IMS DB ONLINE TRAINING – DAY 4

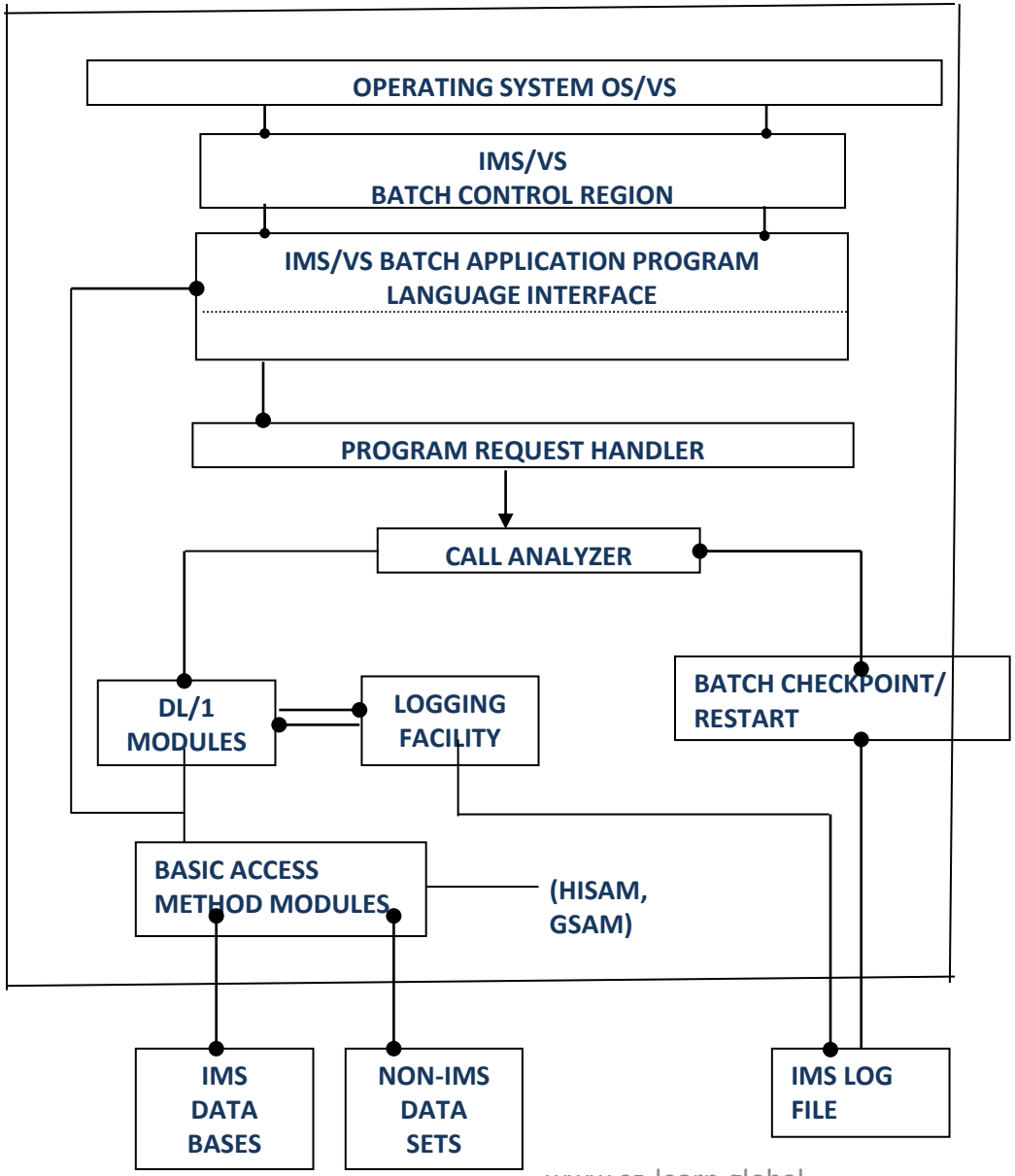
Hierarchical Structure Processing Rules

Accessing hierarchical data structures in correct sequence is guided by three rules:-

- **Top to bottom.** Determine whether the segment you are presently accessing has any unprocessed dependents.
- **Left to right.** If the segment you are accessing has dependents, you process the left-most unprocessed dependent next.
- **Front to back.** If the segment you are accessing has no dependents (or has no more unprocessed dependents), check to see if it has twins. If it does, process the twins next in a front-to-back sequence.

At any point, all three rules must be applied before going on to another segment. When you have satisfied rule three, you will have finished one hierarchic path in the structure. Then move up one level in the structure and reapply the rules.

IMS BATCH ENVIRONMENT



IMS BATCH ENVIRONMENT

- IMS Region is started by Operating System and loads IMS modules
- Application program must have been previously linked with the language interface while compilation process
- When Application program issues CALLs to IMS DB, then control passes to language interface which then pass control to program request handler
- The call analyzer receives control from the program request handler where, depending upon what type of call has been issued, control is passed to the appropriate call processor module. This activity is logged using the logging facility
- If your program requests a checkpoint, them the batch checkpoint/restart module's services are requested and its information is logged via the logging facility.
- Unlike an IMS online application program, a batch IMS application program can access non – IMS data sets. In that case, your request is passed directly to the basic access method modules for processing.
- Upon termination of your application program, control is passed back to the IMS/VS batch control region and your program is terminated. The control region concludes its processing, and control returns to the operating system that will terminate the IMS/VS batch control region.

DLI – DATA LANGUAGE INTERFACE

- It is a set of programs interfacing Application program and the Database

- It establishes connectivity between the Application program and the database and assist program in transferring data to and from the database

COBOL-IMS DB PROGRAMMING

ENTRY Statement

ENTRY Statement establishes connectivity between the Application program and the database thru PCB of PSB.

- It's the first executable statement in the program immediately following procedure division.
- Addresses of IMS DBs are passed to the program when this statement is executed by ZOS.
- Application program returns control back to IMS region by issuing a GOBACK.

COBOL-IMS DB PROGRAMMING

ENTRY Statement

ENTRY 'DLITCBL' USING PCBMASK1, PCBMASK2,

(or)

PROCEDURE DIVISION USING PCBMASK1, PCBMASK2,

PCBs are passed in the same order they are listed in the PSB.

For BMP, IO-PCB must be the first as given below followed by PCBMASKs

ENTRY 'DLITCBL' USING IO-PCB, PCBMASK1, PCBMASK2,

Note: We need to code 1 PCBMASK for every PCB defined in PSB.

COBOL-IMS DB PROGRAMMING

CBLTDLI STATEMENT

CBLTDLI is an I/O for IMS DB. IMS Services are requested via a DLI call. In a COBOL program, CALL 'CBLTDLI' is used whenever a DLI service is required

CBLTDLI accepts parameters like

- Function code
- Status codes (PCBMASK)
- I-O Area
- Segment Search Arguments (Keys/Search fields)

to do the database operations

COBOL-IMS DB PROGRAMMING

GOBACK STATEMENT

GOBACK sends control back to DLI that performs closing of IMS DB properly and then terminate the Batch program.

Note: STOPRUN should never be used as it abruptly stops the execution of the program and doesn't sends control back to IMS.

CBLTDLI CALL

CALL 'CBLTDLI' USING PARM-COUNT, FUNCTION, PCB-MASK, IO-AREA, SSA.

(OR)

CALL 'CBLTDLI' USING FUNCTION, PCB-MASK, IO-AREA, SSA.

PARM-COUNT: How many parameters follows the Parm count field

FUNCTION: Type of call (Get, Insert, Replace, Delete)

IO-AREA: Denotes the working storage structure (or) copybook of segment fetched. This storage used to send data to and from the database. Length will be exactly equal to the length of the segment.

SSA: Segment Search Argument - Used to denote the search criteria

PLITDLI for PLI, ASMTDLI for Assembler

DLI FUNCTION

Function Code (4 byte) : This is the first parameter in the list. This contains the file operation that we are intended to do against the database like Read, Insert, Update or Delete.

DL/I Function code	Function Description
GU	Get Unique – Allows direct retrieval of segment (Random Processing)
GHU	Get Hold Unique – Allows direct retrieval of a segment with segment locking.
GN	Get Next – Allows sequential retrieval of a segment.
GHN	Get Hold Next – Allows Sequential retrieval of a segment with segment locking.
GNP	Get Next within Parent – Allows sequential retrieval of a segment under the previously retrieved parent segment.
GHNP	Get Hold Next within Parent – Allows sequential retrieval of a segment under the previously retrieved parent segment with segment locking.
ISRT	Insert – Allows the adding of a segment occurrence.
REPL	Replace – Allows the data in a segment to be replaced by new data.
DLET	Delete – Allows the deletion of a segment occurrence and all segments below it in the hierarchy.

PCBMASK

This is the Second parameter in the list. After the execution of DLI call, IMS subsystem fills this structure with the details of execution. PCBSTATUS is an important field that is used to check whether the call is Successful or not.

```
01 PCBMASK.
    05 DBD-NAME          PIC X(08).  -> Contains DBD Name
     05 SEG-ID           PIC X(02).  -> Contains current Segm Lvl
     05 STATS            PIC X(02).  -> Contains Status code
     05 PROCOPT           PIC X(04). -> Processing Options
     05 FILLER            PIC X(04).
     05 SEGMENT-NAME      PIC X(08).  -> Contains Segment Name
     05 LENGTH-FDBK       PIC S9(05) COMP.  -> Length of key
                                               retrieved
     05 NUMBER-SENSEGS    PIC S9(05) COMP.     -> No.of Sensitive SEGM
     05 KEY-FDBK-AREA     PIC X(21).           -> Longest retrieved key
                                                      value
```

I-O AREA

(User defined Structure): This is the third parameter in the list. This is the structure of the segment occurrence that we are intended to operate. It basically contains the data to be passed or retrieved from the database.

It denotes the working storage structure (or) copybook of segment fetched. This storage used to send data to and from the database. Length will be exactly equal to the length of the segment.

SSA (SEGMENT SEARCH ARGUMENT)

This is the fourth parameter in the list and is optional. Used to improve the search criteria to retrieve specific segment type or segment occurrence very quickly. There can be 1 SSA every level and hence a total of maximum 15 SSAs are allowed in a single CALL statement.

Types of SSA:-

Unqualified - Specifies only the Segment Type to be retrieved and not the segment Occurrence (actual data). It is 9 bytes long.
Used for Sequential processing as we are not mentioning any particular value

Qualified - Specifies the Segment Type along with a condition (Field Name, Relational Operator, Value)
Used for Random processing as we are searching with a particular value in a database

Executing an IMS Program

Under an IMS Region, batch initialization module DFSRRC00 is invoked via JCL which in turn loads the Application program and the DLI modules required to serve it.

Under IMS subsystem, a batch DLI program executes as a subprogram

It is a standard that program name is same as psb name. These are specified in PARM operand of EXEC statement.

Executing an IMS Program

IMSVS.RESLIB – Contains IMS load modules that make up IMS software

IMSVS.PGMLIB – Contains Application program load modules that are specified in STEPLIB/JOBLIB

IMSVS.DBDLIB – Contains load module of DBDs

IMSVS.PSBLIB – Contains load module of PSBs

IMSVS.MACLIB – Contains definition of macros used in DBDGEN and PSBGEN process

IMSVS.PROCLIB – Contains IMS catalogued procedures supplied with IMS

IMSVS.ACBLIB – Used to store combination of DBD and PSB to before an Application program is executed

DLI FUNCTION – GN CALL

This call does sequential processing and retrieves the next segment occurrence from the current cursor position. This follows the hierarchy in the logical view of database given in PCB.

If this call is given as a first call of a program, it retrieves the first root segment occurrence of the database. Further GN call retrieves any child occurrences in a hierarchical fashion.

This call by default sets parentage to whichever segment is retrieved.

GN call with a qualified SSA is like Skip-Sequential processing.

Status Codes

Blank - Denotes successful retrieval of segment occurrence

Without any SSA

GA – Cursor move up one level across different Segment types

GK – Cursor move within same level across different Segment types

GB - End of Database is reached (No more Segment Occurrences to be read)

DLI FUNCTION – ISRT CALL

Loading an Empty Database

Loading should be done in increasing order of keys (if present) and should be in hierarchical order.

PROCOPT in PCBMASK should be LS (Load Sequential).

Unqualified SSA should be used with the segment name to let DLI know to establish the parent child relationship and levels.

Status Codes

Blank - Denotes successful insertion of Segment occurrence

LB – Segment already exists (when you try to load same segment twice)

LC – Key value out of sequence (Segments that are loaded are not in hierarchical sequence)

LD – No parent for the dependent segment to be loaded. We cannot load a child segment unless the parent has been loaded

LE – Segment types out of sequence

End of Day 4