

VAX 软件手册

中国科学院沈阳计算所113组译

《小型微型计算机系统》编辑部

译 者 前 言

《VAX软件手册》译自美国DEC公司1982~1983年版的《SOFTWARE HANDBOOK》一书。自从VAX—11系列问世以来，受到广大用户的热切关注。目前国内VAX—11系列用户迫切需要有关资料，我们邀请部分同志将本手册译成中文，以供急需。

参加本资料翻译工作的有（以姓氏笔划为序）：万宝诚、王佳宏、乔贵文、任文岚、刘瑾亮、张玉声、张树田、李彤、金石龙、赵国泰、高庆茂、郭春祥。

赵国泰、武占峰和编辑部部份同志承担了全部资料的校对工作。由于水平有限，错误和不当之处在所难免。切望读者批评指正。

《小型微型计算机系统》编辑部

1984年4月

前 言

我们，DIGITAL，认为您买了我们的计算机，您是正在为未来投资。因此，我们设计的计算机系统不但满足您今天的需要，而且还预见到了明天。系统的设计使您的应用不断扩充。当然，投资是根本。我们以最佳的投资制造我们的计算机。VAX正体现了这种精神。

在这本手册中，我们将会遇到“VAX体系结构”这个术语。为了真实地评价VAX体系和软件是什么样的投资，您需要明白关于体系结构的一些知识——即什么是体系结构和为什么它是如此重要。

VAX体系结构确定VAX处理机如何运行有关软件。这意味着本手册叙述的所有软件可以在VAX处理机系列任何一个成员上运行，从新的VAX—11/730到VAX—11/782双机系统。它也意味着在一个VAX类型处理机上开发运行的所有软件，也可以在其他类型的VAX处理机上运行。因为任何未来的VAX系统将符合同样的体系结构，所以您的软件工程开发得到VAX的保护。

软件需求是体系结构设计的核心。实际上，VAX体系结构和VAX/VMS操作系统是互相渗透一起设计的。我们确信，VAX体系结构提高了VAX/VMS操作系统的效能，并且操作系统利用了VAX处理机的优点。每一个VAX处理机提供32位虚地址寻址、完善的存贮管理和保护机构，以及加速进程调度和协助同步的硬件；上述所有的机构都被VAX/VMS所利用。这样，VAX就可以作为大型和复杂应用的多用户系统，并且它可以同时编译和运行巨型程序（可达4千兆字节）。从分类的观点看，这种能力通常只有大型的、昂贵的主机才具有。它意味着您今天在VAX上所作的投资为明天扩充留有充分的余地。

VAX体系结构也被设计成便于提高程序性能。例如所有VAX语言处理程序都采用了有力的可变长指令集和多种数据类型。其结果是编译程序产生紧凑的和有效的代码，并且执行迅速——因而您的应用程序不但执行得快而且好。

当然还不止VAX体系结构适用于您的投资；它们也融合到软件本身之中。在本手册叙述了我们设计的软件的强大功能。VAX/VMS操作系统易学易用；它提供一个强有力的工具来帮助和革新程序开发。VAX语言处理程序在性能和特性上走向工业化，用一种语言写的程序可以调用用另外一种语言写的过程。VAX信息管理软件为了管理您的数据提供一个前所未有的，完整的系统。网络选件将允许延伸您的应用并且可以选用您需要的任何形式。所有这些，您将看到一个令人满意的效果。

最后，当我们设计VAX体系结构和VAX/VMS软件时，我们不仅为未来负责，而且我们认识到我们许诺是针对成千的已经向我们投资的PDP—11顾客；为此，我们设计成VAX与PDP—11兼容。您在PDP—11上的投资受到保护，因为VAX给它提供了扩充的余地。即使您还没有PDP—11机，而当您把投资放在我们的机器上，就会了解它的精美之处，我们坚持暂作您的投资后盾。

第一部分

绪 论

目 录

第一部份 绪 论

第一章 VAX软件简介

系统介绍.....	1
用户进程.....	2
虚存和存贮管理.....	2
交换和调度.....	4
系统进程和系统服务.....	4
进程间通讯和同步.....	6
输入/输出	6
实时环境.....	7
I/O驱动程序.....	8
通讯服务.....	9

程序设计语言.....	9
程序开发工具.....	9
数据和文件管理实用程序.....	12
系统管理实用程序.....	14

第二章 系统用户

引言.....	17
进入系统.....	17
文件.....	19
逻辑名.....	22
程序开发.....	24

第二部分 程序开发

第三章 DIGITAL命令语言(DCL)

DCL简介	29
命令格式.....	29
关于语言命名命令的约定.....	31
命令过程.....	32
终端功能键.....	32
命令.....	33

第四章 程序设计支撑设施

引言.....	54
DEC标准编辑程序(EDT).....	55
交互文本编辑程序(SOS)	61
面向批处理的文本编辑程序(SLP)	64
连接程序.....	67
VAX查错程序(DEBUG)	70
VAX运行系统库	75

VAX SORT/MERGE.....	79
文件格式实用程序(DSR)	82
代码管理系统选件.....	85

第五章 程序设计语言

引言.....	87
VAX公用语言环境.....	87
VAX-11 BASIC.....	89
VAX-11 COBOL	97
VAX-11 FORTRAN	105
VAX-11 PASCAL	112
VAX-11 PL/1	114
VAX-11 C.....	115
VAX-11 BLISS-32	118
VAX-11 BLISS-16	126
VAX-11 CORAL66	128
VAX-11 DSM	129

VAX—11 MACRO	132
PDP—11 FORTRAN W/VAX	
TO RSX.....	134
MACRO—11	135

第六章 信息管理

引言.....	137
VAX信息系体结构	137
VAX—11 DATATRIEVE ...	140
VAX FMS.....	146
VAX—11 公共数据字典(CDD)	148
VAX—11 RMS.....	150
VAX—11 DBMS	150

第三部份 VAX/VMS 系统设计和应用

第八章 虚存和存贮管理

引言.....	179
虚存.....	179
进程.....	183
进程控制结构.....	184
映象.....	185
调页.....	186
进程间共享页.....	189
交换.....	191
在系统空间调页.....	191

第九章 进程调度和交换

引言.....	192
调度.....	192
交换.....	197

第十章 特殊事件管理

引言.....	201
条件处理程序.....	201
致命错误及系统瘫痪.....	202
出口管理.....	204
异步系统自陷.....	204

第七章 数据通讯

引言.....	155
DIGITAL网络体系结构.....	157
DECNET通讯软件	157
DEC—VAX阶段 III 通讯软件.....	158
DIGITAL 命令语言文件管理.....	162
记录管理服务文件管理.....	163
任务间通讯实例.....	166
任务信息.....	169
程序设计过程.....	170
网间产品.....	172
包交换网 (PACKET NET) 产品	175

第十一章 系统服务

引言.....	208
事件标志服务.....	209
异步系统自陷服务.....	212
逻辑名服务.....	214
输入／输出服务.....	216
进程控制服务.....	222
记时或和时间转换服务.....	227
条件管理服务.....	229
存贮管理服务.....	230
改变状态服务.....	232
锁管理服务.....	233

第十二章 输入／输出服务

引言.....	235
程序设计接口.....	235
辅助控制进程.....	237
设备驱动程序.....	237
处理I/O请求	237
I/O 队列.....	238
I/O 完成.....	239
记录管理服务.....	240

RMS文件组织	240
RMS记录存取方式	242
文件和记录属性	244
RMS实用程序	249
使用VAX—11 RMS	250

第十三章 I/O驱动程序

引言.....	253
设备驱动程序组成.....	253
FORK进程.....	255
一般设备活动.....	256
行式打印机QIO请求实例.....	259

第十四章 进程间通讯

第四部份 配置考虑

引言.....	274
公共事件标志.....	274
信箱.....	275
DECNET	276
全局段.....	276
锁管理服务.....	277
共享磁盘文件.....	278

第十五章 PDP—11兼容性

引言..... 279

实现考虑..... 280

第十六章 系统管理员

引言.....	287
系统开工和运行.....	287
用户环境测试包.....	288
帐目系统.....	288
管理公共文件和卷宗.....	291
系统的综合控制.....	293
监督系统活动.....	294
发觉和处理错误.....	295

第十七章 双处理机系统

双处理机系统.....	297
软件.....	298
程序设计考虑.....	300

附录A 公用助记符..... 302

术语解释 306

第一章 VAX软件简介

本章摘要：这一章是VAX软件述评，包括服务、控制、和能力。在每一个段落中，对VAX/VMS某一专门问题都给出相当精辟的简要概述，在后面的各章节给出更详细的解释。

题目包括：

- 系统介绍
- 虚存与实存的管理
- 进程的定义
- 调度和交换
- 系统服务、I/O控制和I/O设备
- 进程间通讯
- 通讯和网际通讯
- 实时能力
- 语言和语言处理程序
- 数据管理设施

系 统 介 绍

VAX是一族高性能多道程序设计计算机系统，它具有32位的体系结构，完善的存贮管理，以及一个程序地址空间实质上不受限制的虚存操作系统。

体系结构的可变长指令集和多种数据类型，包括十进制数和字符串，提供高效的位功能。该指令系统专门地实现许多高级语言的结构成份（constructs）和操作系统的某些功能。

VAX系列的每个机型都是多用户系统，既可以开发程序，又可以执行应用系统。它是一个优先级调度、事件驱动系统：即赋予的优先级和在系统中进程的活动决定了它们需要服务的级别。实时作业根据它们优先级和执行能力接受服务，而对于一般的执行进程由系统管理CPU时间的分配和存贮器的驻留。

VAX系统具有高可靠性。硬件和软件的固有保护装置确保了数据的可用性。联机诊断、测定错误和记录错误用来验证系统的完整性。倘若电源、硬件、或者软件发生故障，硬件和软件的许多特性能提供快速的诊断和自动恢复。

系统不但具有灵活性并且有可扩充性。虚存操作系统使程序员能够写大型程序，而该程序在小的或大的内存配置上均可运行，毋须程序员去定义复盖或者修改程序，以便于扩充的内存。DIGITAL命令语言使得用户能够修改或容易扩充他们的命令功能，并且允许用户提供用户自己的命令接口。

为了理解本手册上描述的操作系统功能，解释几个基本术语是值得的。用户必须首先明白程序映象（Image）和进程的概念，并且知道它们之间的区别。请注意，在这一章介绍的几乎所有的概念和特点在后面的段落和章节中都有更详细地论述。

用 户 进 程

一个程序映象是把源语言模块翻译成目标模块，再把这些目标模块连接在一起所产生的一个可执行程序。映象通常以文件形式存贮在磁盘上。当一个用户运行一个映象时，操作系统把映象文件的一份拷贝读到物理存贮器中，以便执行它。

一个过程是实现解决一个问题的逻辑描述；即，它是一个算法的静态定义。一个映象包括许多由连接程序把它们组织在一起的过程和数据。连接指的是各模块交叉链接问题以及虚地址分配问题。

一个映象执行的环境 (Environment) 是它的上下文 (context)。一个映象的完整环境 (context) 不仅包括在任何时刻的执行状态 (通常称作硬件环境)，也包括确定它的资源分配的优先级和份额，诸如设备所有权，文件存取，以及最大的实存分配。某些软件信息其中包括某些关键地址和某些在后面叙述的软件数据结构，构成软件的环境 (context)。一个映象环境和该映象在这个环境上的执行叫做进程。

工作集 (Working Set)

当一个进程执行时，仅它的页面的一个子集需要在物理存贮器。(一个页面内含512个字节，它既是一个内存物理页面的大小，也是磁盘盘区的大小)。这个页面子集叫做进程的工作集。余下的进程页面驻留在辅助存贮器。在一个进程竞争中央处理机资源之前，它的工作集必须驻留在内存。

平衡集 (Balance Set)

在物理存贮器驻留的一组进程称作平衡集。这组进程对内存的需求与该系统可用的内存相平衡。在一个进程执行期间的任何时刻，它的整个工作集可以写到辅助存贮器，腾出的内存可以为其它用户使用。这称作交换 (Swapping)。

软件进程控制 (Software Process Control)

VAX/VMS操作系统给每个进程提供了软件定义以便控制该进程，并且提供进程工作集。该操作系统为定义一个进程提供两个关键的数据结构：软件进程控制块 (PCB) 和进程首部 (Process header)。通过进程标识 (Process Identification)，系统为每一个进程提供一个唯一的标识符。

虚 存

VAX/VMS虚地址空间包括 2^{32} 个字节，分为系统和进程地址空间，每一个空间各有 2^{31} 个字节。VAX/VMS系统认为一个进程所需要的物理内存和该进程定义的虚地址空间是不同的。进程的虚地址空间是该进程可以寻址的内存单元的范围。

进程虚地址空间分为程序区和控制区。程序区包括当前正执行的映象。控制区包含系统为了控制进程所需要的信息，以及用户栈，核心态、执行态、管态的栈。仅控制区的一小部份用来保存系统控制的环境，余下的为用户用。

- 读或者修改数据
- 按有意义的形式产生输出

高级语言程序员通常使用它们特殊语言的I/O设施完成记录和文件操作。这些操作正是使用VAX—11—RMS设施实现的。VAX—11 MACRO程序员在他们的程序内部可以直接使用VAX—11 RMS例程。

VAX—11 RMS例程是这个操作系统的必备部分。对于这些程序，程序员既不需要执行任何特殊的连接也不需要说明这个例程的全面入口点。而且，调用VAX—11 RMS程序与VAX的调用标准一致。

VAX 高级语言

VAX-11 BASIC

VAX—11 BASIC产品对VAX用户给出一种高度交互地程序设计环境和一种高性能的开发语言。它把完全与VAX环境结合为一体的VAX语言所提供的良好性能与PDP—11 BASIC—PLUS—2和RSTS/E BASIC—PLUS语言的特点结合起来。

VAX—11 BASIC语言是一种高级的可扩充的工具语言。它提供了强有力的数学和字符串处理设施，支撑符号字符，以及RMS的全部索引、顺序和相对的I/O操作。

把VAX—11 BASIC可以看作一种解释程序，也可以看作一种编译程序。快速的RUN命令和对于无标号语句的直接执行（直接方式）的支撑，给VAX—11 BASIC用户一种解释程序的“感觉”，然而，本产品也可用于编译方式，它与其他VAX编译程序一样产生本机方式的目标模块。在两种情况下，VAX—11 BASIC系统都产生执行时间极快的VAX本机方式指令。典型编译速度是每分钟3000行，执行同一程序通常要比PDP—11系统快五倍。

以下为VAX—11 BASIC语言一般概述。

一般特性

VAX—11 BASIC系统在RUN或编译方式下按行产生本机的VAX指令。产生的代码利用了VAX/VMS 本机方式的优点，包括：

- 直接调用操作系统的服务例程，甚至在立即方式中也可以直接调用。
- 对于DEC网络通信软件的访问是透明的。
- 直接调用公共运行库和标准的系统实用程序，包括VAX—11 SORT/MERGE。
- 采用VAX过程调用标准直接调用各别编译用任何语言书写的本机方式过程。
- 程序的尺寸可达到2兆字节。
- 所有的模块都是与位置无关的（PIC），并且可作为完全可重入代码运行。
- VAX符号排错程序完全支撑VAX—11 BASIC语言

VAX—11 BASIC系统产生的代码使用标准的VAX/VMS追踪设施确定运行时的错误根源。如果出现一个致命性程序错误，则打印出标明产生错误的模块号和行号的英文信息。英文正文、追踪、以及相结合的BASIC的HELP实用程序等提供了一个强有力的程序排错（DEBUG）环境。

工作集交换程序

工作集交换程序是一个小进程,它负责把进程工作集移进和移出平衡集。工作集交换程序提供内存,因而它们可以被调度执行。

工作集交换出现在两个阶段:

- 把不可执行的程序或者低优先级的进程从平衡集中交换出去,为交换进来的进程腾出内存
- 交换进来的进程从可执行的非驻留状态变成可执行的驻留状态

工作集交换程序也执行初始进程建立。由于进程建立是利用交换到内存的外壳(Shell)进程实现的,所以进程建立需要交换程序一点额外的工作。外壳(Shell)进程建立新进程的初始环境(context)和虚存。

进 程 调 度

VAX/VMS操作系统为调度进程定义了32个软件优先级。低的16个优先级(0到15)用于一般进程,高的16个优先级(16到31)留给实时进程。总是挑选最高优先级的可执行驻留进程来执行。实时进程的优先级是由用户建立的,并且不由系统改变。而一般进程的优先级可以由系统改变以期达到最佳响应。

进程调度程序通过

- 维护一个进程可以达到的每一种状态的队列
- 对系统事件作出反应

作出调度决策。

系统事件的出现引起系统中一个或多个进程改变状态。调度程序利用把进程的软进程控制块(PCB)从一个状态队列移到另外一个适当队列的办法反映出这种改变。

系 统 进 程

VAX/VMS系统的所有功能或是通过进程或是通过过程实现的,而这些进程或过程又可由用户进程或者由许多系统进程调用。一个系统进程可以是下面三种类型之一:

- 完全进程(Full Process)
- 小进程(Small Process)
- 分叉进程(Fork Process)

完全进程是用户进程。

小进程在它的虚地址空间中根本就没有程序区,然而有一个压缩的环境。它们像完全进程一样的方式被调度,但是必须驻留在内存。例如,工作集交换程序就是一个小进程。

分叉进程有一个最低限度的环境;它们由一个称为分叉块的压缩控制块所定义。分叉进程在软件中断级执行,并且立即被分派执行。分叉进程一直执行到它们被较高优先级的分叉进程所抢占或它们执行完。设备驱动例程就是分叉进程的一个例子。

系 统 服 务

系统服务是一些可以被用户进程调用的执行程序中的过程，它提供了系统资源的可控共享。由于系统以用户名义执行服务，所以控制需要存取特权数据基的那些功能。

仅当请求的进程有足够的特权，并且不违反保护规则时，系统服务请求才兑现。

事件标志服务

相关事件系统服务是这样一些服务，它允许一个进程或一组协同进程 (Cooperating Processes) 读、等待、控制事件标志。每个进程的软件进程控制块(PCB)包括两簇(Clusters) 每簇32个事件标志，这些事件标志都局部于这个进程。此外，诸协同进程组可以建立而且联结着两个附加的事件标志簇。这些簇为同一组号的全部有关进程共用。

锁管理服务

锁管理服务为协同进程同步存取公共资源提供了一个机构。它有六把锁，每一把锁提供一个不同级的共享。资源被定义成树形结构名字表。在名字表中所采用的层次的深度决定了在确定和存取该资源的粒度 (granularity) 的级别。

异步系统自陷 (AST) 服务

通过事件 (譬如I/O完成) 可以中断进程的执行，以便去执行选定的子例程。由于它们异步于进程的执行，所以称作异步系统自陷 (AST)。由于提供系统服务，所以进程可以控制AST的管理。

逻辑名服务

逻辑名服务为管理和存取字符串逻辑名和等价名对提供一个综合的技术。逻辑名为系统和应用程序的输入和输出操作提供设备独立性。为了把一个应用程序从一个单CPU系统移植到多CPU系统，逻辑名重新赋值是极方便、极灵活的设施。

I/O系统服务

I/O服务直接执行输入和输出操作，不通过由VAX/VMS记录管理服务 (RMS) 提供的文件管理。I/O服务：

- 执行物理、逻辑和虚输入/输出操作
- 格式化输出行，把二进制数值转换成ASCII字符串，以及用ASCII字符串置换变量数据
- 执行网络操作
- 排队消息到系统进程
- 建立信箱，它是进程间通讯的虚设备

进程控制服务

进程控制系统服务允许用户建立、删除和控制进程的执行。

记时器和时间转换服务

记时器服务为特定时刻或者已经过了一段规定的时间间隔之后调度事件程序。时间转换服务提供设置、取和格式化二进制时间值的方法供记时器服务使用。

条件管理服务

条件管理程序是一些过程，当一个硬件或软件条件在映象执行期间出现时，这些过程可被选来接受控制。条件管理服务为专门目的选定条件管理程序。

存贮管理服务

存贮管理系统服务使一个进程能控制虚存和实存的使用。其服务包括：

- 允许一个映象增加或减少可用的虚存数量
- 控制虚存的调页和交换
- 建立和存取包含共享代码和数据的存贮器文件

改态服务

改态服务把一个进程的存取态改成更高的特权态去执行一个特别的例程。使用这些服务需要特权。

进程间通讯和同步

VAX/VMS操作系统为进程间彼此通讯和同步它们的执行提供各种各样的方法。为进程间通讯所选的方法受到多方面因素的影响，它们包括：进程之间直接的协同的层次，通讯的效率，以及在网络环境中的灵活性。

进程间通讯可以通过使用下面的方法达到：

1. 利用共享数据基隐含通讯。这种方法是最有效的，但是要求进程间直接互通。
2. 利用信箱或DECnet一般化通讯。信箱是一个虚设备，进程可以对那个设备发送消息和从那个设备读消息。可以在一个单节点或多节点环境把DECnet用于进程间通讯。然而，这种方法开销最大。
3. 共享文件

进程间同步的方法之一是利用公共事件标志簇。每一簇包括32个事件标志，一个进程可以等待同一组的其他进程置一个事件标志，置这个标志表明后者执行了一个正是前者等待的功能。一个进程可以最多联结两个公共事件标志簇。

同步的另外一个方法是使用锁管理服务。通过排队锁请求，互通进程可以同步资源的存取。有六把锁，每一把都提供一个不同级的存取共享。

VAX/VMS 输入和输出

I/O处理系统由几个相互关联的成份组成，它使程序员能够选择合适的程序设计接口和处理方法。I/O请求处理软件利用硬件能使I/O传输与计算重叠、快速环境(Context)转换以及在多重优先级中产生中断等优点，确保尽可能高的数据吞吐量和快速中断响应。

I/O接口

I/O程序设计接口是：记录管理服务（RMS），用于通用文件和记录的处理；以及系统服务，用于直接I/O处理。用户可以通过高级语言语句，如OPEN，CLOSE，GET和PUT或在VAX—11 MACRO汇编程序中用CALL语句引用RMS过程。也利用CALL语句引用I/O系统服务。

RMS过程具有设备独立性，并且对所有外部设备提供文件结构存取，不管它是网络上的还是远程的设备。最通用的存取是使程序能够处理逻辑记录，而RMS能自动提供逻辑记录结块和解块的能力。RMS用户亦可在文件结构卷上，如磁盘、磁带上实现它们自己的记录结块，或控制缓冲区的分配或优选专门记录的处理。

I/O系统服务不但提供与设备无关程序设计，也提供与设备有关程序设计。用户在文件结构和非文件结构设备上执行它们自己的记录结块。在文件结构卷上，不但虚块寻址而且逻辑块寻址都是可能的，虽然后者需要有私人卷宗的特权或拥有权。具有足够优先级的用户可以利用逻辑块寻址执行直接I/O操作，用于定义他们自己的文件结构和在磁盘上、磁带设备上的存取方法。

RMS和I/O控制进程——叫做辅助控制进程——处理结构文件的I/O请求。ACP为特定类型设备提供文件结构和卷宗存取控制。在这个系统上有三种类型的ACP：磁盘、磁带和网络通讯链路。

I/O请求处理

通过排队I/O（QIO）请求系统服务产生所有I/O请求。如果一个程序调用RMS过程，则RMS就以该程序的名义调用QIO系统服务。排队I/O请求处理是极快的，其原因在于：

- 通过执行初始化请求和事后请求最少代码，使设备装置利用率最佳
- I/O传输的同时进行重叠查询，使磁盘控制器利用率最佳

处理机的多中断优先级提高了中断响应，因为它们使软件有最少数量的代码在高优先级执行，而用低优先级代码管理请求的检验和完成后的通知。

VAX/VMS 实时环境

由于VAX硬件和VAX/VMS软件是一起开发的，它确保了一个高级实时多任务计算系统。如果要执行实时任务，VAX系统的如下的一些固有的系统属性使得VAX系统成为强有力的实时应用系统。

- 为高效的进程调度程序提供16个实时进程优先级
- 进程环境快速切换
- 快速硬件中断处理
- 中断转向VAX/VMS设备驱动程序
- VAX/VMS操作系统支持PDP—11系统设备和设施，使顾客能够为他们原有设备增添支持范围
- 提供变换I/O页面并连结设备中断向量的易用设施

由于实时应用要求快速响应，所以把应用与操作系统的内核直接接口是很重要的。图1—1

说明了VAX/VMS操作系统的层次。

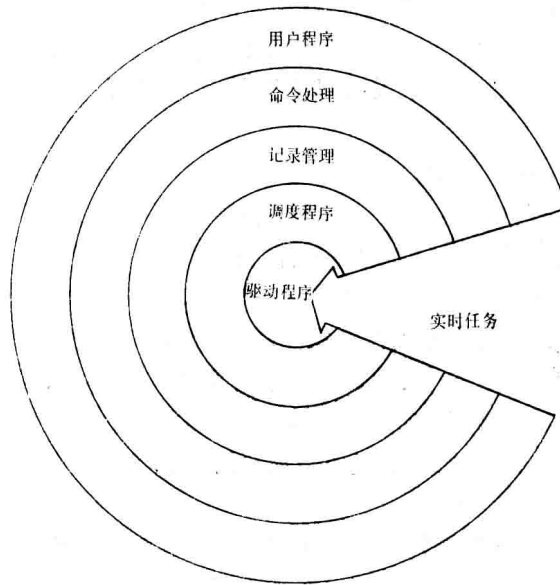


图1—1 VAX/VMS 操作系统

VAX/VMS操作系统的外层是比较成熟的、通用的，这就保证了使用容易、功能强。这些层由命令过程，记录管理服务，用户程序等组成。最深层构成上述的实时系统。

I/O 驱动程序

一个VAX/VMS驱动程序是由一组表和例程组成，它们控制与VAX系统接口的外部设备的I/O操作。一个驱动程序包括：

- 定义VAX/VMS操作系统之外的外部设备
- 为系统过程定义驱动程序，该系统过程把驱动程序和它的设备数据基变换并装入到系统虚存
- 在系统开工时和电源故障之后，启动设备（和／或它的控制器）
- 把I/O操作的软件请求翻译成具体设备的命令
- 激活设备
- 响应由设备产生的硬件中断
- 报告设备错
- 把从该设备得来的数据和状态回送到用户

如果一个I/O操作的细节需要翻译成由专门类型的设备才认识的术语，操作系统才转移控制设备驱动程序。这种处理称作与设备有关的处理。因为不同类型的外部设备希望有不同的命令和装置，所以在VAX系统每一个设备类型都需要有它自己支撑的驱动程序。该设备驱动程序执行所有与设备有关的处理。除了由DIGITAL软件支持的范围很广的外部设备之外，用户也可以开发专用设备驱动程序。

通 讯 服 务

DECnet是DIGITAL软件产品、协议、接口、以及支撑服务的一个系列。它支持把DIGITAL计算机系统连接到分布式处理网络。VAX/VMS操作系统对使用单一VAX系统和使用DECnet/VAX通讯软件提供同样的接口。把DECnet/VAX这套软件添加到VAX/VMS, 在保存了这些接口的同时使系统之间能够通讯。因此, 用户应用可以从单一VAX系统扩展到多重节点网络, 并为已存在的网络所承认, 而不必重写应用程序。网络对应用程序员是透明的。事实上, 应用程序员可以把网络计算机当作一个公共的资源。

使用DECnet通讯软件, 可以构造各种不同类型的计算机网络, 使远程通讯, 资源共享, 和分布计算更容易。DIGITAL网络系统结构(DNA)是公共网络结构, 根据它构造所有DECnet软件产品。DECnet通讯软件是高度模块化的, 具有高度灵活性, 能适应广泛的应用领域。

DIGITAL的网间系列软件包括与其他厂家生产的计算机进行批处理和交互通讯的软件产品。VAX系统的网际产品模拟IBM和CDC主机支持和承认的通讯协议。由于为VAX计算机增添了操作环境的品种和数量, 这些共有的特性又为VAX系统增加了灵活性。

程 序 设 计 语 言

VAX/VMS操作系统支持许多主要的语言, 包括FORTRAN, COBOL, BASIC, 和PL/I语言。在维护编译程序竞争性和执行性能的前提下, 对编译程序提出增加工业标准的要求。

应用程序不局限于一种单一的语言: 如必要, 可能是几种语言的组合, 以便最有效地完成计算机作业。由于一些语言可以调用一个其他的语言, 所以程序员在一个应用程序中可以容易地融合一种以上的语言。这意味着: 用哪一种特殊语言可以最有效完成哪一个例程就用哪个语言写, 然后再按需要归并程序。

VAX/VMS操作系统包含下面可用的VAX语言:

VAX-11 BASIC	VAX-11 BLISS-32
VAX-11 COBOL	VAX-11 BLISS-16
VAX-11 FORTRAN	VAX-11 CORAL66
VAX-11 PASCAL	VAX-11 DMS
VAX-11 PL/I	VAX-11 MACRO (汇编)
VAX-11 C	

此外, 还有主机开发方式程序设计环境, 它包括支持PDP-11 FORTRAN IV/VAX TO RSX, 以及MACRO-11语言处理程序。这些语言处理程序产生兼容方式目标代码, 使得对许多类型的应用来说, 一台VAX计算机“看上去好像”一台PDP-11计算机。

VAX 程序开发工具

VAX程序开发工具包括文本编辑程序、编译程序、库、连接程序以及VAX符号查错

程序 (DEBUG)。也包括 PATCH、ANALYZE、MESSAGE、以及 MAIL 等实用程序。所有程序开发的实用程序包括编译程序和 DEBUG，既可以交互方式也可以在批处理方式下使用。

库被广泛地用来构成可执行的程序映象。在本机方式程序设计环境中，程序员可以建立汇编宏定义库，目标模块库、以及可共享映象库、该系统也包含公共运行系统库，这个库对所有 VAX 程序设计语言提供公共库功能

所有程序对操作系统和它的实用程序的接口有完全一致的调用标准。系统程序员可以把新的库过程加到运行系统库，联机安装它们，不必修改已存在的程序和实用程序，因为所有变量用标准数据结构传递。

通过系统服务和命令语言逻辑命名设施把用户程序写成完全与设备无关。所有文件和设备可以用任意定义的逻辑名来标识，逻辑名可以在运行时赋值。

除编辑程序和 MAIL 实用程序外，其他程序开发实用程序均不适用于宿主开发环境。许多实用程序在第 4 章有更详细的叙述。

编辑程序

有三个文本编辑程序：EDT、SOS 和 SLP，程序员可以使用任何一个或全部。EDT (DIGITAL 标准编辑程序) 是一个交互编辑程序，利用从硬拷贝或显示终端敲入的命令，程序员可以建立和修改文本文件。它提供有效的和功能极强的字符、字、行和缓冲区编辑能力。此外，EDT 对 VT100 和 VT52 终端用户支持屏幕编辑。一个窗口通向文本，在这个窗口可以使用插入、删去、和再定位命令，以及把整个文本缓冲区 (编辑文件) 移到另外一个缓冲区的能力使它成为一个很有吸引力的工具。可以编写编辑过程 (宏和程序) 以便在编辑对话期间建立一个特殊的环境。检查跟踪文件保护对话，避免偶然地损失。

SOS 也是一个交互文本编辑程序。用户可以插入、删除、和置换某些行，搜索和置换字符串，或者一次修改文本的一个字符。行可以用行号，相对位置，或者上下文来识别。一组邻接的行可以从一个地方复制和传输到另外一个地方。编辑可以在一个文件内以任意次序完成。编辑程序参数可以置成用户规定的值，并且可以显示当前值。用户规定的参数可以在编辑程序开工时自动设置。

SLP 是一个程序文本编辑程序，它使一个用户藉助包含修改命令的文件能够修改一个存在的文件。命令文件为以后或者为另外一个系统提供一个可靠的办法重复修改一个文件。SLP 在源文件和检查跟踪 (audit trail) 列表中，提供修改文件的格式记录。在大型程序设计工程的追踪阶段，常使用这个特性。

连接程序

VAX/VMS 连接程序接受一个或多个由汇编或编译程序产生的基本目标模块，解决在它们之间引用的符号和过程，分配虚存，并且产生一个可执行的程序映象。

VAX/VMS 连接程序与众不同之处在于它可以使程序员建立一个可共享的映象，可共享映象还可以与其他模块连接。由于可共享映象在运行时由映象激活程序分配虚存，它们在程序开发中发挥惊人的经济效益；可以直接修改可共享映象，不必与使用它的所有程序重新连接。

连接程序不仅接受目标模块和可共享映象作为输入，而且接受目标模块和可共享映象库