一种简单的实时多任务操作环境的实现

王 普 张亚庭
(北京工业大学自动化系，100022)

摘 要 介绍了一种在DOS环境下实现实时多任务操作的简单方法，任务数仅限于二个，内存占用小，任务切换速度快，有非常好的实时性，文中给出了全部程序清单.

关键词 实时多任务，操作系统，中断服务
分类号 TP 316.1

目前，在个人电脑上使用较多的多任务环境主要有UNIX操作系统，OS2操作系统和WINDOWS平台．这些多任务环境都具有处理任务量多的特点，但也有学习使用难度大的困难．在许多应用系统中，往往只需同时进行两个任务，构成前后台分时工作，前台完成显示工作，后台多为通迅、控制或打印等．因此在DOS环境下用C语言开发应用软件时，常常希望能简单地实现分时操作，以便开发的应用软件独立性好，维护方便，开发容易．本文介绍一种操作十分简单的实现方法，该方法程序简单，使用方便，内存占用小，仅适用于两个任务的分时操作．此方法已成功应用于实际系统中．

1 方法与实现

1.1 基本原理

分时操作总是建立在时间中断的基础上，使用计算机的实时钟为中断源，这样最小时间片为976.562 μs．在任务的切换上，为了程序的简单和通用，没有使用386以上CPU的任务转换功能，而是在640 K基本内存中实现两个任务之间的不停切换．

首先，在内存中建立一个驻留数据区，该区分为两部分，分别存放2个任务程序的入口，这样当需要从任务A(前台任务)切换任务B(后台任务)时，先将任务A的运行入口全部存入驻留区的A区内，然后再从驻留区的B区内将任务B的运行入口取出，执行任务B的程序．上述任务的切换工作是在实时钟中断服务子程序中完成，中断服务子程序和驻留数据通常是在运行autoexec.bat批处理文件时一起驻留在内存中．在开中断前，需先将任务B装入内存，并将任务B的程序入口放入驻留区，而后运行任务A的程序，在任务A中编入开中断指令，起动实时钟定时器，定时到时，将自动切换到任务B．

1.2 驻留数据区分配与后台任务

驻留数据区共10个字节，2个字节存放任务切换状态字，4个字节存放任务A入口，

收稿日期：1997—01—14

4个字节存放任务B入口．4个字节的入口内容只是该任务堆栈段寄存器(SS)值和堆栈寄存器(SP)值，实际上每个任务自身运行环境都是转到另一任务前压到自己的堆栈区中，这样只要有了堆栈入口就可以恢复原状态运行．

后台任务一般先行驻留，但有时后台任务需要能随时更换，如控制程序，这时可以先在内存中划分出一个区域，将该区域的首地址放入某一未用中断向量中，这样后台任务就可随时覆盖这一区域．

1.3　实时钟中断服务子程序

实时钟中断服务子程序是整个体系的核心部分，主要完成两个任务切换工作，中断服务子程序总是以中断返回指令IRET退出，只要在退出前将新任务程序入口移入堆栈，就可跳到新任务上．另外还需解决任务B的首次进入问题，这也是通过对任务切换状态字判断后完成．图1为中断服务子程序框图．

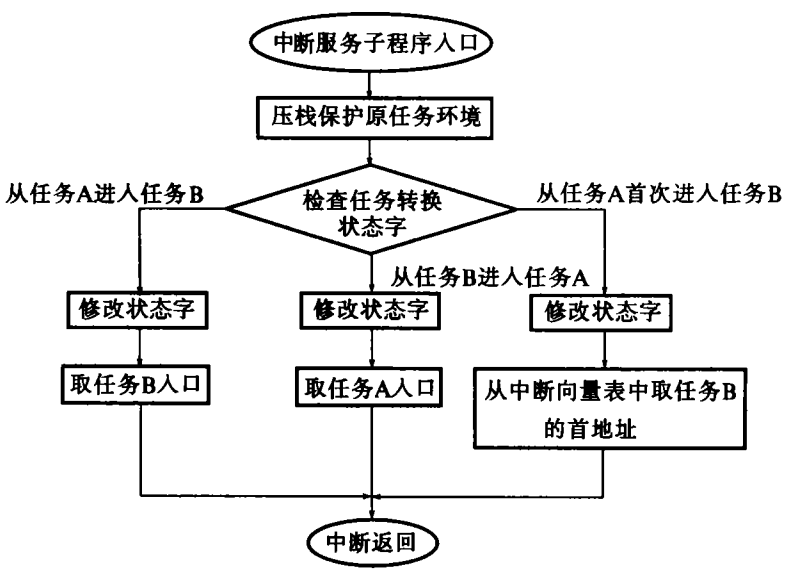

图1　中断服务子程序框图

1.4　任务间数据交换

任务之间总是需要进行数据交换的，交换的方法有2种：第1种仍是利用DOS的驻留功能，即开设一个常驻数据区，数据区大小根据需要决定，然后将数据区的地址放入中断向量表中，这样2个任务就可以共享同一数据区；第2种方法是2个任务独自建立数据区，但统一定义变量名，同时都生成一个变量地址表，这样任务A就可以通过任务B提供的变量表访问任务B的数据区．方法1适用于大量数据交据交换，方法2数据交换速度快．

1.5　前台任务编程

任务A(前台任务)可以说具有主程序作用，何时开始分时操作，何时关闭都在任务A中进行．起动操作为实时钟设置中断允许，以及开中断，具体程序为：

```
        cli
    ;  8259A OPEN
         MOV AL, 0H
        OUT  0A1H, AL
    ;       TIMER CMOS SETUP
        MOV   AL, 0AH
        OUT   70H, AL
        MOV   AL, 26H
        OUT   71H, AL
        MOV   AL, 0BH
        OUT   70h, al
        MOV   AL, 42H
        OUT   71h, al
        MOV   al, 0ch
        OUT   70h, al
        IN   AL, 71h
        ST1
```

关闭操作为关中断(CFI). 如用 C 语言编程，可使用嵌入汇编.

2 结论与应用

本文介绍方法的应用前提一是限于 2 个任务，二是任务分前后台，三是应用软件不能再使用硬中断. 本方法的优点有：①完全建立在 DOS 环境上，无需其它支持；②分时操作的控制及其方便；③程序短，内存占用很小；④实时性好，最小时间片 976.562 μs；⑤易学习，应用方便.

将此放法成功的应用在 JPHC 工控组态软件上，控制组态程序作为后台务，显示、报警、打印、记录为前台任务，这样在一台工业 PC 机上同时完成了现场实时控制和显示管理工作，最重要的是所有控制和显示管理工作都能组态建立，大大方便了用户使用.

附：中断服务子程序：

```
;=========================================
;INTER PROGRAME
;PROGRAM BE KEEPED IN INT70
;=========================================
    DOSSEG
     MODEL SMALL
;STACK1  SEGMENT PARA STACK 'STACK'
;STACK1  ENDS
;
;DATA    SEGMENT
```

```
;DATA    ENDS
;*****************************
;set Vector address
S_VECT MACRO INTER,SEG_ADDR, OFF_ADDR
    PUSH  DS
    MOV   AX,SEG_ADDR
    MOV   DS,AX
    MOV   DX,OFFSET OFF_ADDR
    MOV   AL,INTER
    MOV   AH,25H
    INT 21H
    POP DS
    ENDM
;-----------------------------------------
INT0  MACRO
    MOV  AL,OCH
    OUT   CMOS_PORT,AL
    JP      SHORT $+2
    IN  AL,CMOS_PORT+1
    MOV   AL,20H
    OUT    20H,AL
    OUT    0a0H,AL
    ENDM
;-----------------------------------------
;get vector address
G_VECT MACRO INTER
    MOV   AL,INTER
    MOV   AH,35H
    INT 21H
    ENDM
;-----------------------------------------
CODE     SEGMNT
ASSUME  CS:CODE,DS:CODE
CMOS_PORT EQU      70H
START:
    G_VECT 70H
    S_VECT 60H,ES,BX
    S_VECT 70H,CS,STARTI
    MOV    DX,28H
```

```
    MOV     AL,0
    MOV     AH,31H
    INT 21H
    MOV     AH,4CH
    INT     21H
STARTI:
    CL1
    PUSH    AX
    PUSH    BX
    PUSH    CX
    PUSH    dX
    PUSH    DS
    PUSH    ES
    PUSH    BP
    PUSH    SI
    PUSH    DI
    MOV     aX,CODE
    MOV     ds,ax
    MOV     BX,OFFSET DATA1
    CMP     WORD PTR DS:[BX],0H     ;FIRST CONTROL
    JE    LOP1
    CMP     WORD PTR DS:[BX],01H        ;DISPLAY
    JE    LOP2
    CMP     WORD PTR DS:[BX],10H    ;CONTROL
    JE    LOP3
    MOV     AH,4CH
    INT 21H
LOP1:
    MOV     WORD PTR DS:[BX],0H
    MOV     AX,SS
    MOV     DS:[BX+2],AX
    MOV     AX,SP
    MOV     DS:[BX+4],AX
    push ds
    mov ax,0
    mov ds,ax
    mov si,61h*4
    mov bx,ds:[si]
    mov es,ds:[si+2]
```

```
    pop  ds
    INT0
    pushf
    push es
    push bx
    iret
LOP2:
    MOV     WORD PTR DS:[BX],10H;10H
    MOV     AX,SS
    MOV     DS:[BX+6],AX
    MOV     AX,SP
    MOV     DS:[BX+8],AX
    MOV     AX,DS:[BX+2]
    MOV     SS,AX
    MOV     AX,DS:[BX+4]
    MOV     SP,AX
lend
    INT0
    POP DI
    POP SI
    POP BP
    POP ES
    POP DS
    POP DX
    POP     CX
    POP BX
    POP AX
    IRET    ;return display
LOP3:
    MOV     WORD PTR DS:[BX],01H
    MOV     AX,SS
    MOV     DS:[BX+2],AX
    MOV     AX,SP
    MOV     DS:[BX+4],AX
    MOV     AX,DS:[BX+6]
    MOV     SS,AX
    MOV     AX,DS:[BX+8]
    MOV     SP,AX
    INT0
```

```
    POP DI
    POP SI
    POP BP
    POP ES
    POP DS
    POP DX
    POP       CX
    POP BX
    POP AX
    IRET      ;return    control
;-----------------------------------------------
DATAI  DW0
    DW0
    DW0
    DW0
    DW0
    DW0
    DW0
    DW0
    DW0
CODE  ENDS
    END    START
```

A Simple Method to Realize the Operation of Multitask

Wang Pu Zhang Yating
(Department of Industrial Automation, Beijing Polytechnic University, 100022)

Abstract A simple method to realize the operation of multitask on DOS environment is introduced, of which the task number is two, only taking small area in internal memory, with fast transfering speed and ideal real-time characteristics. The whole programming flow is presented in this paper.

Keywords computer programme, real-time multitask, operating system