



Introducción al JCL y a SDSF

Roberto Gómez Cárdenas
rogomez@itesm.mx

Lámina 1 Roberto Gómez C.



JCL

- Job Control Language (JCL) le indica al sistema que programa ejecutar, seguido de una descripción de las entradas y salidas del programa.
- ¿Qué puedo hacer con JCL?
 - Someter un trabajo al sistema operativo.
 - Solicitar recursos necesarios para correr un trabajo.
 - Controlar el sistema para procesar el trabajo.
- ¿Qué se necesita para escribir JCL?
 - Acceso al mainframe.
 - Un ID válido TSO.

Lámina 2 Roberto Gómez C.



¿Qué es un trabajo?

- Algo que se desea completar con la ayuda de la computadora
 - p.e. copiar un data set, ejecutar un programa o procesar varios “job steps”
- Necesario proporcionar la información que requiere el trabajo e indicar a la computadora que hacer con esta información.
- Un trabajo consiste de enunciados que controlan la ejecución de un programa o procedimiento, solicita recursos y define entradas y/o salidas.

Lámina 3 Roberto Gómez C.



¿Qué información incluye?

- El programa o procedimiento a ser ejecutado
- Datos de entrada
- Datos de salida
- Reportes de salida
- También proporciona información acerca de a quien pertenece el trabajo y a que cuenta cargar el trabajo.

Lámina 4 Roberto Gómez C.



Sintaxis básica de enunciados JCL (1)

//NOMBRE OPERACIÓN OPERANDO,OPERANDO,OPERANDO, COMENTARIOS

campo de nombre	campo de operación	campo de operando	campo comentario
--------------------	-----------------------	-------------------	------------------

- **campo nombre**
 - identifica al enunciado de tal forma que otros enunciados o el sistema pueda hacer referencia a él.
 - debe ir después de la segunda diagonal
 - puede variar de 1 a 8 caracteres en longitud y solo puede contener cualquier carácter alfanumérico o @ \$ #
- **campo operación**
 - especifica el tipo de enunciado: JOB, EXEC, DD o un comando de operador

Lámina 5 Roberto Gómez C.



Sintaxis básica de enunciados JCL (2)

//NOMBRE OPERACIÓN OPERANDO,OPERANDO,OPERANDO, COMENTARIOS

campo de nombre	campo de operación	campo de operando	campo comentario
--------------------	-----------------------	-------------------	------------------

- **campo operando**
 - contiene parámetros separados por comas
 - parámetros están compuestos de keywords y variable
- **campo comentarios**
 - opcional
 - pueden extenderse a través de la columna 80 y pueden ser incluidos si existe un campo de operando

Lámina 6 Roberto Gómez C.



Los enunciado de JCL

- **JOB:** proporciona un nombre al sistema para el trabajo batch.
 - Opcionalmente puede incluir información de bitácoras y algunos parámetros.

- **EXEC:** proporciona el nombre del programa a ejecutar.
 - Pueden existir múltiples enunciados EXEC en un trabajo.
 - Cada enunciado EXEC dentro del mismo job es un job step (paso del trabajo)

- **DD (Data Definition):** proporciona entradas y salidas al programa a ejecutar dentro del enunciado EXEC.
 - Este enunciado liga un data set u otro dispositivo de E/S a un ddname codificado en el programa.
 - Los enunciados DD estan asociados con un job step en particular.

Lámina 11 Roberto Gómez C.



Ejemplo JCL

```
//MYJOB    JOB 1
//MYSORT   EXEC PGM=SORT
//SORTIN   DD DISP=SHR,DSN=ZPROF.AREA.CODES
//SORTOUT  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
           SORT FIELDS=(1,3,CH,A)
/*
```

Instrucción	Significado
MYJOB	Nombre trabajo sistema asocia con esta carga de trabajo
MYSORT	Nombre del paso que le indica al sistema ejecutar el programa SORT
SORTIN	En el enunciado DD es el ddname. Este es codificado dentro del programa SORT como entrada. El nombre del data set es ZPROF.AREA.CODES. El data set puede ser compartido con otros procesos (DISP=SHR).

Lámina 12 Roberto Gómez C.



Continuación ejemplo JCL

```
//MYJOB    JOB 1
//MYSORT   EXEC PGM=SORT
//SORTIN   DD DISP=SHR,DSN=ZPROF.AREA.CODES
//SORTOUT  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
           SORT FIELDS=(1,3,CH,A)
/*
```

Instrucción	Significado
SORTOUT	Es el ddname de la salida del programa SORT
SYSOUT	SYSOUT=* especifica que se debe enviar mensajes de salida al área de impresión de salida de JES. Es posible enviar la salida a un data set.
SYSIN	DD * es otro enunciado de entrada. Especifica que lo que se sigue son datos o enunciados de control. En este caso, es la instrucción sort que indica al programa SORT que campos de los datos SORTIN serán ordenados.

Lámina 13
Roberto Gómez C.



Los parámetros del ejemplo

nombre del trabajo

nombre del paso

```
//MYJOB    JOB 1
//MYSORT   EXEC PGM=SORT
//SORTIN   DD DISP=SHR,DSN=IBMUSER.AREA.CODES
//SORTOUT  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSIN    DD *
           SORT FIELDS=(1,3,CH,A)
/*
```

nombre entrada programa

nombre salida programa

especifica si la entrada serán datos o enunciados de control

donde enviar mensajes salida del sistema

Lámina 14
Roberto Gómez C.



El enunciado JOB

- Informa al sistema operativo del inicio de un trabajo.
- Proporciona la información de accounting necesaria y establece los parámetros de ejecución.
- Cada trabajo debe comenzar con un enunciado JOB
 - //nombre-trabajo JOB
- El nombre-trabajo es un nombre descriptivo asignado al trabajo por el usuario que se despliega en la salida del trabajo
 - cualquier nombre de 1 a 8 caracteres alfanumérico y nacionales (\$ @ #)
 - primer caracter debe ser alfabético o nacional

Lámina 15 Roberto Gómez C.



Operandos adicionales del enunciado JOB

//jobname JOB USER=userid, TIME=m, MSGCLASS=class, NOTIFY=userid

- USER=userid
 - identifica, al sistema, el usuario ejecutando el trabajo
- TIME=m
 - total de minutos máquina permitido a un trabajo ejecutar
- MSGCLASS=class
 - clase de salida para el log del trabajo
- NOTIFY=userid
 - usuario que recibirá un mensaje TSO cuando el trabajo termine

Lámina 16 Roberto Gómez C.



Parámetros JOB (1)

- **REGION=**
 - solicita recursos específicos de memoria a ser asignados al trabajo
- **NOTIFY=**
 - envía notificación de que el trabajo se completó a un determinado usuario
- **USER=**
 - especifica que el trabajo se ejecuta bajo la autoridad del ID de usuario especificado
- **TYPRUN=**
 - retarda o detiene la ejecución, para ser liberado después
- **CLASS=**
 - dirige un enunciado statement para ser ejecutado en una cola de entrada en particular

Lámina 17 Roberto Gómez C.



Parámetros JOB (2)

- **MSGCLASS=**
 - dirige la salida del trabajo a una cola de salida en particular
- **MSGLEVEL=**
 - controla el número de mensajes del sistema a ser recibidos
- **EJEMPLO:**
 - `//MYJOB JOB 1,NOTIFY=&SYSUID,REGION=6M`

Lámina 18 Roberto Gómez C.



El enunciado EXEC

- Identifica el programa de aplicación o catalogado o procedimiento de flujo que este trabajo va a ejecutar y le indica al sistema como procesar el trabajo.
- Sintaxis

```
//stepname EXEC procedure,REGION=####K
```

```
//stepname EXEC PGM=program,REGION=####K
```

Lámina 19 Roberto Gómez C.



Opciones enunciado EXEC

- Sintaxis
 - //stepname EXEC procedure,REGION=####K
 - //stepname EXEC PGM=program,REGION=####K
- Donde
 - stepname: palabra opcional de 1 a 8 caracteres usada para identificar el trabajo
 - EXEC: indica que se desea invocar un programa o procedimiento catalogado
 - procedure: nombre el procedimiento catalogado a ser ejecutado
 - program: nombre del programa a ser ejecutado
 - REGION=####K: monto del almacenamiento a asignar al trabajo

Lámina 20 Roberto Gómez C.



Otros parámetros EXEC

- PARM=
 - parámetros conocidos por y pasados al programa
- COND=
 - operadores lógicos para controlar la ejecución de otros EXEC en el trabajo
 - existen enunciados IF, THEN, ELSE, que son superiores al uso de COND; sin embargo muchos trabajos JCL usan este enunciado.
- TIME=
 - impone un límite de tiempo
- Ejemplo
 - //MYSTEP EXEC PGM=SORT

Lámina 21 Roberto Gómez C.



Programas a ejecutar

- //EXEC PGM=pgmname
- Es una versión compilada y ligada de un conjunto de enunciados de lenguaje que están listos para llevar a cabo una tarea determinada.
- También conocido como un ejecutable load module.
- Debe residir en un dataset particionado.

Lámina 22 Roberto Gómez C.



El enunciado Data Definition DD

- Un enunciado DD debe ser incluido después del enunciado EXEC para cada data set usado en el paso.
- El enunciado proporciona el nombre del data set, unidad de E/S, probablemente un volumen específico a usar, y la disposición del data set.
- El sistema asegura que los dispositivos de E/S puedan ser asignados al trabajo antes que la ejecución inicie.
- El enunciado también puede proporcionar al sistema información varia acerca del data set
 - su organización, longitud registro, bloqueo, etc.

Lámina 23 Roberto Gómez C.



Sintaxis del enunciado DD

- Sintaxis
 - //ddname DD operando,operando,etc.
- ddname
 - un nombre de 1 a 8 caracteres proporcionado al enunciado DD
- DD
 - identificador del DD
- operando
 - parámetros usados para definir el dataset de entrada o de salida

Lámina 24 Roberto Gómez C.



Parámetros enunciado DD

- Enunciado DD cuenta con más parámetros que los enunciados JOB y EXEC.
- Entre los más comunes encontramos
 - DSN=
 - el nombre del data set
 - puede incluir creación de data sets temporales o nuevos, así como una referencia al nombre del data set
 - DISP=
 - disposición de data sets
 - si el data set existe o necesita ser creado o si puede ser compartido por más de un trabajo
 - SPACE=
 - cantidad de espacio en disco requerido por el nuevo data set

Lámina 25 Roberto Gómez C.



Parámetros enunciado DD (2)

- SYSOUT=
 - define la ubicación de impresión (y la cola de salida o el data set)
- VOL=SER=
 - nombre del volumen, disco o cinta
- UNIT=
 - disco sistema, tipo especial dispositivo, o esotérico (nombre local)
- DEST=
 - rutea la salida a un destino remoto
- LABEL=
 - se espera la etiqueta de una cinta
 - una cinta puede almacenar múltiples data sets
 - cada data set en la cinta es una posición en el archivo
 - el primer data set en la cinta es el archivo 1

Lámina 26 Roberto Gómez C.



Parámetros enunciado DD (3)

- DCB=
 - data set control block,
 - tiene varios subparámetros
 - LRECL= longitud de registro lógico, número de bytes/característica en cada registro
 - RECFM= formato del registro, fijo, lógico, variable, etc
 - BLOCKSIZE= se almacenan registros en un bloque de este tamaño
 - DSORG= organización del data set (secuencial, particionado, etc)
- *,DLM=
 - todo lo que sigue son datos de entrada (incluyendo //) hasta que dos caracteres alfanuméricos o caracteres especiales especificados se encuentren en la columna 1

Lámina 27 Roberto Gómez C.



Parámetros enunciado DD (4)

- DUMMY =
 - resulta en una entrada nulo o tirar datos escritos a este ddname.
- *
 - siguen datos de entrada o enunciados de control.
 - es un método para pasar datos a un programa desde el stream JCL.
 - los caracteres /* marcan el final de los datos.

Lámina 28 Roberto Gómez C.



Disposición de Data Sets

- La función DISP es una de los más importantes enunciados de DD.
- Entre otras cosas el parámetro DISP advierte al sistema acerca del data set necesario para un trabajo, previniendo conflictos de uso del mismo data set por parte de otros trabajos.
- El parámetro completo cuenta con los siguientes campos
 - DISP=(status, normal end, abnormal end)
 - DISP=(status, normal end)
 - DISP=status
 - donde status puede ser NEW, OLD, SHR o MOD

Lámina 29 Roberto Gómez C.



El parámetro status

- NEW
 - indica que un nuevo data set va a ser creado.
 - el trabajo tiene acceso exclusivo al data set.
- OLD
 - indica que el data set ya existe y que el trabajo tiene acceso exclusivo mientras esta corriendo.
- SHR
 - indica que el data set ya existe y que varios trabajos puede tener acceso a él mientras están corriendo.
 - todos los trabajos deben especificar SHR.
- MOD
 - data set ya existe y el trabajo actual tiene acceso exclusivo.
 - si se abre el data set para salida, la salida será añadida al final del data set.

Lámina 30 Roberto Gómez C.



Los parámetros normal y abnormal end

- Indica que hacer cuando el data set (la disposición) si el trabajo termina normal o anormalmente.
- Opciones para los dos parámetros
 - DELETE: borra y descataloga el data set
 - KEEP: guarda pero no cataloga el data set
 - CATLG: guarda y cataloga el data set
 - UNCATLG: guarda el data set pero lo descatalogaba
 - PASS: permite que un “job step” posterior especifique la disposición final.
- La disposición por defecto (para un final normal y anormal) son dejar el data set como estaba antes de que el trabajo empezara.

Lámina 31 Roberto Gómez C.



Creando nuevos data sets

- Si el parámetro DISP tiene un valor de NEW se debe proporcionar más información incluyendo
 - Un nombre del data set.
 - El tipo de dispositivo para el data set.
 - Un “volser” si se trata de un disco o una cinta etiquetada.
 - Si se usa un disco, la cantidad de espacio para ser asignado.
 - Si se trata de un data set, el tamaño del directorio debe ser especificado.
 - Opcionalmente se pueden especificar parámetros DCB
 - alternativamente el programa que escribirá el data set puede proporcionar dichos parámetros

Lámina 32 Roberto Gómez C.



Otros parámetros Data Set

- Volser
 - formato para un enunciado DD es VOL=SER=xxxxxx, donde xxxxxx es el volser
- Tipo dispositivo (device type)
 - existen varias formas de hacer esto
 - UNIT=xxxx es la más común
 - xxxx puede ser un dispositivo IBM (como un 3390) o una dirección específica de dispositivo (p.e. 300), o un número esotérico definido por la instalación (como un SYSDA)
- Nombre miembro (member name)
 - una librería (o un PDS) puede ser tratado como un data set por varias aplicaciones y utilerías

Lámina 33 Roberto Gómez C.



El parámetro SPACE del enunciado DD

- Requerido para asignar data sets en un DASD.
- Identifica el espacio requerido para el data set.
- Antes que data set será creado en el disco, el sistema debe saber cuanto espacio requiere el data set y como se mide este espacio.
- En su formato básico se cuenta con dos parámetros:
 - unidad de medida
 - cantidad de espacio

Lámina 34 Roberto Gómez C.



Parámetros básicos SPACE

- Unidad de medida
 - tracks, cilindros o el tamaño de bloque promedio
- Cantidad de espacio: tres subparámetros
 - tamaño de la primera extensión
 - tamaño de cada extensión secundaria (opcional)
 - si existe, indica que un data set particionado (librería) esta siendo creado

Lámina 35 Roberto Gómez C.



Ejemplos formatos parámetro SPACE

- Existen varios formatos y variantes, por ejemplo
 - SPACE=(TRK,10) 10 tracks, no extensiones secundarias
 - SPACE=(TRK,(10,5)) 10 tracks primarios, 5 tracks por cada extensión secundaria
 - SPACE=(CYL,5) posible usar CYL (cilindros) en lugar de tracks
 - SPACE=(TRK,(10,5,8)) PDS con 8 bloques de directorios
 - SPACE=(1000,(50000,10000))
50,000 registros primarios con 1,000 bytes cada uno

Lámina 36 Roberto Gómez C.



Concatenación enunciados DD

- Un nombre ddname puede contener múltiples enunciados DD.
 - esto se conoce como concatenación
- El siguiente JCL indica que los data sets son concatenados

```
//DATAIN DD DISP=OLD,DSN=MY.INPUT1
//      DD DISP=OLD,DSN=MY.INPUT2
//      DD DISP=SHR,DSN=YOUR.DATA
```

cuando el programa de aplicación lee el final de MY.INPUT1, el sistema abre automáticamente MY.INPUT2 y empieza a leerlo.

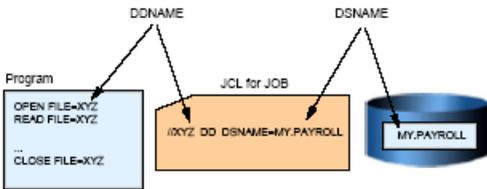
- Concatenación solo aplica a data sets de entrada.
- Los data sets son automáticamente procesados en secuencia.

Lámina 37
Roberto Gómez C.



Nombres simbólicos archivos

- Z/OS usa nombres archivos simbólicos.
- Aplica una redirección entre un nombre relacionado con un data set y el data set actual usado durante la ejecución del programa.



- Cuando el programa se escribe, nombre XYZ es seleccionado para referenciar al data set.
- El programa puede ser compilado y almacenado como ejecutable.
- Cuando alguien quiere correr el ejecutable, un JCL debe ser usado para relacionar el nombre XYZ al data set.
- El nombre simbólico usado en el programa es un DDNAME y el nombre real del data set es un DSNAME.

Lámina 38
Roberto Gómez C.



DDNAMES Reservados

- Posible seleccionar casi cualquier nombre para un DD
 - recomendable usar un nombre significativo dentro límite de 8 caracteres
- Existen algunos nombres de DD reservados que un programador no puede usar
 - //JOB LIB DD ...
 - //STEP LIB DD ...
 - //JOB CAT DD ...
 - //STEP CAT DD ...
 - //SYS ABEND DD ...
 - //SYS DUMP DD ...
 - //SYS MDUMP DD ...
 - //CEEDUMP DD ...

Lámina 39 Roberto Gómez C.



Procedimientos JCL (PROCs)

- Algunos programas y tareas requieren de una gran cantidad de JCL que un usuario puede introducir con facilidad
 - estas funciones pueden residir en librerías de procedimientos
- Un miembro de la librería de procedimiento contiene parte del JCL para una determinada tarea, usualmente la parte fija, no alterable del JCL
 - el usuario del procedimiento proporciona la parte variable del JCL
- Un procedimiento JCL es como un macro.
- También conocido como procedimiento catalogado (cataloged procedure).

Lámina 40 Roberto Gómez C.



Ejemplo procedimiento JCL (PROC)

```
//MYPROC
//MYSORT
//SORTIN
//SORTOUT
//SYSOUT
//
```

```
PROC
EXEC PGM=SORT
DD DISP=SHR,DSN=&SORTDSN
DD SYSOUT=*
DD SYSOUT=*
PEND
```

- Enunciados PROC y PEND son únicos a los procedimientos
 - identifican principio y fin del procedimiento
- PROC es precedido por una etiqueta o nombre
 - en el ejemplo es MYPROC
- Las variables de sustitución de JCL son la razón por la cual JCL PROCs son usados
 - en ejemplo &SORTDSN es la única variable

Lámina 41
Roberto Gómez C.



Llamando al procedimiento (PROC)

```
//MYJOB    JOB 1
/*-----*
//MYPROC   PROC
//MYSORT   EXEC PGM=SORT
//SORTIN   DD DISP=SHR,DSN=&SORTDSN
//SORTOUT  DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//         PEND
/*-----*
//STEP1    EXEC MYPROC, SORTDSN=ZPROF.AREA.CODES
//SYSIN    DD *
           SORT FIELDS=(1,3,CH,A)
```

- Cuando MYJOB es enviado a ejecución se llevan a cabo la sustitución.
 - Se debe proporcionar valor de &SORTDSN
 - La variable &SORTDSN y su valor debe ser colocados en una línea diferente, continuación del enunciado EXEC
 - La secuencia //SYSIN DD * seguida del enunciado SORT serán añadidos al final del JCL substituido

Lámina 42
Roberto Gómez C.



Enunciado de sobre-escritura en JCL PROC

- Cuando un enunciado entero JCL PROC necesita ser reemplazado, un enunciado de reemplazo JCL PROC puede ser usado.
- El enunciado cuenta con la siguiente forma

`//stepname.ddname DD ...`

Lámina 43 Roberto Gómez C.



Ejemplo reemplazo

- Reemplazo del enunciado SORTOUT DD en MYPROC
- SORTOUT es dirigido a un data set secuencial recién creado.

```

//MYJOB JOB 1
/*-----*
//MYPROC      PROC
//MYSORT      EXEC PGM=SORT
//SORTIN      DD DISP=SHR,DSN=&SORTDSN
//SORTOUT     DD SYSOUT=*
//SYSOUT      DD SYSOUT=*
//           PEND
/*-----*
//STEP1 EXEC MYPROC,SORTDSN=ZPROF.AREA.CODES
//MYSORT.SORTOUT DD DSN=ZPROF.MYSORT.OUTPUT,
//              DISP=(NEW,CATLG),SPACE=(CYL,(1,1)),
//              UNIT=SYSDA,VOL=SER=SHARED,
//              DCB=(LRECL=20,BLKSIZE=0,RECFM=FB,DSORG=PS)
//SYSIN DD *
SORT FIELDS=(1,3,CH,A)

```

Lámina 44 Roberto Gómez C.



Sometiendo un trabajo a procesamiento batch

- ISPF editor command line
 - SUBmit y presionar Enter.
- ISPF command shell
 - SUBmit 'USER.JCL'
 - el data set es secuencial.
- ISPF command line
 - TSO SUBmit 'USER.JCL'
 - donde el data set es secuencial.
- ISPF command line
 - TSO SUBmit 'USER.JCL(MYJOB)'
 - donde el the data set es una librería o un data set particionado conteniendo un miembro MYJOB.
- TSO command line
 - SUBmit 'USER.JCL'

Lámina 45
Roberto Gómez C.



Ejemplo creación JCL (1)

- JOB statement
 - Crear un miembro usando editor ISPF
 - Crear enunciados JCL
 - JOB statement
 - Accounting information
 - Execution classes

```

EDIT      MIRIAM.PRIVATE.JCLLIB(JOB1) - 01.05          Columns 00001 00072
Command ==>                                         Scroll ==> HOLE
***** Top of Data *****
000001 //MIRIAM2 JOB TS,MIRIAM,NOTIFY=8SYSUID,MSGCLASS=T,
000002 //MSGLEVEL=(1,1),CLASS=A
000003 //STEP1 EXEC PGM=IEFBR14
000004 /*-----*
000005 /* THIS IS AN EXAMPLE OF A NEW DATA SET ALLOCATION
000006 /*-----*
000007 //NEWDD DD      DSN=MIRIAM.IEFBR14.TEST.NEWDD,
000008 //              DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
000009 //              SPACE=(CVL,(10,10,45)),LRECL=80,BLKSIZE=3120
***** Bottom of Data *****

```

Lámina 46
Roberto Gómez C.



Ejemplo creación JCL (2)

- EXEC statement
 - Region size

```

EDIT      MIRIAM.PRIVATE.JCLLIB(JOB1) - 01.05          Columns 00001 00072
Command ---> _                                         Scroll ---> HALE
***** Top of Data *****
000001 //MIRIAM2 JOB 19,MIRIAM,NOTIFY=8SYSUID,MSGCLASS=T,
000002 // MSGLEVEL=(1,1),CLASS=A
000003 //STEP1 EXEC PGM=IEFBR14
000004 //-----*
000005 // * THIS IS AN EXAMPLE OF A NEW DATA SET ALLOCATION
000006 //-----*
000007 //NEWDD DD   DSN=MIRIAM.IEFBR14.TEST.NEWDD,
000008 //           DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
000009 //           SPACE=(CYL,(10,10,45)),LRECL=80,BLKSIZE=3120
***** Bottom of Data *****

```

Lámina 47
Roberto Gómez C.



Ejemplo creación JCL (3)

- DD statement
 - DD name (referenciado en el programa)
 - DSN= (nombre data set, tal y como esta catalogado en disco)

```

EDIT      MIRIAM.PRIVATE.JCLLIB(JOB1) - 01.05          Columns 00001 00072
Command ---> _                                         Scroll ---> HALE
***** Top of Data *****
000001 //MIRIAM2 JOB 19,MIRIAM,NOTIFY=8SYSUID,MSGCLASS=T,
000002 // MSGLEVEL=(1,1),CLASS=A
000003 //STEP1 EXEC PGM=IEFBR14
000004 //-----*
000005 // * THIS IS AN EXAMPLE OF A NEW DATA SET ALLOCATION
000006 //-----*
000007 //NEWDD DD   DSN=MIRIAM.IEFBR14.TEST.NEWDD,
000008 //           DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
000009 //           SPACE=(CYL,(10,10,45)),LRECL=80,BLKSIZE=3120
***** Bottom of Data *****

```

Lámina 48
Roberto Gómez C.



Ejemplos sometiendo JCL para procesamiento

```

EDIT ---- userid.SORT.JCL ----- LINE 00000000 COL 00
COMMAND ==> SUBMIT                               SCROLL ==>
***** TOP OF DATA *****
//userid JOB 'accounting data',
      .
      .

```

TSO/E command line:

```

----- TSO COMMAND PROCESSOR -----
ENTER TSO COMMAND OR CLIST BELOW:

==> SUBMIT 'userid.SORT.JCL'

ENTER SESSION MANAGER MODE ==> NO   (YES or NO)

```

After READY mode message:

```

.
.
.
READY
SUBMIT 'userid.SORT.JCL'

```

Lámina 49
Roberto Gómez C.



Entendiendo SDSF

- System Display and Search Facility.
- Después de someter un trabajo, es común usar SDSF para revisar la salida para del termino exitoso, o revisar y corregir errores JCL.
- SDSF permite desplegar salidas retenidas en el área de spool de JES.
- Bastante del trabajo enviado a JES para trabajos en batch (y otros trabajos) nunca es impresa
 - es inspeccionado usando SDSF y borrada o usada conforme se necesita.

Lámina 50
Roberto Gómez C.



Funciones adicionales SDSF

- Ver el sistema de bitácoras y buscar por cualquier string.
- Introducir comandos de sistema
 - en primeras versiones del sistema operativo, solo el operador podía introducir los comandos
- Controlar procesamiento de trabajos (hold, release, cancel, and purge jobs).
- Monitorear trabajos mientras son procesados.
- Desplegar salida trabajo antes de decidir a imprimirlos.
- Controlar el orden en que los trabajos son procesados.
- Controlar el orden en que los trabajos serán impresos.
- Controlar impresoras e iniciadores.

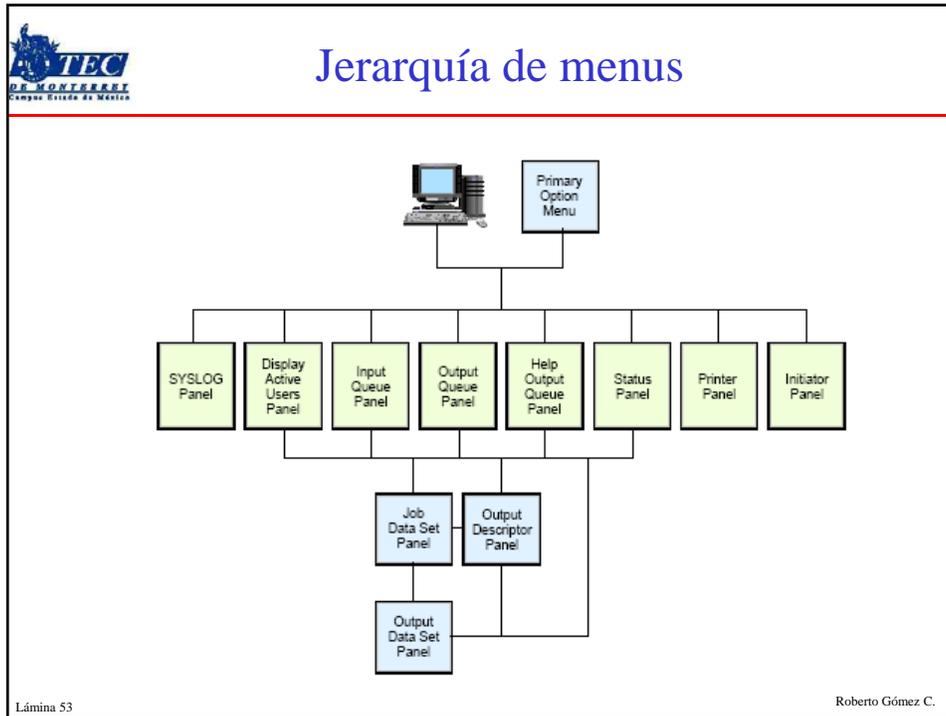
Lámina 51
Roberto Gómez C.



Primer menu de opciones



Lámina 52
Roberto Gómez C.



Viendo los data sets

- Se pueden ver los data sets de salida de JES durante la ejecución del job en batch
 - son almacenados en el JES spool data set
- Se pueden ver los data sets en las siguientes colas
 - I Input
 - DA Execution queue
 - O Output queue
 - H Held queue (cola de retención)
 - ST Status queue
- Para las colas de salida y retención, no se pueden ver los data sets de JES que se configuraron para ser automáticamente purgados a través de asignar MSGCLASS para que no guarde salida alguna.
- Dependiendo en la selección de MSGCLASS en la definición del trabajo, los sysouts pueden ubicarse en la cola de salida o en la de retención.

Lámina 54 Roberto Gómez C.



Ejemplo vista archivos salida JES2

```

Display Filter View Print Options Help
-----
SDSF HELD OUTPUT DISPLAY ALL CLASSES LINES 44          LINE 1-1 (1)
COMMAND INPUT ==>                                     SCROLL ==> PAGE
PREFIX-* DEST-(ALL) OWNER-* SYSNAME-
NP  JOBNAME JobID  Owner  Prty C ODisp Dest          Tot-Rec Tot-
?_  MIRIAM2  JOB26044 MIRIAM  144 T HOLD LOCAL          44

```

- Desplegado de una lista de trabajos lanzados y cuya salida fue dirigida a la cola de retención.
 - MSGCLASS = T
- En este caso solo un trabajo ha sido lanzado y ejecutado.
 - Por lo tanto solo se despliega un trabajo en la cola de retención.
- Introducir un comando “?” en la columna NP despliega los archivos de salida generados por el trabajo 7359.

Lámina 55
Roberto Gómez C.



Los data sets del job

```

Display Filter View Print Options Help
-----
SDSF JOB DATA SET DISPLAY - JOB MIRIAM2 (JOB26044)    LINE 1-3 (3)
COMMAND INPUT ==>                                     SCROLL ==> PAGE
PREFIX-* DEST-(ALL) OWNER-* SYSNAME-
NP  DDNAME  StepName ProcStep DSID Owner  C Dest          Rec-Cnt Page
    JESMSG LG JES2          2 MIRIAM T LOCAL          20
    JESJCL  JES2          3 MIRIAM T LOCAL          12
    JESYSMSG JES2        4 MIRIAM T LOCAL          12

```

- Despliega tres ddnames:
 - Mensajes de bitácoras
 - El archivo JES2 JCL
 - El archivo de envío de mensajes de sistema de JES2
- Opción útil cuando se están viendo trabajos con varios archivos dirigidos a SYSOUT y se quiere desplegar un determinado archivo asociado con un paso en específico.
- Para ver todos los archivos, en lugar de un carácter “?”, se debe teclear “S” en la columna NP.

Lámina 56
Roberto Gómez C.



El data set de bitácoras

```

JES2 JOB LOG -- SYSTEM SC64 -- NODE

13.19.24 JOB26044 ---- WEDNESDAY, 27 AUG 2003 ----
13.19.24 JOB26044 IRR010I USERID MIRIAM IS ASSIGNED TO THIS JOB.
13.19.24 JOB26044 ICH70001I MIRIAM LAST ACCESS AT 13:18:53 ON WEDNESDAY,
AUGU
13.19.24 JOB26044 $HASP373 MIRIAM2 STARTED - INIT 1 - CLASS A - SYS SC64
13.19.24 JOB26044 IFF403I MIRIAM2 - STARTED - ASID=0027 - SC64

13.19.24 JOB26044 - --TIMINGS
(MINS.)--
13.19.24 JOB26044 -JOBNAME STEPNAME PROCSTEP RC EXCP CPU SRB
CLOCK
13.19.24 JOB26044 -MIRIAM2 STEP1 00 9 .00 .00
.00
13.19.24 JOB26044 IEF404I MIRIAM2 - ENDED - ASID=0027 - SC64
13.19.24 JOB26044 -MIRIAM2 ENDED. NAME=MIRIAM TOTAL CPU TIME=
13.19.24 JOB26044 $HASP395 MIRIAM2 ENDED

```

Lámina 57
Roberto Gómez C.



Continuación archivo

```

----- JES2 JOB STATISTICS -----
27 AUG 2003 JOB EXECUTION DATE
11 CARDS READ
44 SYSOUT PRINT RECORDS
0 SYSOUT PUNCH RECORDS
3 SYSOUT SPOOL KBYTES
0.00 MINUTES EXECUTION TIME
1 //MIRIAM2 JOB 19,MIRIAM,NOTIFY=&SYSUID,MSGCLASS=T,
// MSGLEVEL=(1,1),CLASS=A
IEFC653I SUBSTITUTION JCL -
19,MIRIAM,NOTIFY=MIRIAM,MSGCLASS=T,MSGLEVE
2 //STEP1 EXEC PGM=IEFBR14
/*-----*
/* THIS IS AN EXAMPLE OF A NEW DATA SET ALLOCATION
/*-----*
3 //NEWDD DD DSN=MIRIAM.IEFBR14.TEST1.NEWDD,
// DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
// SPACE=(CYL,(10,10,45)),LRECL=80,BLKSIZE=3120
4 //SYSPRINT DD SYSOUT=T
/*

```

Lámina 58
Roberto Gómez C.



Fin del archivo

```
ICH70001I MIRIAM  LAST ACCESS AT 13:18:53 ON WEDNESDAY, AUGUST 27, 2003
IEF236I ALLOC. FOR MIRIAM2 STEP1
IGD100I 390D ALLOCATED TO DDNAME NEWDD  DATACLAS (      )
IEF237I JES2 ALLOCATED TO SYSPRINT
IEF142I MIRIAM2 STEP1 - STEP WAS EXECUTED - COND CODE 0000
IEF285I  MIRIAM.IEFBR14.TEST1.NEWD  CATALOGED
IEF285I  VOL SER NOS= SBOX38.
IEF285I  MIRIAM.MIRIAM2.JOB26044.D0000101.?  SYSOUT
IEF373I STEP/STEP1  /START 2003239.1319
IEF374I STEP/STEP1  /STOP 2003239.1319 CPU  OMIN 00.00SEC SRB  OMIN
00.00S
IEF375I JOB/MIRIAM2 /START 2003239.1319
IEF376I JOB/MIRIAM2 /STOP 2003239.1319 CPU  OMIN 00.00SEC SRB  OMIN
00.00S
```

Lámina 59 Roberto Gómez C.