JES INTRODUCTION

JES INTRODUCTION

- JES Introduction
- JES Resource Initialization
- Stages of Job Processing
- JES Shared Spool

¿QUÉ ES EL JES?

° En el sistema operativo z/OS, JES maneja (o administra) las colas de entrada y salida de jobs, y los datos.

° JES maneja los siguientes aspectos del procesamiento batch para z/OS:

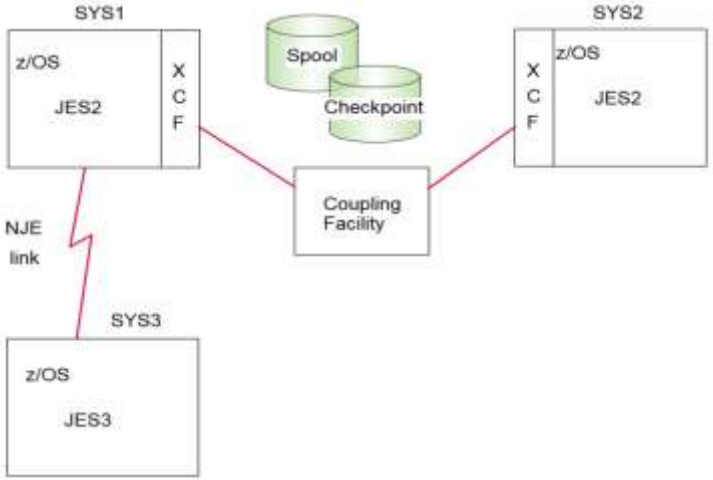
- Recibe jobs en el sistema operativo.
- Los programa para procesarlo por el z/OS.
- Controla su procesamiento de salida.

JES es el componente del sistema operativo que provee manejos suplementarios de funciones de jobs, datos y tareas como la planificación de ejecución, control del flujo del job y spooling.

El z/OS tiene dos versiones de subsistemas de entrada de job: JES2 y JES3.

De éstos dos, el JES2 es el más común de todos.

JOB ENTRY SUBSYSTEM: JES2 / JES3



JES Resource Initialization

° JES Resource Initialization.
° Stages of job processing.
° Shared spool.

JES RESOURCE INITIALIZATION

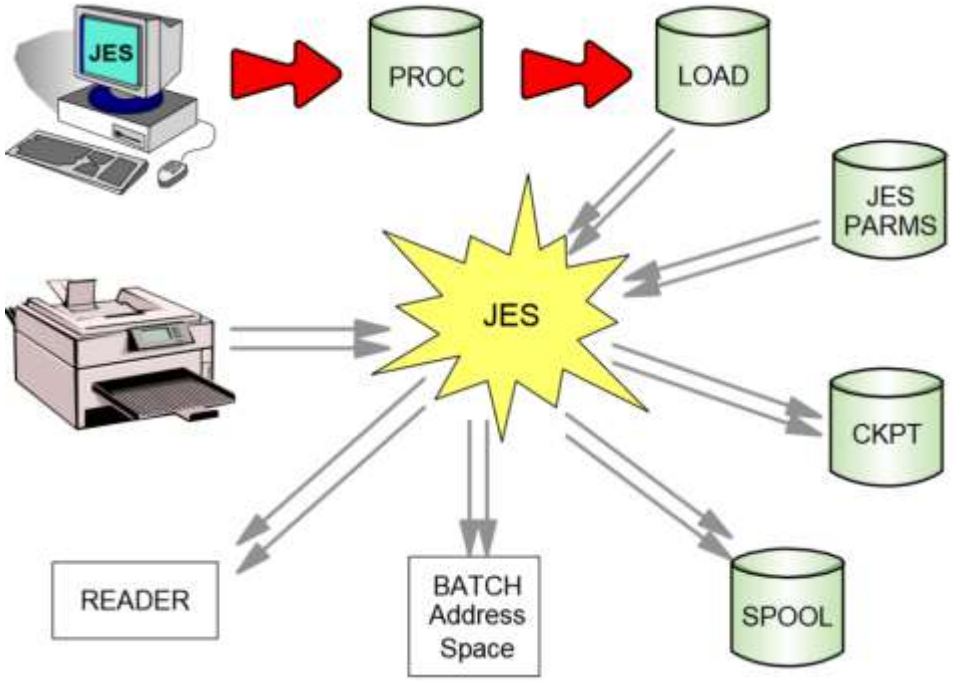
JES Introduction

JES Resource Initialization

Stages of Job Processing

JES Shared Spool

JES RESOURCE INITIALIZATION



¿QUÉ ES UN PROCESAMIENTO BATCH?

° **Muchas de las tareas ejecutando en z/OS consiste de programas llamados** trabajos (jobs) batch.

° Procesamiento Batch **se usa para programas que pueden ejecutarse:**

- Con mínima interacción humana.

- A un tiempo programado o basado en necesidades.

° **Después que una tarea (job) fue enviada al sistema para su ejecución, normalmente no hay otra interacción humana con la tarea hasta que se complete.**

¿QUÉ ES UN PROCESAMIENTO BATCH?

No hay una contrapartida bien directa del procesamiento batch (en lote) del z/OS con los sistemas PC o UNIX.

Procesamiento batch es para aquellos programas usados frecuentemente que se pueden ejecutar con una mínima interacción humana.

Típicamente se ejecutan a una hora planificada o básicamente cuando es necesario.

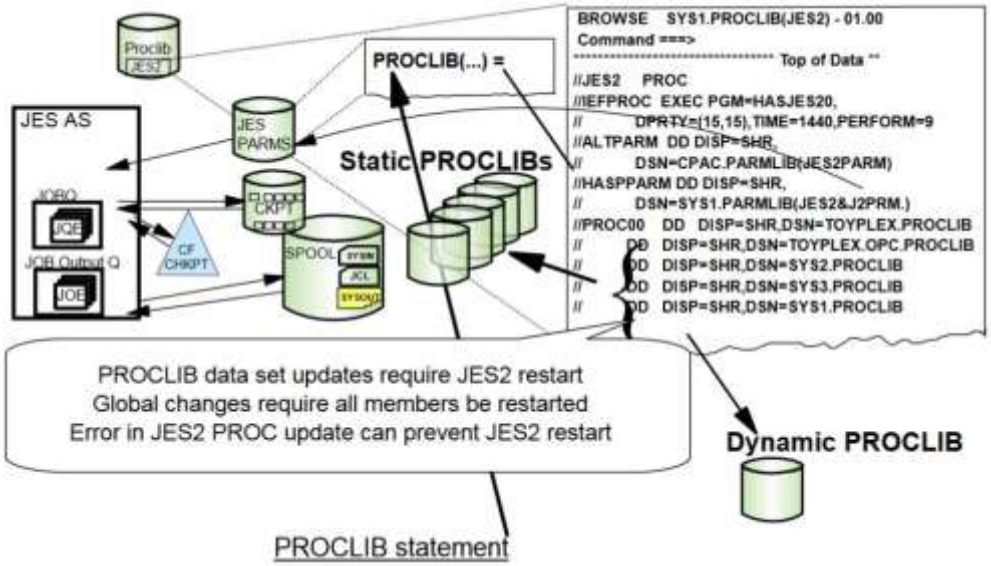
(tal vez la comparación más cercana es con procesos ejecutando debido a un comando AT o CRON en UNIX, aunque existen diferencias).

La tarea ejecutando en z/OS se llama job, y el lenguaje de control de tareas (job control language o JCL) se usa para describir ciertos atributos del job al z/OS.

Un job batch en z/OS consiste en programas que ejecutan en ambientes descriptos por el JCL.

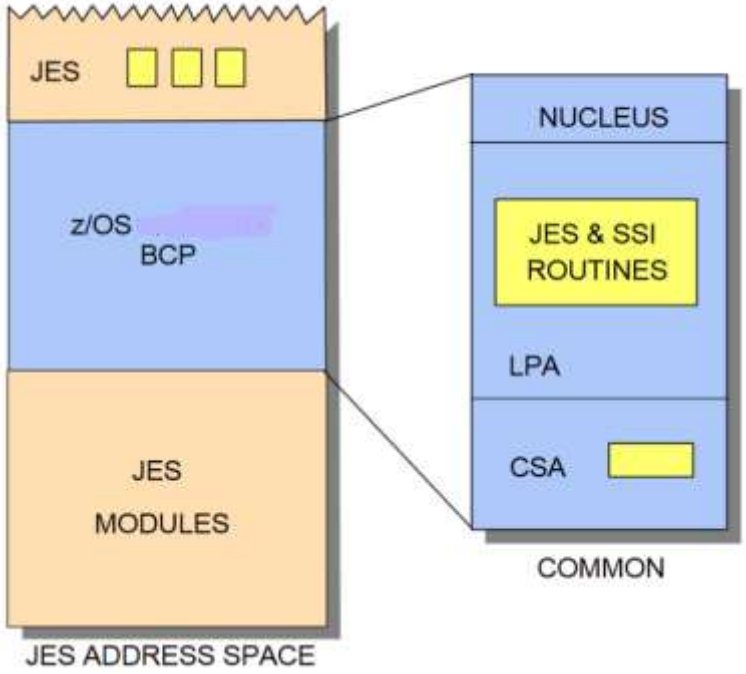
Después de haber enviado el JCL al sistema, normalmente no hay más interacción humana con el job hasta que termine.

JES2 RESOURCE INITIALIZATION DEFINITIONS

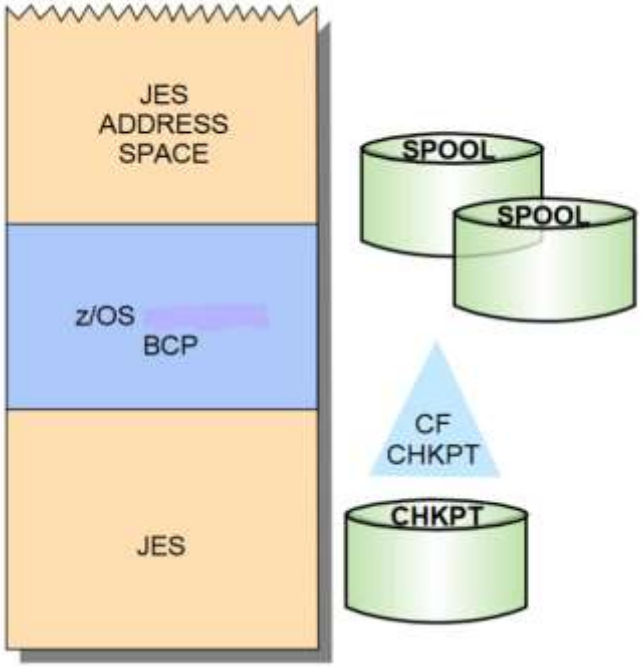


JQE: Job Queue Elements.
JOE: Job Output Elements.

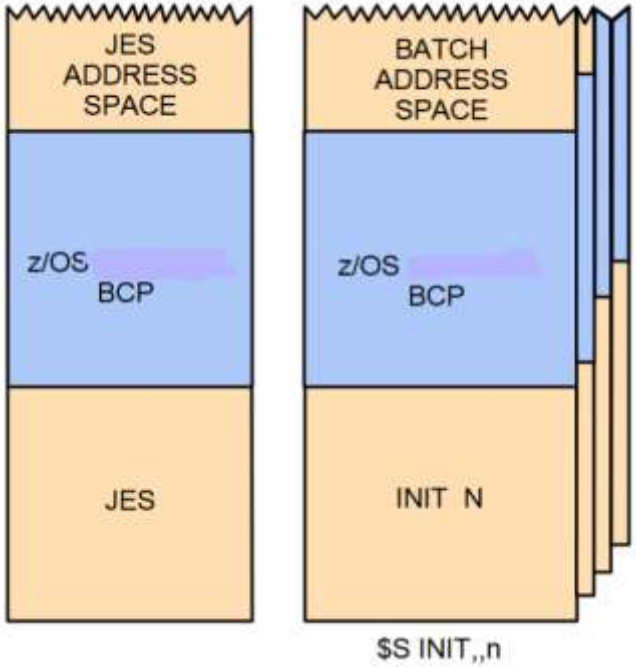
STORAGE INITIALIZATION



SPOOL AND CHECKPOINT INITIALIZATION



BATCH ADDRESS SPACE CREATION



STAGES OF JOB PROCESSING

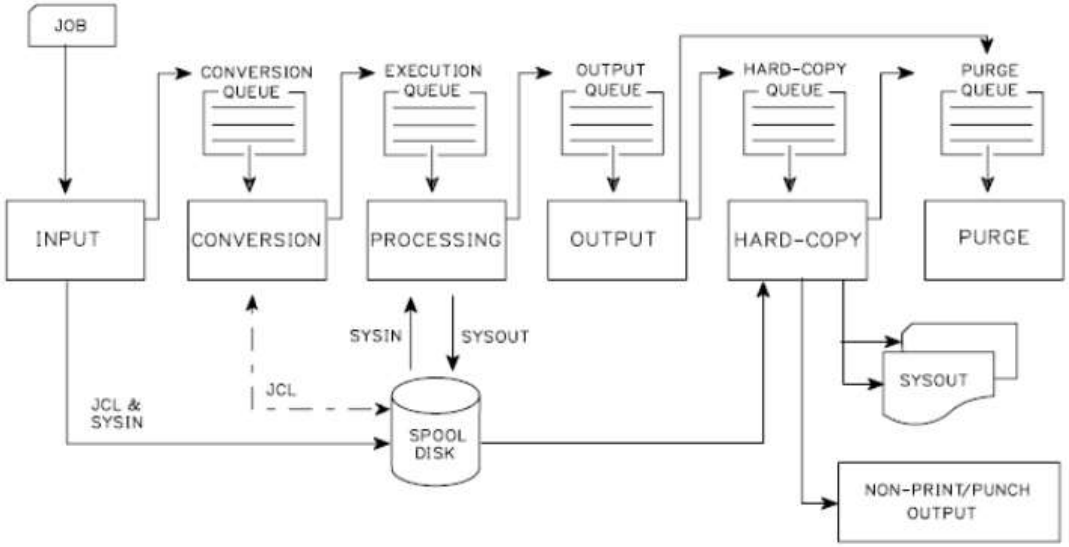
JES Introduction

JES Resource Initialization

Stages of Job Processing

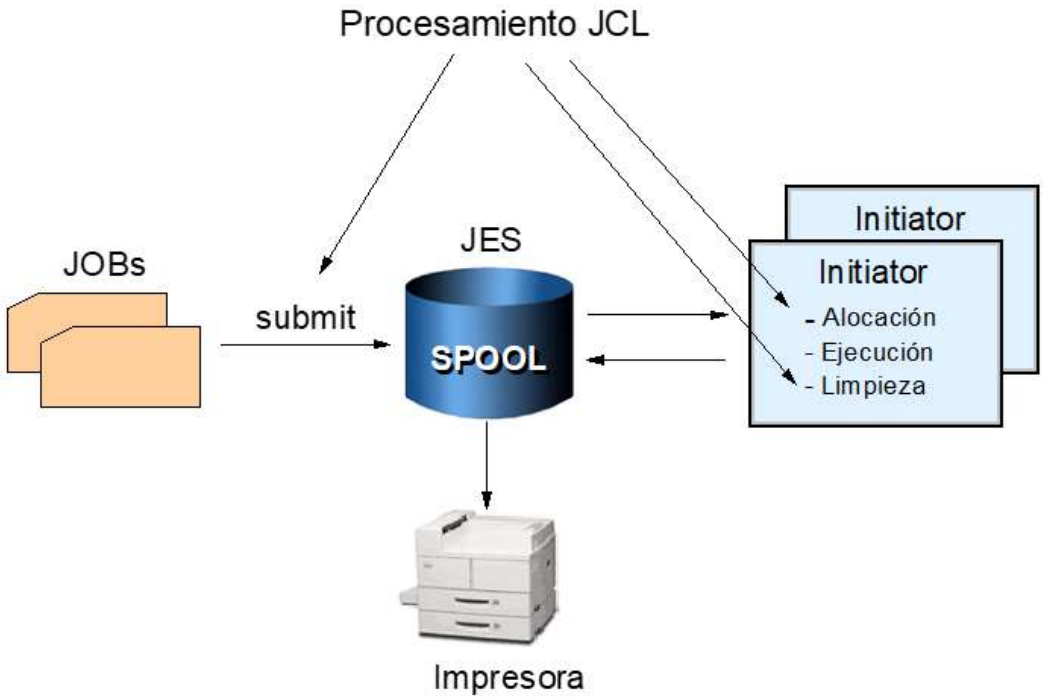
JES Shared Spool

FASES DEL PROCESAMIENTO DE UN JOB



Each queue is input to specific JES2 processers
(represented by PCEs - Process Control Elements)

FLUJO BATCH (simplificado)



FLUJO BATCH (simplificado)

El **initiator** (iniciador) es una parte dentro del z/OS que lee, interpreta y ejecuta el JCL. Ejecuta normalmente en varios address spaces (espacios de direcciones), como múltiples initiators.

Un initiator maneja la ejecución de jobs batch, uno a la vez, en el mismo address space.

Si diez initiators se activan (en diez address spaces), entonces diez jobs batch pueden ejecutarse a la vez.

JES realiza algún proceso de JCL, pero el initiator es la clave del trabajo.

Los jobs que se muestran en el dibujo representan JCL y tal vez datos entremezclados con el JCL.

Entrada de código fuente para el compilador es un ejemplo de datos (sentencias fuente) que pueden entremezclarse con JCL.

Otro ejemplo, es una información de contabilidad del job que prepara la liquidación semanal de pagos para diferentes divisiones de la compañía (presumiblemente, el programa aplicativo es el mismo para las divisiones, pero los datos de entrada los archivos de resumen pueden ser diferentes).

(el diagrama representa los jobs como tarjetas perforadas (usando el clásico símbolo para tarjetas) aunque las tarjetas perforadas reales como entrada no es ya usado).

¿QUÉ HACE UN INITIATOR?

° Para ejecutar múltiples jobs asincrónicamente, el z/OS usa initiators para:

- Asegurar que los jobs no tengan conflicto en el uso de data set.

- Asegurar que dispositivos de uso exclusivo (unidad de cinta) sean asignados correctamente.

- Ubicar programas ejecutables pedidos por los jobs.

- Limpieza de bloques de control después de la terminación del job, y pedir el siguiente job.

° Prevenir que dos usuarios accedan al mismo dato al mismo tiempo es crítico para el z/OS, y la habilidad de hacer esto es una de las características definidas del sistema operativo.

¿QUÉ HACE UN INITIATOR?

Para ejecutar múltiples jobs en forma asincrónica, el sistema realiza una serie de funciones:

° Selecciona jobs de las colas de entrada (JES hace esto).

° Asegura que múltiples jobs (incluyendo usuarios de TSO y otras aplicaciones interactivas) no tengan conflicto en el uso de data sets.

° Asegura que dispositivos de uso unitario, como las unidades de cinta, se asignen correctamente.

° Encontrar el programa ejecutable solicitado por el job.

° Limpieza de los bloques de control creados una vez que termine el job, y solicitar el siguiente job.

¿QUÉ HACE UN INITIATOR?

Mucha de esta tarea es hecha por el initiator, basado en la información de JCL de cada job.

La función más compleja es la de asegurar que no haya conflicto debido al uso de data sets.

Por ejemplo, si dos jobs tratan de grabar el mismo data set en el mismo momento (o uno lee mientras el otro graba), ahí hay un conflicto.

Este evento normalmente resultaría en corrupción de datos.

El principal propósito del JCL es el de decirle a un initiator qué es lo que necesita para el job.

Si el job A y el job B deben ambos grabar en un data set particular, el sistema (a través del initiator) no permite ejecutar ambos jobs al mismo tiempo.

En cambio, cualquier job que comienza primero causa que si un initiator intenta ejecutar otro job, que espere hasta que el primero termine.

G&L GROUP

FLUJO DEL JOB A TRAVÉS DEL SISTEMA

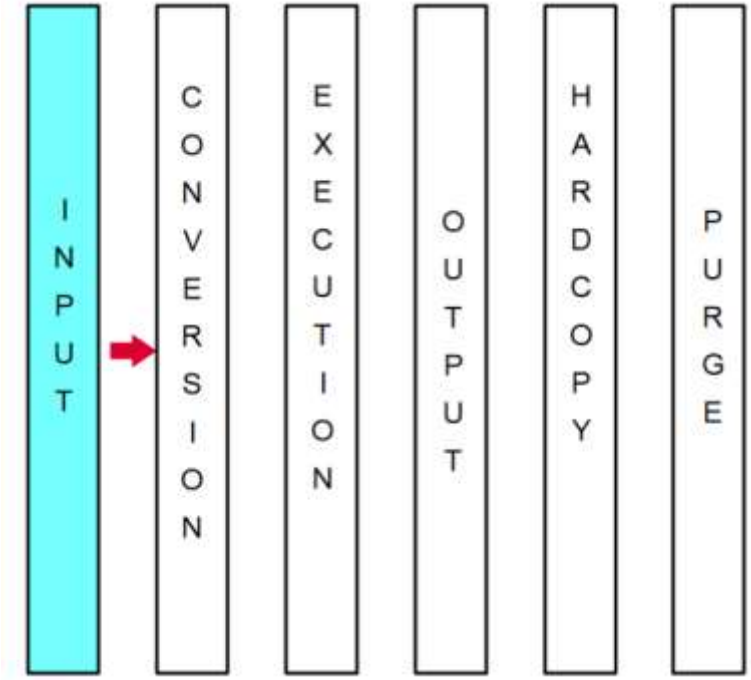
° Durante la ejecución, un job pasa por las siguientes fases:

- Input.
- Conversión.
- Procesamiento.
- Output.
- Print (a impresora o display en terminal).
- Purge.

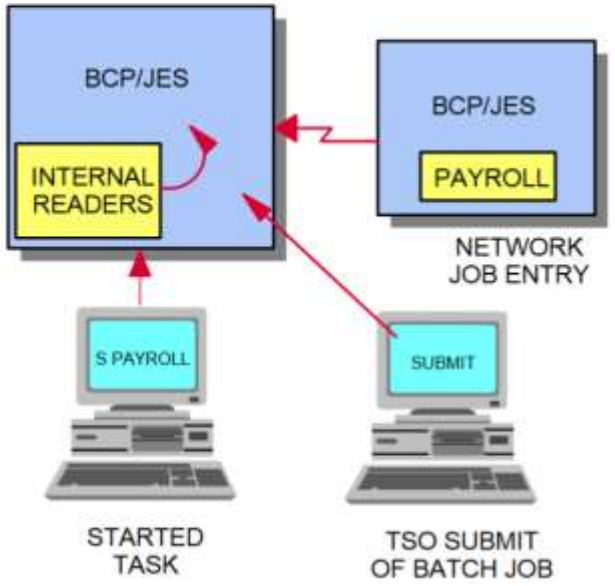
Durante la vida de un job, el JES y el programa de control base del z/OS controlan diferentes fases de todo el proceso.

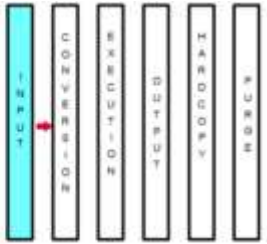
Las colas de job contienen jobs esperando ejecución, los que están ejecutando, esperando para que sus procesos de salida se produzcan, teniendo sus salidas ya procesadas, y esperando su borrado por parte del sistema.

STAGES OF JOB PROCESSING - INPUT

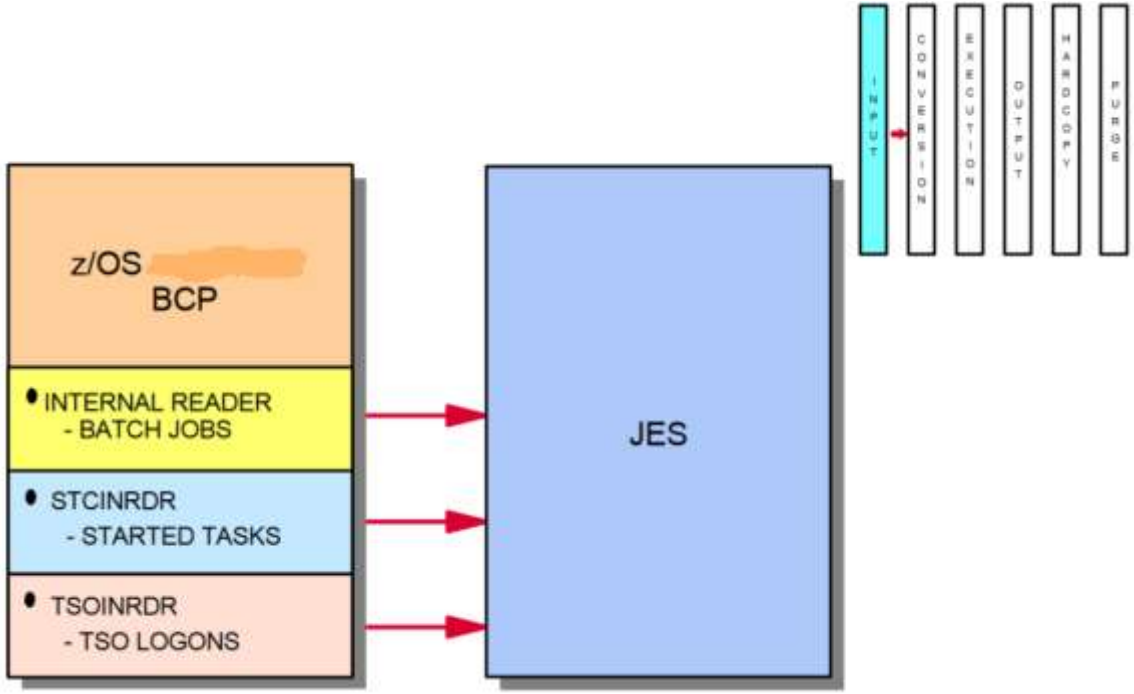


JOB ENTRY

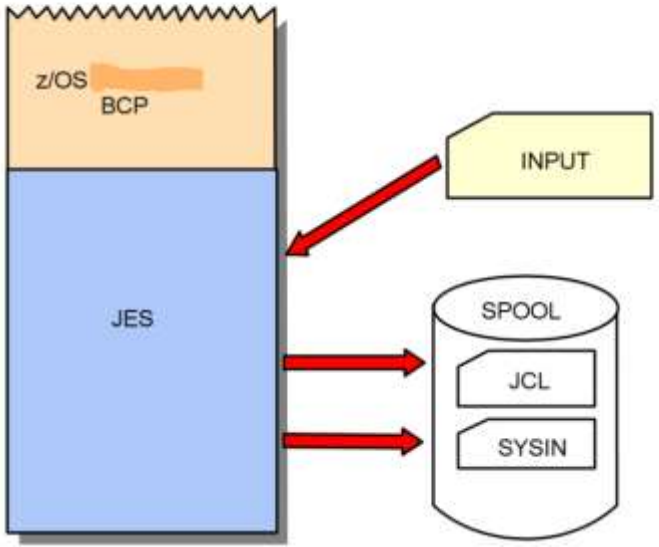

BCP/JES
INTERNAL READERS
BCP/JES
PAYROLL
NETWORK JOB ENTRY
S PAYROLL
SUBMIT
STARTED TASK
TSO SUBMIT OF BATCH JOB

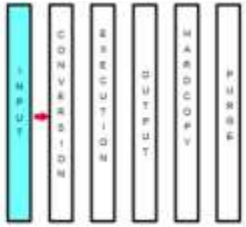

INPUT
CONVERSION
EXECUTION
OUTPUT
HARDCOPY
PURGE

INTERNAL READERS

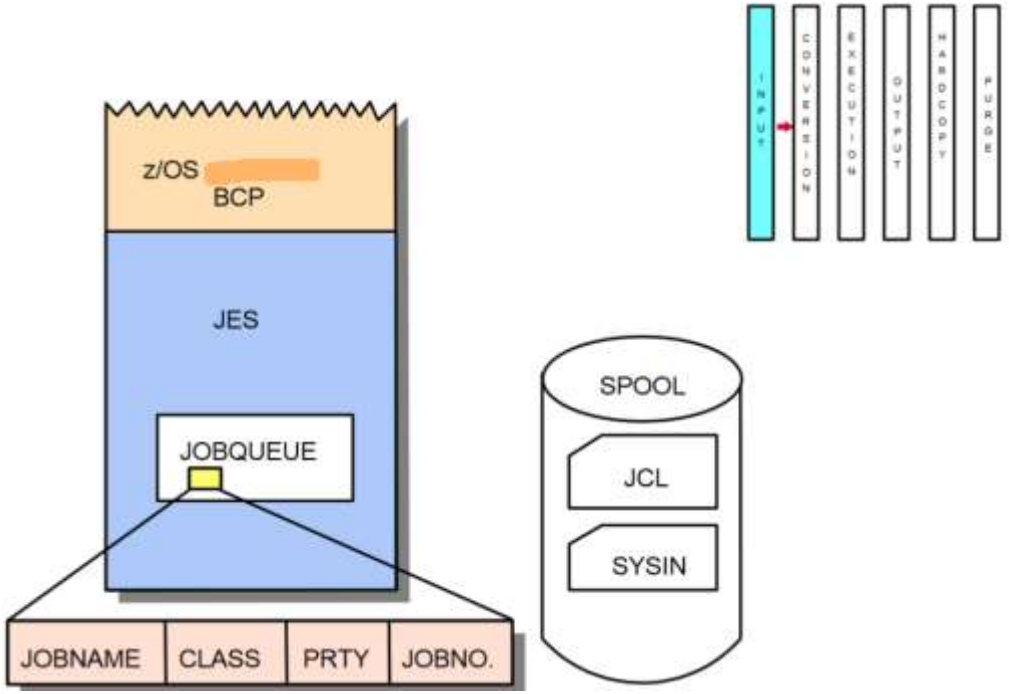


BATCH JOB SPOOLING

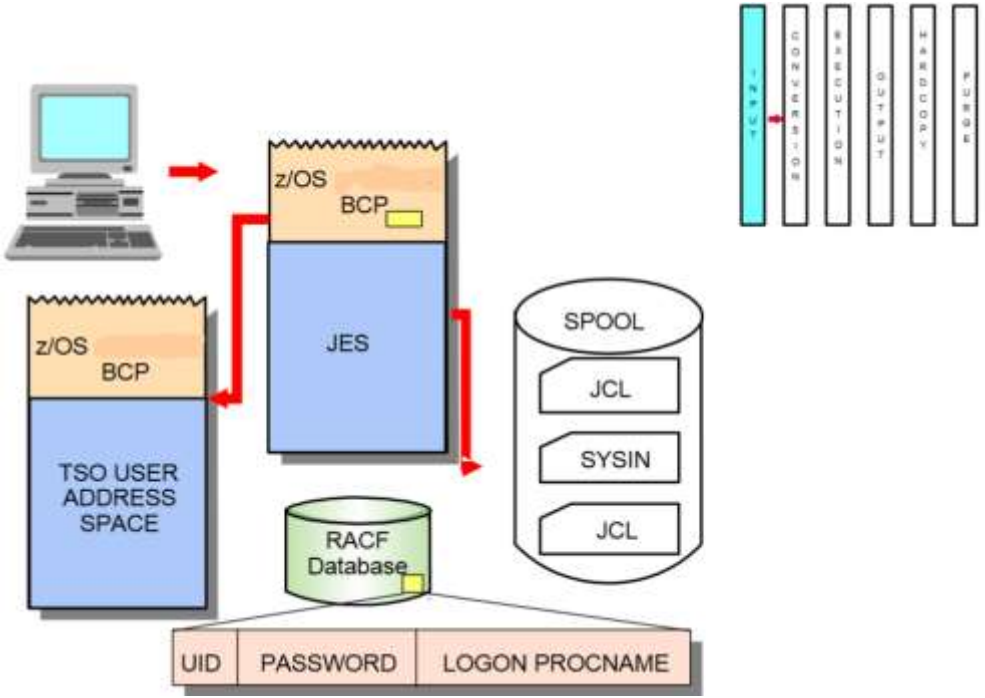




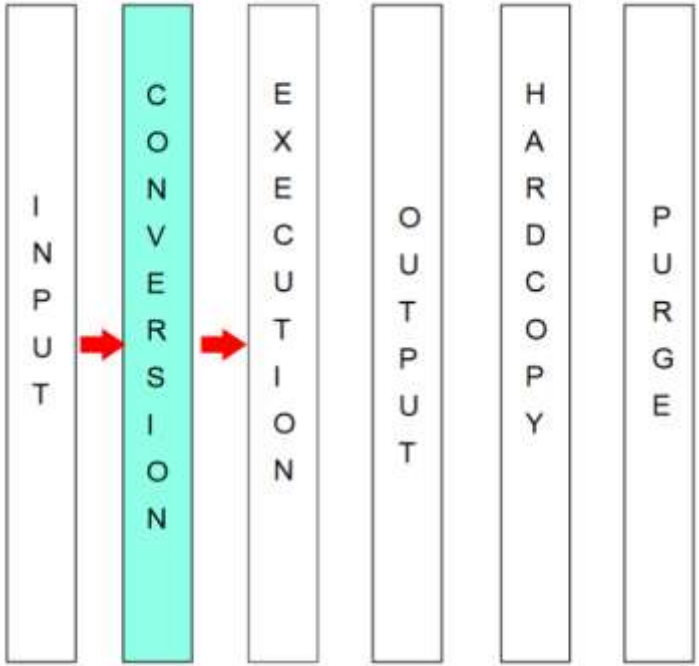
BATCH JOB JOBQUEUE ENTRY



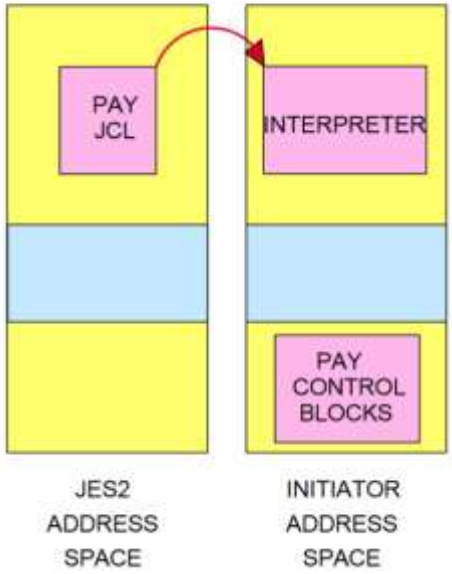
TSO LOGON SPOOLING

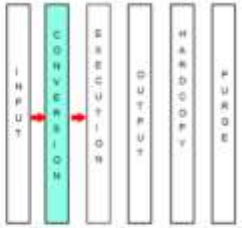


STAGES OF JOB PROCESSING - CONVERSION



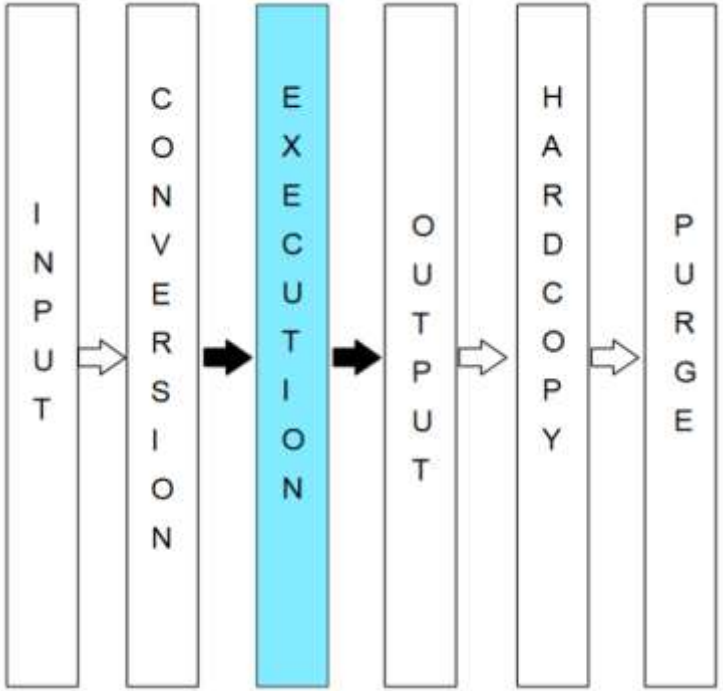
JOB INTERPRETER - JES2



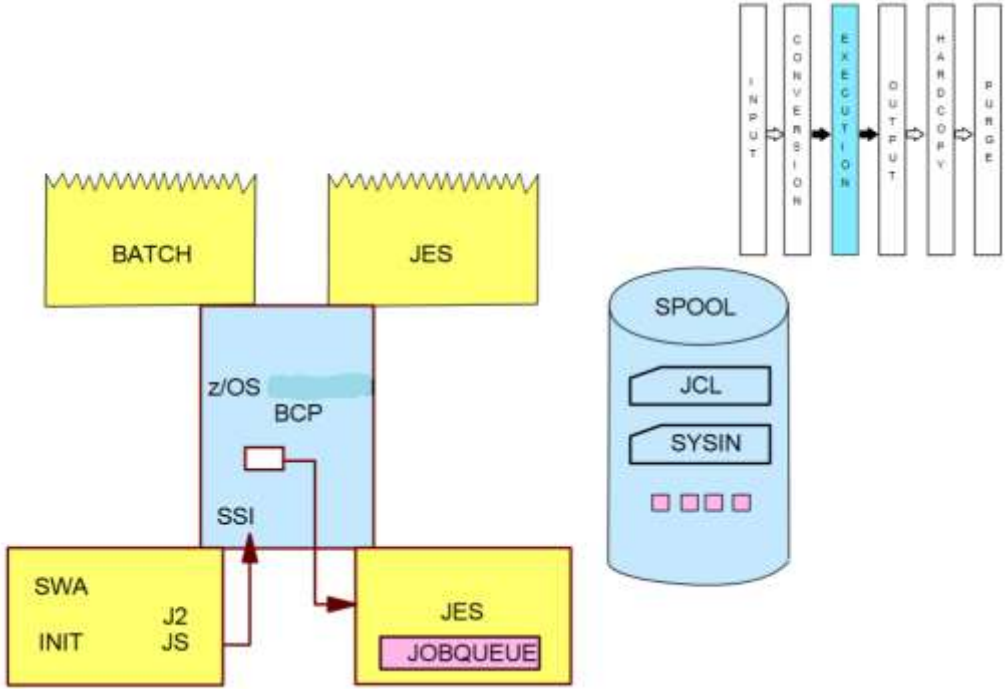


1. JES2 GIVES JOB TO IDLE INITIATOR
2. INITIATOR ATTACHES THE INTERPRETER TO BUILD CONTROL BLOCKS
3. CONTROL BLOCKS USED TO START JOB IN INITIATOR

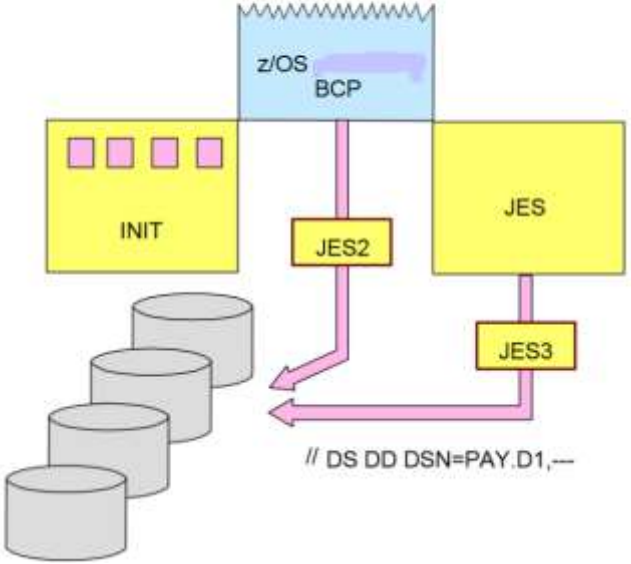
STAGES OF JOB PROCESSING - EXECUTION

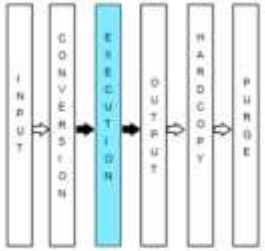


SSI

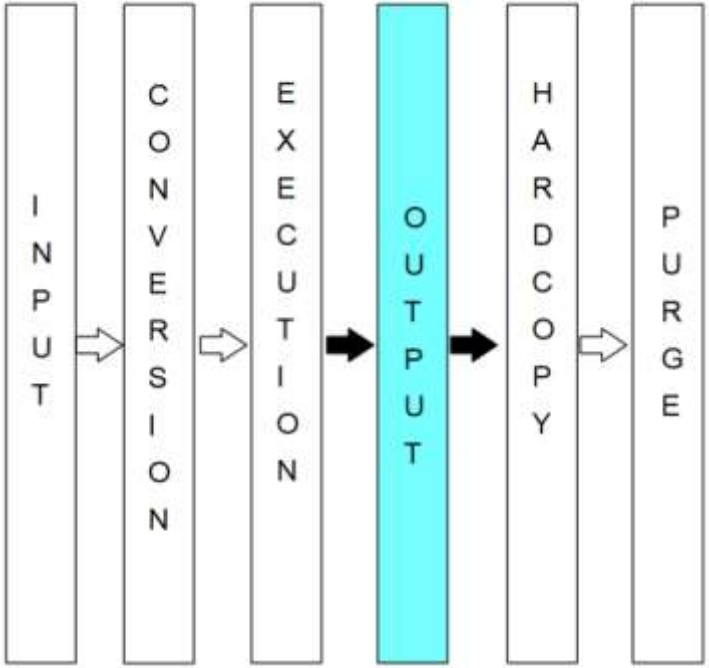

INPUT
CONVERSION
EXECUTION
OUTPUT
HARDCOPY
PURGE
BATCH
JES
z/OS
BCP
SSI
SPOOL
JCL
SYSIN
SWA
INIT
J2
JS
JES
JOBQUEUE

DEVICE ALLOCATION

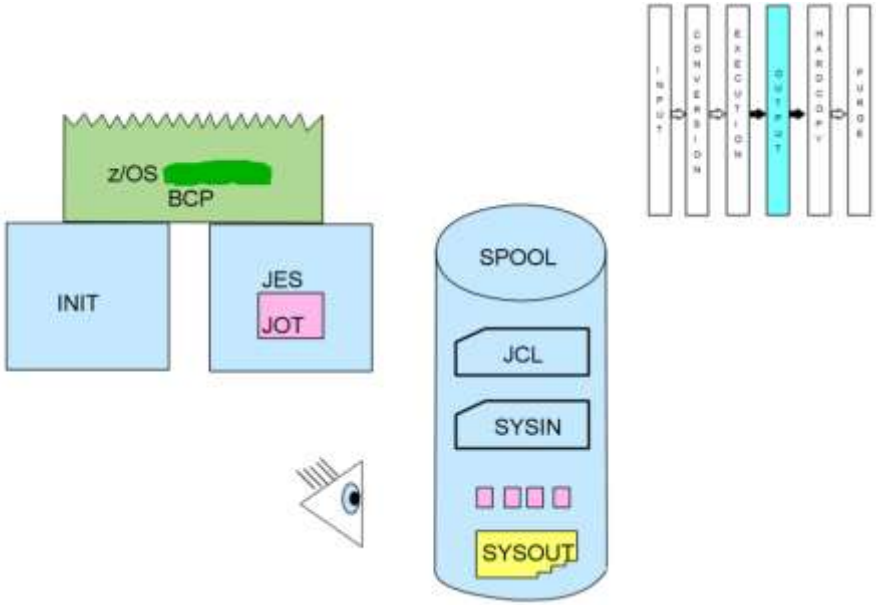




STAGES OF JOB PROCESSING - OUTPUT

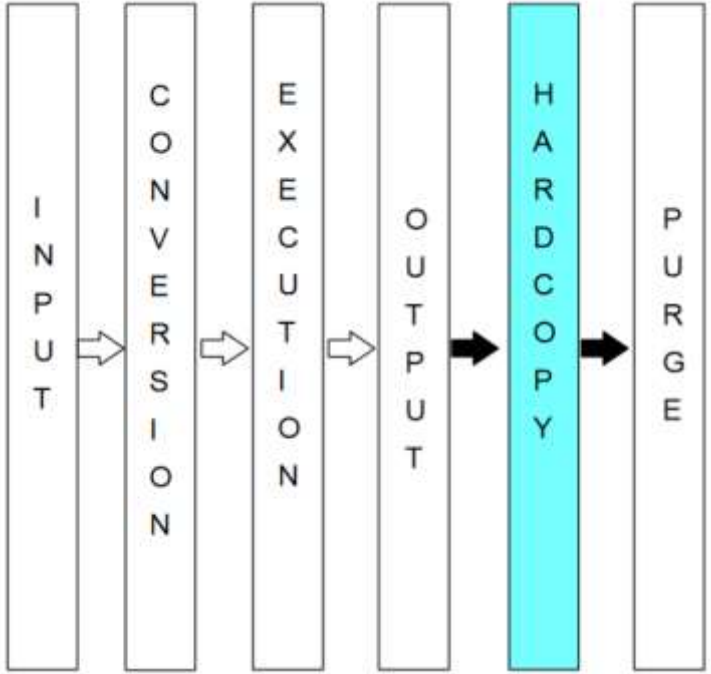


JOB OUTPUT PROCESSING

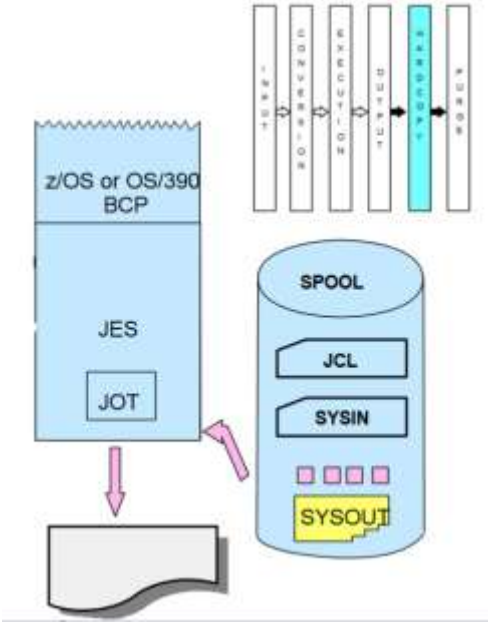


JOT: Job Output Table.

STAGES OF JOB PROCESSING - HARDCOPY

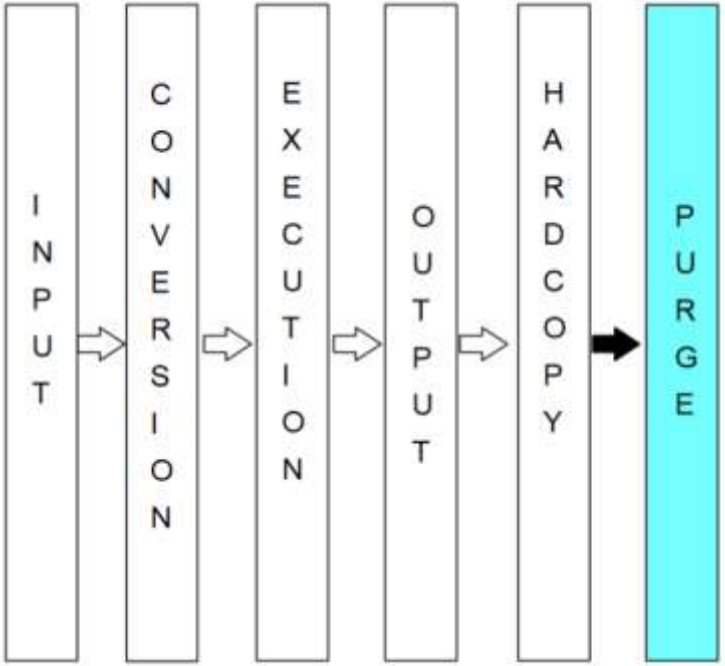


HARDCOPY PROCESSING

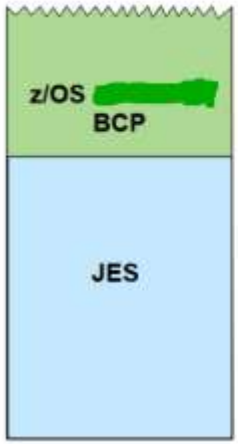


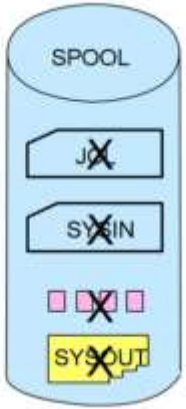
JOT: Job Output Table.

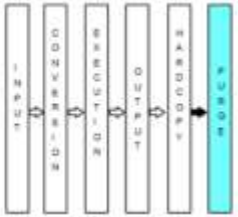
STAGES OF JOB PROCESSING - PURGE



PURGE PROCESSING







(Las distintas fases).

FASE DE ENTRADA (INPUT)

JES2 acepta jobs, en un formato de conjunto de tarjetas de entrada, desde dispositivos de entrada, desde otros programas a través de lectores internos ‘virtuales’ (internal readers) y desde otros nodos en una red de entrada de job (job entry network).

El internal reader es un programa que pueden usar otro programas para enviar jobs, sentencias de control, y comandos de JES2.

Cualquier job ejecutando en z/OS puede usar un internal reader para pasar un grupo de datos de entrada al JES2. JES2 puede recibir múltiples jobs simultáneamente a través de varios internal readers.

El programador de sistemas define internal readers a ser usados para procesar todos los Jobs batch, no para started Tasks (STCs, tareas generadas mediante el comando START) y pedidos de TSO.

JES2 lee el conjunto de entrada y asigna una identificación de job a cada sentencia de JCL JOB.

JES2 coloca el JCL del job, sentencias de JES2 (opcional), y datos de SYSIN, en data sets en disco (DASD) llamados spool. JES2 entonces selecciona jobs desde los data sets de spool para procesamiento y subsiguiente ejecución.

FASE DE CONVERSIÓN

JES2 usa un programa convertidor (converter) para analizar las sentencias de control del job.

Además, el converter toma el JCL del job y lo mezcla con el JCL desde una librería de procedimientos.

La biblioteca de procedimientos puede estar definida en la sentencia de JCL JCLLIB, o librerías del sistema o de usuario definidas en la sentencia de JCL PROCxx en el procedimiento de arranque del JES2.

Entonces, JES2 convierte las sentencias de JCL en forma compuesta en un texto convertidor/interprete que tanto el JES2 como el Interpreter pueden reconocer.

A continuación, JES2 almacena el texto convertido/interpretado en el data set de spool. Si JES2 detecta cualquier error de JCL, emite mensajes, y coloca al job en proceso de salida en lugar de ejecución.

Si no hay errores, JES2 encola el job para ejecución.

FASE DE PROCESO O EJECUCIÓN

En la fase de proceso, JES2 responde a los pedidos de los jobs desde los initiators.

Jes2 selecciona jobs que estén esperando ejecutar desde una cola de jobs y los envía a los Initiators.

Un initiator es un programa del sistema que pertenece al z/OS, pero controlado por JES o por el workload manager (WLM) que comienza un job alocando los recursos necesarios que le permiten competir con otros jobs que ya están en ejecución.

Los initiators de JES2 arrancan (started) por el operador o por el JES2 automáticamente cuando se inicializa en sistema.

Se definen al JES2 mediante sentencias de inicialización propias del JES2.

La instalación asocia cada initiator con una o más clases de job para obtener un uso eficiente de los recursos disponibles del sistema.

El initiator selecciona jobs que coincidan con las clases asignadas a ese initiator, teniendo en cuenta la prioridad en la cola de jobs.

FASE DE PROCESO O EJECUCIÓN

Initiators de WLM se arrancan automáticamente basado en los objetivos de performance, la importancia relativa de la carga batch, y la capacidad del sistema para ejecutar más trabajo.

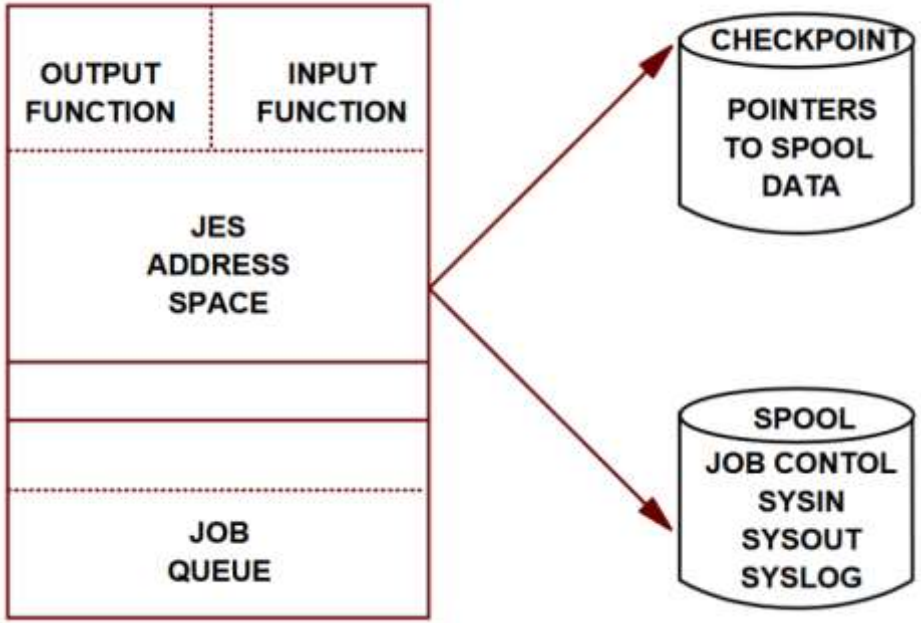
Los initiators seleccionan jobs basado en su clase de servicio y la orden en que está disponible para ejecución.

FASE DE BORRADO (PURGE)

Cuando se completa todo el procesamiento para un job, JES2 libera en espacio de spool asignado al job, marcando el espacio disponible para alocación de los siguientes jobs.

Entonces JES2 emite un mensaje al operador indicando que el job ha sido borrado del sistema.

JES PROCESSING SUMMARY



JES Shared spool

JES Introduction

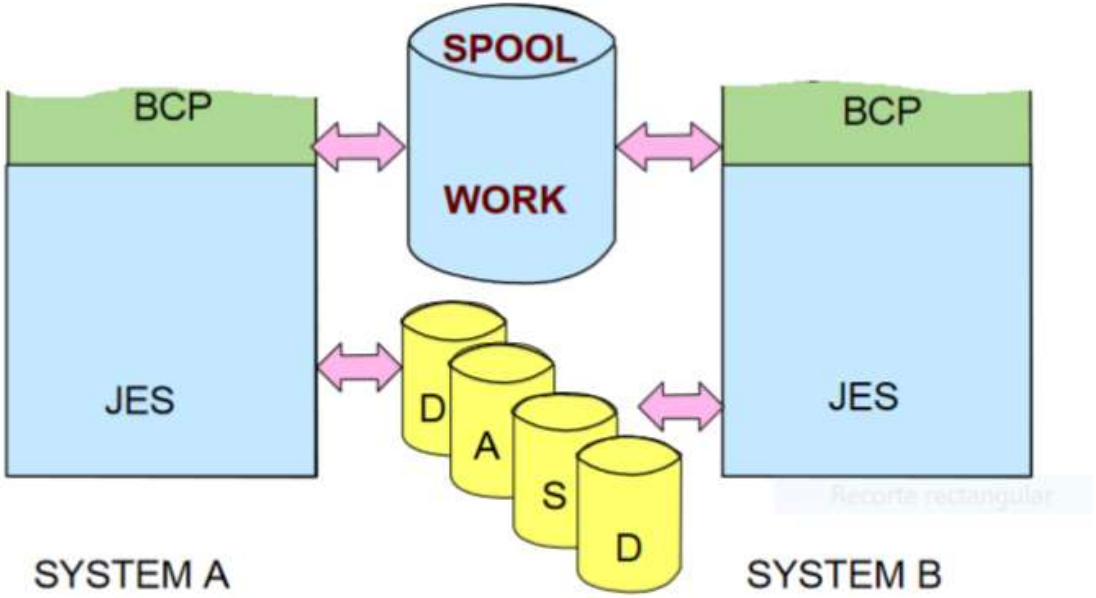
JES Resource Initialization

Stages of Job Processing

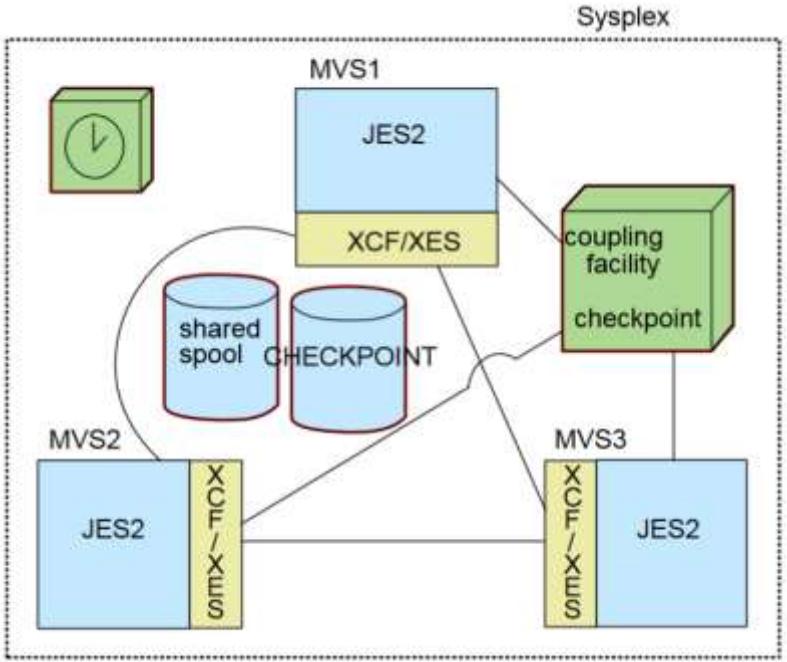
JES Shared Spool

G&L GROUP

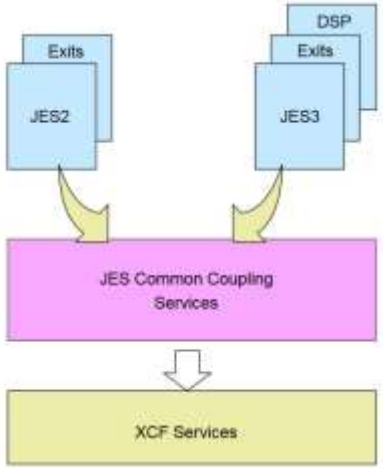
JES Communication



JES2 IN A MAS


Sysplex
MVS1
JES2
XCF/XES
coupling facility
checkpoint
shared spool
CHECKPOINT
MVS2
JES2
XCF/XES
MVS3
XCF/XES
JES2

JES COMMON COUPLING SERVICES



(DSP: Dynamic Support Programs, are multiprogrammed system components that are scheduled by the job segment scheduler (JSS) and implement a JES3 function. A DSP can be related to job execution, such as main service or output service, or it can be a background utility, such as the dump job facility).

¿QUÉ ES EL WORKLOAD MANAGER?

° Workload manager (WLM):

- Es el componente del z/OS que administra la carga de trabajo en el sistema de acuerdo con los objetivos de negocio de la compañia, como por ejemplo, los tiempos de respuesta.

- También maneja el uso de los recursos del sistema, como los procesadores y la memoria, para cumplir con esos objetivos.

Antes de la introducción del WLM, la única manera de informar al z/OS sobre los objetivos del 'negocio', fué para el programador de sistema la traducción desde objetivos de alto Nivel sobre qué trabajo necesita hacerse en términos extremadamente técnicos que el sistema pudiera entender.

Esta traducción requería personal altamente conocedor, y proactivo, propenso a errores, y eventualmente en conflicto con los objetivos originales de negocio.

¿QUÉ ES EL WORKLOAD MANAGER?

Más adelante, era difícil predecir los efectos del cambio de un definición del sistema, que podría ser necesario, por ejemplo, luego de un incremento de capacidad del sistema.

Esto podría resultar en alocación no balanceada de recursos, esto es, suministrando un recurso mientras otro lo necesita.

Esta forma de operación, conocida como modo compatibilidad, se volvió inmanejable a medida que fueron introducidas nuevas cargas de trabajo, y tantos sistemas múltiples fueron administrados juntos en procesos de 'parallel sysplex' y ambientes de datos compartidos.

Cuando el sistema opera en modo objetivo (goal), el WLM provee pocas, simples y más consistentes elementos externos del sistema que reflejan el objetivo para el trabajo expresado en términos comúnmente usados en objetivos de negocio, y el WLM y el Service Request Manager (SRM) hace coincidir recursos que cumplan aquellos objetivos mediante el monitoreo y adaptación constante.

Workload Manager provee una solución para manejar distribución de carga, balanceo de carga, y distribuyendo recursos entre las cargas de trabajos en competencia.

RESUMEN

° Procesamiento Batch es una función fundamental del z/OS.

° z/OS comparte con el JES la administración de los jobs y recursos.

° JES recibe jobs en el sistema, schedules them para procesamiento, y controla su salida.

° JES maneja jobs en colas.

° Un Initiator establece el ambiente necesario para la ejecución de un job batch. Múltiples Initiators permiten la ejecución el paralelo de jobs batch.

° Durante la vida de un job, tanto el JES como el z/OS controlan diferentes fases de todo el proceso del job.

FIN.