

一种新型的多任务执行程序 AMX

高存宝 贾新 陈大为 (北方交通大学自动化所 100044)

摘要:本文介绍了一种基于DOS操作系统的新一代实时多任务系统—AMX。阐述了它的基本特征、一般操作及其软件开发过程。

关键词:AMX 实时多任务 DOS

一、AMX 概述

DOS虽然是当今较为流行的一种操作系统,在世界上拥有很多的用户,但它却是一个单任务操作系统,不提供多任务的机制,这就给实时多任务应用系统的开发人员带来了一个难题。为了解决这一难题,KADAK公司开发了一种基于DOS操作系统的多任务执行程序—AMX。它提供了对复杂实时多任务的简单解决办法。AMX管理一组应用程序模块(称为任务)的顺序执行。AMX是通过一组程序实现多任务机制,它与汇编语言,C语言完全兼容,对于汇编它提供了一组AAxxxx的过程,而对于C语言,它提供了丰富的库函数,以头文件“AM831CF.H”的形式出现,因此它使系统设计者只需关注于使用编程语言来设计应用程序而不必过多地考虑多任务的特性。

AMX在其最小形式时仅需不到6K字节的程序存储区(ROM)及1K字节的数据存储区(RAM),这比起一般的实时多任务操作系统来说是大大节省了内存,这也是其最大的优点之一。

二、AMX 的一般操作和系统结构

每个基于AMX的系统都由AMX执行程序 and 一组应用任务以及中断服务过程组成,这些程序驻留于微机内存,构成整个系统。

图1表示了AMX系统的一般操作过程。执行开始在于用户区,以便在AMX系统初始化前,进行硬件配置及建立起与应用有关的配置。

一旦所有用户初始化完成,程序调用AMX入口程序ajentr。AMX系统中有一特殊的系统配置模块(由AMX根据用户编辑的参数文件生成),在其内定义的操作参数可预先定义特定任务和定时器。这些任务和定时

器由AMX根据系统配置模块中的定义在初始化时自动创建。AMX初始化其本身并置应用任务和定时器处于空闲状态。

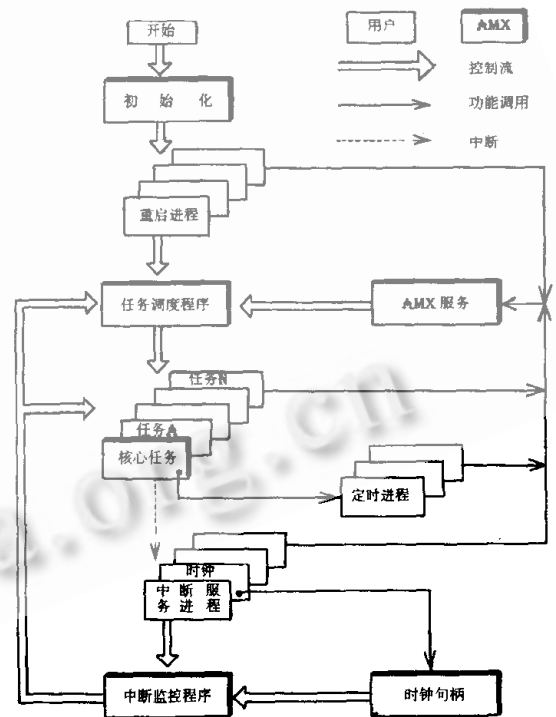


图1 AMX一般操作

一旦AMX初始化所有内部变量和结构以后,它执行一系列用户提供的重新启动进程。该进程由用户系统配置模块中的重新启动进程表提供。这些进程能请求AMX服务过程以便开始执行任务,并初始化间隔定时器。这是AMX具有的一大特点。

系统初始化后,AMX 进入任务调度。任务调度的方法具有一般抢占式多任务操作系统的特点。即通过占先权、优先级驱动算法,总使最高优先级的就绪任务控制处理器做有用工作。任务总是处于下列状态之一:空闲、就绪、运行、等待、暂停。当任务创建时,AMX 赋给它一个任务控制块,并置它为空闲状态。任务执行优先级可通过用户参数文件(扩展名为 .UP)由系统设计者定义。任务优先级范围为 0 (最高)到 255(最低)。优先级 0 和 128 到 255 由 AMX 保留使用。AMX 核心任务在 0 优先级执行,高于所有其他的任务。应用任务能赋以从 1 到 127 的一个优先级。如果一个以上的任务赋予相同的优先级,AMX 将按时间顺序赋予相对优先级,即先创建任务具有较高优先级。一个任务一旦执行,便可通过过程调用来请求 AMX 提供服务,如果一任务要等待一事件,AMX 服务进程将挂起任务并需要 AMX 任务调度程序来强迫执行下一个高优先级的任务。当 AMX 从正在执行的任务中收到运行一更高优先级任务的请求时,将切换任务。例如,执行任务可通过发信息请求 AMX 启动更高优先级的任务。

同样,AMX 也提供了一系列的中断监控程序,定时装置以及信息排队机制来更好地为用户提供服务。中断监控程序保证任务在开中断执行时允许系统对实时事件予以及时的响应,并且在中断服务程序需要执行一些比被中断任务优先级高的任务时,挂起被中断的任务并打上标记,然后激活高优先级的任务。

AMX 程序包还包括时钟处理程序,为实时应用提供定时控制。AMX 时钟处理过程独立于特定的硬件配置。如果 AMX 定时设施被利用,配置中必须包括实时钟。

AMX 最有效特性之一是 AMX 具有对任务执行的请求进行排队的能力。且其信息排队装置因允许信息排在任一四级优先级上而得到进一步加强。任务能从大量调用者那里得到已由 AMX 按优先级分类的信息。系统设计者可在任务定义中确定每个应用任务的邮箱需求。对每个任务可指定四级邮箱优先级,也能为那些邮箱指定所需的信息嵌套深度。此外,AMX 具有一些空信息信封,可被 AMX 用来进行任务间的信息传递。调用者信息首先被移到空信封内,随后将以请求的优先级加进目标任务邮箱中。这些特点使得任务之间彼此可方便地进行通信。

AMX 系统能按与启动相同的顺序形式终止。任务能通过调用 AMX 退出进程 ajexit 启动终止。AMX 退出

进程执行一系列的用户退出过程,使之恢复到 AMX 系统开始前存在的初始环境。一旦所有退出进程运行完毕,AMX 返回到初始调用它的那一点上。

AMX 还提供了一组管理程序来简化事件同步、资源操作和存储分配过程。这些程序是时间/日期管理程序、信号灯管理程序、事件管理程序、信息交换管理程序、缓冲管理程序、存储管理程序、循环链管理程序以及连接链管理程序。这组程序满足了用户对多任务应用系统的管理要求,为 AMX 在实时系统中的应用提供了更为广阔的空间。系统设计者可根据需要决定最适合用于特定应用的 AMX 管理程序。

三、AMX 软件开发

AMX 系统应用程序的各个部分可以作为一个独立的软件过程用 C 或汇编语言编写。为了实现多任务的特性,开发人员必须提供 AMX 系统配置模块。它预先定义 AMX 必须支持的最大任务数、间隔计时器、消息信封、信号、事件组以及 AMX 所支持的信息交换和缓冲池。

AMX 提供一个配置创建程序来简化系统配置模块的建立,用以在 IBM PC 及其兼容机上进行软件开发。此应用程序提供一种全屏幕交互式编辑 AMX 系统配置模块的方法。创建程序将产生一个描述 AMX 系统的汇编语言磁盘文件以及用来支持 AMX 使用的选择项所必需的结构。只有系统中需要的选择项将被包括在系统中。

使用系统配置创建程序有助于缩短整个系统的生成时间,主要是省去了系统配置模块的手工操作过程。为应用开发人员提供了很大的便利。

AMX 能够安装在运行 IBM PC DOS 或 Microsoft MS-DOS 的 IBM PC 及其兼容机上用于软件开发。它提供了与不同 C 和汇编语言的接口,可使用以下开发工具进行 AMX 系统开发和测试:

Microsoft :	Watcom :
MASM Assembler	C Compiler (for 80X86)
LINK Linker	Linker
LIB Librarian	Library Manager
DEBUG Debugger	WVIDEO Debugger
Code View Debugger	
C Compiler	
Borland :	Lattice C

TASM Assembler C Compiler (for 80X86)
 TLINK Linker
 TLIB Librarian
 Turbo Debugger
 Turbo C and C + + Compiler

下面以 Microsoft 的开发工具为例说明 AMX 应用程序的开发过程。

1. 用文本编辑软件编辑扩展名为 .up 的用户参数文件。该文件由关键字系列构成,形式为 ...xxx。每个关键字后必须跟着一个或几个参数。如

```
... UPT NME, UMS, CLP, CLF, NTK, NTM
... STK ESS, ISS
... OPT TSLICE, NSM, NEV, NMX, NBP,
ROMS, CAT, BKPT
```

该文件定义了多任务应用程序所需的各种参数,如用户参数表,堆栈,任务优先级,重启进程,退出进程,间隔定时器,信号灯,缓冲池等。是开发 AMX 应用程序过程中极其重要的一个环节。

2. 使用配置生成软件建立系统配置模块。如前所述,为了便于开发,AMX 提供了一个配置创建程序 AM831CG.EXE。可方便地将用户编辑的参数文件与系统提供的配置样板文件 AM831CG.CT 结合在一起以生成系统配置模块文件(扩展名为.ASM)。

运行配置生成软件的命令语句如下:

```
C: > AM831CG upfname. up AM831CG. CT Scfname. asm
```

upfname. up 用户编辑的参数文件的文件名

Scfname. asm 生成的系统配置模块的文件名

3. 将系统配置模块文件编译成目标文件.obj,可输入如下命令:

```
C: > MASM Scfname
```

4. 编辑应用程序的源文件,并将其编译或汇编成相应的目标文件。由于 AMX 提供了丰富的 C 语言库函数和汇编语言过程,使得用户在编程时可直接调用而不必过多地考虑多任务的特性。这就如同在其他多任务操作系统环境下(如 UNIX)直接使用系统调用编程一样方便。

在使用 C 语言编程时,其 Microsoft C 编译命令如

下:

```
C: > CL / Fo filename.obj /Gs /Alfw /Zp /Ze /Wo /
c filename. c
```

在上述选择项中,只有/Alfw 必须用到。其他选择项可以根据用户的特殊要求而变换。

5. 用 Microsoft 连接程序连接该系统。

```
C: > LINK AJ831PRF + Scfname + filename +
AJ831MC, AMXSYS, AMXSYS, AMX831. LIB +
LLIBCE. LIB;
```

上面说明的系统也可用下述方法进行连接。用任何一种文本编辑程序产生一个包含下述内容的连接说明文件 AMXSYS.LNK:

```
AJ831PRF + { object file list } + AJ831MC
AMXSYS/MAP/NOE
AMXSYS
AMX831. LIB + LLIBCE. LIB
```

然后使用如下命令进行连接: C: > LINK @AMXSYS.LNK

AMX831PRF.OBJ 为系统提供的段前缀模块,它必须为目标文件列表的第一个文件,以便强制用户系统中段排序的定义。用户的 AMX 系统配置模块须紧跟在它后边。目标模块 AJ831MC.OBJ 必须包含于用户的目标模块列表中。该模块强制性地用 KADAK 提供的虚拟栈检测过程替代 Microsoft 库中相应过程。AMX831.LIB 为 AMX 库。它的存在使 C 可访问所有 AMX 的 ajxxxx 过程。LLIBCE.LIB 为 Microsoft C 的大模式库。该命令生成可在 DOS 下运行的 AMXSYS.EXE 文件。

四、结束语

以上简单介绍了 AMX 的系统特征和软件开发过程。由于 AMX 是一组基于 DOS 的多任务执行程序,它具有操作容易,编程简单的特点,可较为方便地利用它编写出满意的实时多任务应用程序,因此必将给广大用户带来诸多的便利。

参考文献

- [1] <<AMX 86 参考手册>>, 机电部六所华胜电子技术开发部
- [2] <<AMX 86 Multitasking Executive>>, KADAK 公司

(来稿时间:1996年12月)