

VAX/VMS 操作系统 中断系统

国防科技大学研究所六〇二教研室

一九八五年六月

前 言

VAX/VMS 操作系统中的“中断系统”规模较为庞大，功能丰富，结构也较为复杂。VAX/VMS 操作系统本身是一个“中断驱动”的系统，因此“中断系统”是整个系统最核心的内容之一。

我们用较大的精力，在学习，开发 VAX/VMS 的基础上，也研究了 VAX/VMS 操作系统，基本搞清了“中断系统”控制结构及主要功能。现将这一部分中的主要内容，用较大的篇幅整理编写出来。

在这部分材料中，主要介绍 VAX/VMS 的“中断系统”包括：中断处理和异常处理两大部分。在“中断处理”中，主要介绍了各类中断的处理，其中“时钟中断”因为有许多特殊的系统功能，所以将其单独地列为一章详细讨论。在“异常处理”中，也主要是介绍各类异常的处理，同时，也将“系统服务和 RMS 服务的分派”单独地列为一章，这种分派本身是嵌在在“VMS 异常处理机构”中完成的。

文章中的错误，请批评指正。

编 者

1984 年 10 月

目 录

第一章 中断系统概述	1
§ 1.1 系统控制块 SCB 及若干寄存器	2
§ 1.2 中断和异常的启动处理及返回	9
第二章 中断处理	13
§ 2.1 软件中断	16
§ 2.2 硬件中断	24
§ 2.2.1 VAX-11/780 硬件结构	24
§ 2.2.2 单总线设备中断处理	29
§ 2.2.3 多总线设备中断处理	36
§ 2.2.4 DR32 设备中断处理简述	43
§ 2.2.5 MA780 中断处理简述	45
§ 2.2.6 VMS 内部与中断有关的机制	46
第三章 时钟管理及有关的服务	48
§ 3.1 硬时钟	48
§ 3.2 软时钟	51
§ 3.3 硬时钟中断服务例程及有关的 服务	52
§ 3.4 软时钟例程、中断处理及有关的 服务	55
第四章 异常处理	63
§ 4.1 硬件异常	64
§ 4.1.1 普通异常的最初启动处理	64
§ 4.1.2 特殊异常的处理	70
§ 4.2 异常处理程序的行文	72

§ 4.3 异常的分派处理	76
§ 4.3.1 分派查找算法	78
§ 4.3.2 多重异常信号的分派	81
§ 4.3.3 异常处理中解返调用栈	86
§ 4.4 软件异常的处理	87
§ 4.4.1 系统检测的软件异常	88
§ 4.4.2 用户进程发出的软件异常信号	89

第五章 系统服务和RMS服务的分派

§ 5.1 系统服务向量	91
§ 5.2 VAX-11 “改变方式”指令CHMX	92
§ 5.3 系统服务和RMS服务的分派	94
§ 5.3.1 改变方式的服务分派	95
§ 5.3.2 RMS服务的分派	105
§ 5.3.3 在每种方式下执行的服务分派	107
§ 5.3.4 用户写的系统服务及其分派	108

第一章 中断系统概述

VAX/VMS 操作系统是一个“中断驱动”的系统，所谓中断，即处理机中断，指的是：暂停处理机正在执行的程序，转去处理引起暂停的中断事件所指示的工作。VMS 操作系统运行在 VAX-11 系列机上，中断系统与机内硬件有密切的关系。VAX-11 硬件对中断系统提供了强有力的支持，因而 VMS 的中断系统功能很强，规模也很大，是整个操作系统的核心。

根据“中断事件”的不同来源，VMS 将传统的“中断”概念细分为两类：“中断、异常”，简单地说，所谓“中断”指的是：与当前进程运行无关的暂停事件。比如说：系统的时钟中断、外部设备中断、重新调度中断等；而所谓“异常”指的是：与当前进程运行相关的暂停事件。比如：进程因为出现了浮点数溢出而导致程序异常、虚地址转换中，因为缺页而导致的转换无效异常，等。VMS 系统不仅在概念上区分了中断和异常，而且在具体实施处理时，也充分考虑了中断和异常的特点，实施了不同的控制和处理，简单地归纳起来，在 VMS 系统中，中断和异常有以下几点区别：

1. 异常一般是由现行指令的执行引起的，而中断是系统中可能与现行指令无关的活动引起的；
2. 异常的处理一般是在产生该异常的现行进程关联中进行，而中断的处理则在系统关联的中断栈上进行；
3. 系统启动异常处理时，中断优先级（IPL）一般不改变，而启动中断处理时总是提高 IPL；
4. 异常出现时，无论处理机的 IPL 多高，都将立即启动处理，而中断则要在 IPL 降至请求级时，才被启动处理；

5. 中断可以被屏蔽。如果某一中断被屏蔽时，出现了中断信号，虽然该中断不被处理，但中断信号仍被记录在中断寄存器中，等到该屏蔽被取消后，原中断信号可得到处理。而异常则不同，虽然大多数的异常不能被屏蔽处理，但如果某一异常被屏蔽时，发生了这类异常，则异常信号将可能丢失；

6. 在处理中断时，处理机的状态寄存器 (PSL) 中的先前方式域，总是被置为内核方式，而处理异常时，先前方式域则直接表示的是异常所处的方式。

本部分材料将详细介绍 VMS 的中断系统，包括中断处理和异常处理两大部分。在本章概述中，主要介绍中断和异常的最初启动过程，以及中断系统有关的一些数据结构等。

§1.1 系统控制块 (SCB) 及若干寄存器

在 VAX-11/780 机上，中断系统使用的系统控制块 SCB (也称中断向量) 由一个 512 字节的页面组成，其中 32 位为一个逻辑单元。主要存放中断或异常处理程序的入口地址。在操作系统生成时，系统控制块 SCB 是可以动态装配的，其物理地址由内部寄存器 PR\$-SCBB 指出，该寄存器的格式见图 1-1。

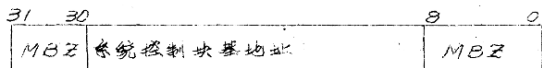


图 1-1 系统控制块基地址寄存器 PR\$-SCBB (只读)

系统控制块基地址寄存器 PR\$-SCBB 由 32 位组成，其中 $\langle 8:0 \rangle$ 位为零，即 SCB 在内存中必须按页面对齐存放。

§ 1.1.1 系统控制块

系统控制块中 32 位的逻辑单元的格式见图 1-2:

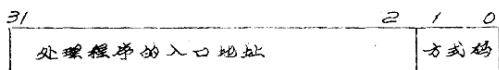


图 1-2 SCB 中每一向量所具有的格式

SCB 中每一个 32 位的向量， $\langle 31:2 \rangle$ 位存放相应的中断或异常处理程序的入口地址（处理程序在内存中以字对齐）， $\langle 1:0 \rangle$ 两位为方式码，在整个 VMS 系统中，相对于进程头联共有四种访问方式，即核方式，执行方式，管理方式，用户方式。每一访问方式在进程头联中，均设有一个对应的方式栈。在系统头联中，另外，设有一个系统栈（也称中断栈）专门用于处理中断事件，SCB 向量中的 $\langle 1:0 \rangle$ 两位表示的方式码，其含义是：

$\langle 1:0 \rangle$ 两位值：

0：表示出现异常和中断时含义：在进程的核方式栈上为该事件服务。若当时已经在中断栈上运行，则无奈在中断栈上为该事件服务；

1：表示在中断栈上为该出现的的事件服务，若该事件是异常，则将 IPL 提高至 31 级；

2：表示该可写控制存储器中为该事件服务，并将向量中的 $\langle 15:2 \rangle$ 位传送入可写控制存储器中的微码；如果未配置该存储器，或者可写控制存储器未被装入相应的微码，则操作是不确定的（打机）；

3：不稳定，VAX-11/780 遇到这种情况时执行打机；特殊地，改变方式指令 CHM%（开管指令）所对应的向量 $\langle 1:$

0> 两位不取从此规定，按指令自身决定的新方式直接选择指令的栈。

VAX-11/780 机上使用的系统控制块 SCB，见图 1-3。表中向量地址从十六进制的 00~50 之间的所有向量，除“电源故障”属于中断外，其余均是异常。十六进制地址 50~1FC 之间的向量均属于中断。图中关于异常的失理在右几章讨论，在图中把异常的中断优先级表示成 32 级，实际上 VAX 系统仅有 0~31 中断，因为异常出现时是无条件被响应的，故此将其视为具有“最高”的中断优先级。有关各异常源、中断源的含义及处理均在以后的各章讨论。

名称	中文名称	类型	中断
00	未用		
04	核心故障	abort/trap	31
06	IO 驱动故障	abort	31
0C	电源故障	中断	20
10	通信/协议故障	Fault	(22)
14	用户协议错误	Fault	*
18	网络/协议层故障	Fault/abort	*
1C	设备非正常启动方式	Fault	*
20	网络访问阻塞	Fault	*
24	设备掉线告警	Fault	*
28	设备告警	Fault	*
2C	新设备启动 SPT 命令	Fault	*
30	设备方式启动	Fault/abort	*
34	输入告警	Trap	*
38	未用		
3C			
40	CHKM 命令	Trap	*
44	CHME 命令	Trap	*
48	CHMS 命令	Trap	*
4C	CHML 命令	Trap	*
50	SBI SILO 故障	中断	25
54	设备故障心跳监测器		26
58	SBI Alert		27
5C	SBI 故障		28
60	特制心跳检测		29
...	未用		
80	未用		
84	未用	软中断	1
88	未用	软中断	2

地址	中断或寄存器	类型	地址
0C	中断向量		
90	IO 驱动处理	软中断	2
94	进入 XDELTA		4
98	IRL0 驱动分叉处理		5
9C	软中断中断		6
A0	IRL0 驱动分叉处理		7
A4	IP0 驱动分叉处理		8
A8	IRL0 驱动分叉处理		10
AC	IRL1 驱动分叉处理		11
B0	未用		...
...	未用		15
C0	前送时钟	中断	24
C4	未用		
...			
F4	控制白场场接收	中断	20
F8	控制色场场接收		20
FC	控制色场场接收		20
100	0号SBI互连软件中断	中断	21
...			...
10C	15号SBI互连软件中断	中断	21
110	0号SBI互连软件中断		21
11C	15号SBI互连软件中断	中断	21
120	0号SBI互连软件中断		21
12C	15号SBI互连软件中断	中断	21
130	0号SBI互连软件中断		21
13C	15号SBI互连软件中断	中断	21
140	0号SBI互连软件中断		21
14C	15号SBI互连软件中断	中断	21
150	0号SBI互连软件中断		21
15C	15号SBI互连软件中断	中断	21
160	0号SBI互连软件中断		21
16C	15号SBI互连软件中断	中断	21
170	0号SBI互连软件中断		21
17C	15号SBI互连软件中断	中断	21
180	0号SBI互连软件中断		21
18C	15号SBI互连软件中断	中断	21
190	0号SBI互连软件中断		21
19C	15号SBI互连软件中断	中断	21
1A0	0号SBI互连软件中断		21
1AC	15号SBI互连软件中断	中断	21

图 1-5 VAX-11/780 系统故障表 SCB

§ 1.1.2 处理机状态字 (PSL)

VAX-11/780 将与过程相联系的若干处理机状态变量集中存放在 32 位的处理机状态字 (PSL) 中。PSL 的第 15 位 ~ 第 0 位可单独称为处理机状态字 (PSW)。PSW 含非特权信息，且 PSW 中的具有特定含义的那些位可由程序自由控制。PSL 的第 31 位 ~ 第 16 位包含着特权状态信息。PSL 的形式与各位的意义如下：

31	30	29	28	27	26	25	24	23	22	21	20	16	15	8	7	6	5	4	3	2	1	0									
CM	TP	MBZ	EPD	IS	CLRM	PREM	MBZ	MBZ	IPL	MBZ	DV	FU	IV	T	N	Z	V	C													
																PSW															

PSL < 3: 0 > 称为条件码，通常反映当前指令执行的结果状态，

N：负号条件码：结果为负数时置 1，否则置零；

Z：零条件码：结果为 0 时置 1，非 0 时清除；

V：溢出条件码：溢出时置 1；

C：进位条件码；

PSL < 7: 4 > 允许自陷的标志位，在特定条件下导致自陷发生，

T：跟踪位：置上该位时，在下条指令执行后便产生一次跟踪自陷；

IV：允许整数溢出自陷位，该位置 1 时，若某指令产生的整数结果不能在所提供的空间中正确表示时，则在执行该指令之后发生整数溢出自陷；

FU：允许浮点溢出自陷位；

DV：允许十进制数溢出自陷位；

PSL < 15: 8 > 没有使用，作为保留位；

PSL <20:16> 为处理机的现行中断优先级 (IPL)。若系统处理机能对某中断作出响应, 该中断的优先级必须高于现行的 IPL。实际上, 所有的软件都在 IPL0 上运行, 这样处理就可以对具有任何优先级的中断请求进行交叉并服务。然而, 对于任一请求的中断服务程序都运行在该请求的 IPL 上, 因此, 暂时封锁了具有较低或同样优先级的中断请求。

PSL <23:22> 为处理机先前方式字段 (PRE)。该字段含有在执行业务最近一次异常时从某一较低的特权方式转变为当前方式时的先前方式字段的值;

PSL <25:24> 为处理机现行方式字段 (CUR), 它确定目前正在执行的程序的特权级别。方式字段的值为前表示的含义是:

- 0: 核方式, 特权最高, 具有执行所有指令的能力
- 1: 执行方式
- 2: 管理方式
- 3: 用户方式, 其特权最低

PSL <26> 为中断栈标志位, 它表示处理机正在使用系统的“中断栈”, 当 IS 位置 1 时, 现行方式总是核方式; 因此, 运行在“中断栈”的软件具有完全的核方式特权。

PSL <27> 为第一部分完成标志位 (FPD), 处理机在执行某些指令时, 使用该标志位, 这些指令是那些在指令执行中的中间点可以被中断或产生页故障的指令。若 FPD 已置位, 当处理机从异常或中断返回时, 它直接从断点处恢复被中断的操作, 而不要再从该指令开始。

PSL <30> 为特权处理跟踪位 (TP)

PSL <31> 为兼容方式位

其余与中断有关的内部寄存器已在有关中断的章节讨论。

§1.2 中断和异常的启动处理及返回

与中断、异常有关的处理程序地址均固定地存放在系统控制块 SCB 的相应单元中，当出现中断、异常时，硬件自动将控制权交给各自对应的处理程序，处理结束后，用特权指令 REI 返回在栈。

§1.2.1 中断、异常的启动处理

系统中出现中断或异常时，硬件自动负责保存部分处理机现场，然后根据系统控制块 SCB 中的向量将控制转给相应的处理程序。这些硬件动作，可以描述为：

{ 关闭中断 } ；

$tmp1 \leftarrow SCB$ [对应的向量单元] ；

若 $tmp1 \langle 1:0 \rangle = 3$ 则 { 打机 } ；

若 $tmp1 \langle 1:0 \rangle = 0$ 且 { 此次出现的是“机心故障”或内
核栈无效异常 }

则 { 打机 } ；

若 $tmp1 \langle 1:0 \rangle \neq 0$ 且 { 此次出现 CHMx 指令异常 }

则 { 打机 } ；

若 $tmp1 \langle 1:0 \rangle = 2$ 则：

{ 若 { 设置可写控制栈入口并进入了微码 } 则 { 进
入可写控制栈入口 }

否则 { 打机 } } ；

若 $PSL \langle IS \rangle = 0$ 则

{ $PSL \langle CURMOD \rangle - SP \leftarrow SP$

* 保存旧 SP 指针。

若 $tmp1 \langle 1:0 \rangle = 1$ 则 $SP \leftarrow ISP$

* 余件成立，取出中断栈 SP，

否则 $SP \leftarrow \text{新方式} - SP$ }

* 新方式栈指针, 做为当前 SP ,

$-(SP) \leftarrow PSL$ * 原 PSL 进栈,

$-(SP) \leftarrow PC$ * 原 PC 进栈,

{ 若有中断或异常的参数, 也压入栈 } * 处理参数也进栈;

$PSL \langle CM, TP, FPD, DV, FIJ, IV, T, N, V, C \rangle \leftarrow 0$

* PSL 相应位清 0;

若 { 中断 } 则 $PSL \langle PRVMOD \rangle \leftarrow 0$

* 出现中断时先若方式为核方式;

否则 $\{ PSL \langle PRVMOD \rangle \leftarrow PSL \langle CURMOD \rangle$

$PSL \langle CURMOD \rangle \leftarrow \text{新方式} \}$

* 除 CHM 异常外, 新方式为核方式

若 { 中断 } 则 $PSL \langle IPL \rangle \leftarrow \text{新 IPL}$

否则

{ 若 $tmp1 \langle 1:0 \rangle = 1$ 则 $PSL \langle IPL \rangle \leftarrow 31$

若 $tmp1 \langle 1:0 \rangle = 1$ 则 $PSL \langle IS \rangle \leftarrow 1$ }

$PC \leftarrow tmp1 \langle 31:2 \rangle \langle 1:0 \rangle$ * 取 SCB 中的入口地址, 该位
字对象。

{ 开放中断 }

从这一硬件动作序列中不难看出, 启动中断和异常的处理,
硬件无非是完成:

- ① 根据向量的 $\langle 1:0 \rangle$ 两位判断合法性, 并选择新的方式栈;
- ② 向新的方式栈压入 PSL 、 PC 以及规定的异常参数;
- ③ 压入新的 PSL
- ④ 由 SCB 向量单元中的入口地址, 转各处理程序;

启动异常, 中断处理时, 方式栈的选择是由硬件自动进行的,
选择的依据是 SCB 向量中的 $\langle 1:0 \rangle$ 两位信息。VMS 系统中共
有四种方式, 对应着四个方式栈, 再加一个系统内部的中断栈。

因此系统中的方式变化，方式栈的变化往往是很复杂的。在启动过程中，方式栈的选择归纳起来有这么几种情况：

- ① 对于大部分的中断启动，均选择中断栈；
- ② 若异常出现在核方式以外的其它方式，且此时又不是 CHMU、CHMS、CHME 指令的异常，则选择核方式栈；
- ③ 若异常出现在核方式且核栈正在使用，则继续选择核栈；
- ④ 若异常出现在核方式，但中断栈正在使用，则无奈选择中断栈。操作系统在中断栈上执行代码时，是不期望会出现异常的。如若真出现了异常则该异常分派程序用宏指令：

`BUGS - INVEXCEPIN`

报告此次出现的非法异常，bugcheck 代码产生一次“VMS 请求的系统崩溃 (Crash)”。

- ⑤ 特殊异常：“机口故障、核栈无效”，它们的处理也不同于一般异常启动处理时，不但将 IPL 提高至 31 级而且还选择中断栈。

§ 1.2.2 中断、异常处理结束时的返回

在 VMS 系统中，中断、异常的结束返回均使用 REI 指令，REI 指令的微操作可以描述为：

$tmp1 \leftarrow (SP)+;$ * 从当前栈中弹出 PC
 $tmp2 \leftarrow (SP)+;$ * 从当前栈中弹出 PSL
 若： $\{ tmp2 < CURMOD > < PSL < CURMOD > \}$ 或
 * 判断 PSL 的合法性

$\{ tmp2 < IS > = 1 \text{ 且 } PSL < IS > = 0 \}$ 或

$\{ tmp2 < IS > = 1 \text{ 且 } tmp2 < CURMOD > \neq 0 \}$ 或

$\{ tmp2 < IS > = 1 \text{ 且 } tmp2 < IPL > = 0 \}$ 或

{ $tmp2 \langle IPL \rangle > 0$ 且 $tmp2 \langle CURMOD \rangle \neq 0$ } 或

{ $tmp2 \langle PRVMOD \rangle < tmp2 \langle CURMOD \rangle$ } 或

{ $tmp2 \langle IPL \rangle \neq PSL \langle IPL \rangle$ } 或

{ $tmp2 \langle PSL - MBZ \rangle \neq 0$ } ;

则 { 发出“保存操作数”异常 } ; * 检查 PSL 的第 21 位是否为 0

若 { $tmp2 \langle CM \rangle = 1$ 且 $tmp2 \langle FPD, IS, DV, FLIV \rangle \neq 0$ } 或

{ $tmp2 \langle CURMOV \rangle \neq 3$ } ;

则 { 发出“保存操作数”异常 } ;

若 $PSL \langle IS \rangle = 1$ 则 { $ISP \leftarrow SP$ } ; * 保存旧的栈指针

否则 $PSL \langle CURMOD \rangle - SP \leftarrow SP$; * 保存旧的栈指针

若 $PSL \langle TP \rangle = 1$ 则 $tmp2 \langle TP \rangle \leftarrow 1$;

$PC \leftarrow tmp1$; * 恢复原 PC, PSL

$PSL \leftarrow tmp2$;

若 $PSL \langle IS \rangle = 0$ 则 { 发出“保存操作数”异常 } ;

{ $SP \leftarrow PSL \langle CURMOD \rangle - SP$ }

若 $PSL \langle CURMOD \rangle \geq ASTLVL$ * 检查 AST 的可交付性

则 { 请求 IPL₂ 级, “AST 交付中断” } ;

{ 检查软件中断 } ;

REI 指令详细的解释参见“VAX Architecture Hand book”。

REI 指令主要功能为, 从栈上恢复中断或异常前的现场 (PSL, PC), 并将控制返回。

另外两条与中断系统有关的指令:

SVPCTX : 保存进程关联;

LDPCTX : 恢复进程关联;

参见“进程调度”部分材料。

第二章 VMS中断系统中的中断处理

VAX-11 处理机有 31 个中断优先级 (IPL), 分为 15 个软件级 (编号 01₁₆~0F₁₆) 和 16 个硬件级 (编号 10₁₆~1F₁₆), 第 0 级为用户级 (用户应用软件, 系统调用和系统服务均运行在 0 级)。VAX 按“先响优先级高的中断”原则响应该并处理中断, 并且优先级高的中断也能打断优先级低的中断响应。

VAX/VMS 的中断源事件包括:

1. 设备结束 (IPL 10~17 十六进制);
2. 设备出错 (IPL 10~17 十六进制);
3. 设备告警 (IPL 10~17 十六进制);
4. 设备存储出错 (IPL 10~17 十六进制);
5. 控制总线端发送或接收 (IPL 12), 十六进制;
6. 间隔时钟 (IPL 18, 十六进制);
7. 已被恢复的存贮口, 总线或处理机错误 (IPL 19-1D, 十六进制);
8. 未被恢复的存贮口, 总线或处理机错误 (IPL 19-1D, 十六进制);
9. 电源失效 (IPL 1E, 十六进制);
10. MTPR #SIRR 指令调用的软件中断 (IPL 01~0F, 十六进制);
11. REI 指令从异常, 中断返回时, 可能请求 IPL 0 级中断 (AST 交付);

中断源事件及相应的中断优先级见图 2-1。

任何中断事件都是由处理机的自动检测的, 然后通过系统控制将控制传给相应的中断服务例程。