

**Enterprise Computing  
Einführung in das Betriebssystem z/OS**

**Prof. Dr. Martin Bogdan  
Prof. Dr.-Ing. Wilhelm G. Spruth**

**WS 2012/13**

**z/OS Betriebssystem Teil 2**

**Job Entry Subsystem**

# Job Entry Subsystem

## JES

In großen Unternehmen ist die Stapelverarbeitung (neben der interaktiven Verarbeitung) eine wichtige Aufgabe. Ein Job Entry-Subsystem (JES) hat die Aufgabe, Stapelverarbeitungsvorgänge zu automatisieren. Diese Funktion ist in den meisten Windows- und Unix-Betriebssystemen nur sehr rudimentär vorhanden. Job Control-Subsysteme werden für Windows- und Unix-Betriebssysteme von unabhängigen Herstellern angeboten. Ein Beispiel dafür bildet COSbatch ([www.cosbatch.com/](http://www.cosbatch.com/)) der Firma Open Systems Management Inc (OSM).

Das z/OS-Betriebssystem stellt eine spezielle Systemkomponente zur Verfügung, die für die Steuerung und Ablaufkontrolle aller Jobs (einschließlich aller TSO-Sitzungen) zuständig ist. Diese Systemkomponente hat den Namen **Job Entry Subsystem** (JES). Aus historischen Gründen existieren davon zwei Varianten: JES2 und JES3, die sich in ihrer Funktion jedoch sehr ähnlich sind.

Beide Varianten haben die gleiche Hauptaufgabe, und sind in der Lage, die Jobverarbeitung auf einem einzelnen Rechner zu steuern. Der wesentliche Unterschied zwischen JES2 und JES3 liegt in der Steuerung einer System-Konfiguration, die aus mehreren Rechnern besteht. Ein einzelner Rechner wird in der Regel mit JES2 gesteuert.

Werden mehrere Mainframe Rechner in einem (als Sysplex bezeichneten) Verbund (Cluster) betrieben, erfolgt die Steuerung häufig durch JES3. Mehrere Rechner arbeiten gemeinsam, und jeder Job wird von JES3 je nach Auslastung einem der Rechner zur Verarbeitung zugewiesen.

# **z/OS Stapelverarbeitung**

**Enterprise Batch Processing erfordert Sophistication und eine umfangreiche Funktionalität, die auf Windows, Unix und Linux Servern nicht verfügbar ist:**

- **Zeitabhängige Ausführung**
- **Maintaining relationships**
- **Administrative und Controlling Funktionen**
- **Einrichtungen für die Steuerung von System Ressourcen**
- **Effektive Multiprogrammierung und Multiprocessing**
- **Accounting Einrichtungen**
- **Umgebung für die Stapelverarbeitung (Batch Execution)**

**Können Sie sich vorstellen, in einem Unix oder Windows System mehr als 10 Batch Jobs gleichzeitig auszuführen und zu steuern ?**

**z/OS kann in einem (als Parallel Sysplex bezeichneten) Cluster zehntausende von Jobs gleichzeitig ausführen, verwalten und steuern .**

**Die Übergabe eines neuen Jobs an das Betriebssystem, das Queuing von Jobs, die Prioritäts- und Ablaufsteuerung zahlreicher Jobs untereinander, Startzeit und Wiederholfrequenz, Durchsatzoptimierung sowie die Zuordnung oder Sperrung von Ressourcen ist die Aufgabe eines Job Steuerungssystems.**

# Aufgaben der z/OS Stapelverarbeitung

**z/OS Jobs können eine Verarbeitungsdauer von Sekundenbruchteilen, Minuten, Stunden und evtl Tagen haben.**

**Ein Job wird submitted, es erfolgt eine lange Serie von Berechnungen, unterbrochen durch zahlreiche Ein/Ausgabe (I/O) Anforderungen, Ergebnisse der Berechnungen werden in eine Datei geschrieben, und der Job wird beendet (terminated) mit einem Return Code (RC), der die erfolgreiche oder nicht erfolgreiche Beendigung der Verarbeitung anzeigt.**

**Beispiele für z/OS Jobs sind:**

- **Complex Data BaseQueries**
- **Cheque and account processing, e.g. debits, credits, loans, credit rating**
- **Data Backups, Compression, Conversion**
- **Offline ATM (Automatc Teller Machine) processing**
- **System and Database maintenance**
- **Data replication and synchronization**
- **Processing exchanged B2B (Business to Business) data**
- **Tape processing**
- **Printing**
- **File system maintenance, healthchecks, compression...**
- **Configuration Jobs (Admin, RACF)**

# Job Ausführung

Die Übergabe (Submission) eines Stapelverarbeitungs-Jobs an das Betriebssystem erfolgt durch einen Benutzer; bei einem Großrechner durch einen spezifisch hierfür abgestellten Bediener (Operator). Unter z/OS wird die Job Submission mit Hilfe des Job Entry Subsystems (JES) automatisiert.

Voraussetzung für die Ausführung eines Jobs ist, dass benutzte Programme und Dateien (Data Sets, Files) bereits existieren.

Die Erstellung und Eingabe (Submission) eines Jobs erfordert die folgenden Schritte:

1. Zuordnung einer Datei, welche das Job Control Programm (JCL Script) enthalten soll.
2. Editieren und Abspeichern der JCL Datei
3. Submission (JES übernimmt die Ausführung des Jobs).

Viele Jobs werden in einem Unternehmen immer wieder ausgeführt. Ein Beispiel ist die monatliche Lohn- und Gehaltsabrechnung und Geldüberweisung für alle Mitarbeiter eines Unternehmens. Hier wird jeden Monat das gleiche JCL Script ausgeführt. Eine „**Cataloged Procedure**“ ist ein JCL Programm, welches vom Benutzer für eine spätere Verwendung zwischengespeichert wird (z.B. in einer vom Benutzer erstellten Library JCLLIB) und bei Bedarf mittels eines JCL Befehls aufgerufen wird.

Das JCL Programm gibt die verwendeten Dateien an. In dem Beispiel der Gehaltsabrechnung sind die sich ändernden Gehaltslisten zusätzlich zu sich nur selten ändernden Dateien wie Personaldaten. In dem JCL Script werden die verwendeten Dateien mittels von DD Statements angegeben. Damit ist ein „late Binding“ der verwendeten Dateien an die auszuführenden Programme möglich.

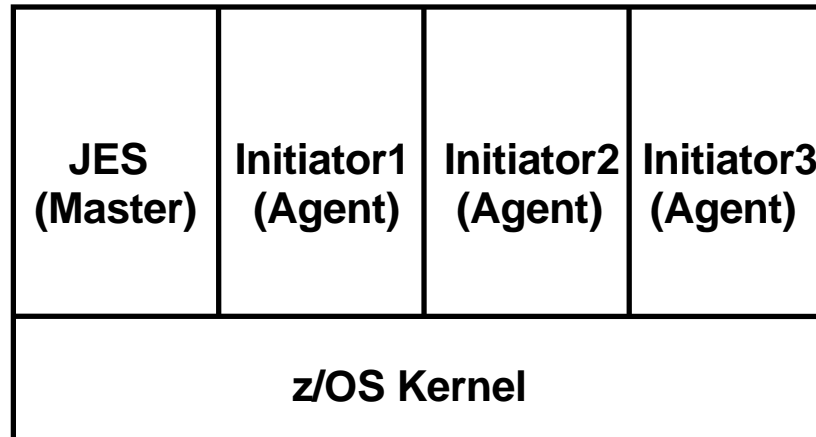
# Job Scheduler, Job Entry Subsystem

**JES arbeitet nach dem Master/Agent-Konzept. Der Master (entweder JES2 oder JES3)**

- **initiiert die Jobs,**
- **ordnet zu Anfang der Bearbeitung die Eingabe und Ausgabe Ressourcen zu**
- **reihet sie je nach Priorität in Warteschlangen ein**
- **übernimmt während der Verarbeitung die Zuordnung von Ressourcen wie CPU's und Hauptspeicher und**
- **kontrolliert den Ablauf der Jobs.**
- **Nach Abschluß der Bearbeitung übernimmt JES die Freigabe der Ressourcen. Sie werden damit für andere Jobs verfügbar.**

**Die Agenten werden unter z/OS als Initiator bezeichnet. Dies sind selbständige Prozesse, die getrennt von JES in eigenen virtuellen Adressräumen (Regions) laufen. Sie**

- **starten die Anwendungsprogramme des Jobs und**
- **senden Statusinformationen an die JES Steuerungskomponente, und/oder**
- **fertigen Laufzeitberichte an.**



## Initiator

Zur Verbesserung der Auslastung unterhält ein z/OS System typischerweise mehrere Initiators. Ein Initiator unterhält eine Input Warteschlange, an die JES einen neuen Job übergibt. Der Initiator fertigt die Jobs in seiner Input Warteschlange unter Prioritätsgesichtspunkten ab. Jeder Initiator verarbeitet in jedem Augenblick nur einen einzigen Job, aber mehrere Initiatoren können mehrere Jobs gleichzeitig verarbeiten.

Jeder Initiator belegt einen eigenen virtuellen Adressenraum, in dem er selbst und ein von ihm gerade bearbeiteter Job untergebracht ist. Das z/OS Stapelverarbeitungs-Subsystem belegt somit einen Reihe von Adressräumen, einen für JES, und weitere für mehrere Initiators.

Wenn ein Job abgeschlossen wird, wird der Initiator des Adressraums aktiv und JES sendet den nächsten Job, der darauf wartet, verarbeitet zu werden.

JES2 kann bis zu 999 Initiators in 999 Address Spaces verwalten.

# Batch Processing Beispiel

Ein Job Scheduler steuert die zeitliche Ablauffolge einer größeren Anzahl von Jobs. Er dient als single point of control für die Definition und das Monitoring der Background Ausführung für einen einzelnen oder eine Gruppe (Cluster) von z/OS Mainframes.

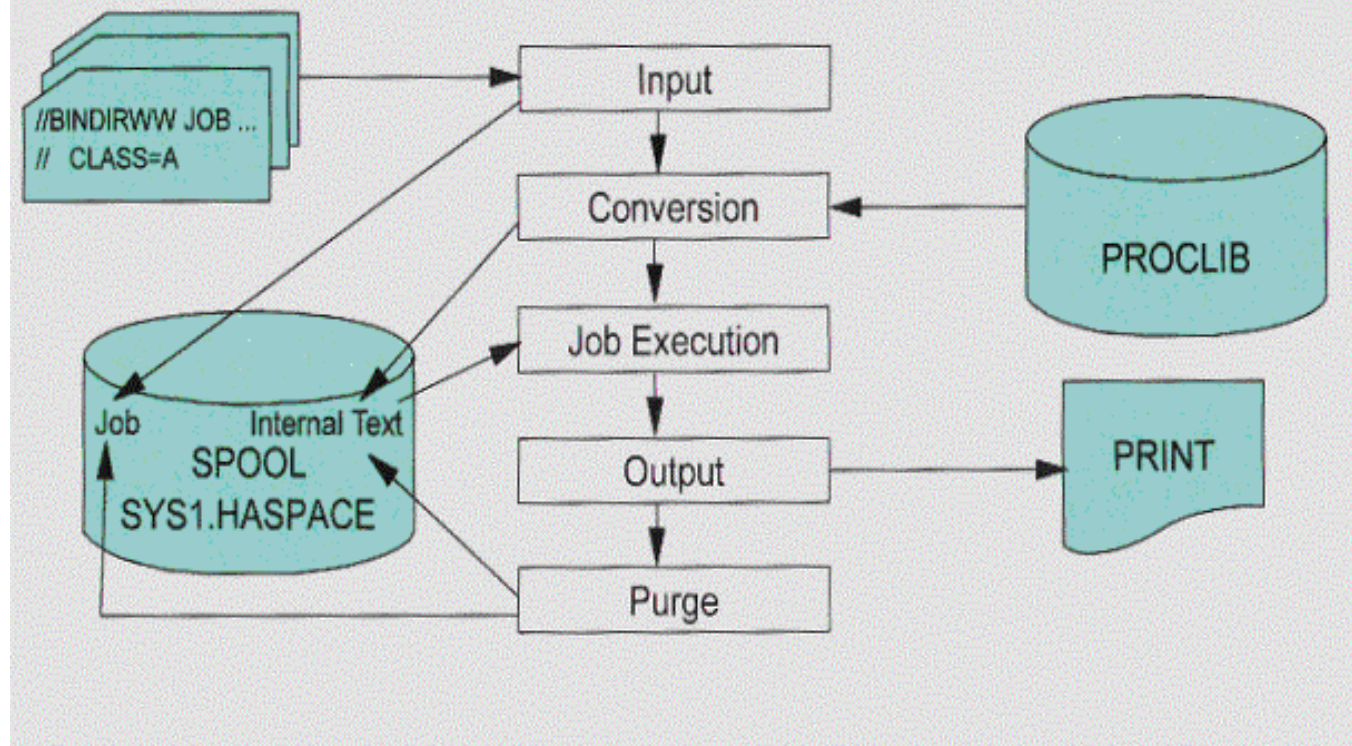
Eine Stapelverarbeitung erfolgt häufig in mehreren Schritten (Job Steps). Der Initiator steuert die Ablauffolge der einzelnen Job Steps.

Beispiel: Monatliche Kreditabrechnung für 300 000 Kundenkonten in einer Bank. Diese könnte z.B. aus den folgenden Schritten (Job-Steps) bestehen:

1. Darlehnskonto abrechnen, Saldo um Tilgungsrate verändern
2. Tilgung und Zinsen im laufenden Konto (Kontokorrent) auf der Sollseite buchen
3. Globales Limit überprüfen
4. Bilanzpositionen (Konten)
5. G+V Positionen (Gewinn- und Verlust Konten)
6. Zinsabgrenzung monatlich für jährliche Zinszahlung
7. Bankmeldewesen (ein Kunde nimmt je 90 000.- DM bei 10 Banken auf, läuft am Stichtag)



# Job Entry Subsystem JES



## Verarbeitungsablauf

Die Ausführung eines Jobs durch einen Initiator erfolgt in 5 Schritten: Input, Conversion, Execution, Output und Purge. Spool Files sind temporäre Dateien, die nach Abschluss der Verarbeitung des Jobs nicht mehr gebraucht werden.

## **INPUT**

Ein Job wird über den Reader (Internal Reader) eingelesen und auf dem Spool.Datenträger abgelegt. Er gelangt in die JES Input Queue und erhält eine Jobnummer.

## **CONVERSION**

Die JCL wird im Falle von Prozeduren ergänzt und auf Gültigkeit überprüft. Es wird ein so genannter Internal Text erstellt, der als Kontrollblockstruktur die Batch.Verarbeitung gegenüber dem JES repräsentiert.

## **EXECUTION**

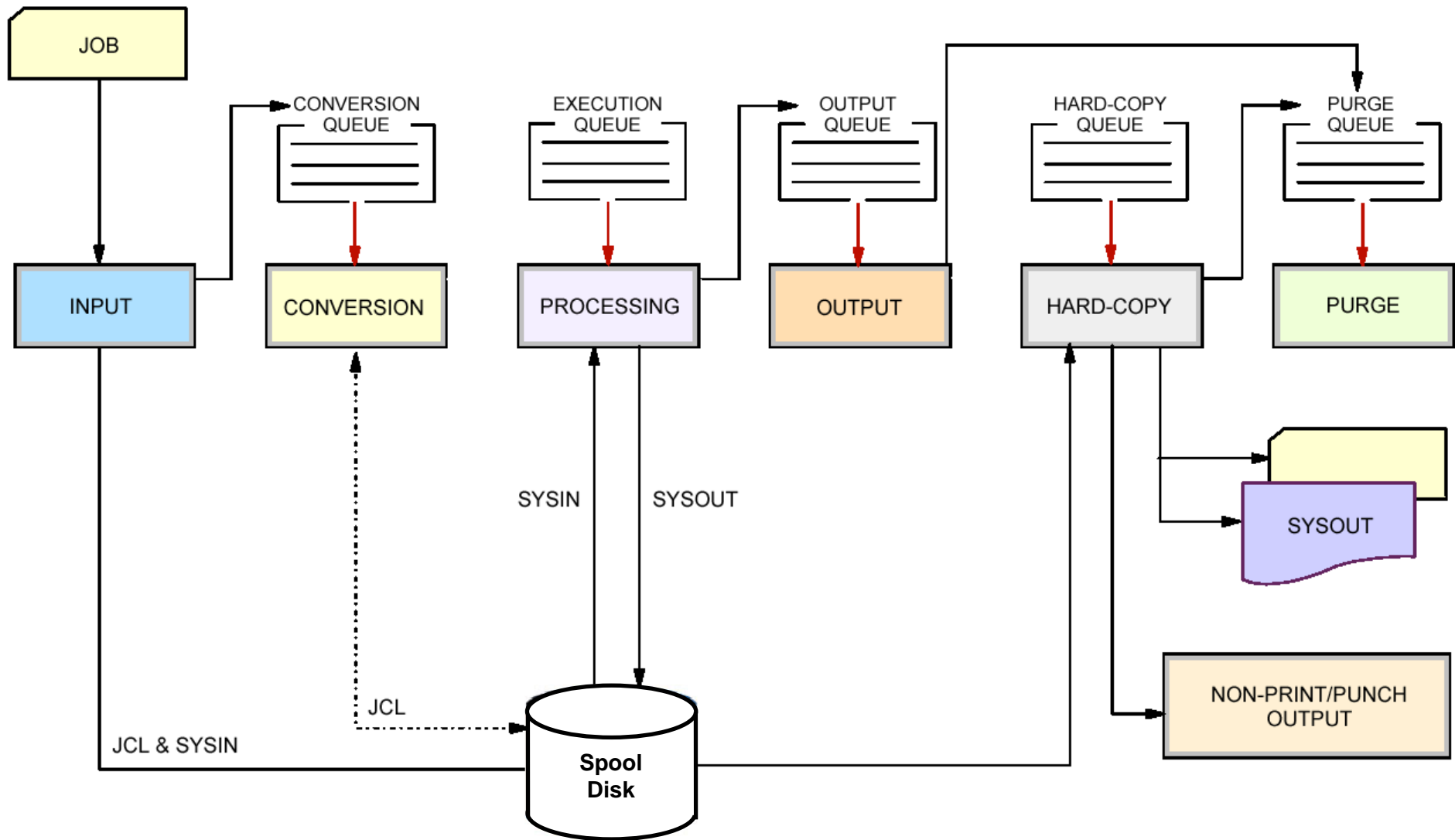
Nach einem bestimmten Algorithmus und auf der Basis von Prioritäten wird ein Job zur Ausführung ausgewählt. Die Kontrollblöcke des ausgewählten Jobs werden dazu einem freien Initiator übergeben.

## **OUTPUT**

Die vom Programm erstellten Listen werden in der JES Output Queue auf dem Spool.Datenträger gehalten, bis ein freier (heute meist virtueller) Drucker bereit ist, die Listen auszudrucken, oder bis der User/Operator den Output löscht (cancelt).

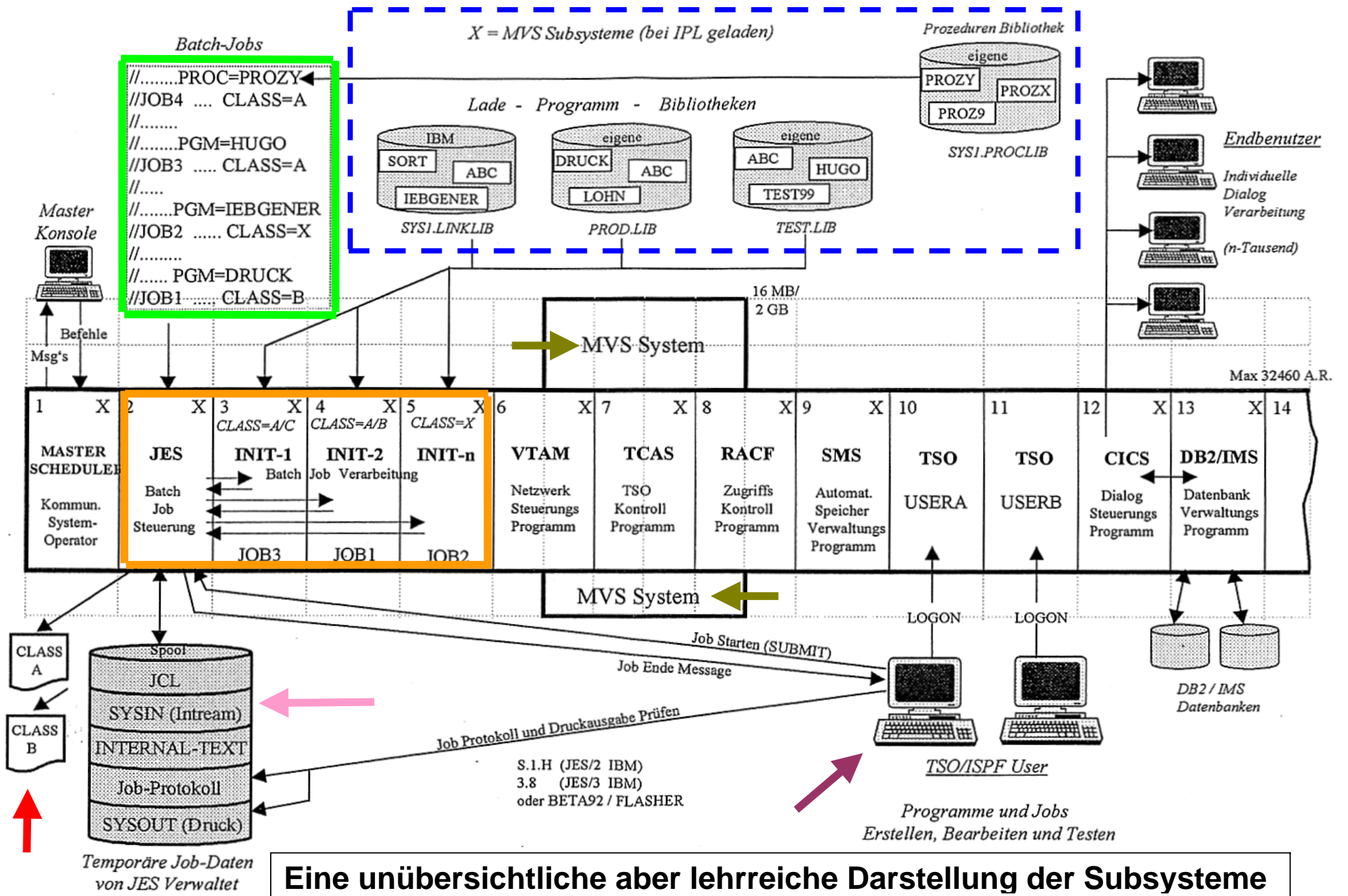
## **PURGE**

Erst wenn der gesamte Output auf einen Drucker ausgegeben wurde, werden die von einem Batchjob belegten Bereiche auf dem Spool.Datenträger gelöscht.










Zur Speicherung aller Informationen über einen Job (z.B. JCL-Records oder Ausgabe-Daten) wird ein dem JES gehörender Dataset, der sogenannte SPOOL-Dataset, benutzt in dem folgende Daten gespeichert werden;

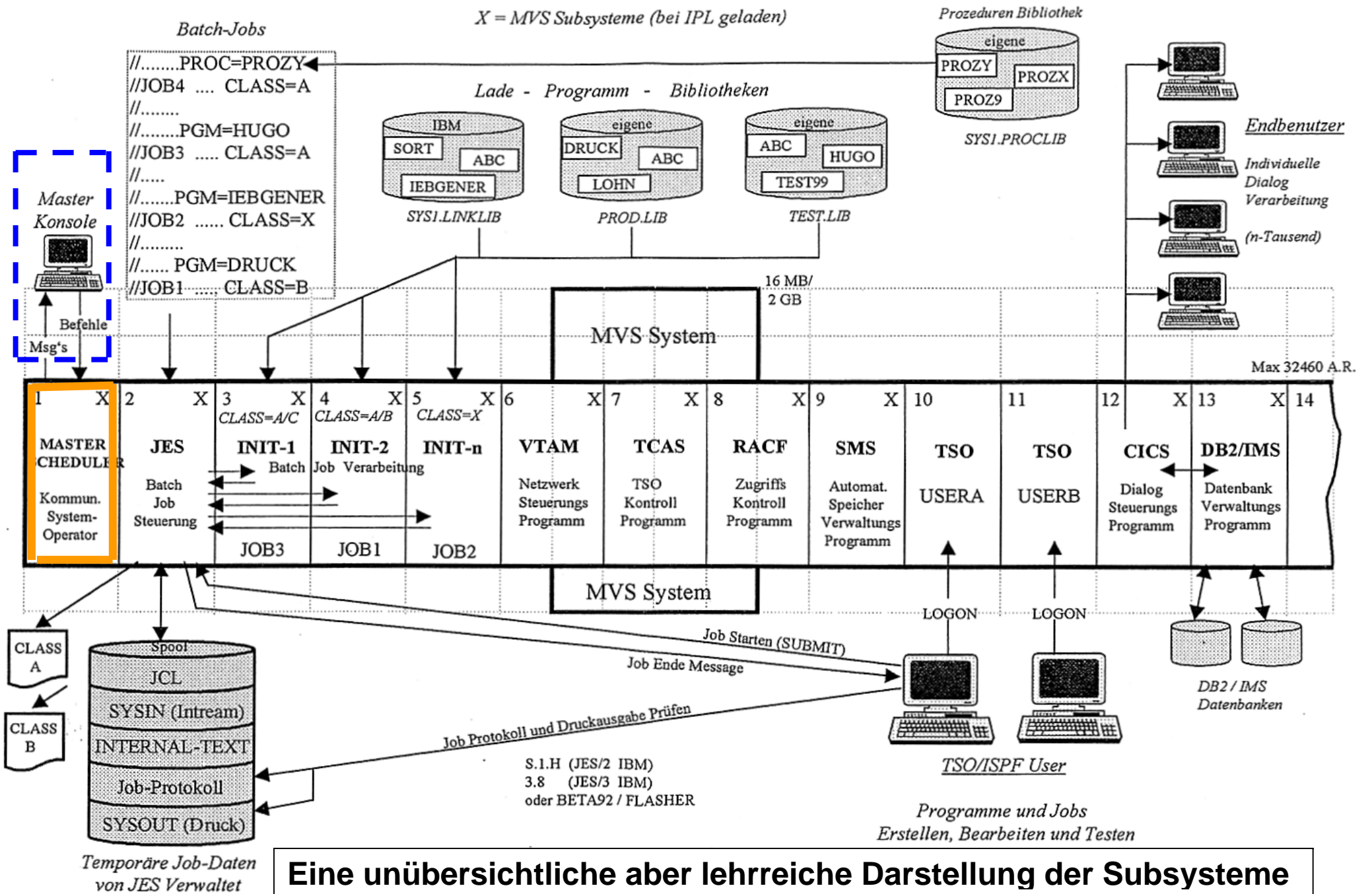
- Original-JCL und ihre Interpretation durch JES,
- Uhrzeit, wann der Job die einzelnen Phasen oder Steps durchlaufen hat,
- Return-Codes für die einzelnen Steps,
- Benutzung aller Systemkomponenten, z.B. Rechenzeit, Speicherplatz, Zahl der Zugriffe auf Platten und Bänder.



# Eine unübersichtliche aber lehrreiche Darstellung der Subsysteme

Die obige Abbildung zeigt als Beispiel ein z/OS System mit einer größeren Anzahl von Subsystemen, die jeweils in eigenen virtuellen adressräumen (Regions) laufen.

-  An Stelle der korrekten Bezeichnung z/OS wird der noch immer geläufige Begriff MVS benutzt.
-  Das Stapelverarbeitungssystem besteht aus einer JES Region und drei Initiator Regions.
-  JES steuert derzeit 4 Batch Jobs mit den Namen PROZY, HUGO, IEBGENER und DRUCK.
-  Die 4 JOBS werden entsprechend ihrer Prioritätsklassifizierung A, B und X von unterschiedlichen Initiators ausgeführt.
-  Das Programm PROZY ist in einer System-Library SYS1.PROCLIB untergebracht. IEBGENER befindet sich ebenfalls in einer Systemlibrary SYS1.LINKLIB. HUGO befindet sich in einer vom Benutzer angelegten Library. TEST.LIB. DRUCK schließlich befindet sich in einer ebenfalls vom Benutzer angelegten Library PROD.LIB.
-  Eine Spool File enthält temporäre Daten, die nach Abschluss der Verarbeitung wieder gelöscht werden können.
-  Ein Interaktiver Benutzer kann mittels seines TSO Terminals (siehe unten) einen neuen Job starten. Sie werden dies in Ihren praktischen Übungen auf unserem Mainframe Rechner ausprobieren.



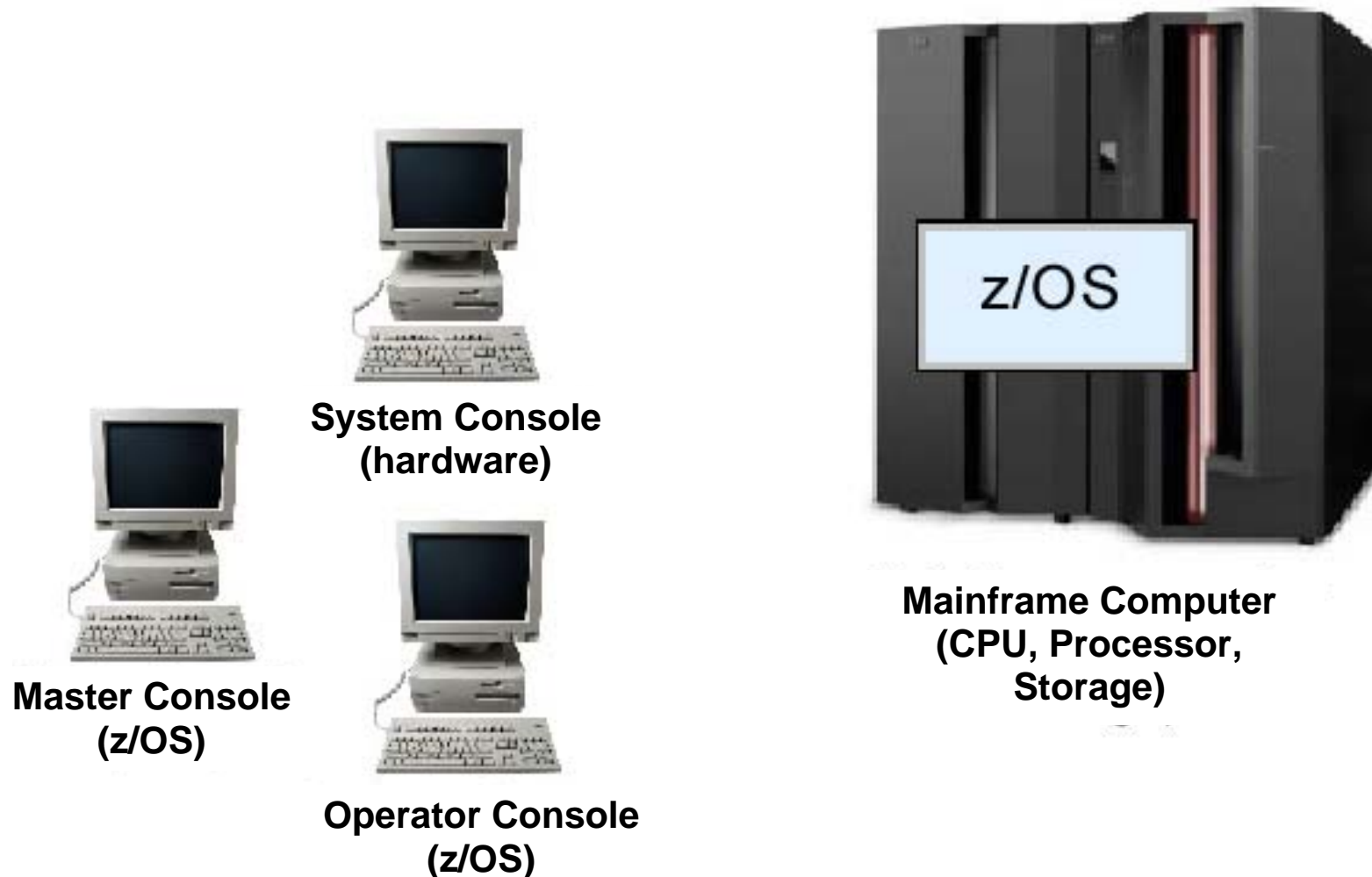
# Master Konsole und Master Scheduler

Wenn Sie beim Booten Ihres PC eine spezielle Taste drücken, kommen Sie in das BIOS Menue. Hier können Sie Kommandos absetzen, um z. B. ein Booten von einer CD zu ermöglichen.

Unter z/OS ist diese Funktionalität nicht nur beim Hochfahren des Rechners, sondern auch während des laufenden Betriebs in der Form der „Master Konsole“ vorhanden. Im Vergleich zum BIOS sind dabei wesentlich erweiterte Funktionen vorhanden.

- — Mit Hilfe der z/OS Master Konsole kann ein System Administrator (Operator) den System Zustand beobachten, und bei Bedarf in das laufende System eingreifen..
- Die Steuerung der Master Console übernimmt eine z/OS Komponente der Master Scheduler. Dieser ist auch beim Hochfahren des Betriebssystems aktiv. Der Master Scheduler etabliert die Communication zwischen dem Betriebssystem und dem Job Entry Subsystem.

Neben der z/OS Master Konsole existieren noch weitere Konsolen, z.B. eine Hardware Konsole nur für die Steuerung der Hardware. In modernen z/OS Versionen können diese Funktionen auf einer einzigen physischen Konsole vereinheitlicht werden.



Die System Console steuert die Hardware des Rechners. Für die Steuerung des z/OS Betriebssystems sind die Master Console und die Operator Console vorgesehen.