CSCI 360 External Subroutines

External subroutines are routines that are created and maintained separately from the program that will be calling them.

Each routine should begin with:

```
rtnName  CSECT
```

where rtnName is the label by which the subroutine will be called.

Pattern for a program with external subroutines

```
MAINPGM  CSECT
*
*
*** MAINPGM code, LTORG, and storage goes here
*
*
SUBRTN1  CSECT
*
*
*** SUBRTN1 code, LTORG, and storage goes here
*
*
SUBRTN2  CSECT
*
*
*** SUBRTN2 code, LTORG, and storage goes here
*
*
         END   MAINPGM
```

Each new control section will automatically begin on a doubleword boundary

LTORG is included within each CSECT to ensure that the literals for each routine are assembled as part of that control section, rather than as a part of the first control section (in the above case MAINPGM).

To call an external subroutine, you must:

1. point register 1 to a parameter list (if needed)
2. get the address of the subroutine into register 15
3. call the routine

Step 2 is accomplished by using a V-type literal:

```
L     R15,=V(SUBRTN)
```

Step 3 is accomplished by using the BALR instruction

Format: label BALR R_1,R_2

The BALR instruction works just like the BAL instruction. R_1 is the address of the instruction immediately following the BALR instruction. R_2 is the address of the routine being called.

Standard Linkage Conventions

1. When control is passed to an external subroutine, register 15 should contain the address of the subroutine, register 14 should contain the address of the next instruction to execute, and register 13 should contain the address of an 18F savearea in the calling routine in which its initial register values will be saved.

2. Parameters are passed to an external subroutine through a parameter list. The address of the parameter list is contained in register 1.

3. Return codes are passed back to a calling routine in register 15.

4. Calculated values are passed back to a calling routine in register 0.

5. The subroutine is responsible for storing the contents of the registers upon entry to the routine and restoring those values before returning control to the calling routine.

Standard Entry Linkage

The following code should be included as the first lines of each CSECT, including the MAIN one:

```
rtnName  CSECT
         STM   14,12,12(13)
         LR    12,15
         USING rtnName,12
         LA    14,name_of_18F_save_area_in_rtnName
         ST    13,4(,14)
         ST    14,8(,13)
         LR    13,14
```

STM 14,12,12(13) saves all of the calling routine's registers, except for register 13, in the calling routine.

LR 12,15 puts the address of rtnName in register 12.

USING rtnName,12 sets register 12 as the base register for rtnName.

LA 14,name_of_18F_save_area_in_rtnName points register 14 to an area in rtnName where its registers will be saved if rtnName calls another routine.

ST 13,4(,14) stores the address of the calling routine's save area in rtnName's save area. The value in register 13 is known as the **backward pointer**.

ST 14,8(,13) stores the address of rtnName's save area in the calling routine's save area. The value in register 14 is known as the**forward pointer**.

LR 13,14 points register 13 to rtnName's save area in case another routine is going to be called.

Format of the 18F register save area

Unused

Backword Pointer

Forward Pointer

Register 14

Register 15

Register 0

Register 1

Register 2

Register 3

Register 4

Register 5

Register 6

Register 7

Register 8

Register 9

Register 10

Register 11

Register 12

Standard Exit Linkage

The following code should be included as the last lines of executable code in each CSECT if no values are being passed back, including the MAIN one:

```
L     13,4(,13)
LM     14,12,12(13)
BR     14
```

L 13,4(,13) loads the address of the calling routine's save area into register 13.

LM 14,12,12(13) reloads all of the calling routine's registers, except for register 13.

BR 14 returns control to the calling routine.

Exit Linkage for a routine passing a return code back in register 15

All of the registers, except for 13 and 15, are going to be restored.

```
L     13,4(,13)
L     14,12(,13)
LM    0,12,20(13)
BR    14
```

L 13,4(,13) loads the address of the calling routine's save area into register 13.

L 14,12(,13) restores the calling routine's register 14.

LM 0,12,20(13) restores registers 0 through 12 of the calling routine.

BR 14 returns control to the calling routine.

Exit Linkage for a routine passing a calculated value back in reg 0

All of the registers, except for 0 and 13, are going to be restored.

```
L     13,4(,13)
LM    14,15,12(13)
LM    1,12,24(13)
BR    14
```

L 13,4(,13) loads the address of the calling routine's save area into register 13.

LM 14,15,12(13) restores the calling routine's register 14 and 15.

LM 1,12,24(13) restores registers 1 through 12 of the calling routine.

BR 14 returns control to the calling routine.