COMP 3400 Mainframe Administration[1]

Christian Grothoff
christian@grothoff.org

http://grothoff.org/christian/

[1] These slides are based in part on materials provided by IBM's Academic Initiative.

Today

- Quick JCL review
- The Job Entry Subsystem (JES)
- The Work Load Manager (WLM)

Question!

What is wrong with the following JCL code?[2]

```
//TESTPGM EXEC PGM=NEWPROG,PARM=‘user1,USER6,myparm,XYZ,400,300,maxuser
//   s,noname,myaccount’
```

How would you code it so that it would work?

How do you test your solution?

[2] NEWPROG is supposed to require nine parameters: user1, USER6, myparm, XYZ, 400, 300, maxusers, noname, and myaccount.

Batch Processing

Much of the work running on z/OS are *batch jobs*, which are programs that run:

- with minimal human interaction
- at a scheduled time or on an as-needed basis
- using significant amounts of resources

After a batch job is submitted to the system for execution, there is normally no further human interaction with the job until it is complete.

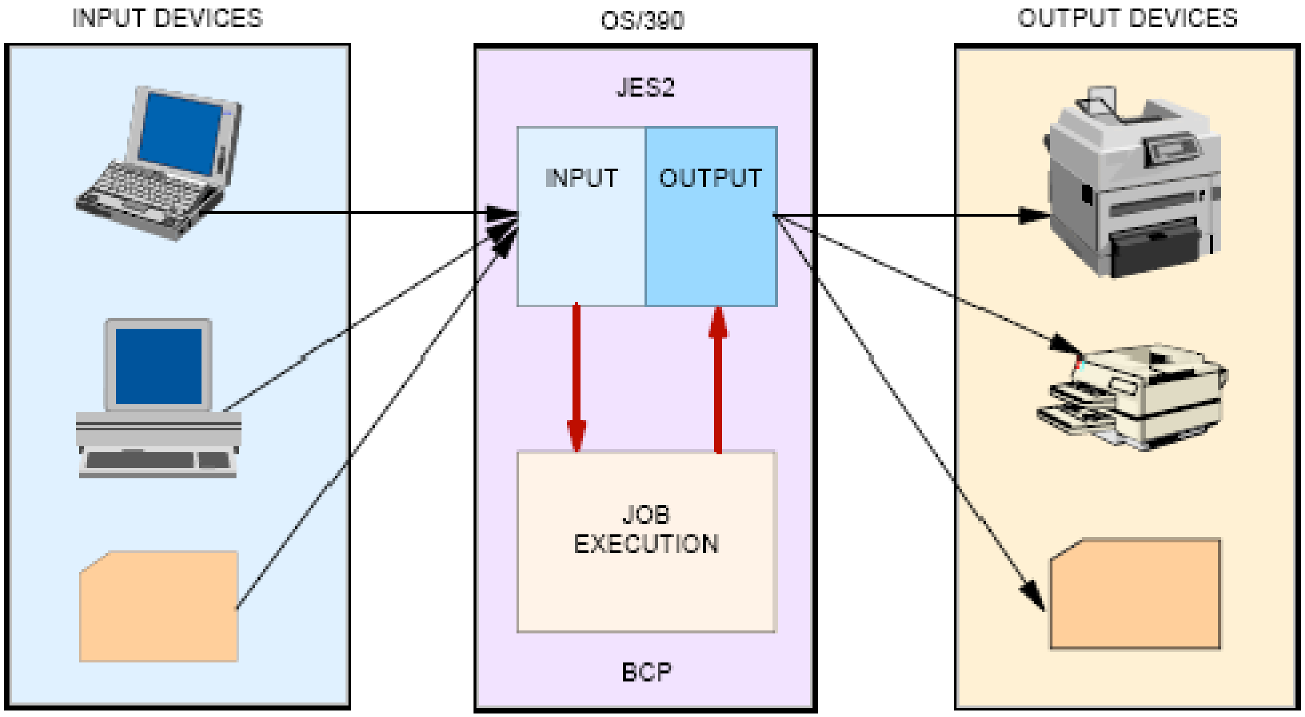
The Job Entry Subsystem (JES)

In z/OS, JES is responsible for:

- Managing input and output job queues and data
- Receiving and queuing of jobs
- Scheduling jobs for processing by z/OS
- Control of output processing (printing, storing, spooling, etc.)

z/OS offers two versions of JES, JES2 and JES3.

JES Overview



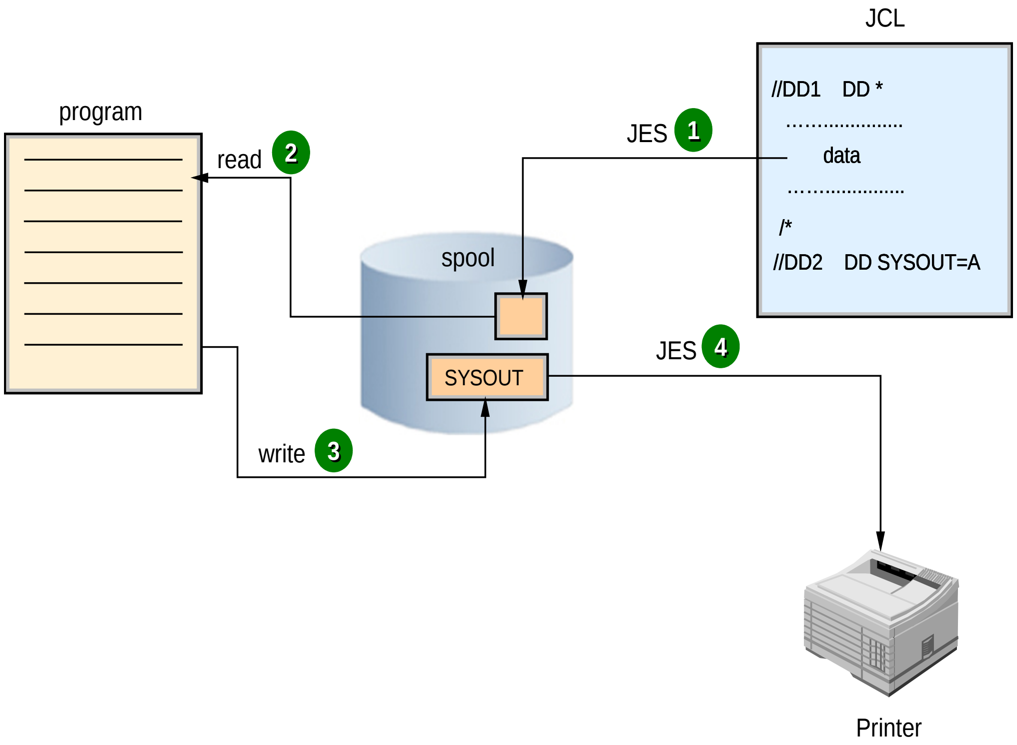
Spooling

Spooling is a method for queuing and holding data for input or output.

- JES uses one or more disk data sets for spooling
- Input jobs and printed output from many jobs are stored in the (conceptually) single spool data set

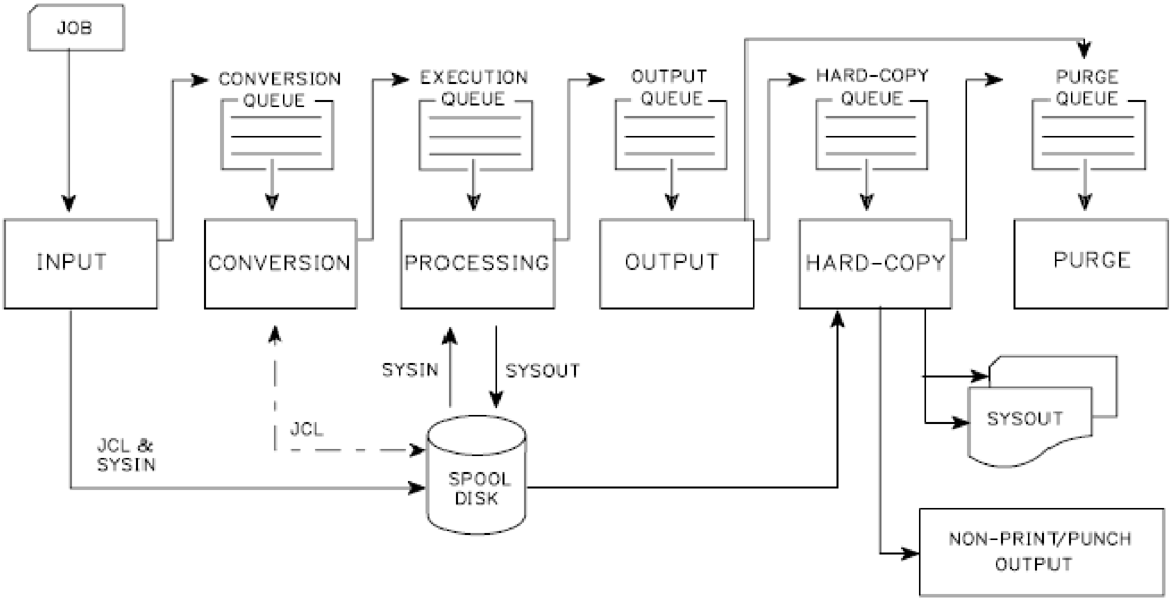
SPOOL $\equiv$ simultaneous peripheral operations online

Spooling



```
//DD1    DD *
  ....................
        data
  ....................
 /*
//DD2    DD SYSOUT=A
```

Phases of Job Processing



Phases of Job Processing

1. Input phase
2. Conversion phase
3. Processing phase
4. Output phase
5. Hardcopy phase
6. Purge phase

1. Input Phase

- Accept job from job entry network (TSO, etc.)
- JES2 reads JCL stream
- JES2 determines the job name
- JES2 loads job's SYSIN data into spool data sets
- Eventually, the job is selected for processing

2. Conversion Phase

- Analysis of the JCL
- Loading of JCL procedures from procedure library
- Validation of the JCL script
- Storage of the (preprocessed, simplified) JCL in the spool

3. Processing Phase

- Initiator requests job from the execution queue
- Initiator prepares inputs for the job
- Initiator runs the job using the Base Control Program (BCP)

4. Output Phase

- JES2 spools SYSOUT data generated by the job
- JES2 also spools other outputs depending on its output classes

5. Hardcopy Phase

- JES2 processes output queues (according to priorities and queue classes)

6. Purge Phase

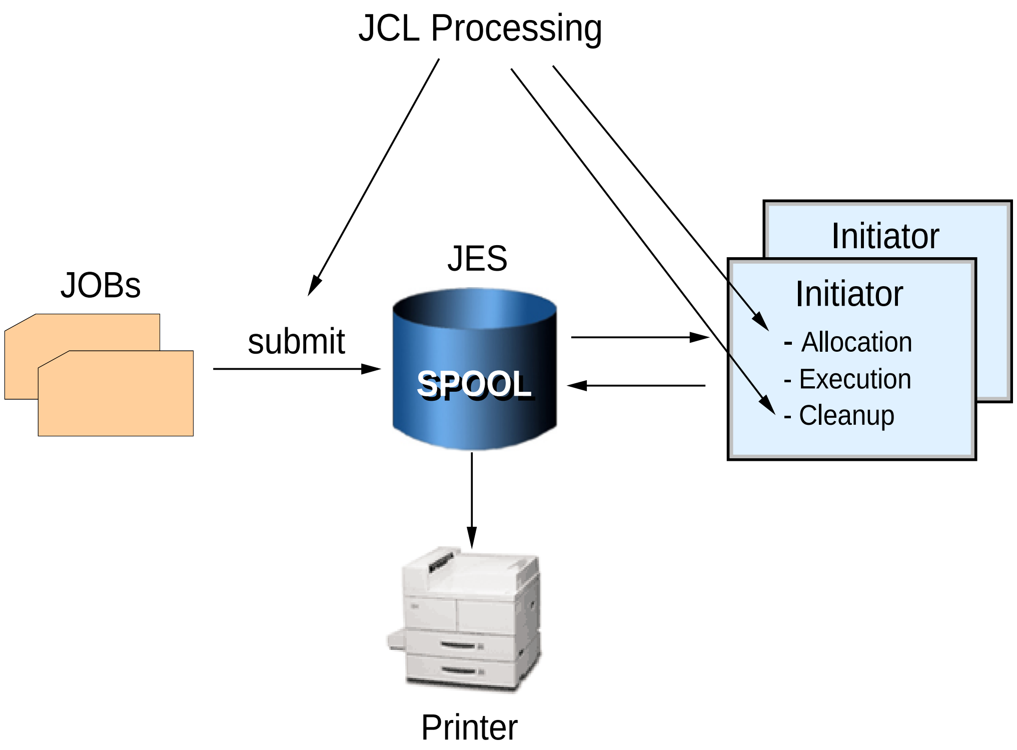
- JES2 releases spool space assigned to the job
- JES2 informs operator that the job has been purged

Initiators

Initiators are JES programs that:

- Request the next job in the queue
- Allocate resources for the job and **ensure that there are no conflicts in data set usage**
  footnotePreventing two users from accessing the same data at the same time is critical to z/OS and the ability to do this is one of the defining characteristics of the operating system.
- Find executable programs requested by the job and give them to z/OS for execution
- Clean up after the job is done

Initiators


JCL Processing
JOBs
submit
JES
SPOOL
Initiator
Initiator
- Allocation
- Execution
- Cleanup
Printer

JES Queues

- JES knows 38 execution class queues (A-Z, 0-9, STC (Started Task Control), TSU (Time Sharing User))
- Each uses an ordered FIFO within priority (optionally priority aged)
- Initiators only process jobs with matching execution class
- JES also has 110 output queues

JES2 on Multiple Systems

- Each JES2 “processor” controls its own job input, scheduling and output processing.
- A Multi-Access Spool (MAS) can be used to share spool and checkpoint data sets among up to 32 JES2 members in a sysplex
- MAS usually lives at the CFs; JES2 processors access the spool as peers
- Checkpoint data can reside at CF or at processors

Most installations use JES2.

JES3

In contrast to JES2, JES3 keeps:

- centralized control over processing
- A global JES3 processor does job selection, scheduling and device allocation for other JES3 systems

⇒ Better deadline scheduling capabilities and improved control

JES2 Start-up Options

Cold-Start All spooled jobs and data are lost (SPOOL space formatted)

All-Member Warm Start Restart of JES2 with no active members (rebuild damaged control blocks)

Single System Warm Start System rejoining sysplex, jobs that were running on system before resume from last checkpoint

Hot Start Restart JES2 after ABEND (jobs running before ABEND continue running)

JES2 Startup

- JES2 should startup automatically with z/OS
- JCL code is used to initialize JES2
- TSO is run by JES2

⇒ If JES2 does not start, you can not logon to start JES2!

⇒ Be very careful when messing with the JES2 initialization

Poly-JES

Poly-JES (also known as secondary JES or alternate JES) can be used to test JES configurations (there can usually only be one JES per z/OS). Two main configurations:

- Poly-JES as a member of your primary MAS (good for testing additional members)
- Poly-JES using a separate JES2 Spool & Checkpoint (safer, especially for more risky changes!)

JES Performance Concerns

- JES spools on fastest DASD? Fast enough?
- Access speed to CF (checkpoint data, etc.) fast enough?
- Workload manager and JES working together nicely?
- JES address space big enough?

Monitor using SDSF, RMF (z/OS facility); know your baseline performance!

The Workload Manager (WLM)

- Manages processing of workload in the system according to business goals
- Service Level Agreements (SLAs) specify business requirements, such as response times and availability
- WLM manages system resources, including processors and storage, to accomplish these goals
- WLM integrates with JES for job scheduling

WLM objectives

General objectives of the WLM are:

- Goal achievement — specifics depend on business goals
- Optimal use of system resources — throughput!
- Good performance from the user's perspective — low response time / high turnaround

WLM goals

The types of business goals for each service class that can be specified in the WLM configuration are:

- Average (!) response time
- Percentage of users meeting response time goal
- Execution velocity:

$$\texttt{Velocity} = 100 \cdot \frac{\texttt{Total Using Samples}}{\texttt{Total Using Samples} + \texttt{Total Delay Samples}}$$

- Discretionary — no specific requirement

WLM operation

Using the goals, the WLM computes a **performance index** (PI):

- PI indicates if performance goals were met, overachieved or missed
- Based on PI, WLM adjust scheduling of resources
- WLM only changes allocations based on PI after measuring for a period of time (usually $>>$ 20 seconds)

Questions

?