.7节，有关INTERRUPTS接纳认可的进一步内容，参见3.6节。

2.2 数据结构

2.2.1 网络连接块

程序必须将有关外出呼叫的信息交给VAX PSI软件。称作网络连接块（NCB）的数据结构就是用来运载这一信息。在程序中定义这个数据结构及其内容，并通过在QIO（IO\$_ACCESS）系统服务的P2参数中给出它的地址来使用它。

使用一个NCB完成虚电路的建立、接受、重定向、拒绝、清除。

要建立虚电路，NCB总要包含下列各项之一：

- 远程DTE地址（对于SVC）
- PVC名字（对于PVC）

另外，NCB可以含有下列选项：

- 远程DTE子地址
- 本地DTE地址
- 本地子地址
- 用户数据
- 诊断编码
- 网络名
- 访问控制信息
- PSDN功能选项：
 - 对方付费
 - 短暂选择（对响应有或没有限制）
 - 流量级别
 - 用户组
 - 包尺寸选择
 - 窗口尺寸选择
 - 本地功能

这些功能的说明参见《P.S.I.引论》，要了解你的包交换数据网所支持的功能参见《公共网络信息》手册。怎样建立一个NCB请看3.2.2节，同时，附录A描述了NCB的结构和项目种类。

当一个外来呼叫到达时，VAX PSI软件构造一个NCB，将信息交到我们的进程（2.3节描述了VAX PSI怎样处理外来的呼叫）。当接受，拒绝，或重定向一个呼叫时，我们可以使用这个NCB，并且，根据呼叫的类型，增加或修改NCB中的某些字段。

2.2.2 邮 箱

可以让所用的每条通道与一个VAX/VMS邮箱建立联系。使用\$CREMBX系统服务建立一个邮箱。在做这之前，用\$ASSIGN系统服务对网络设备分配一条通道。

通道与邮箱相联系是很可取的，因为VAX PSI使用邮箱来运载NCB，而NCB含有我们

需要了解的有关外来呼叫的信息。VAX PSI 也使用这个邮箱来通知我们其它网络事件（例如，当清除一个呼叫时）。如果不使用邮箱，仅能间接地接收这些事件的通知。

有关邮箱更多的内容，参见附录 B，它描述了邮箱的结构和邮箱报文的类型。

当清除一个呼叫时，可以利用 NCB 确定一个诊断码。确定一组具有适当意义的诊断码作为我们自己使用。

2.3 VAX PSI怎样处理外来呼叫

来自另一个DTE的外来呼叫请求所提供的路由信息可能包括：

- 发出呼叫请求的远程DTE地址。
- 该呼叫请求要到达的DTE的本地子地址。
- 封闭用户组（CUG）或双边封闭用户组（BCUG）的名字。主叫DTE和被叫DTE都必须属于这个组。
- 对于特定PSDN的具体用户数据。

当外来呼叫到达时，VAX PSI：

1. 利用外来呼叫请求中的路由和其它信息构造一个网络连接块（NCB）。
2. 将NCB中的路由信息与VAX PSI SERVER数据基中建立的记录相匹配。2.3.1节描述了VAX PSI SERVER数据基。

2.3.1 VAX PSI SERVER数据基

为了对外来呼叫定义适当的目的，系统管理员建立VAX PSI SERVER 数据基。为了接受外来呼叫，VAX PSI必须能够将NCB中的路由信息与X25—SERVER数据基中的目的项相匹配。

有二个VAX PSI SERVER数据基：

- X25—SERVER
- X29—SERVER

在X25—SERVER目的数据基中，有下列几种目的：

- 自宣布X.25网络进程，它将NCB发送到X.25程序。
- 具有目标的X.25目的，它导致VAX PSI创建一个进程来运行一个命令过程。

另外，如果系统管理员将X29—SERVER模块的状态置为ON，那么，对于具有X.29呼叫标识的外来呼叫，VAX—PSI 建立了一个专门的目的，X29__SERVER。与这一目的相关的数据基是第二个数据基，X29—SERVER数据基。

注意：这一目的是X29__SERVER，在X29和 SERVER 之间是一下横线，不是一个连字符。

有关VAX PSI怎样处理具有X.29呼叫标识的外来呼叫，参见《VAX P.S.1, X.29程序员指南》。

程序可以通过发送 \$QIO (IO\$_ACPCONTROL) 系统服务，宣布其自身为一个网络进程。这使得该程序，不通过系统管理员的干预，将目的记录直接进入到了 SERVER 数据基。程序可以对要处理哪些呼叫做出自己的决定。

为了建立一个具有目标的目的，系统管理员必须对该目的，提供一个唯一的字母数字串的名字。

另外, 如果一个目的是与一个目标相联系的, 那么系统管理员要提供一个目标记录。目标记录确定, 当一个外来呼叫匹配该目的时, 要运行的命令过程, 以及完成这一呼叫的用户。该命令过程能执行一个用户程序。

系统管理员可能提供进一步的信息, 如:

- 该目的的优先级 (缺省为零)。
- 对该目的有效的本地DTE子地址的范围。
- 用户组的名字, 该目的能够接受这个用户组的外来呼叫。
- 该目的能够接受外来呼叫的远程DTE地址。
- 呼叫掩码和值。VAX PSI首先用该掩码与外来呼叫中的用户数据进行逻辑与(AND)。

然后VAX PSI将其结果与值相比较。

怎样建立目的细节, 参见《VAX P.S.I.管理指南》。

2.3.2 VAX PSI怎样处理一个外来呼叫

图2—1给出VAX PSI怎样使用X25—SERVER数据基中的目的来处理一个外来呼叫。

对任何外来呼叫, VAX PSI使用呼叫包中给出的远程DTE地址, 以及若指定的话, 本地DTE子地址, 用户组名, 用户数据, 和X.25功能。VAX PSI将这些信息与X25—SERVER数据基中的目的记录相匹配。

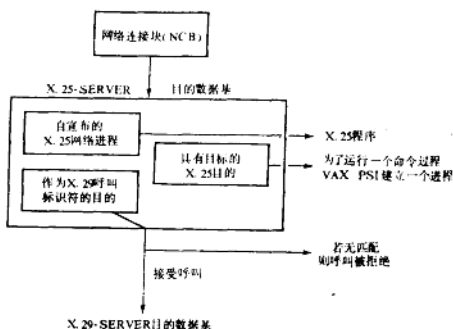


图2—1 VAX PSI对外来呼叫的处理

VAX PSI按下列步骤处理对X25—SERVER数据基中目的的外来呼叫:

- 若搭成一个匹配, VAX PSI则将该外来呼叫与该目的相结合。
- 若搭成多个匹配, 则VAX PSI将该外来呼叫与具有最高优先级的那个目的相结合。
- 若未搭成匹配, VAX PSI拒绝该呼叫。

注意, X29_SERVER的优先级为127, 它位于优先级范围的中间 (由低0 到高255)。因而, VAX PSI首先将外来呼叫与具有较高优先级的X25—SERVER目的相匹配, 即先于X.29目的。一旦搭成匹配, VAX PSI建立一个邮箱, 将NCB放入这个邮箱, 然后通过运行LOGINOUT映象来建立一个进程。LOGINOUT映象检验计帐信息, 如果正确, LOGINOUT,

1. 使SYS\$NET等同于含有NCB的邮箱。

2. 激活LOGIN.COM (若存在)。

3. 起动与该目的相结合的目标所确定的命令文件。

若记录信息不正确,则清除该呼叫。若该呼叫不清除,则含有NCB的邮箱一直存在。

当一个到达的外来呼叫是与X29—SERVER目的相配匹,VAXPSI 接受该呼叫。VAXPSI根据该目的是否指定了目标标识来决定采取什么措施。若没有找到目标标识或没有找到相匹配的目的,则VAX/VMS登录序列启动,要求在远程终端上的用户输入用户名和口令。

2.3.3 用网络进程处理外来呼叫

通过将进程宣布为一个网络进程,我们不使用VAX PSI网络管理,就能决定 哪些外来呼叫我们要处理。

为了将进程宣布为网络进程,发出一个具有功能码IO\$_ACPCONTROL的QIO。这样将为进程建立一个目的地记录。通过进一步的发出具有功能码 IO\$_ACPCONTROL 的\$QIO,我们可以对该进程增加更多的目的。

例如:第一个ACP控制功能可能是选取子地址为21的外来呼叫;而第二个 ACP 控制功能可能是选取子地址为42的外来呼叫。那么现在,该进程就能接受子地址为21或42的所有外来呼叫(配置数据基中将含有两个与该进程相联系的记录)。

一旦我们宣布一个进程作为网络进程以后,VAX PSI 将所有与该目的记录相配匹的外来呼叫放入一个邮箱中。这个邮箱是与发送 ACP 功能的通道相联系的。然后该进程通过发生具有功能码IO\$_READVBLK的QIO系统服务调用,重新由 这个邮箱获得外来呼叫。

去配该通道则由配置数据基中删除该目的信息。

为了显示所有网络进程的目的地记录,使用下列命令:

处理X.25外来呼叫的目的记录

```
NCP>SHOW MODULE X25—SERVER KNOWN DESTINATIONS
```

处理X.29外来呼叫的目的记录。

```
NCP>SHOW MODULE X29—SERVER KNOWN DESTINATIONS
```

2.4 MACRO编程

系统服务宏产生参数列表,CALL指令调用这些系统服务。这些宏是位于系统库SYS\$LIBRARY:STARLET.MLB 中的;当汇编一个源程序时,对未解决的引用自动搜索这个库。VAXPSI的专用符号和宏定义包含在库SYS\$LIBRARY:PSILIB中。在所写的任何MACRO 应用程序中都要含有这个库,来说明PSI的专用符号,如2.1.2节所示。

为了理解这一节所给出的材料,需要知道MACRO汇编语言编程的规则。《VAX MACRO语言参考手册》和《VAX—11MACRO程序设计指南》中含有必要的内容。

2.4.1 参数列表

在第四章的服务说明中,给出了系统服务所要求的参数。对每个系统服务的“宏格式”给出每个参数的位置关系和关键字名。

所有参数都是长字。列表中的一个长字必须在它的低字节中含有该列表中其余参数的数量。它剩下的三个字节必须为0。

在系统服务的宏指令中,如果省略了某个参数选项,则宏将对该参数 提供一个缺省值。

对于调用系统服务的编程有二种通用的宏方式:

\$名字__G

\$名字__S

使用宏的方式依赖于怎样为系统服务构造参数表。

\$名字__G方式, 要求在程序的某个地方构造一个参数列表, 并将这个列表的地址作为一个参数指定给系统服务。(对每个系统服务提供一个建立参数列表的宏。)使用这种方式, 对于该宏的多次调用, 能够使用同一个参数列表, 若需要可以作些修改。

\$名字__S方式, 要求以宏指令方式将参数提供给系统服务。在程序执行期间, 这个宏产生将该参数列表压入调用栈的编码。使用这种方式, 我们可以利用寄存器包含或指向参数, 所以可以写可重入的程序。

\$名字__G宏方式产生一个CALLG指令; \$名字__S宏产生一个CALL指令。所调用的服务, 依赖于标准调用进程的约定。系统服务保存除R0和R1以外的所有寄存器, 在将控制返回到程序之前恢复保存的寄存器。

《VMS系统服务参考手册》描述了怎样使用每种宏方式编写系统服务调用。

2.5 高级语言编程

VAX/VMS 所支持的每种高级语言都提供了调用外部过程以及将参数传递给这个外部过程的机构。然而, 各语言中的机构类型和所使用的术语是不同的。

VMS系统服务是接受参数的外部进程。向系统服务传递参数的方法有三种:

- 通过值。该参数传递的是真正的值(一个代表数值的符号或一个数)。
 - 通过基址。该参数是一个区域或字段的地址, 在这个字段中含有这个值。由基址传递的参数, 通常表示为与某个区域或字段相联系的标号(事实上, 一个公共的错误是在没有指出是值传递的情况下, 传递的是一个数值。若编译程序认为该数值是一个地址, 那么, 以该数值为0或1为例, 当映象访问虚地址0或1时将产生一个运行时间访问错误。)
 - 通过描述符。该参数也是一个地址, 但是一个专门的数据结构, 称其为字符串描述符。
- 2.5.1中说明了一个描述符的格式。

每个系统服务的高级语言格式为:

SYS\$ASSIGN (Devnam, chan, [acmode], [mbxnam])

SYS\$DASSGN (chan)

SYS\$QIO ([efn], chan, func, [iosb], [astadr], [astprm], [p1], [p2], [p3], [p4])

每个参数的说明见第四章。每个服务的说明指出了怎样传递每个参数。短语‘一个地址’和‘字符串描述符’分别表示地址参数和描述符参数。词‘指针’, ‘数’, ‘值’, 或‘掩码’指示是由值来传递的参数。

当系统服务将控制返回到我们的程序时, 它将返回的状态值放入到通用寄存器R0中。在其低字中的值表明或者该服务成功地完成或者由于某些错误使得该服务不能完成某些功能或全部功能。对系统服务每次调用以后, 程序应该负责检验该服务是否成功地完成。也可以测试专门的错误条件。(有关返回状态值的更多内容见2.1.1.1节)。

类似于用MACRO 进行程序设计, 在我们的高级语言程序中也需要使用PS1的专用符

号。为了实现这一目的，先写一个含有VAX PSI库和符号（如2.1.2节所示）的MACRO程序，然后当编译这个高级语言程序时连接这个模块。

2.5.1 描述符

一个字符串描述符是一个四字（8个字节）区，它含有数据串长度和数据开始地址。在大部分情况，编译程序自动产生描述符和数据；在某些情况中，我们需要自己定义全部字段（见所用语言的用户指南）。

《VAX进程调用和条件处理标准》全面地解释了描述符，在《VAX体系结构手册》和《VAX系统例程参考手册》中都有其说明。描述符由下列字段组成：

- 长度。确定ASCII字符数据的数量或该缓冲区的字节个数；该值是放入长字的低字中。在一些情况下，我们可能在程序执行期间想要将某个值移到这个字段。
- 类型。确定该参数的数据类型，系统服务省略这个字节。
- 类别。确定该描述符的类别，系统服务省略这个字节。因此，动态串描述符是作为定长串描述符来处理的。
- 数据地址。以相应于所用语言的方式，给出数据的开始地址。可能必须指定与数据相联系的基地名或标号。
- 数据。若该描述符作为对该服务的输入，指定数据。若该描述符作为由该服务的输出，则分配足够的空间以容纳由该服务返回的数据（该数据不是描述符的一部分）。

第三章 系统服务的使用

3.1 系统服务的使用介绍

使用系统服务:

- 建立和清除通信。
- 处理外来呼叫。

表 3—1 按顺序给出系统服务及其使用说明。

表3—1 系统服务

调 用	用 途
建立和清除通信的系统服务	
\$ ASSIGN	分配一条通道。
\$ QIO (IO\$_ACCESS)	建立一条虚电路。
\$ QIO (IO\$_DEACCESS)	——清除一条虚电路。——接纳清除的认可。
\$ DASSCN	去配这条虚电路。
处理外来呼叫的系统服务	
\$ QIO (IO\$_ACCESS; IO\$_M_ACCEPT)	接受建立一条虚电路的请求。
\$ QIO (IO\$_ACCESS; IO\$_M_AOBR)	拒绝建立一条虚电路的请求。
\$ QIO (IO\$_ACCESS; IO\$_M_REDIRECT)	重定向建立一条虚电路的请求。
\$ QIO (IO\$_WRITEVBLK)	传送数据。
\$ QZO (IO\$_READVBLK)	接收数据。
\$ QIO (IO\$_NETCONTROL)	传送一个中断或复位请求。中断、复位请求、或重起动的接纳认可。
\$ QIO (IO\$_ACPCONTROL)	宣布一个网络进程。

由3.2.1到3.6给出了如何使用每个系统服务的说明。对于永久虚电路, 3.10节含有关于使用这些系统服务的附加内容。

图3—1到3—10给出了系统服务到包的转换。这些图也说明了寄存器0(R0), I/O状态块(IOSB), 和邮箱(当使用时)的内容。

3.2 建立和清除通信

3.2.1 分配一条传输通道

使用\$ ASSIGN调用获得一条通道, 并将一个邮箱与这条通道相联系。

首先明确地分配一条到设备NWA0: 的通道。VAX PSI 软件建立一个新的被调设备NWA_n: , 这里n是一个唯一的部件号, 并分配到这个设备的通道。使用\$ GETDVI找到分配的真正部件号。\$ ASSIGN从不分配一条到NWA0: 的通道, 对每条虚电路总要分配一条这样的通道, 但不能显式地对NWA_n: 设备分配另一条通道。

对于一条永久虚电路(PVC), 类似于对SVC, 分配一条到设备NWA0: 的通道, 并且, 当使用\$ QIO (IO\$_ACCESS)时, 在NCB中确定该PVC的名字。

VAX PSI使用一个邮箱来通知用户程序外来呼叫, 中断, 和其它网络事件的到达(例

如，呼叫被清除，网络失败，远程DTE失败）。

如果我们的程序是处理外来呼叫，必须使用一个邮箱来接收其到达的通知。然后，我们能够使用这个邮箱接收网络事件报文。

如果我们的程序仅是执行外出呼叫，并且，我们不想要网络事件的直接通知，那么我们不需要建立一个邮箱。注意：由于没有邮箱，我们则不能接收中断。

如果我们要求一个邮箱，在分配该通道以前，使用\$CREMBX 系统服务建立一个邮箱。有关\$CREMBX的使用信息见《VMS系统服务手册》。使用\$ASSIGN调用将这个邮箱与该通道联系起来。该邮箱一直与该通道保持联系，直到我们删除该邮箱，或去配该通道。对于邮箱结构和邮箱报文的类型见附录B。

为了方便起见，存在一个运行时间库例程，LIB\$ASN__WTH__MBX，它创建一个临时邮箱，分配一条到它的通道，并分配一条到NWA 0：的通道。这个例程建立一个唯一的邮箱，因而该任务的多个拷贝所使用的邮箱等效于使用不同的邮箱。若我们建立一个具有一个逻辑名的邮箱，所有的任务将使用相同的邮箱，它们的邮箱报文之间相互干扰。对于这个例程的完整说明见《VMS运行时间库例程手册》。

3.2.2 建立一条虚电路

使用\$QIO (IO\$_ACCESS) 操作建立一条到远程 DTE 的虚电路，并有选择地请求网络功能。如果我们订购了短暂选择功能，我们就能使用这个呼叫发送 128 个字节的用户数据而不是 61 个字节。在PVC上发送和接收数之前也使用这个呼叫。

虚电路建立使用网络连接块 (NCB)。对于 NCB 结构及其项目类型的说明见附录 A。在它的描述符中总要确定NCB的长度。

通过进入相应的类型码（括号中给出的），使用NCB来确定下列必须有的信息：

- 对SVC，确定：
 - 对SVC 的远程DTE地址 (PSI\$C__NCB__REMOTE)
 - 网络名——除非已建立缺省或仅能访问一个网络 (PSI\$C__NCB__NETWORK)
- 对PVC，确定PVC名字 (PSI\$C__NCB__PVCNAM)
- 另外，我们能使用NCB确定下列信息选项：
 - 远程DTE子地址 (PSI\$C__NCB__REMSUBADR)
 - 本地子地址 (PSI\$C__NCB__LOCSUBADR)
 - 接收限额 (PSI\$C__NCB__RCV__QUOTA)
 - 用户数据 (PSI\$C__NCB__USERDATA)
 - 对方付费 (PSI\$C__NCB__REVCHG)
 - 短暂选择 (PSI\$C__NCB__FSEL)
 - 具有响应限制的短暂选择 (PSI\$C__NCB__FSEL__RES)
 - 流量级别 (PSI\$C__NCB__THRUCLS)
 - 最小流数级别 (PSI\$C__NCB__MIN__THRNCLS)
 - 用户组 (PSI\$C__NCB__CNG)
 - 包尺寸选择 (PSI\$C__NCB__PKTSIZE)
 - 窗口尺寸选择 (PSI\$C__NCB__WINSIZE)
 - VAX PSI不支持且PSDN支持的本地功能 (PSI\$C__NCB__LOCFAC)

- 目的优先级 (PSI\$C_NCB_PRIDST)
- 重定向优先级 (PSI\$C_NCB_PRIRED)
- 网络用户标识 (PSI\$C_NCB_NET_USER_ID)
- 收费信息请求 (PSI\$C_NCB_CHARGING_INFO)
- 收费信息——货币单位 (PSI\$C_NCB_CHARGE_MON)
- 收费信息——段计数 (PSI\$C_NCB_CHARGING_SEG)
- 收费信息——期限 (PSI\$C_NCB_CHARGE_TIME)
- 访问选择的远程端口 (PSI\$C_NCB_RPOA)
- 被叫线路地址修改的原因 (PSI\$C_NCB_ADDR_MOD_RSN)
- 呼叫重定向——原因 (PSI\$C_NCB_CALL_REDIR_RSN)
- 呼叫重定向——原始DTE (PSI\$C_NCB_CALL_REDIR_ORIG)
- 传输延迟选择 (PSI\$C_NCB_TRANSIT_DELAY)
- 主叫地址扩充 (PSI\$C_NCB_CALLING_EXTENSION)
- 被叫地址扩充 (PSI\$C_NCB_CALLED_EXTENSION)
- 加快数据的协商 (PSI\$C_NCB_EXPEDITE)
- 累积传输延迟 (PSI\$C_NCB_CNM_TRST_DLY)
- 累积端对端传输延迟 (PSI\$C_NCB_CNM_TRST_DLY_R)
- 端对端传输延迟 (PSI\$C_NCB_ETE_TRST_DLY)
- 最大可接受的传输延迟 (PSI\$C_NCB_MAX_TRST_DLY)

根据QIO能够确定某些选项。对每个QIO参见第四章中的相应节。

注意，如果我们是一个双边封闭用户组的成员，仅需要指定CUG字段来标识远程目的。

NCB是一个用户生成的数据结构，它由一个长度可变的项目串组成（类型码放在一个项目的某个字段中）NCB结构及其项目类型的说明见附录A。A.5节给出了一个NCB的例子。对于NCB中必给字段的一个概要见第四章。

NCB中的网络名使得我们能够指定希望使用的PSDN。

图3—1到3—3给出了一个虚电路请求被远程DTE和网络接受和拒绝。

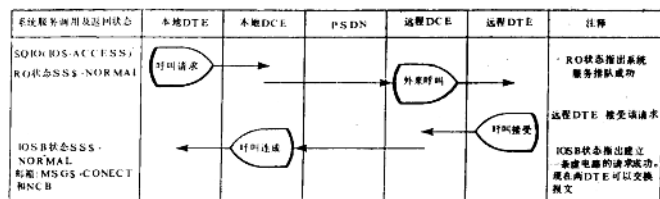


图3—1 建立一条虚电路——呼叫被接受