

第 6 章 过程实验

实验目的

- 了解过程的分类和功能
- 练习过程的书写组织和调用

实验内容

- 了解过程及其分类
- 定义流内过程
- 定义编目过程
- 修改过程参数

实验介绍

- 本实验主要锻炼学生编写流内过程和编目过程的能力，并通过在调用过程的时候改变 DD 语句参数或者 EXEC 语句参数来灵活地使用同一个过程实现不同的功能。

6.1 了解过程及其分类

6.1.1 过程的定义语句

► PROC 语句

PROC 语句标志过程的开始，用来定义被调用的过程名，例如定义一个名为 PROCA 的过程：

```
//PROCA PROC
```

► PEND 语句

PEND 语句标志过程的结束，放在流内过程结束处，也可以为 PEND 语句指定一个名字：

```
// PEND (注：无名) 或者 //ENDPROC PEND (注：名字为 ENDPROC)
```

6.1.2 过程的调用方式

以下两行都可以调用过程 PROCA：

```
//JSTEP EXEC PROC=PROCA
```

```
//JSTEP EXEC PROCA
```

6.1.3 过程的分类

流内过程： 写在某作业中，放在第一个调用该过程的作业步之前，流内过程以 PROC 开始，以 PEND 结束，过程可由本作业中过程定义之后的任意一个 EXEC 语句调用。

编目过程： 储存在过程库（分区或扩展分区数据集）的成员中，每个成员中只能存储一个过程实体，成员名即为过程名。任何作业都可以调用该过程。

6.2 定义流内过程

流内过程以 PROC 开始，以 PEND 结束，过程由一个 EXEC 语句调用。流内过程举例：

```
//MYJOB JOB 377-44-1247,D.ROSE
//PROCA PROC
//PSTEP1 EXEC PGM=MYPROC
//DDIN DD DSN=INDATA,DISP=SHR
//DDOUT DD SYSOUT=A
// PEND
//STEP1 EXEC PGM=PROG1
//DD1 DD DSN=DATA1,DISP=SHR
//DD2 DD SYSOUT=A
//STEP2 EXEC PROCA
```

回答下面的问题

下列哪个 EXEC 语句调用了该过程？

-
- A. EXEC PROCA
 - B. EXEC PGM= PROCA
 - C. EXEC PGM=MYPROG

思考：将下列语句按正确顺序排列成流内过程

-
- A. //PSTEP1 EXEC PGM=MYPROG
 - B. //STEP1 EXEC PROCA
 - C. //MYPROC PROC
 - D. //MYJOB JOB NOTIFY=&SYSUID
 - E. // PEND
 - F. //DD1 DD DSN=DATASET, DISP=SHR

6.3 定义编目过程

编目过程是指将过程内容放在某过程库中某成员中的过程。该过程库可以是分区数据集或扩展分区数据集，通常系统过程被放置在系统过程库中，用户过程被放置在用户库中。

举例如下：在用户指定的过程库（TE01.PROC.LIB）中定义编目过程 NEWDS，内容如下，该编目过程包括两个作业步。STEP1 使用 IEBGENER，在 USER02 卷上新建一个输出数据集。输入数据集由调用者动态指定。STEP2 将 STEP1 新建的数据集作为输入数据集，控制语句由调用者动态指定。

编目过程内容如下 (TE01.PROC.LIB(NEWDS))

```
//NEWDS      PROC   DDN=TE01.PS1
//STEP1      EXEC   PGM=IEBGENER,TIME=(2,1)
//SYSPRINT   DD     SYSOUT=*
//SYSUT2     DD     DSN=&DDN,DISP=(NEW,CATLG,DELETE),
//            SPACE=(TRK,(1,1)),VOL=SER=USER02,
//            RECFM=FB,LRECL=80,UNIT=SYSDA
//SYSIN       DD     DUMMY
//STEP2      EXEC   PGM=IEBGENER,TIME=(2,1)
//SYSUT1     DD     DSN=&DDN,DISP=SHR
//SYSUT2     DD     SYSOUT=*
//SYSPRINT   DD     SYSOUT=*
```

调用以上编目过程的作业可以组织如下：

```
//TE01A      JOB      ACCT#,TE01,MSGLEVEL=(1,1),NOTIFY=TE01
//MYLIB      JCLLIB   ORDER=TE01.PROC.LIB
//PROCSTEP   EXEC     PROC=NEWS,DDN=TE01.PS1
//STEP1.SYSUT1 DD *
    TOM WANG 21 3500
    MIKE LIU  25 8000
//STEP2.SYSIN DD *
    GENERATE MAXFLDS=3
    RECORD FIELD=(4,8,,1),FIELD=(4,3,,15),FIELD=(2,13,,25)
```

6.4 修改过程参数

修改过程参数：

- 修改 DD 语句
- 修改 EXEC 语句

6.4.1 修改 DD 语句

修改 DD 语句的参数使之在执行时生效，包括：

- 更改 DD 参数


```
//procstepname.ddname DD parameter=value
```
- 还原 DD 参数默认值


```
//procstepname.ddname DD parameter=
```
- 使 DD 语句无效


```
//procstepname.ddname DD DUMMY
//procstepname.ddname DD DSN=NULLFILE
```
- 增加 DD 参数


```
//procstepname.newddname DD parameter=value
```

示例 1：更改 DD 参数，过程内容及过程调用语句(更改了 DD 语句)如代码 1，实际运行的过程语句如代码 2

代码 1 更改 DD 语句示例

```
//TRANSACTION PROC
//PSTEP1      EXEC    PGM=PROG1,TIME(1,30)
//DD1         DD      DSN=INTRAN,DISP=SHR,UNIT=3590,VOL=SER=USER01
//DD2         DD      DSN=MASTER,DISP=SHR
//PSTEP2      EXEC    PGM=PROG2,TIME=5
//DD3         DD      DSN=NEWTRAN,DISP=(OLD,DELETE)
//            PEND
//STEP        EXEC    TRANSACTION
//PSTEP1.DD1   DD      DSN=NEWTRAN, UNIT=3390,VOL=SER=USER02
```

代码 2 更改 DD 语句结果

```
//PSTEP1 EXEC PGM=PROG1,TIME(1,30)
//DD1 DD DSN=NEWTRAN,DISP=SHR,UNIT=3390,VOL=SER=USER02
//DD2 DD DSN=MASTER,DISP=SHR
//PSTEP2 EXEC PGM=PROG2,TIME=5
//DD3 DD DSN=&&VALID,DISP=(OLD,DELETE)
```

示例 2: 还原 DD 参数，过程内容及过程调用语句(还原了 DD 语句)如代码 3，实际运行的过程语句如代码 4

代码 3 还原 DD 参数默认值示例

```
//TRANSACTION PROC
//PSTEP1 EXEC PGM=PROG1,TIME(1,30)
//DD1 DD DSN=NEWTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR,UNIT=3480,VOL=SER=987762
//PSTEP2 EXEC PGM=PROG2,TIME=5
//DD3 DD DSN=&&VALID,DISP=(OLD,DELETE)
// PEND
//STEP EXEC TRANSACTION
//PSTEP1.DD2 DD UNIT=,VOL=SER=
```

代码 4 还原 DD 参数默认值结果

```
//PSTEP1 EXEC PGM=PROG1,TIME(1,30)
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR
//PSTEP2 EXEC PGM=PROG2,TIME=5
//DD3 DD DSN=&&VALID,DISP=(OLD,DELETE)
```

示例 3: 使 DD 语句无效，过程内容及过程调用语句(使 DD 语句无效)如代码 5，实际运行的过程语句如代码 6

代码 5 使 DD 语句无效示例

```
//TRANSACTION PROC
//PSTEP1 EXEC PGM=PROG1,TIME=(1,30)
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DSN=MASTER,DISP=SHR,
// UNIT=3390,VOL=SER=USER01
//PSTEP2 EXEC PGM=PROG2,TIME=5
//DD3 DD DSN=##VALID,DISP=(OLD,DELETE)
// PEND
//STEP EXEC TRANSACTION
//PSTEP1.DD2 DD DUMMY
```

代码 6 使 DD 语句无效示例结果

```
//PSTEP1 EXEC PGM=PROG1,TIME=(1,30)
//DD1 DD DSN=INTRAN,DISP=SHR
//DD2 DD DUMMY
//PSTEP2 EXEC PGM=PROG2,TIME=5
//DD3 DD DSN=##VALID,DISP=(OLD,DELETE)
```

*问题

定义过程 MYPROC 如下

```
//MYPROC PROC
//STEP1 EXEC PGM=PROGA
//DD1 DD DSN=A,DISP=SHR
//DD2 DD DSN=OUTA,DISP=(NEW,CATLG),
// UNIT=SYSDA,SPACE=(TRK,(1,1))
//STEP2 EXEC PGM=PROGB
//DD3 DD DSN=B,DISP=SHR
//DD4 DD DSN=RPT,DISP=(NEW,CATLG),
// UNIT=SYSDA,SPACE=(TRK,(1,1))
// PEND
```

要求更改 DD2 的 UNIT 参数为 3390，并令 DD4 中的 SPACE 参数无效。请在下面写出相应的 JCL 语句。

```
//JSTEP EXEC MYPROC
```

6.4.2 修改 EXEC 语句

- ▶ 更改 EXEC 语句参数

```
//JSTEP EXEC 过程名,参数名.作业步名=要改动的值
```

//STEPNAME EXEC procname, parameter.procstepname=value

示例：//JSTEP1 EXEC TRANSACT, TIME. PSTEP1=1

- ▶ 还原 EXEC 语句参数默认值

```
//JSTEP1 EXEC 过程名, 参数名.作业步名=
```

//STEPNAME EXEC procname, parameter.procstepname=

示例：//JSTEP1 EXEC TRANSACT, TIME. PSTEP1=

- ▶ 增加 EXEC 语句参数

```
//JSTEP1 EXEC 过程名, 新参数名.作业步名=新值
```

//STEPNAME EXEC procname, newparameter.procstepname=value

示例：//JSTEP1 EXEC TRANSACT, REGION. PSTEP1=0

6.4.3 符号参数

&+变量名称组成符号参数, 它可以理解为编程语言中的变量, 它的值可以在过程定义时指定, 或者在过程运行时指定。用法如下:

```
//MYPROC PROC DS=TE02.JCL.LIB
//STEP1 EXEC PROG=PROG1
//DDN DD DSN=&DS
//MYPORC PEND
//STEP1 EXEC MYPROC, DS=TE02.RUNTIME.JCL.LIB
```

实验 1 编写一个简单的流内过程并调用, 实现显示指定顺序数据集中的内容, 要求将显示的内容放在 A 类设备上 (SYSOUT=A)。在指定顺序数据集中读取内容, 从第 2 列取 5 个字符显示在第 1 列上, 从第 10 列取 4 个字符显示在第 7 列上, 从第 16 列取 6 个字符显示在第 15 列上。

[提示] 使用 IEBGENER 来实现

实验 2 改写上述实验, 要求第一次调用流内过程时增加 TIME 参数, 最多执行 4 分钟, 而第二次调用该过程时还原 TIME 参数默认值。

实验 3 编写编目过程, 复制已经存在的某顺序数据集 (yourid.PS) 内容到新建的某分区数据集的某成员 (yourid.P0(MEM)) 中, 要求该顺序数据集及分区数据集成员在调用过程时指定。

[提示] 使用 IEBGENER 来实现

6.5 过程综合实验

在上述实验的基础上，根据要求编写流内过程和编目过程并调用它们。

实验 4 按要求编写流内过程及编目过程并调用它们。

说明：yourid 表示用户 TSO 帐号，'Sxxxxxx' 中的 x 部分输入自己的学号，注意：yourid.EXER.TXT 的内容如下（需要自己创建该顺序数据集）。

```
013224 TOM
033197 JACK
013208 MARY
023216 LILY
033219 IRMA
023218 FRANK
```

实验要求：

步骤一 编写作业 1，要求如下

1. 作业步一：执行 SORT 实用程序，将 yourid.EXER.TXT 的内容进行字符升序排序，排序字段为前 6 个字符，将排序结果存入新文件 yourid.Sxxxxxx.TXT 中，要求新文件放在 USER02 卷上并对新文件进行编目
2. 作业步二：执行 IEBGENER 实用程序，将 yourid.Sxxxxxx.TXT 的内容按照年级的不同 (01xxxx, 02xxxx, 03xxxx 分别为不同的年级) 分割成三部分，形成三个 MEMBER，放入 yourid.Sxxxxxx.P01 中
3. 设置运行时间为 2 分钟

步骤二 编写作业 2，要求如下

1. 将作业 1 的第二个作业步改写为流内过程（放在作业 2 中），过程名为 MYPROC
2. 使用符号参数 &OUTDS，指定 IEBGENER 的输出数据集。

提示：SYSUT2 DD DSN=&OUTDS

3. 在作业 2 中调用流内过程 MYPROC，调用时给 OUTDS 赋值 yourid.Sxxxxxx.P02

步骤三 编写作业 3

1. 将作业 1 的第二个作业步改写为编目过程，放在 yourid.PROC.LIB(MYPROC) 中

2. 在作业 3 中调用编目过程，调用时给 OUTDS 赋新值 yourid.Sxxxxxx.P03（可以是新文件，也可以是已经存在的文件）
