CMS 环境的有效运用

石化总公司北京设计院 阎 敖

摘要:*IBM* 中型计算机,大多配有 *VM/SP* 和 *CMS* 系统。对于程序开发者和用户而言,有效地运用 *CMS* 环境下的各种设施和机制,将会取得缩短研制周期,加快题目运算速度的效果。通过对引进软件的初步消化和实践,本文就文件的编辑、源程序的编译和调试,提出一些有益的措施,并对虚存做些分析。

一、文件的编辑

系统产品编辑-XEDIT 是 IBM 公司多年开发的成果,而且随着版本的更新,功能也更丰富灵活。因此熟练地掌握编辑的各种子命令和编辑宏,会给程序开发者代来很大方便。例如在 VM/SP 第五版的编辑系统中,可以使用 SI(Structure Input)前缀区子命令。在一行输入结束按 ENTER 键后,便自动添加一行,同时光标自动跳下一行的开头处,可以继续输入。当全部输入结束时,只按 ENTER 键,便结束了 SI 命令。这就可以减少由输入到编辑状态往复转换的时间。

至于屏幕的布局、色调和程序功能键的功能,还可以通过编辑宏进行设计。编辑宏文件的类型须为 XEDIT,文件名称可由用户给定。编辑宏文件可以存放在用户虚机所能访问的任意小盘上。例如一个编辑宏文件,它的文件名称为 XM,便可以下列命令调用

X FN FT FM (PROFILE XM)

其中,X 为 XEDIT 命令的缩写

FX FT FM 分别为被编辑文件的名称,类型和模式

XM 为编辑宏文件的名称

PROFILE 为 XEDIT 命令的选项

此条命令执行后,被编辑文件调入编辑系统后,首先执行编辑宏文件中的命令,构成所设计的屏幕形式,为功能键赋予指定的功能等。下页列出一个简单的编辑宏文件,它的文件标识为 XM XEDIT A。

FILE: XM XEDIT A

```
/*****************************/
/*                           */
/*   T-IE IJAP ENVIROVMENT   */
/*   Editor profile          */
/*****************************/
Trace O
'SET CMDLINE BOTIOM        '
'SET CURLINE ON 1          '
'SET SCALE OFF             '

'SET PF1 1ELP MENU         '
'SET PF3 FILE              '
'SET PF4 SPLIT CURSJR      '
'SET PF5 =                 '
'SET PE6 ?                 '
'SET PF7 BACKWARD          '
'SET PF8 FORWARD           '
'SET PF9 MACRO XM2         '

'SET PF13 HELP MENU        '
'SET PF15 FILE             '
'SET PF16 SPLIT CURSJR     '
'SET PF17 =                '
'SET PE18 ?                '
'SET PF19 BACKWARD         '
'SET PF20 FORWARD          '

'SET COLOR ARROW  WHITE           '
'SET COLOR PREFIX YELLOW          '
'SET COLOR CJRLINE RED            '
'SET COLOR MSGLINE YELLOW REV     '
'SET COLOR PENDING RED REV        '
```

```
'SET COLOR CMDLINE YELLOW      '
'SET COLOR TOF     WHITE       '
'SET MSGLINE ON 1 OVERLAY      '
'SET RESERVED 2 YELLOW NOHIGH  '
```

这个文件和 XM XEDIT A 一样，都是用 REXX 语言编制的。在按下 PF9 键后，这些编辑子命令和语句，依次执行。

'SAVE'将当前正在编辑的文件存盘，且仍保持在编辑子系统内。

TRANSFER FN FT FM'将当前正在编辑文件的名称(FN)，类型(FT)和方式(FM)，这三个已经给定的编辑变量传送到程序栈中。

PU／／ fn ft fm 这是 REXX 语言中的一个语句，它将程序栈中刚存入的三个值，读到变量 fn, ft 和 fm 内。

address CMS 'PRINT' fn ft fm 发出 CMS 环境中的打印命令，打印正在编辑的文件。

如上所述，我们已经有了两个编辑宏文件，在编辑宏 XM 执行后，再按 PF9 键，便将正在编辑的文件打印。如果将 XM，改为保留名称 PROFILE，则用 XEDIT 命令编辑文件时，便首先执行 PROFILE，这个编辑宏文件，而无须再用选项 PROFILE，在开始时，我们使用命令：

X FN FT FM (PROFILE XM

将文件调入编辑子系统，并首先执行编辑宏 XM。现在由于改为 PROFILE XEDIT A，所以只用

X FN FT FM

这个文件用 REXX 语言编写的。它是以／＊开头，以＊／结尾的一串字符，作为开始。接着是停止跟踪 TRACE O，然后是一系列在 XEDIT 子系统中所要执行的子命令，并且以单引号括入。这些编辑子命令都是 SET 命令和它的选项和参数。这个文件由三部分组成。

第一部分是屏幕上各种不同行的设置和显示，例如：

・SET MSGLINE ON 1 OVERLAY'

即将系统信息行设置在屏幕的第 1 行，如有系统信息便在此行显示，没有信息就不显示。

・SET RESERVED 2 YELLOW NOHIGH .....PFG＝PRINT.....'

即将屏幕上的第二行作为保留行，不再显示文件内容，而是显示一行短横，中间有

PF9＝PRINT

的提示，并且为黄色，正常光亮。

第三部分是对编辑屏幕上的一些区域和行进行彩色配置。例如

SET CURLINE ON 10'

即将当前行安排在第 10 行。又如

'SET COLOR CURLINE RED'

即将当前行着为红色。

第三部分是设定程序功能键的功能，例如

'SET PF1 HELP MENU'

即将程序功能键 PF1 设定为助援菜单，在 XEDIT 子系统中，按下 PF1 键便显示编辑子命令的菜单，可以查询各子命令。又如

'SET PF9 MACRO XM2'

即将程序功能键 PF9 设定为另一个编辑宏，XM2 为这个编辑宏文件的 称。按下 PF9 键，便执行 XM2 这个编辑宏文件上的编辑子命令。执行这些命令，就是将当前正在编辑的文件打印。编辑宏文件 XM2 XEDIT A 的内容如下

FILE:XM2　　XEDIT　E

```
/*****************************/
/*                           */
/*  T-IE IJAP ENVIROVMENT    */
/*  Print Macro              */
/*                           */
/*****************************/
Trace o
'SAVE'
'TRANSFER FN FT FM'
pull fn ft fn
address CMS 'PRINT' fn ft fm
```

这个文件和 XM XEDIT A 一样，都是用 REXX 语言编制的。在按下 PF9 键后，这些编辑子命令和语句，依次执行。

'SAVE'将当前正在编辑的文件存盘，且仍保持在编辑子系统内。

'TRANSFER FN FT FM'将当前正在编辑文件的名称(FN)，类型(FT)和方式(FM)，这三个已经给定的编辑变量传送到程序栈中。

PULL fn ft fm 这是 REXX 语言中的一个语句，它将程序栈中刚存入的三个值，读到变量 fn，ft 和 fm 内。

address CMS 'PRINT' fn ft fm 发出 CMS 环境中的打印命令，打印正在编辑的文件。

如上所述,我们已经有了两个编辑宏文件,在编辑宏XM执行后,再按PF9键,便将正在编辑的文件打印。如果将XM,改为保留名称PROFILE,则用XEDIT命令编辑文件时,便首先执行PROFILE这个编辑宏文件,而无需再用先项PROFILE。在开始时,我们使用命令:

X FN FT FM (PROFILE XM)

将文件调入子系统,并首先执行编辑宏XM.现在由于改为PROFILE XEDIT A,所以只用

X FN FT FM

便执行同样功能。但是只要将文件调入编辑子系统,便执行PROFILE XEDIT。如果对于某些文件不需要用XM编辑宏,或者用另外编辑宏文件,就仍应保持各个编辑宏文件的名称,而仍用编辑命令和选项PROFILE调入。

这两个,或类似的编辑宏文件,如果运用在应用程序上,作为输入数据文件,或者计算结果输出文件,便可以给用户提供比较合适的屏幕布置和醒目的色彩,以及一些功能键的功能,便于查阅校核和打印等。同时,调用文件进入编辑子系统的命令,还可以嵌套在执行文件(文件类型为EXEC)中,使用起来就更为方便。

二、源程序的编译与调试

源程序在编译和调试时,会出现一些错误,根据出错信息,对源程序进行相应修改。如此反复,直到调通为止。这段工作是相当耗费时间和精力的。但是如果能够熟练地运用CMS环境下的各种机制和命令,便可以加快速度。下面提出几点有益的措施:

1.源程序比较大时的编译与调试方法

源程序比较大时,可以分为几个程序段进行编译,节省时间,而且容易发现错误。进一步还可以利用SPOOL机制和程序功能键,编成执行文件,进行编译,并把出错信息和源程序,调上屏幕,对照修改。下面两个执行文件,就可以起到这种作用。

第一个文件 BYF EXEC A

```
ETRACE DFF
CP PURGE RDR
ERASE BA ERR A
CP SET PF9 BYF ε1
CP SET PF1 BYF1 ε1
CP SP CON START *
FORTVS ε1
```

第二个文件是 BYF1 EXEC A

```
εTRACE OFF
CP SP CON CLOSE
READCARD BA ERR A
εSTACK SCREEN 2
ESTACK X ε1 FORTRAN A
X BA ERR A
ERASE BA ERR A
```

假如有一个FORTRAN源程序文件的名称为COLUNT。在键入BYF COLUNT后,另一个文件便开始执行,先清除读卡机上的文件,以便接受和以后读到小盘,再将控制台屏幕显示信息形成文件,并且当此文件关闭后,传输到读卡机。接着将程序功能键PF9和PF10分别设置为BYF COLUNT和BYF1 COLUNT,以后再按PF9和PF10键,便分别执行这两个执行文件。最后对于FORTRAN源程序COLUNT进行编译。

如果在编译时,有错误信息在屏幕上接连显示,便可以键入HX命令使其中止返回CMS环境。再按下PF10键,则在命令行上出现BYF1 COLUNT,再按ENTER键,第二个文件便开始执行。首先关闭控制文件,再从读卡机上读到小盘上,文件标识为BA ERR A。通过反STACK语句,将两条编辑子命令送进栈。当执行X BA ERR A时,先将屏幕划分为二,接着将源程序文件COLUNT FORTRAN A,调上屏幕,于是屏幕上有出错信息,也有源程序,可以分别用编辑子命令进行编辑,对照出错信息,修改源程序。如图-1所示。当然也可以将源程序改为LISTING文件。

```
COLUNT FORTRAN A1 F 80 TRUNC-72 SIZE-948 LINE-5 COL-1 ALT-0
---- C     * * * * * * * * * * * * * * * * * * * *
----        SUBROUTINE COLUNT(NO,ID,LEQN)
---- C     * * * * * * * * * * * * * * * * * * * *
----
      ..+..1..+..2..+..3..+..4..+..5..+..6..+..7.>
----        INCLUDE(DATAS)
---- C
----        CHARACTER ID,KM,KS,KE,KO,IB1
--->
                                    XEDIT 2 FILES
BA ERR A1 F 132 TRUNC-132 SIZE-32 LINE-5 COL-1 ALT-0
-----
      ..+..1..+..2..+..3..+..4..+..5..+..6..+..7.>
--- ISN 4          IMENSION ID(80),IVAL(27)
--- TEXT NOT RECOGNIZABLE AS A FORTRAN STATEMENT. STATEMENTIS IGNORED.CHECK
```

```
--- STATENRT SPECIFICATION.
```

图 1　源程序和编译错误信息屏

2.分屏交互调试是提高效率的一种好方法。

在 CMS 系统中一般都配有 ISPP、IPF 和 PDF 交互软件。在这些软件支持下，可以将屏幕分为两部分，上半部对程序进行交互调试，下半部可以翻阅程序或 LISTING 文件进行对照检查和修改，这是相当方便的。但在具体使用时，有以下几点值得注意：

(1) 在进入交互环境后，到达交互调试屏幕时，屏上 LOAD 提示字段，只允许键入一个目标文件名(类型为 TEXT)。一般为主程序段文件名。但对于程序分段和分段文件的命名，带来的问题是：如果分段和命名不当，就会产生有一些子程序找不到，而显示错误信息，不能装配，影响调试。解决这个问题的简而易行的原则是：

主程序及其直接调用的子程序，最好划分在同一程序段内，如不能做到，则应以直接调用的子程序段的文件名。

另一个方法，是以子程序文件包含的任意一个子程序名为文件名，编译后装配时，如果显示出那些子程序来定义，即错误信息为：

```
DMSLIO 20IW  The following names are undefined:
.....
Ready (0004) ;
```

根据未定义的子程序名，查找它们所在的文件，就将这个文件名改为这个未定义的子程序名。再重点编译和装配，再改变文件名。如果重复几次之后，仍有这个错误，就需要改变子程序的划分，尽量将被调用的子程序和调用它的主(子)程序划分在同一个程序段中(即同一个文件中)。总之，最终一定会找到合适的程序划分，和各个子程序段的文件名。

(2) 调试所用的目标文件，在编译时要求选用参数 TEST。至于优化编译参数，最好先选用 0 级优化，虽然对于已说明而未使用的变量能够在调试过程中发现，但并未对循环语句、转向语句等进行优化。因此，对于调试命令，如 AT 等，没有影响。待 0 级优化调通之后，再逐步升级到 3 级优化。当然对于大型程序而言，需要 3 级优化，尽管编译时间延长一些，但能够得到运行较快目标文件，在算题时便可以节省机时。

(3) 准备一个执行文件，包括：

· 声明必要的库，如 TEXT 库等，

· 定义输入和输出文件，

· 如果对于输入数据文件和输出结果文件，有某些约定，也要编入这个文件中。下面是本例应用的执行文件，文件标识为 KLQC EXEC A。如果只给定输入数据文件名，(例如 PROC2)，则输入文件标识为 PROC2 DATA A，输出文件标识为 PROC2 OUT A。如果不打算使用这种约定，则需给定输入数据文件和输出文件的标识。

```
FILE: KLQC    EXEC    A

εTRACE OFF  B
GLOBAL  TXTLIB VFORTLIB CMSLIB VALTLIB
εFN = ε1
εFT = ε2
εFM = ε3
εOFN = ε4
εOFT = ε5
εOFM = ε6
εIF εFT = εTHEN εFT = DATA
εIF εFM = εTHEN εFM = A
εIF εOFN = εTHEN εOFN = εFN
εIF εOFT = εTHEN εOFT = OUT
εIF εOFM = εTHEN εOFM = A

FIL 5 DISK εFN εFT εFM
FIL 5 DISK εOFT εOFM (LRECL 132)
```

这个执行文件是在进入交互环境之后，于命令和执行文件处理(COMMAND AND EXEC PRDCESSING)屏上，“下面输入命令”提示字段，键入该文件名和要求的参数，再按 ENTER 键，便依次执行文件上的命令，为程序调试做好准备。

(4) 进入交互调试和分屏。交互调试是从“前台作业选项菜单(FOREGROUND SELECTION MENU)”中，选择“VS FORTRAN 交互调试(VS FORTRAN INTERACTNE DEBUG)”之后，便显示“前台 VS FORTRAN 交互调试”的屏幕。在 FILE ID (文件标识)提示字段的箭头右侧，只输入一个被调试程序的 TEXT 文件，本例为 KLQ，再按 ENTER 键，便进入交互调试状态。此时，将光标移动到屏幕的中间位置，按 PF2 键，屏幕便从光标所在位置将屏幕分为两部分。上半屏幕为交互调试，下半屏幕为“ISPF／PDF 主选项菜单”，可以选用 BROWSE 选项，将 KLQ LISTING 文件，或源程序 KLQ FORTRAN 文件调入下半屏幕，使用 BROWSE 子命令，对照上半屏幕调试进程和信息，进行翻阅查找出错原因和语句。

3.程序的二、三级优化，是比较困难的

如果能够在编写程序时，注意一些方法，调试时就会相对地容易。一些简而易行的编程原则列举于后。

(1) 消除说明而未使用的变量和数组，和没用的语句标号

(2) 表达式中的变量，包括下标变号和函数，在其值未改变时，如需重复使用，最好赋值予一个变量后，再重复使用这个变量。

(3) 循环语句的循环体中，与循环无关的语句，要移动到循环体之外。

(4) 循环体中如嵌套循环语句时，循环次数多的放在内层，循环次数少的放在外层。

(5) 在 IP 语句中的逻辑表达式不宜过长。

诸如此类的原则很多，仅举几个，余者不在赘述。

三、虚存的设定

虚机虚存的大小，对于程序运行快慢和系统效率的影响是相当大的。因此，系统支持虚机用户以 4KB 步长调节虚存。这就意味着，我们应该把虚存调节到此程序运行实际需要的虚存。为了说明这一点，引起足够的重视，现从虚存和处理机管理两方面进行一些初步分析。

1.虚机存储管理

虚机系统 VM／SP 对存储管理，实际上是一个多虚拟存储管理，沿用了 System／370 的 24 位动态地址转换的硬件 DAT 和段表页表为基础，通过 CP 程序实现的。由于 DAT 的支持，可在 CP 专用盘卷的 PAGE 和 TEMP 区，将每个磁道划分为若干 4KB 区块，称为页(PAGE)，以柱面、磁道和页(CCHHR)为存取地址进行页面存取。同时在实存中划分一块动态可分页区 DPA，同样划分为若干 4KB 区块，称为页框(PAGE FRAME)，与虚存页面相对应。

当虚机登录时，CP 按虚机内存(虚存)大小在 PAGE 区分配页面，PAGE 区不够时，再 TEMP 区补足。同时在内存 FREE 区建立虚机控制块(VMBLOCK)，段表(SEGTBL)，页表(PGTBL)，换页表(SWPTBL)和内存映象表(CORTBL)。所有登录虚机的虚机页面都在竞争实存页框。如果可分配的页框数目低于某个低限值时，CP 便在已分配实存页框上的虚存页面中，判断不活动页面(inactive page)，并且将它们调出，腾出一部分页框。当 CP 检查出运行程序段要访问的虚存页面不在实存页框上时，便将所需页调入实存页框。在进行页调时，CP 对于控制块和表进行相应修改，随时记录虚机页面在实存页框上分配情况，准确地转换虚实地址，完成页面调入调出操作。

不活动页面保留在 PAGE 和 TEMP 区。如果在虚机运行过程中的某个时刻，不活动页面发生了改变，CP 将其移动到 PAGE 区。在执行页调时，PAGE 区优于 TEMP 区。为了保持 PAGE 区的一定空闲率，每隔一段时间执行一次页迁移，将一定数量的页面再 PAGE 区迁移到 TEMP 区。通过以上分析，可以归纳为三点：

(1) 虚存越大，虚存页面相应增多，要求分配的页框数也越多，引起页框竞争加剧，促使页调操作更为频繁。

(2) 虚存越大，占用 PAGE 区越大，页迁移次数和页面也增加，而以 TEMP 区直接页调不如从 PAGE 区页调有利，耗用较多 CPU 资源。

(3) 虚存越大，通过通道-控制器-磁盘路径更为频繁，等待各种资源的时间只会随之增加。

2.处理机管理

虚机系统是以分时多道程序系统的处理机时间片方式为基础，通过 CP 根据虚机运行特徵、优先级别和可供分配的系统资源进行处理机时间片分配的。

在虚机运行时间片的末尾，CP 根据虚机从虚机控制的终端发出或接受操作的频度，把虚机作业分为两类：一类是 I／O 繁忙的交互作业，另一类是使用 CPU 多的非交互作业。这两类作业分别收容在系统设置的 Q1(队 1)和 Q2(队 2)。为了满足一些只使用 CPU 而无人机交互的完全非交互作业使用大量 CPU 时间的需要，系统设置了 Q3(队 3)。

在虚机作业获得申请的系统资源和等待的某些动作已经完成，便可以运行，从 Q1、Q2 和 Q3 进行调度。

当虚机的终端产生中断为虚机响应时，便进入 Q1，当虚机用完处理机时间片便退出 Q1 进入入选表(eligible table)。CP 根据调度策略可以从入选表中选出虚机作业送入 Q2。虚机作业在入选表和调度表(schedule list)的排列和作业调度的选取，都是以截止优先级 DP (Deadline Priority)大小为顺序的。DP 值是用表决定虚机作业获得下一个处理机时间片所需间隔时间的。DP 值越大，间隔时间越短。反之，则越长；也就是说 DP 值越大的虚机作业的时间短，速度快；反之，则时间长，速度慢。

截止优先级 DP 是由调度程序在虚机作业退出队列时按下式进行计算的。DP 值的计算是以虚机作业退出

时间、页调忙闻、处理机利用率、系统负荷、登录虚机数目、虚机优先级和当前可供分配的系统资源为依据的。

DP＝TOD+UBF

式中　DP 截止优先级

TOD 当前日历时间

UBF 用户偏差系数(User Bias Factlr)

UBF＝UBR+Q2DF

式中　UBR 用户偏差比(User Bias Ratio)

＞1 所获资源超过所需资源

UBR　＝1 所获资源等于所需资源

＜1 所获资源少于所需资源

Q2DF Q2的延迟系数

决定虚机调度的另一个重要因素是它的工作页面区(Working set)的大小。工作页面区是为避免过多的页面调度所必须送入实存的用户页面的数目。工作页面区是虚机作业逗留在Q2中所要访问虚存页面的函数。这些页面在虚机作业退出Q2时，调度程序便计算它的工作页面区。只有计算的工作页面区不超过当时实存中可分配页框数目时，该虚机作业才能进入Q2。

Q1虚机作业用完处理机时间片，尚未结束作业，便退出Q1插入选表中。当CP在入选表中依照DP值顺序，从大到小检索虚机作业的工作页面区时，如果工作页面区超过可分配页框时，即使它的DP值大也不能进入Q2。CP继续向下检索，直到一个虚机作业。它的工作页面区不超过当前可分配页框，才将其选入Q2。

Q2虚机作业用完处理机时间片，退出Q2按DP值大小插入入选表。如果此时该虚机发生人机交互，即便I/O，便被送入Q1。如果Q2虚机作业连续接受8次Q2处理机时间片，而无一次Q1处理机时间片，则为Q3收容。Q1虚机作业与Q2虚机作业比较，在DP值相同或者Q2虚机作业的DP值稍大时，Q1虚机作业提前从入选表送入调度表。这是Q1虚机作业唯一优先之处。

各队虚机作业获得处理机时间片的长度和频度的关系如下式表达。

Q2时间片＝B×Q1时间片

Q2获得CPU的频度＝1/B×Q1获得CPU的频度

Q3时间片＝B×Q2时间片

Q3获得CPU的频度＝1/B×Q2获得CPU的频度

由此可知，在相当长的时间内，虚机作业始终在Q1或Q2中运行，获得处理机时间片的总和是相等的。同样，Q2和Q3也是如此。但这并不是说，在Q2和Q3中虚机作业耗用的处理机时间片的总和就相等。这是因为虚机作业退出队列时，总要有一定的系统开销。粗略地说，开销的大小与驻留实存中少量的页面成正比。Q3虚机作业退出队列的次数要比Q2虚机作业退出队列的次数少得很多。因此对于占用虚存较大的程序在Q3中运行耗用处理机时间片的总和要比Q2虚机作业运行少用一倍左右。

通过以上分析，可以归纳以下几点：

(1) 占用虚存大的虚机作业的存储资源，不易满足。因此用户偏差比较小，DP值下降，在入选表中排列靠后，退队时间较长。

(2) 占用虚存大的虚机作业，离队时驻留页框上的页面相对的多，计算出的工作页面区也相对的大，而不易进入Q2；相反，虚存小的虚机作业可能提前进入Q2，得到较多的CPU时间。

(3) 占用虚存较大的程序，以在Q3运行为好。

综上所述，在程序优化调通之后，最好还要设定一个较为合适的虚存，取得最佳效果。甚至可以根据题目大小和经验，修改主程序中的数组大小，再次编译，设定虚存而后算题。

如果题目很大，虚存相应增大，尽管耗用资源较多，也是值得的。如果设定的虚存超过题目实际需要的很多，不仅影响系统效率，而且算题的速度要减慢许多。

参考资料：

VM/SP System Programmers Guide

IBM/VM Interactive Producteivity Facility System Reference

▲