Enterprise Computing
Einführung in das Betriebssystem z/OS

Prof. Dr. Martin Bogdan
Dr. rer. nat. Paul Herrmannn
Prof. Dr.-Ing. Wilhelm G. Spruth

WS 2009/2010

Teil 3

z/OS Betriebssystem

System z und S/390 Betriebssysteme

- **z/OS IBM große Installationen (OS/390, MVS)**
- **z/VSE IBM mittelgroße Installationen**
- **z/VM IBM Virtualisierung, Software Entwicklung**
- **TPF IBM spezialisierte Transaktionsverarbeitung**
- **UTS 4 Amdahl based on System V, Release 4 (SVR4)**
- **OSF/1- Hitachi Open System Foundation Unix**
- **Linux Public Domain**

Alle System z bzw. S/390 Betriebssysteme sind Server Betriebssysteme, optimiert für den Multi-User Betrieb

TPF
Transaction Processing Facility

American Airlines Sabre Group

- **30 000 Reisebüros**
- **3 Mill. registrierte Einzelkunden**
- **400 Fluggesellschaften**
- **50 Mietwagenfirmen**
- **35 000 Hotels**
  - **Eisenbahngesellschaften**
  - **Fähren**
  - **Kreuzfahrten**

- **Amadeus System der Deutschen Lufthansa**

- **Mehr als 20 000 Transaktionen / s**

More than 90 % of all airline reservation transactions and 90 % of all credit card authorization are done using TPF

Transaction Processing Facility
TPF

13. Oktober 2006. Die Firma Worldspan, ein weltweiter Anbieter von Reise-Reservierungs-systemen, hat sich für den Einsatz von sechs IBM System z9 Enterprise Class (EC) Mainframe-Servern entschieden. Worldspan will damit sein Angebot an elektronischen Datendiensten erweitern, um circa 700 Anbietern von Reiseangeboten und Millionen von Reisenden weltweit eine gemeinsame Plattform anbieten zu können.

Worldspan setzt die neuen IBM System z9 EC Server ein, um sowohl Reisebüros als auch Anbietern von Online-basierten Reisediensten die Möglichkeit zur Nutzung des weltweiten Global Distribution System (GDS) zu geben, über das zum Beispiel die Bestellung und Buchung von Reiseprodukten von Flugzeugtickets, Hotels, Mietwagen und andere Reisedienstleistungen durchgeführt wird.

Durch die Nutzung der Software „IBM Transaction Processing Facility“ (TPF) ist Worldspan in der Lage, 17.000 Kundenanfragen pro Sekunde auf den Mainframes zu beantworten.

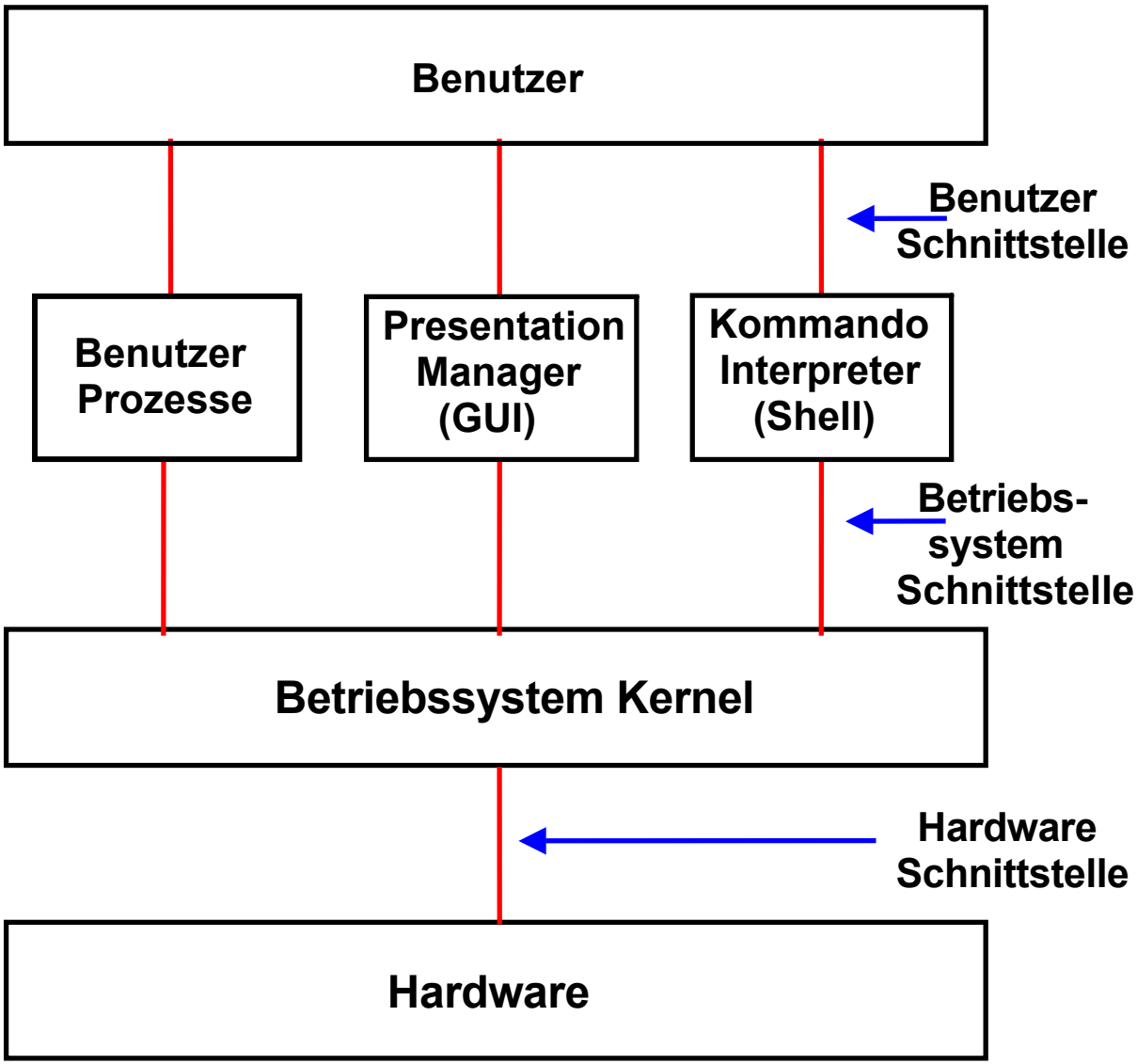
z/OS und OS/390

Anfang 1966 als OS/360 (reines Stapelverarbeitungssystem) eingeführt - Fred Brooks

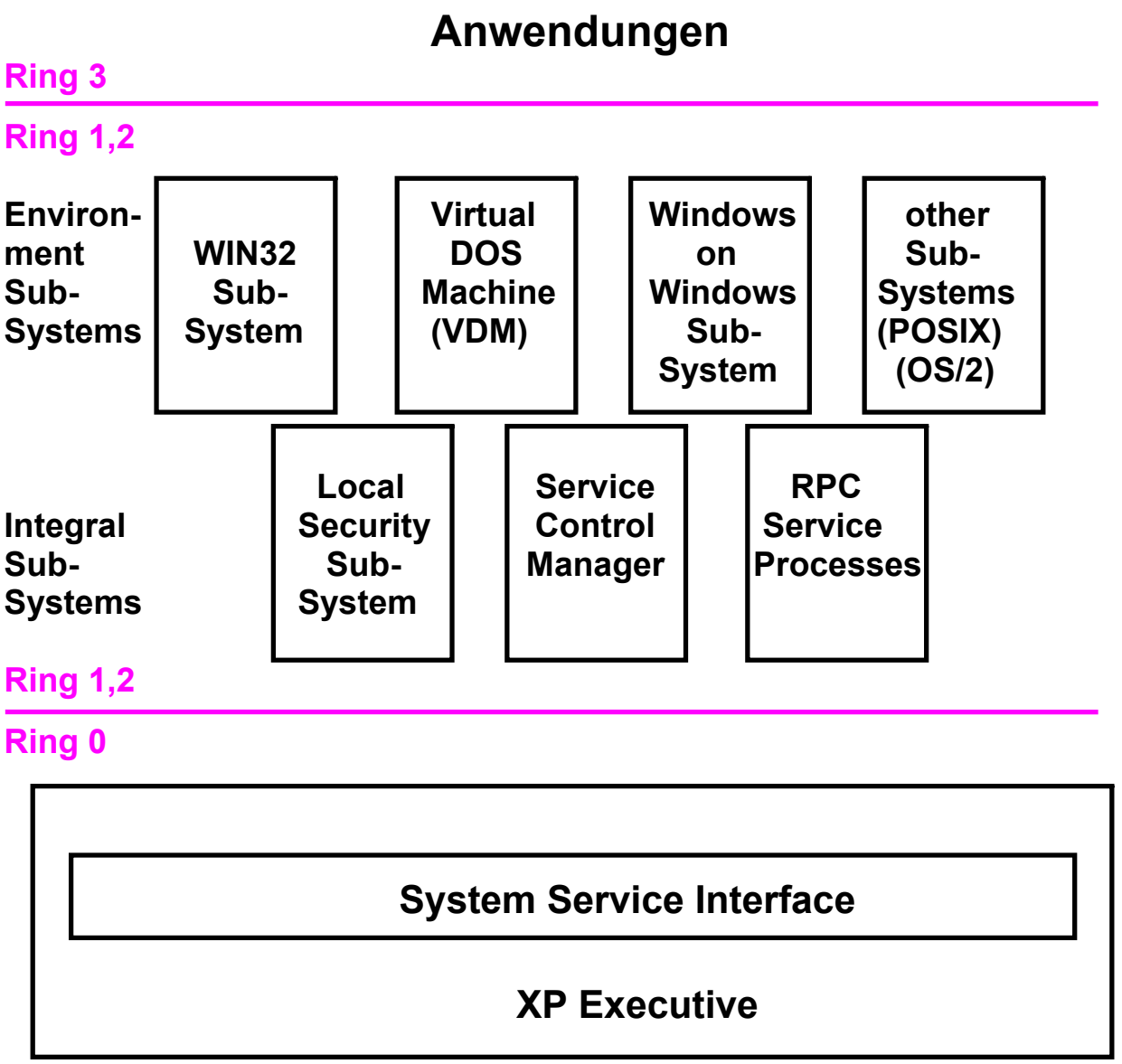
Spätere Namenswechsel OS/360 → MFT → MVT → MVS

1996 Namensänderung MVS nach OS/390 - Bündeln von mehr als 70 Komponenten: Vereinfachung der Installation und der Wartung

2000 Namensänderung OS/390 nach z/OS - 64 Bit Adressierung



Schichtenmodell der Rechnerarchitektur Windows, Linux, Unix, z/OS

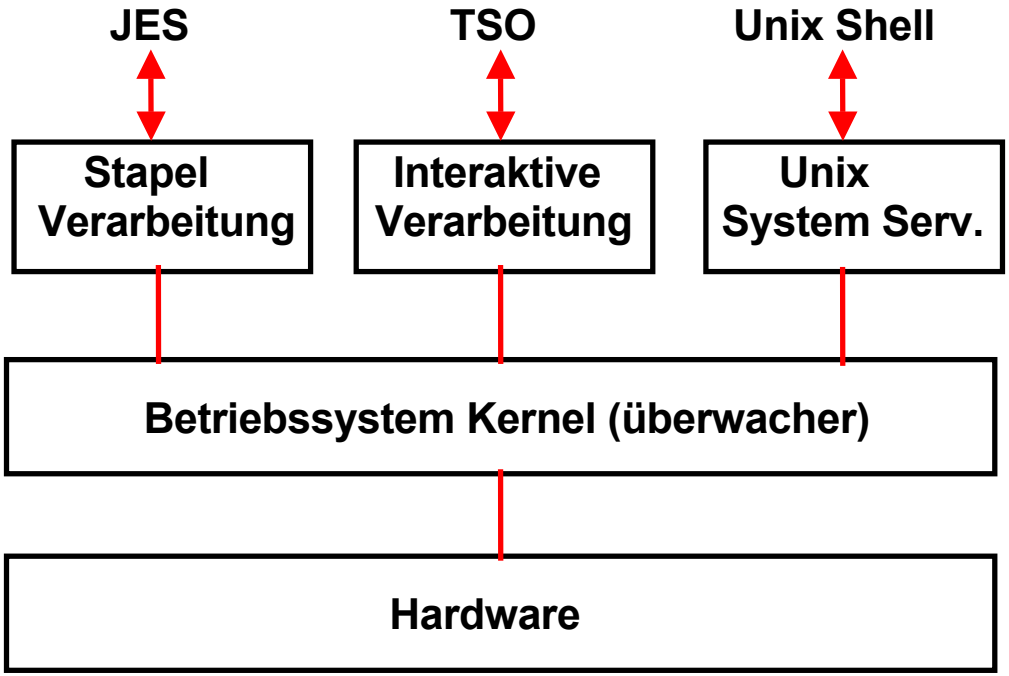


Windows XP Überwacher Erweiterte Systemfunktionen

Virtual DOS Machine verarbeitet 16 Bit Anwendungen.

Windows on Windows Subsystem verarbeitet 16 Bit Windows Apps.

Das Posix ‘Subsystem erlaubt es, Windows wie ein Unix Betriebssystem zu betreiben.

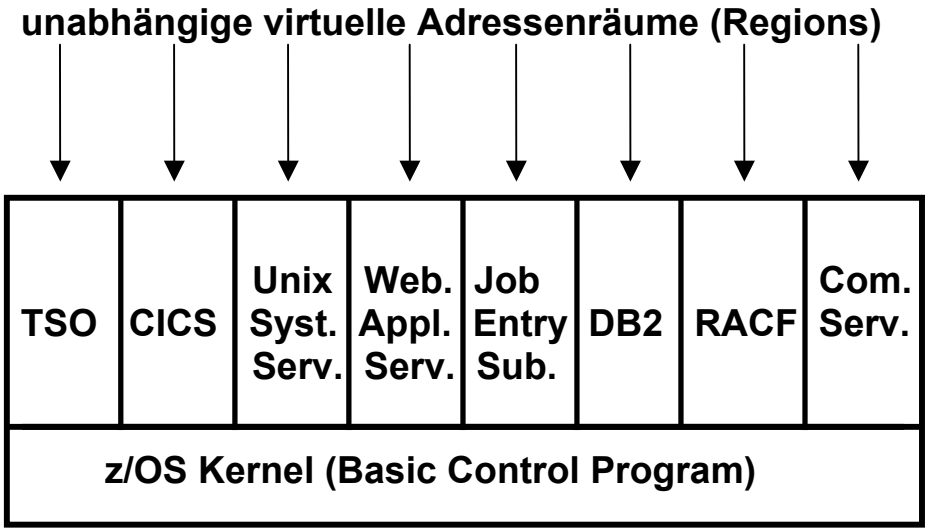


Die drei wichtigsten z/OSSubsysteme:

- **JES (Job Entry Subsystem) für die Stapelverarbeitung**
- **TSO (Time Sharing Option) für die interaktive Verarbeitung (z.B. Programmentwicklung)**
- **Unix System Services, Posix kompatibles Unix Subsystem**

z/OS Grundstruktur

z/OS Grundstruktur

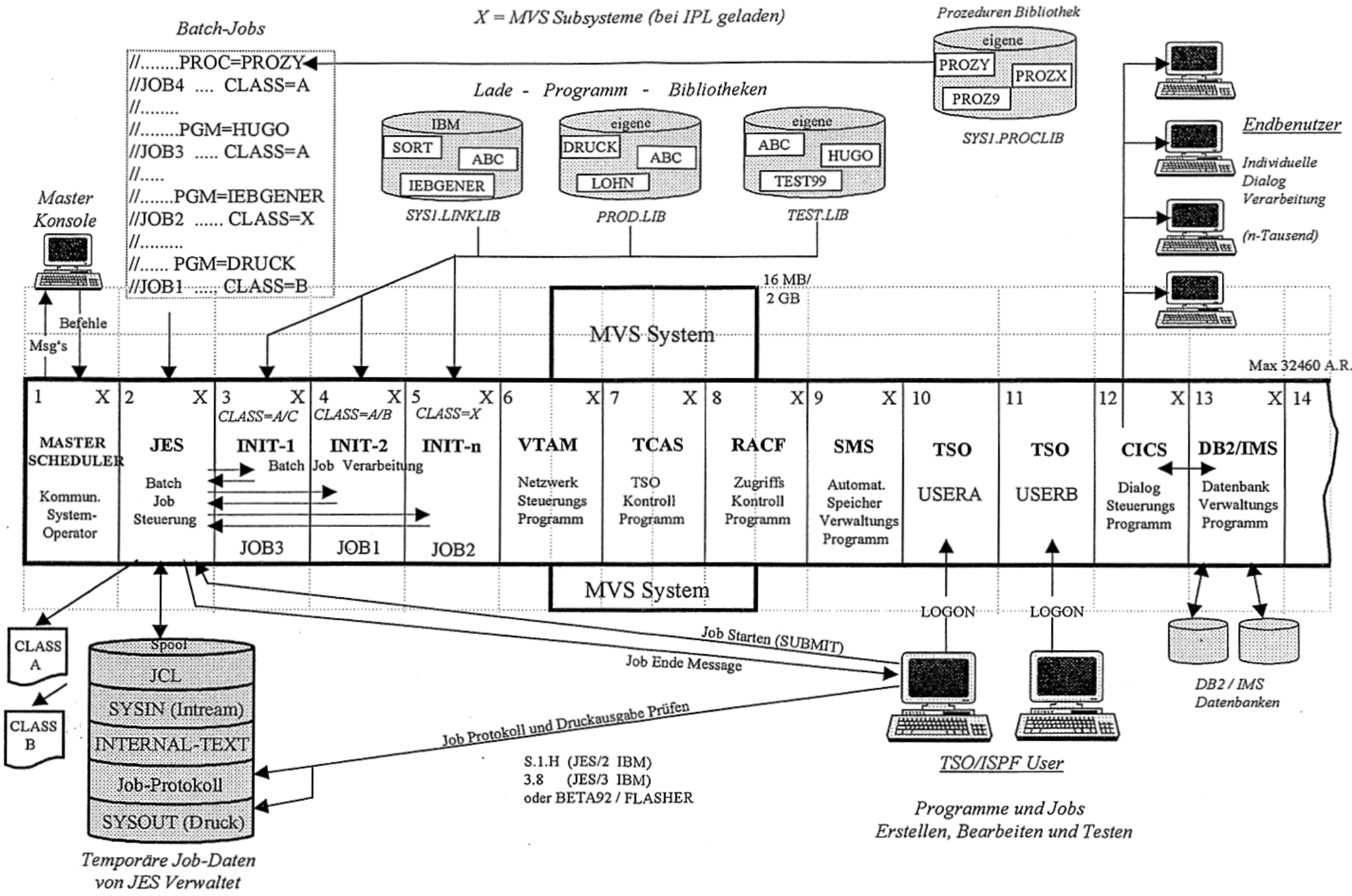


Einige der (zahlreichen) Subsysteme sind:

- **CICS** Transaktionsverarbeitung
- **TSO** Shell, Entwicklungsumgebung
- **USS** Unix kompatible Shell, Entwicklungsumgebung
- **WAS** WebSphere Web Application Server
- **JES** Job Entry Subsystem
- **DB2** relationale Datenbank
- **RACF** Sicherheitssystem
- Communications Server

Der z/OS Kernel unterstützt eine Vielzahl von virtuellen Adressenräumen, die im z/OS Jargon als Regions bezeichnet werden.

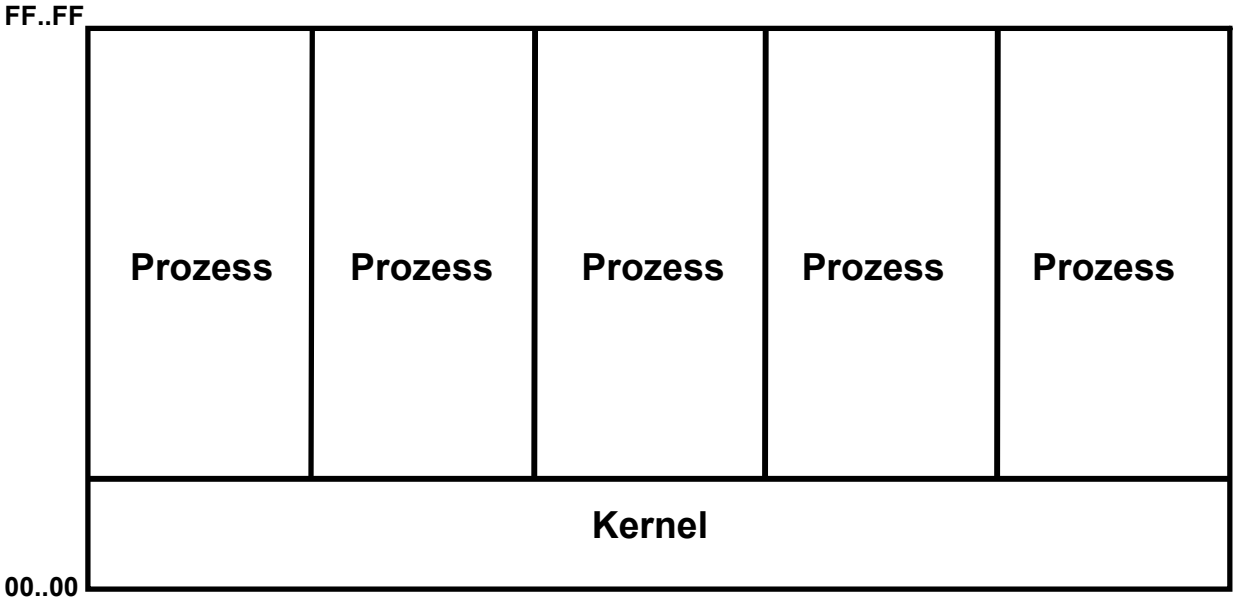
Einige der Regions beherbergen Subsysteme, die Teil des Betriebssystems sind, aber im Benutzerstatus laufen.


X = MVS Subsysteme (bei IPL geladen)
Batch-Jobs
//.....PROC=PROZY
//JOB4 .... CLASS=A
//.....
//.....PGM=HUGO
//JOB3 ..... CLASS=A
//.....
//.....PGM=IEBGENER
//JOB2 ..... CLASS=X
//.....
//..... PGM=DRUCK
//JOB1 ..... CLASS=B
Prozeduren Bibliothek
eigene
PROZY
PROZX
PROZ9
SYS1.PROCLIB
Lade - Programm - Bibliotheken
IBM
SORT
ABC
IEBGENER
SYS1.LINKLIB
eigene
DRUCK
ABC
LOHN
PROD.LIB
eigene
ABC
HUGO
TEST99
TEST.LIB
Endbenutzer
Individuelle Dialog Verarbeitung
(n-Tausend)
Master Konsole
Befehle
Msg's
16 MB/ 2 GB
MVS System
Max 32460 A.R.
1 X MASTER SCHEDULER
Kommun. System-Operator
2 X JES
Batch Job Steuerung
3 X INIT-1
CLASS=A/C
4 X INIT-2
CLASS=A/B
5 X INIT-n
CLASS=X
Batch Job Verarbeitung
JOB3
JOB1
JOB2
6 X VTAM
Netzwerk Steuerungs Programm
7 X TCAS
TSO Kontroll Programm
8 X RACF
Zugriffs Kontroll Programm
9 X SMS
Automat. Speicher Verwaltungs Programm
10 TSO USERA
11 TSO USERB
12 X CICS
Dialog Steuerungs Programm
13 X DB2/IMS
Datenbank Verwaltungs Programm
14
MVS System
LOGON
LOGON
Job Starten (SUBMIT)
Job Ende Message
Job Protokoll und Druckausgabe Prüfen
CLASS A
CLASS B
Spool
JCL
SYSIN (Intream)
INTERNAL-TEXT
Job-Protokoll
SYSOUT (Druck)
Temporäre Job-Daten von JES Verwaltet
S.1.H (JES/2 IBM)
3.8 (JES/3 IBM)
oder BETA92 / FLASHER
TSO/ISPF User
Programme und Jobs Erstellen, Bearbeiten und Testen
DB2 / IMS Datenbanken

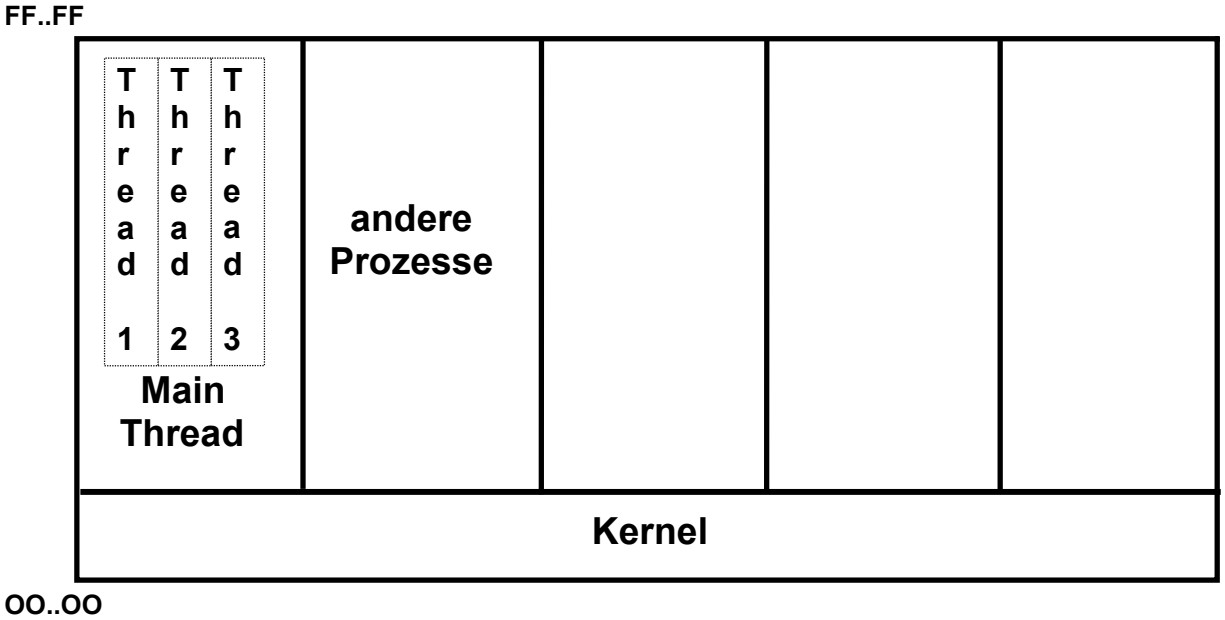
z/OS Subsysteme

CICS Transaktions Manager
IMS Transaktionsmanager
DB2 Datenbank
IMS Datenbank
VSAM index-sequentielles Dateisystem
XCF
JES2/3
Security Server (RACF, DCE Security, Firewall)
Netview, Systemview
WebSphere
UNIX Services
Distributed Computing Services (DCE, NFS, DFS, FTP)
LAN Server
Run Time Language Support
- **C/C++**
- **COBOL**
- **Object Oriented Cobol**
- **PL/1**
- **Fortran**
- **Assembler**

C/C++ Open Class Library
JDK, JVM



Prozess Ansatz



Thread Ansatz

Prozesse laufen in getrennten virtuellen Adressenräumen. Threads sind unabhängige Ausführungseinheiten, die innerhalb des gleichen virtuellen Adressenraums ablaufen. Ein Wechsel zwischen Threads erfordert deutlich weniger Aufwand als ein Wechsel zwischen Prozessen.

z/OS Prozessverwaltung

Ein z/OS Prozess besteht aus mehreren Arbeitseinheiten, die als „Tasks“ bezeichnet werden. Eine „Main Task“ generiert Subtasks; eine Subtask entspricht in etwa einem Thread in Unix oder Windows. Wenn ein Programm gestartet wird, erstellt z/OS hierfür einen Main Task Control Block (TCB). Das Programm kann weitere Subtasks erstellen mit Hilfe des ATTACH System Aufrufes (entsprechend dem Unix fork Befehl).

Da der ATTACH Overhead relativ groß ist, implementieren zeitkritische Subsysteme wie z.B. CICS ihr eigenes Subtasking. Im Kernelmodus existieren zusätzliche Mechanismen, die als SMB’s und SRB’s bezeichnet werden. Windows kennt im Kernelmodus einen ähnlichen Mechanismus, der als „Fibers“ (light weight threads) bezeichnet wird. Fibres werden durch die Anwendung gescheduled. Die CICS Portierung auf Windows benutzt Fibers.

Programmarten

Subsysteme sind Programmprodukte wie Datenbanken und Transaktionsmonitore, die Laufzeitumgebungen für eigentliche Benutzerprogramme zur Verfügung stellen.

Benutzerprogramme können sein:

- klassische z/OS (bzw. OS/390) Hintergrundprogramme
- Kundenanwendungen, die unter der Kontrolle von Subsystemen wie TSO,CICS, IMS oder Websphere ablaufen, oder
- UNIX-Programme, die die UNIX System Services unter z/OS ausnutzen.

Systems Management Funktionen werden für die Steuerung und Überwachung des Ablaufes benötigt. Es gibt sehr viele solcher Funktionen von der IBM und von Drittanbietern zur Überwachung des Betriebssystems und, da sich das Betriebssystem in weiten Bereichen selbst steuert, zur Überwachung der Subsysteme und der Kundenanwendungen.

Stapelverarbeitung
Job Scheduling

Zwei Arten der Datenverarbeitung

Interaktive Verarbeitung – Beispiel Excel:

Sie arbeiten mit einer großen Excel Tabelle und stoßen eine Berechnung an, die viele Sekunden oder Minuten dauert.

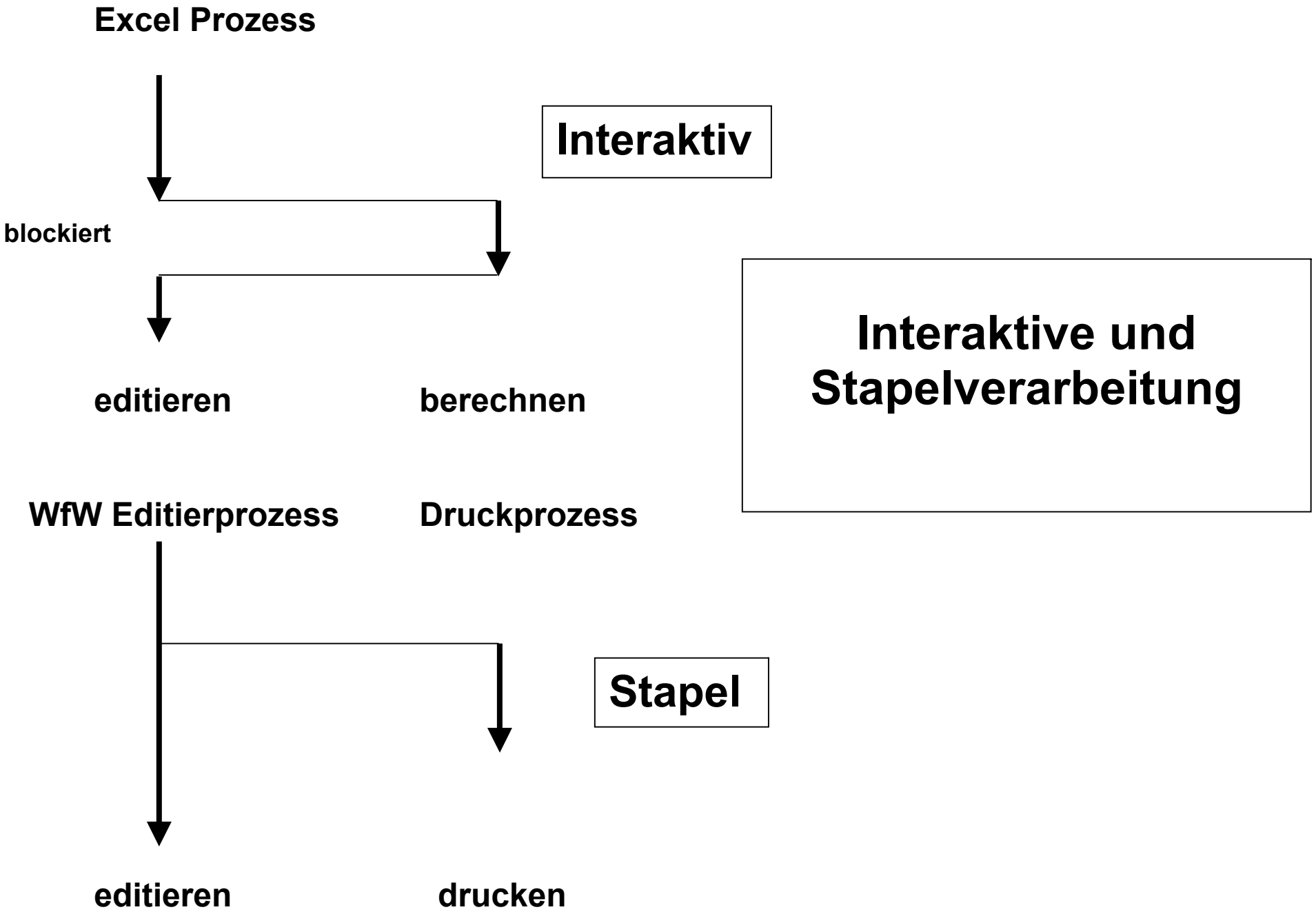
Während der Berechnung blockiert der EXEL Prozess, reagiert z.B. nicht auf Tastatur Eingaben.

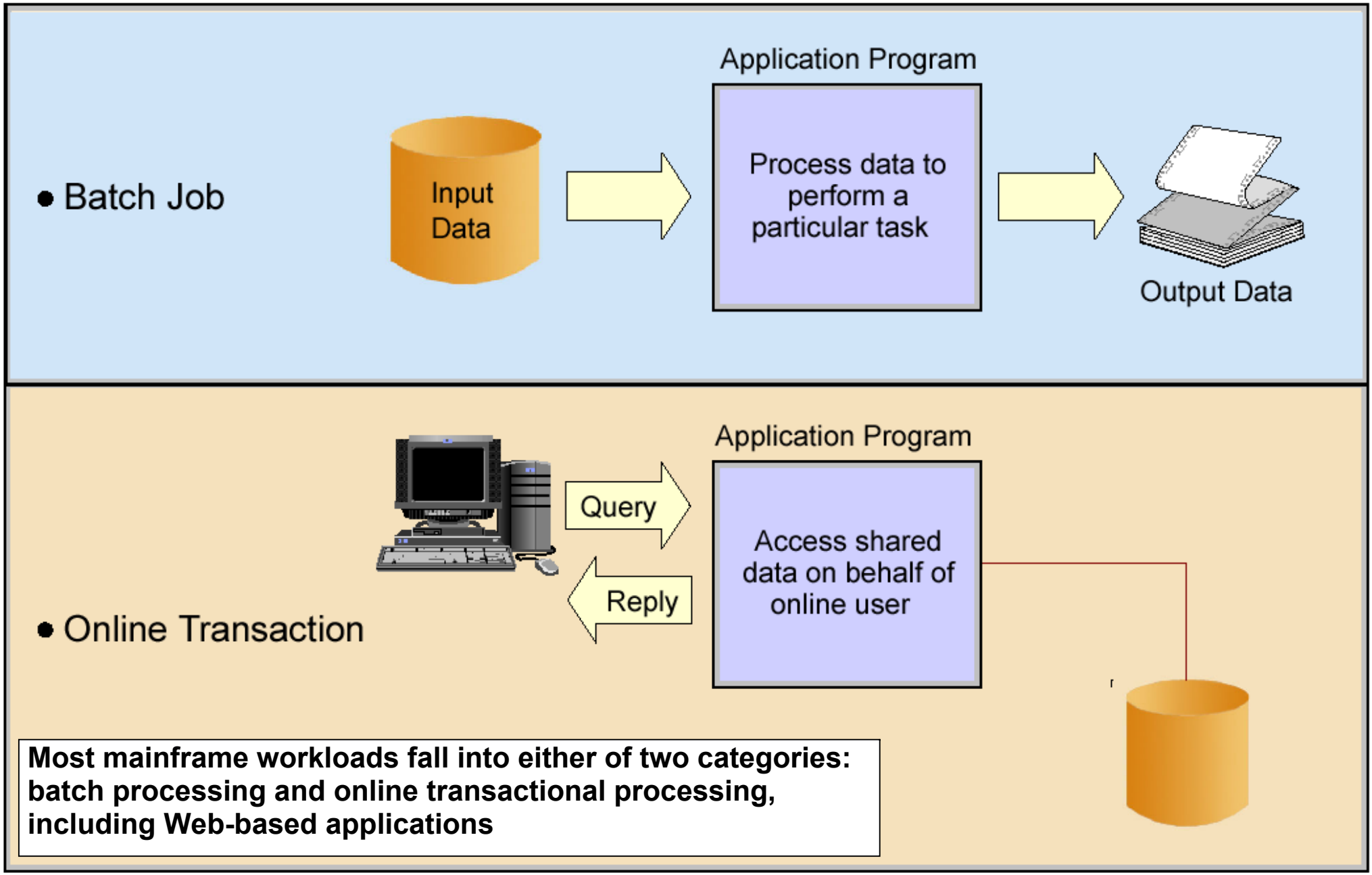
Stapelverarbeitung – Beispiel Word for Windows:

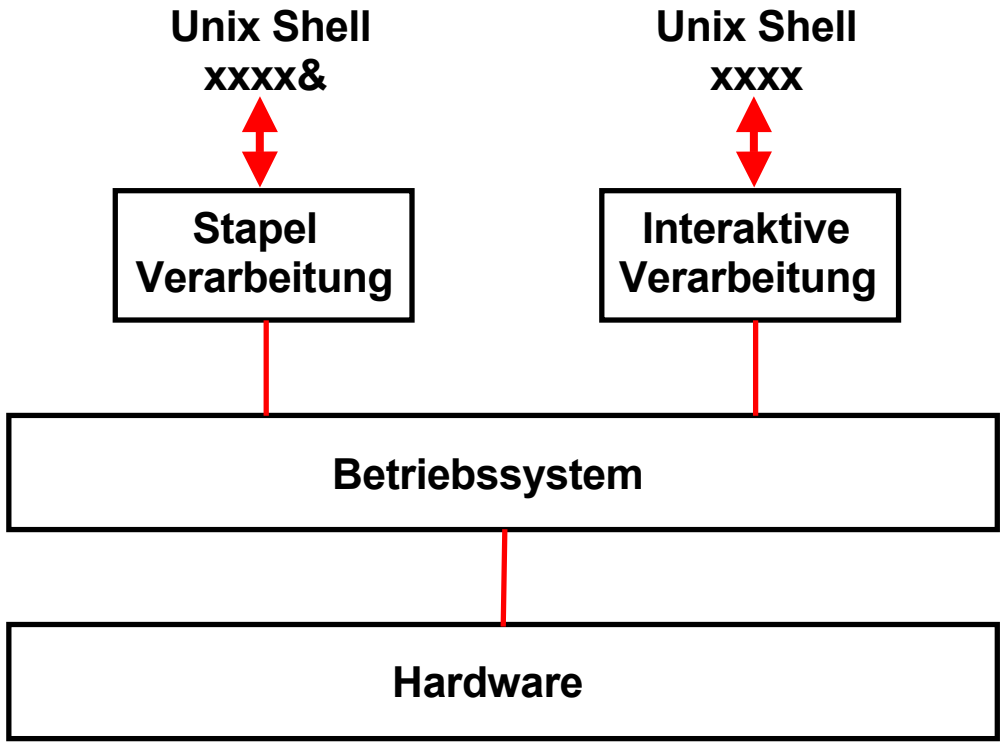
Sie editieren ihre Diplomarbeit mit Word for Windows. Dies ist ein interaktiver Prozess. Sie beschließen, die ganze Arbeit auf Ihrem Tintenstrahldrucker probeweise auszudrucken. Dies dauert viele Sekunden oder Minuten.

Während des Druckens blockiert der Word Prozess nicht. Sie können die Diplomarbeit weiter editieren.

Hierzu setzt Word for Windows neben dem Editierprozess einen getrennten Druckprozess auf, der als Stapelberareitungsprozess (batch job oder background Prozess) bezeichnet wird. Der Scheduler/Dispatcher des Betriebssystems stellt zeitscheibengesteuert beiden Prozessen CPU Zeit zur Verfügung.


Excel Prozess
Interaktiv
blockiert
editieren
berechnen
Interaktive und Stapelverarbeitung
WfW Editierprozess
Druckprozess
Stapel
editieren
drucken


• Batch Job
Input Data
Application Program
Process data to perform a particular task
Output Data
• Online Transaction
Query
Reply
Application Program
Access shared data on behalf of online user
Most mainframe workloads fall into either of two categories: batch processing and online transactional processing, including Web-based applications



Unix Verarbeitung - Grundstruktur
Zugriff über Telnet und Shell auf einen Unix Rechner

Unter Unix kann die Stapelverarbeitung (Batch Processing) als Sonderfall der interaktiven Verarbeitung betrieben werden. In der Shellsprache werden Batch-Aufträge durch ein nachgestelltes „&“ gekennzeichnet.

Stapelverarbeitung unter Linux

Der cron-Daemon ist eine Jobsteuerung von Unix bzw. Unix-artigen Betriebssystemen wie Linux, BSD oder Mac OS X, die wiederkehrende Aufgaben (cronjobs) automatisch zu einer bestimmten Zeit ausführen kann.

Hierzu werden Skripte und Programme zu vorgegebenen Zeiten gestartet. Der auszuführende Befehl wird in einer Tabelle, der so genannten crontab, gespeichert. Jeder Benutzer des Systems darf eine solche crontab anlegen.

Diese Tabelle besteht aus sechs Spalten; Die ersten fünf dienen der Zeitangabe (Minute, Stunde, Tag, Monat, Wochentage), die letzte enthält den Befehl. Die einzelnen Spalten werden durch Leerzeichen oder Tabulatoren getrennt.

Häufig führt der Cron-Daemon wichtige Programme für die Instandhaltung des Systems aus, wie zum Beispiel Dienste für das regelmäßige Archivieren und Löschen von Logdateien.

Beim Hochfahren eines Linux Systems wird als Erstes cron gestartet. cron wiederum startet einen Shell Prozess.

Griechisch chronos (χρόνος) bedeutet Zeit.

Cron

Cron is the name of a batch processing program that enables unix users to execute commands or scripts (groups of commands) automatically at a specified time/date. It is normally used for sys admin commands, like running a backup script, but can be used for anything. A common use today is connecting to the internet and downloading your email.

Vixie cron and ISC Cron are modern versions, included with most Linux distributions.

Cron is the software which most closely resembles the z/OS Batch Processing facilities. However, the z/OS Job Entry Subsystem (JES) provide functions which go far beyond those offered by cron.

Job Scheduler
Batch Processing

Eine Stapelverarbeitung erfolgt häufig in mehreren Schritten .

Beispiel: Monatliche Kreditabrechnung in einer Bank. Diese könnte z.B. aus den folgenden Schritten 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)

Ein derartiger Vorgang wird als Auftrag oder „Job“ bezeichnet. Ein Job besteht aus einzelnen „Job Steps“.

Ein Job Scheduler steuert die zeitliche Ablauffolge einer größeren Anzahl von Jobs.

Job scheduler

A job scheduler is an enterprise software application that is in charge of unattended background executions, commonly known as batch processing.

Synonyms are batch system, Distributed Resource Management System (DRMS), and Distributed Resource Manager (DRM).

Today's job schedulers typically provide a single point of control for definition and monitoring of background executions in a single or in a cluster of computers. Increasingly job schedulers are required to orchestrate the integration of real-time business activities with traditional background IT processing, across different business application environments.

Job scheduling should not be confused with process scheduling, which is the assignment of currently running processes to CPUs by the operating system.

Job Scheduler

Eine ganze Reihe von Software Unternehmen stellen Job Scheduler für Apple, Windows, Unix und Linux Plattformen her. Beispiele sind:

- **Dollar Universe der Firma Orsyp, http://www.orsyp.com/en.html**
- **OpenPBS (Portable Batch System), http://www.openpbs.org,**
- **open source package originally developed for NASA in the early 1990s**
- **PBS Pro der Firma PBS Grid Works, http://www.pbsgridworks.com, enhanced commercial version von OpenPBS).**
- **Global Event Control Server der Firma Vinzant Software, http://www.vinzantsoftware.com/**
- **ActiveBatch der Firma Advanced –System Concepts, http://www.advsyscon.com/**

Beispiel eines DOS 6.22 Jobs

auftrag.bat

```
f:
cd programme
able
baker
charlie
cd \
c:
```

f:\programme

able.exe	xxx.dat
baker.exe	yyy.dat
charlie.exe	zzz.dat
	aaa.dat

able

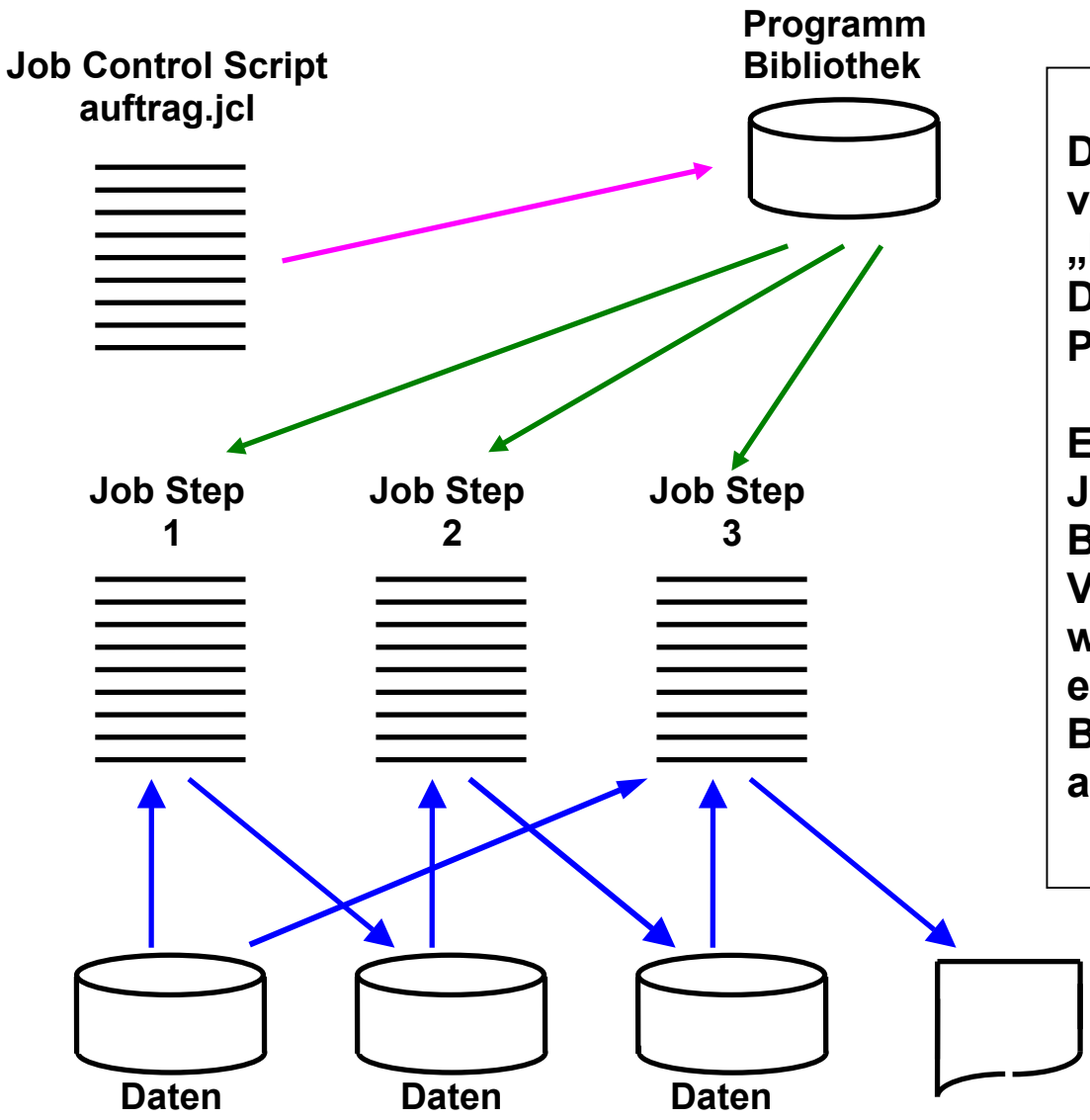
```
read xxx.dat
write yyy.dat
end
```

baker

```
read yyy.dat
write zzz.dat
end
```

charlie

```
read zzz.dat
read xxx.dat
write aaa.dat
end
```



Da das JCL Programm die verwendeten Dateien angibt, ist ein „late Binding“ der verwendeten Dateinen an die auszuführenden Programme möglich.

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.

Konzept eines z/OS Jobs

Die wichtigsten Sprachen

- **Cobol**
- **PL/1**
- **(Assembler)**
- **Fortran**
- **C/C++**
- **Java**

Skriptsprachen

- **TCL/TK**
- **Pearl**
- **PHP**
- **REXX**
- **JCL (z/OS Job Control Language)**

```
/* A short program to greet you                          */
/* First display a prompt                                */

say ‘Please type your name and then press ENTER:‘
parse pull answer              /* Get the reply into answer */

/* If nothing was typed, then use a fixed greeting     */
/* otherwise echo the name politely                     */

if answer=‘‘   then say ‘Hello Stranger! ‘
           else say ‘ Hello ‘ answer ‘!‘
```

REXX

Interpretative Scriptsprache, verfügbar für die meisten Betriebs-Systeme, vergleichbar zu Pearl oder Tcl/tk. Weit verbreitet bei den Benutzern von z/OS Rechnern. Vollwertige Programmiersprache.

```
//SPRUTHC  JOB (123456),'SPRUTH',CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1),
//            NOTIFY=&SYSUID,TIME=1440
//PROCLIB  JCLLIB ORDER=CBC.SCBCPRC
//CCL      EXEC PROC=EDCCB,
//           INFILE='SPRUTH.TEST.C(HELLO1)'
//           OUTFILE='SPRUTH.TEST.LOAD(HELLO1),DISP=SHR'
```

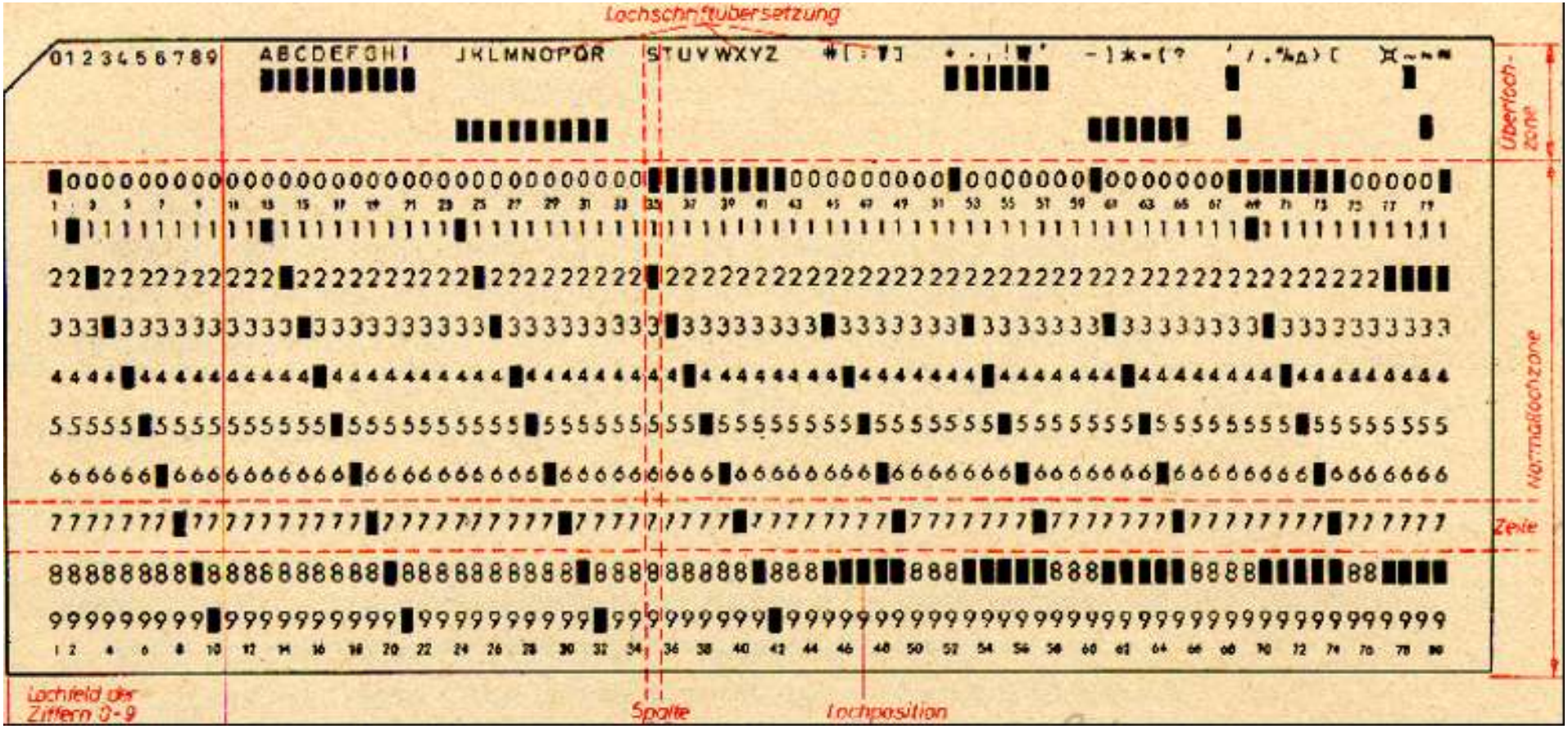
Ein einfaches JCL Script

JOB Statement markiert den Anfang eines Jobs

EXEC Statement bezeichnet Prozedur, die ausgeführt werden soll

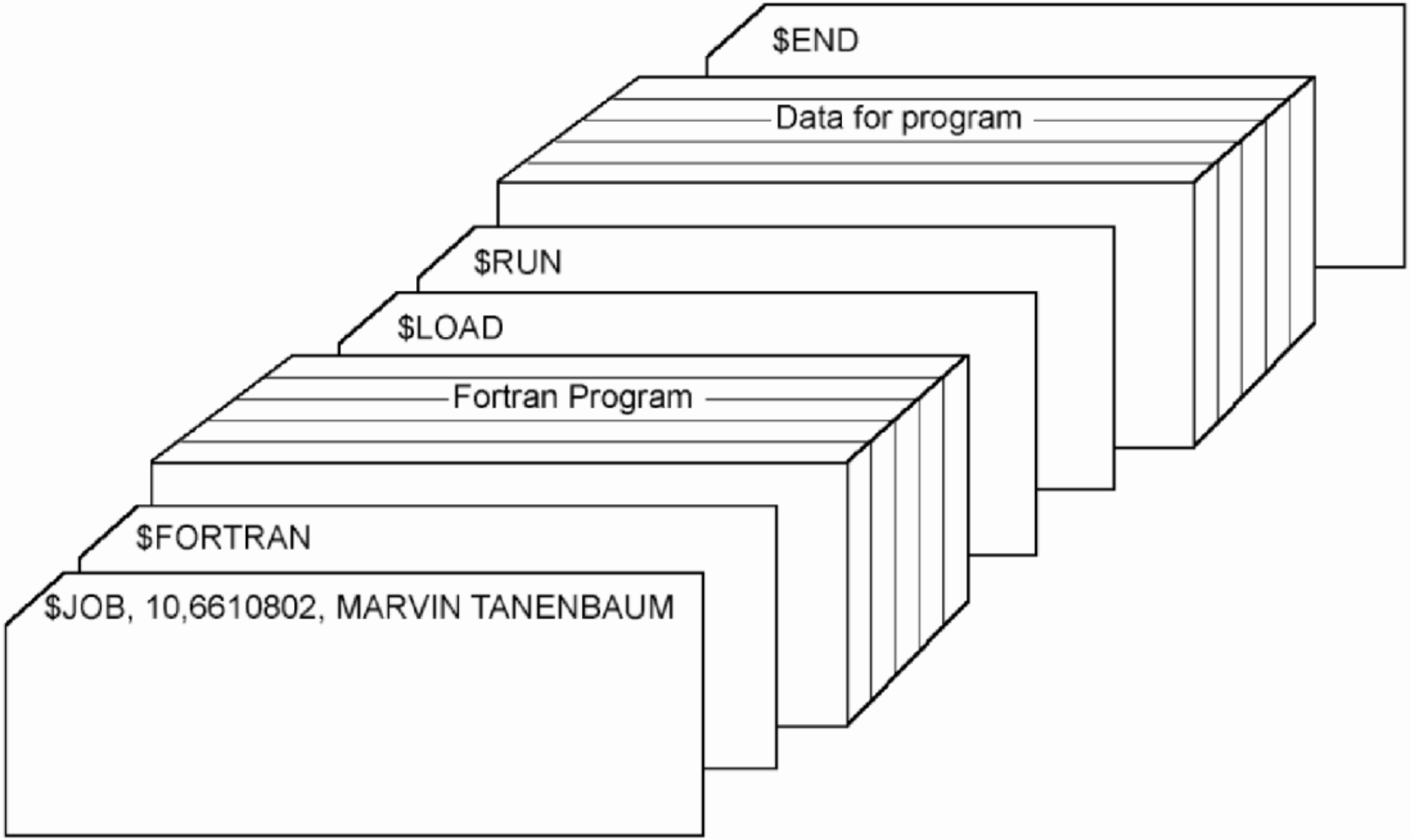
PROC Statement gibt die Adresse einer Procedur an

DD Statement bezeichnet die zu benutzenden Dateien (hier nicht verwendet)



IBM Lochkarte

v


$END
Data for program
$RUN
$LOAD
Fortran Program
$FORTRAN
$JOB, 10,6610802, MARVIN TANENBAUM

Batch Processing(1)

Batch in z/OS:
A job is submitted, it performs a long series of things (calculations, I/Os), writes ist results into a file, exits with a Return Code (RC). Examples:

- **Complex Queries (SQLJ instead of JDBC)**
- **Cheque and account processing**
  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 (wsadmin, RACF)**

Batch Processing(2)

Enterprise Batch requires sophisticated functionality (which is not available in distributed environments)

- **Time dependent execution**
- **Maintaining relationships**
- **Adminstrative and controlling functions**
- **Ability to control & manage system resources**
- **Efficient multi tasking / multi user**
- **Accounting abilities**
- **Batch Execution environment**

Batch Processing(3)

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 Parallel Sysplex zehntausende von Jobs gleichzeitig ausführen, verwalten und steuern .

Job Ausführung

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

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

1. **Zuordnung einer Datei, welche das Job Control Programm enthalten soll. (Kann unter ISPF geschehen).**

2. **Editieren und Abspeichern der JCL Datei (unter ISPF oder TSO Shell Kommandos)**

3. **Submission**

Da das JCL Programm die verwendeten Dateien angibt, ist ein „late Binding“ der verwendeten Dateinen an die auszuführenden Programme möglich.

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.

Submission eines Jobs

Ein „Job Control Programm“ (Äquivalent einer Bat File unter DOS 6.22) besteht aus einer Reihe von prozeduralen Befehlen, und wird in einer Script Sprache, der „Job Control Language“ (JCL) erstellt.

Die JCL hat ihren Ursprung im Lochkartenzeitalter. Jeder JCL Befehl beginnt mit den beiden Zeichen „ // „ und hat keinen Delimiter (z.B. „ ; „ in C++), sondern statt dessen eine feste Befehlslänge von genau 80 Zeichen (genau genommen 72 Zeichen plus 8 folgende Steuerzeichen).

Beispiel für einen JCL Befehl:

```
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400)
```

RECFM, FB, LRECL und BLKSIZE sind Schlüsselwörter der JCL Sprache.

Der Befehl besagt, daß die hiermit angesprochene Datei (bzw. ihr Data Control Block, DCB) ein Fixed Block (FB) Record Format (RECFM) hat (alle Datensätze haben die gleiche Länge), dessen Länge (Logical Record Length LRECL) 80 Bytes beträgt, und daß für die Übertragung vom/zum Hauptspeicher jeweils 5 Datensätze zu einem Block von (Blocksize BLKSIZE) 400 Bytes zusammengefaßt werden.

Literatur

M.Winkler: „MVS/ESA JCL“. Oldenbourg, 3. Auflage, 1999

Job Scheduler, Job Entry Subsystem

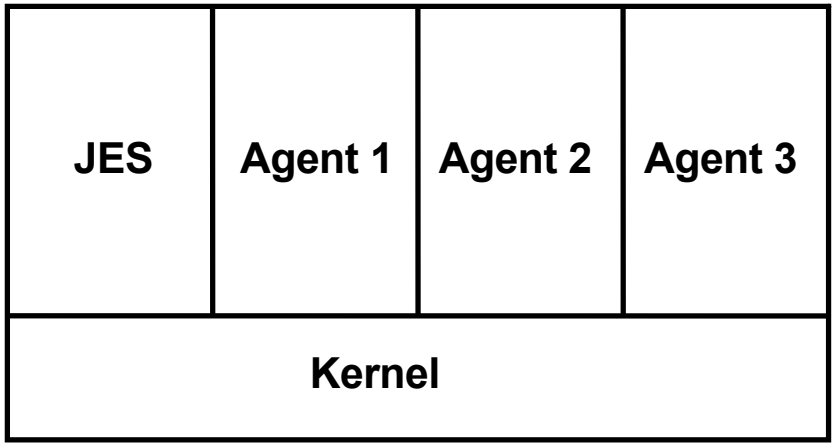
Job-Scheduler arbeiten nach dem Master/Agent-Konzept.

Der Master (Job entry subsystem, entweder JES2 oderJES3)

- **initiiert die Batches,**
- **reiht sie je nach Priorität in Warteschlangen ein und**
- **kontrolliert deren Ablauf.**

Die Agenten sind selbständige Prozesse, die in eigenen virtuellen Adressräumen laufen. Sie

- **führen die Jobs aus und**
- **senden Statusinformationen an die Steuerungskomponente und/oder**
- **fertigen Laufzeitberichte an.**



Job Entry Subsystem (JES)

Auf einem Großrechner laufen in der Regel zahlreiche Jobs gleichzeitig und parallel (hunderte oder tausende von z. T. langlaufenden Jobs während eines Tages).

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

Unter Windows ist dies z.B. der „Windows Scripting Host“. Unter Unix existieren verschiedene Software Produkte, z.B. LSF Batch (for clusters) und viele andere.

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.

JES bewirkt:

- **Zu Anfang der Bearbeitung die Zuordnung von Eingabe und Ausgabe Ressourcen**
- **Während der Verarbeitung die Zuordnung von Ressourcen wie CPU´s und Hauptspeicher**
- **Nach Abschluß der Bearbeitung die Freigabe der Ressourcen. Sie werden damit für andere Jobs verfügbar.**

Jobs werden an das Betriebssystem über JES gesteuerte Queue Server (Initiator) übergeben. Jeder Initiator bedient einen einzigen virtuellen Adressenraum, in dem er auch selbst untergebracht ist..

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

TSO

Server Zugriff

Unterschied zwischen Einzelplatzrechner und Client/Server Betriebssystemen.

Windows, Linux und und Unix werden für beides eingesetzt. z/OS ist ein reinrassige Server Betriebssysteme. Andere Beispiele für Server Betriebssysteme: Tandem NonStop, DEC OpenVMS.

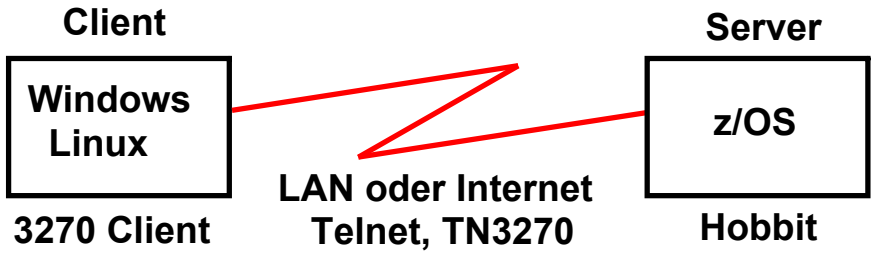
Ein Server Zugriff benötigt spezielle Client Software. Drei Client Alternativen:

1. **Selbstgeschriebene Anwendungen:**

   Sockets, RPC, Corba, DCOM, RMI

2. **Zeilenorientierte Klienten:**

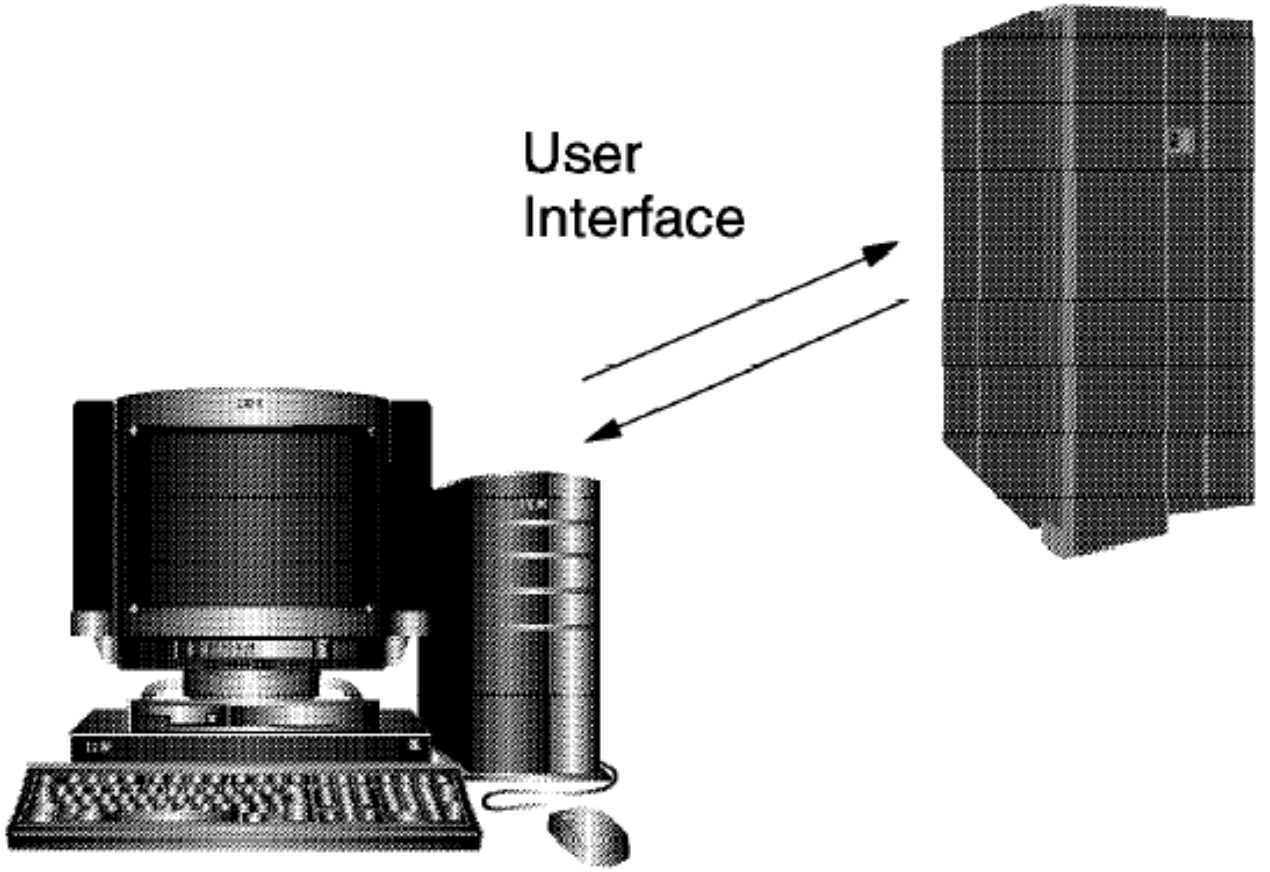
Unix Server	**Telnet Client, FTP**
z/OS Server	**3270 Client (3270 Emulator)**
VMS Server	**VT 100 Client**



3. **Klienten mit graphischer Oberfläche:**

Windows Server	**Citrix Client**
WWW Server	**Browser Client**
SAP R/3 Server	**SAPGUI Client**
Unix Server	**XWindows, Motif**
z/OS und OS/390 Server	**Servlet, JSP Client**

TSO/E
Time Sharing Option/Extended



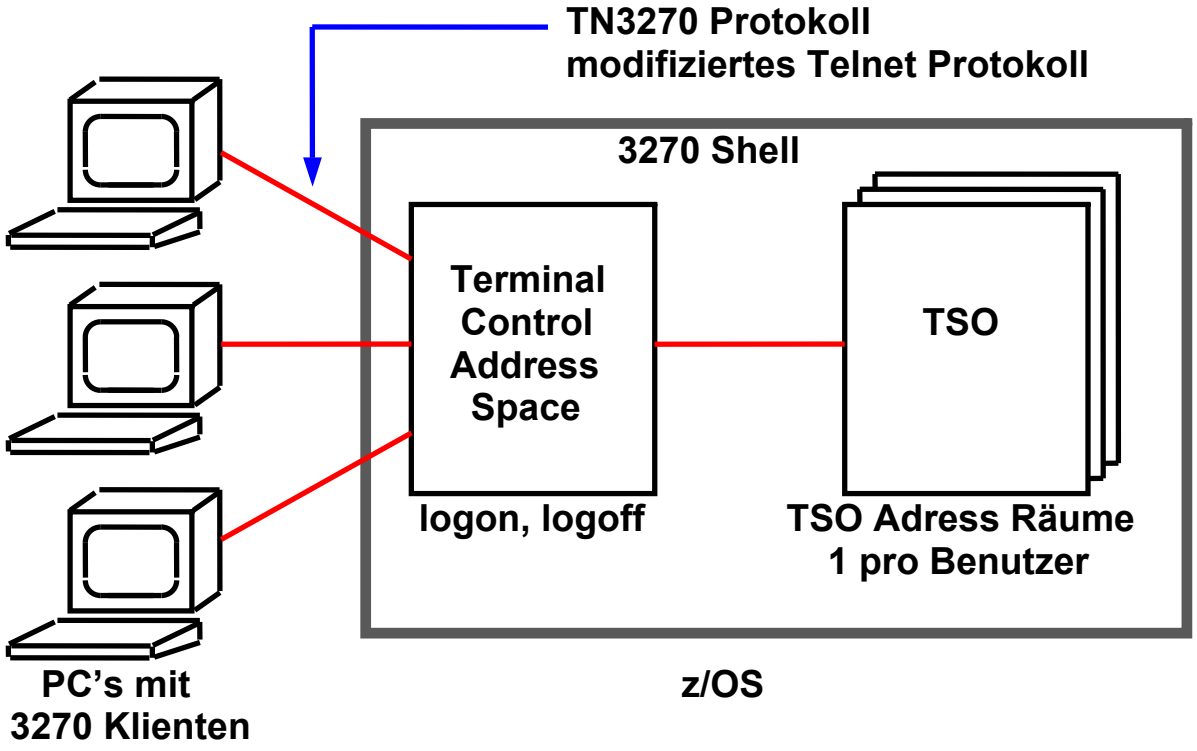
TSO ist eine zeilenorientierte Shell, vergleichbar mit der Windows „ „DOS Eingabeaufforderung“ oder den Unix/Linux Bourne, Korn, Bash oder C-Shells.

TSO („Time Sharing Option“)

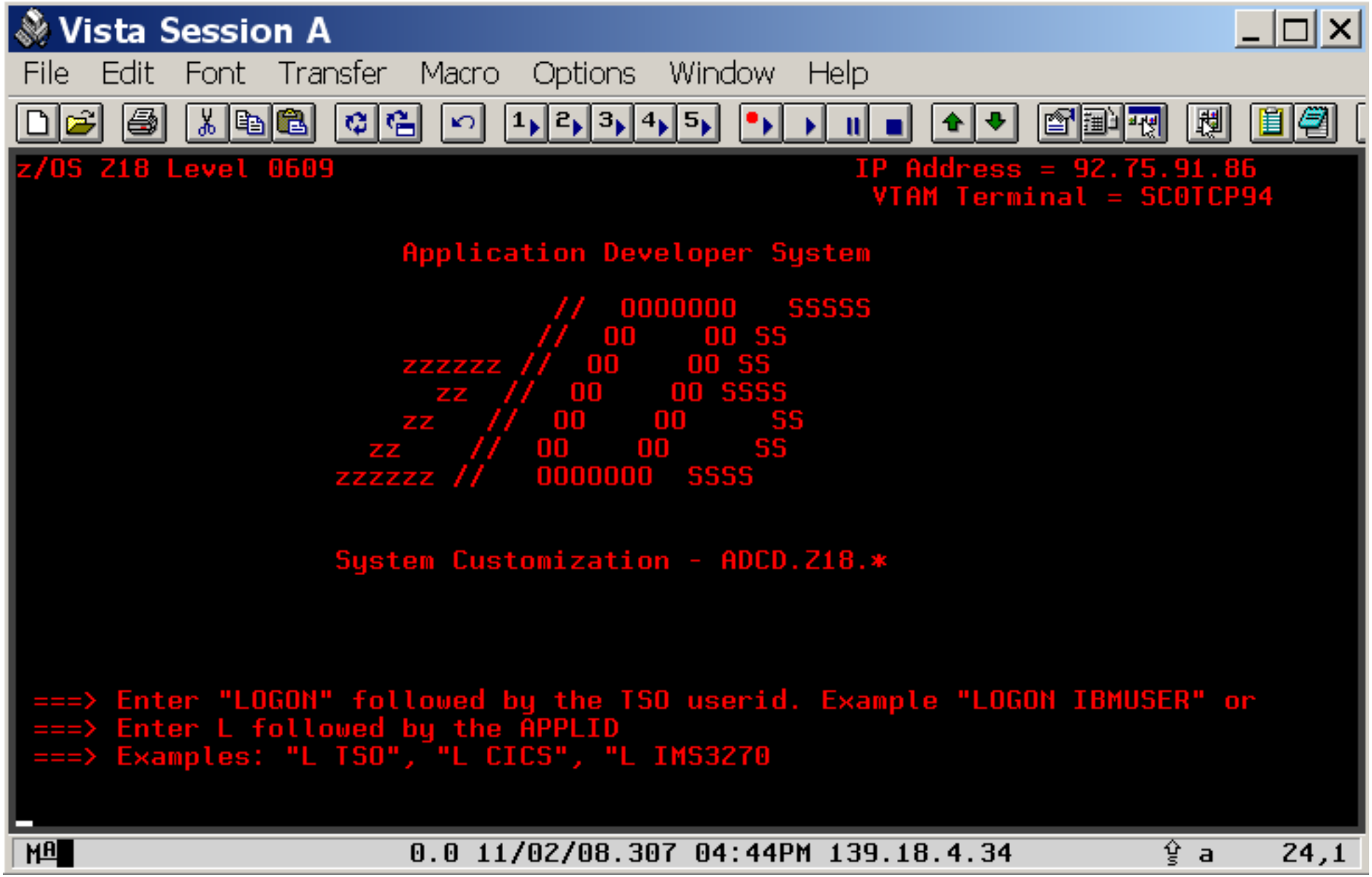
Vergleichbar mit dem ähnlichen UNIX Time Sharing. Hauptsächliche Anwendung: Software Entwicklung und Test,
sowie System Administration.

Systemadministratoren (Systemprogrammierer) verwenden TSO um Files zu editieren, Steuerungen vorzunehmen, Systemparameter zu setzen und einen Job Status zu überprüfen.

„Full Screen“ und Command Level Schnittstelle verfügbar. Die Full Screen Komponente wird als ISPF - Interactive System Productivity Facility bezeichnet.

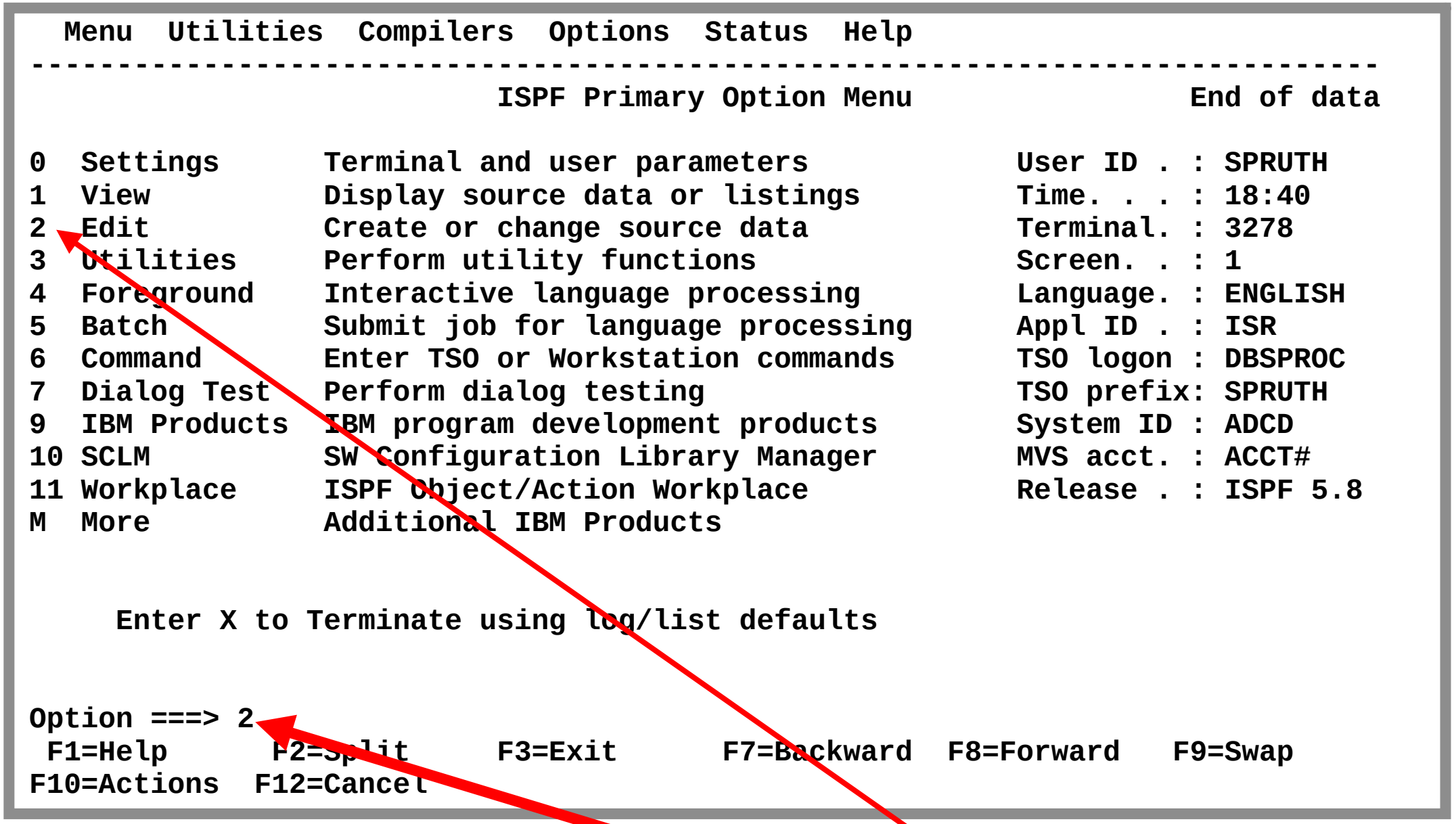


Der TSO Terminal Control Prozess (TCAS) arbeitet in einem eigenen Adressenraum. Er nimmt Nachrichten von den einzelnen TSO Klienten entgegen und leitet sie an den für den Benutzer eingerichteten separaten TSO Adressenraum weiter.



139.18.4.30 oder jedi.informatik.uni-leipzig.de

```
z/OS Z18 Level 0609                                   IP Address = 92.75.227.134
                                                       VTAM Terminal = SC0TCP33

                        Application Developer System

                                 //  OOOOOOO   SSSSS
                                //  OO    OO SS
                            zzzzzz //  OO    OO SS
                              zz  //  OO    OO SSSS
                             zz   //  OO    OO     SS
                           zz    //  OO    OO     SS
                         zzzzzz //   OOOOOOO  SSSS


                        System Customization - ADCD.Z18.*



 ===> Enter "LOGON" followed by the TSO userid. Example "LOGON IBMUSER" or
 ===> Enter L followed by the APPLID
 ===> Examples: "L TSO", "L CICS", "L IMS3270

L TSO
```



===> ist das Cursor Symbol. Eingabe einer „2“ ruft den „ISPF“ Editor auf.

Detaillierte Screen by Screen Tutorials sind zu finden unter http://jedi.informatik.uni-leipzig.de

Datasets

Dateisystem

Ein Dateisystem verbirgt die Eigenschaften der physischen Datenträger (Plattenspeicher, CD, evtl. Bandlaufwerke) weitestgehend vor dem Amwendungsprogrammierer.

Unix, XP

Dateien sind strukturlose Zeichenketten, welche über Namen identifiziert werden. Hierfür dienen Dateiverzeichnisse, die selbst wie Dateien aussehen und behandelt werden können.

Dateien werden sequentiell gelesen. Ein Direktzugriff wird mit Hilfe von Funktionen programmiert, die gezielt auf ein bestimmtes Zeichen in der Datei vor- oder zurücksetzen.

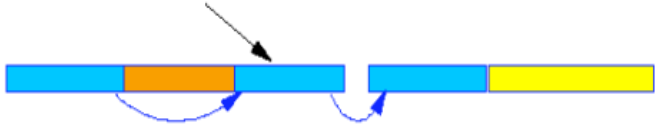
z/OS

Das Dateisystem (Filesystem) wird durch die Formattierung eines physikalischen Datenträgers definiert. Bei der Formattierung der Festplatte wird die Struktur der Datei festgelegt. Es gibt unterschiedliche Formattierungen für Dateien mit direktem Zugriff (DAM), sequentiellen Zugriff (SAM) oder index-sequentiellen Zugriff (VSAM).

Dateizugriffe benutzen an Stelle eines Dateiverzeichnisse „Kontrollblöcke“, welche die Datenbasis für unterschiedliche Betriebssystemfunktionen bilden.

UNIX, XP

- Dateien sind strukturlose Zeichenketten
- Zugriffsmethode: READ(fileid, buffer, length)
  - offset, length



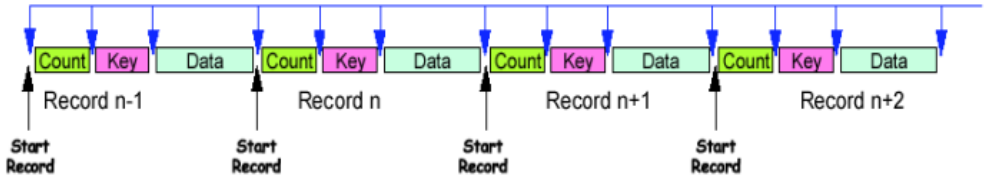
  - Fixed Block Architecture

- Zugriffsmethode: "raw"
  - Implementiert bei Datenbankanwendungen ausserhalb des Dateisystems

z/OS

- Record orientierter Zugriff
- Zugriffsmethode: GET recordid, buffer
  - Teil des Betriebssystems
  - sequentiel, indiziert, random
  - EXCP: Execute Channel Program: direct

- Count-Key-Data



- Dataset
  - Werden allokiert mit fester Grösse
  - Erweiterungen möglich über "Extends"
  - Können über mehrere Volumes gehen: "Spanned"

Vergleich Datenorganisation

Data Sets	Files
Access Methods	UNIX Services
z/OS data sets	HFS files
Record-oriented	Byte-oriented
(F(B), V(B), U)	
Two general methods	Hierarchical file structure
VSAM (ESDS,KSDS,RRDS,LDS)	directory/subdir/filename
Non-VSAM (SAM, PDS-E)	Path info max. 1023 char.
	File name max. 256 char.
UPPER CASE names	Mixed case names,
	case sensitive names
(max. 44 chars.)	(max. 256 chars.)
Location/attr. stored in catalog	Stored in z/OS container data sets

Data Sets und Files in z/OS

z/OS verwendet zwei vollständig unterschiedliche Verfahren um Daten auf Plattenspeichern anzuordnen;

- **Traditionale MVS Data Sets und**
- **Unix Files.**

Benutzung eines z/OS Systems

Erste Aufgabe: Zuordung (allocate) von Dateien

Frage: Was soll das ? Habe ich unter Windows noch nie gemacht.

Falsch. Wenn Sie unter Windows eine neue Datei erstmalig anlegen, müssen Sie entscheiten, ob sie unter C: oder D: gespeichert wird.

Wenn Ihr Rechner über 24 Plattenspeicher verfügt,
z.B. C:, D:, E:, ...Z: ,
wird die Verwaltung schon etwas schwieriger.

Was machen Sie wenn Ihr z/OS Rechner über 60 000 Plattenspeicher verfügt ?

Sie verwenden für die Verwaltung eine z/OS Komponente System Managed Storage (SMS).

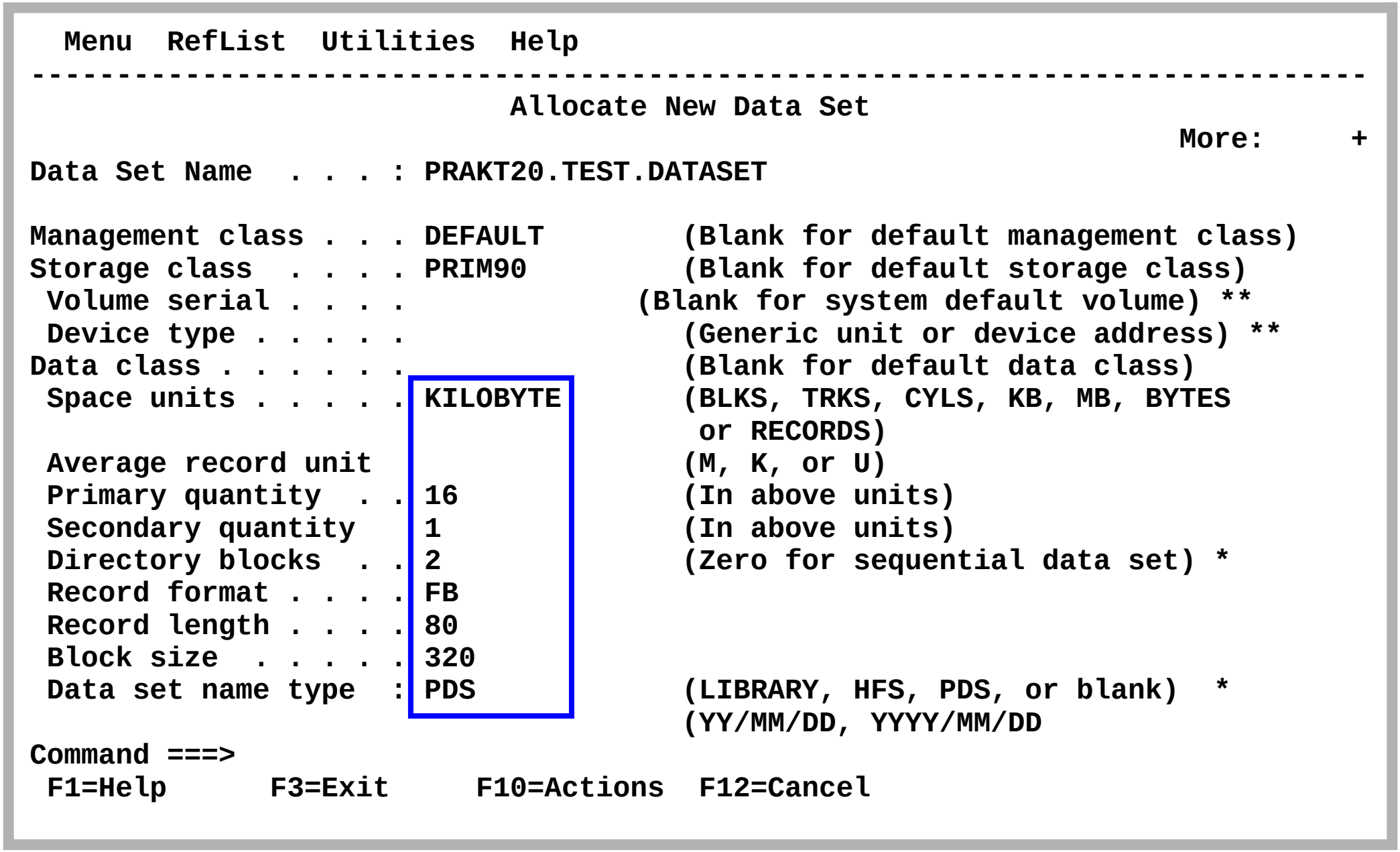
Wenn Sie Platz für eine neue Datei brauchen, melden Sie dies bei SMS an. Dieser Vorgang wird als Allocation bezeichnet.

Erstellen einer Datei: Entscheidungen

Benutzer, die Datenverarbeitung betreiben, haben beim Umgang mit den Daten täglich viele Entscheidungen zu treffen. Zusätzlich zum Umgang mit Themen, die nur die Daten oder die Anwendung betreffen, müssen sie die Speicherverwaltungsmaßnahmen der Installationen kennen. Sie müssen sich außerdem mit den Themen auseinandersetzen, die Format, Bearbeitung und Positionierung der Daten betreffen:

- **Welchen Wert soll die Blockungsgröße haben? Wieviel Speicherplatz ist erforderlich?**
- **Welcher Einheitentyp soll verwendet werden? Sollen die Daten in den Cache geschrieben werden? Soll eine Fehlerbehebung durchgeführt werden?**
- **Wie oft soll eine Sicherung oder Migration durchgeführt werden? Soll die Sicherung/Migration erhalten bleiben oder gelöscht werden?**
- **Welche Datenträger stehen für die Dateipositionierung zur Verfügung?**

Wenn die Verwendung von Datenverarbeitungsservices vereinfacht werden soll, müssen dem System einfachere Schnittstellen zur Verfügung gestellt werden. Insbesondere JCL ist einer der Bereiche, in denen Vereinfachungen vorgenommen werden.


Menu RefList Utilities Help
Allocate New Data Set
More: +
Data Set Name . . . : PRAKT20.TEST.DATASET
Management class . . . DEFAULT (Blank for default management class)
Storage class . . . . PRIM90 (Blank for default storage class)
Volume serial . . . . (Blank for system default volume) **
Device type . . . . . (Generic unit or device address) **
Data class . . . . . (Blank for default data class)
Space units . . . . KILOBYTE (BLKS, TRKS, CYLS, KB, MB, BYTES
or RECORDS)
Average record unit (M, K, or U)
Primary quantity . . 16 (In above units)
Secondary quantity 1 (In above units)
Directory blocks . . 2 (Zero for sequential data set) *
Record format . . . . FB
Record length 80
Block size 320
Data set name type : PDS (LIBRARY, HFS, PDS, or blank) *
(YY/MM/DD, YYYY/MM/DD
Command ===>
F1=Help F3=Exit F10=Actions F12=Cancel

DFSMS Klassen

Data Facility Storage Management System

Beim Neuanlegen einer Datei müssen angegeben werden:

Data Class

File System Attribute wie Record Format, Record Länge, Schlüssellänge bei VSAM Dateien

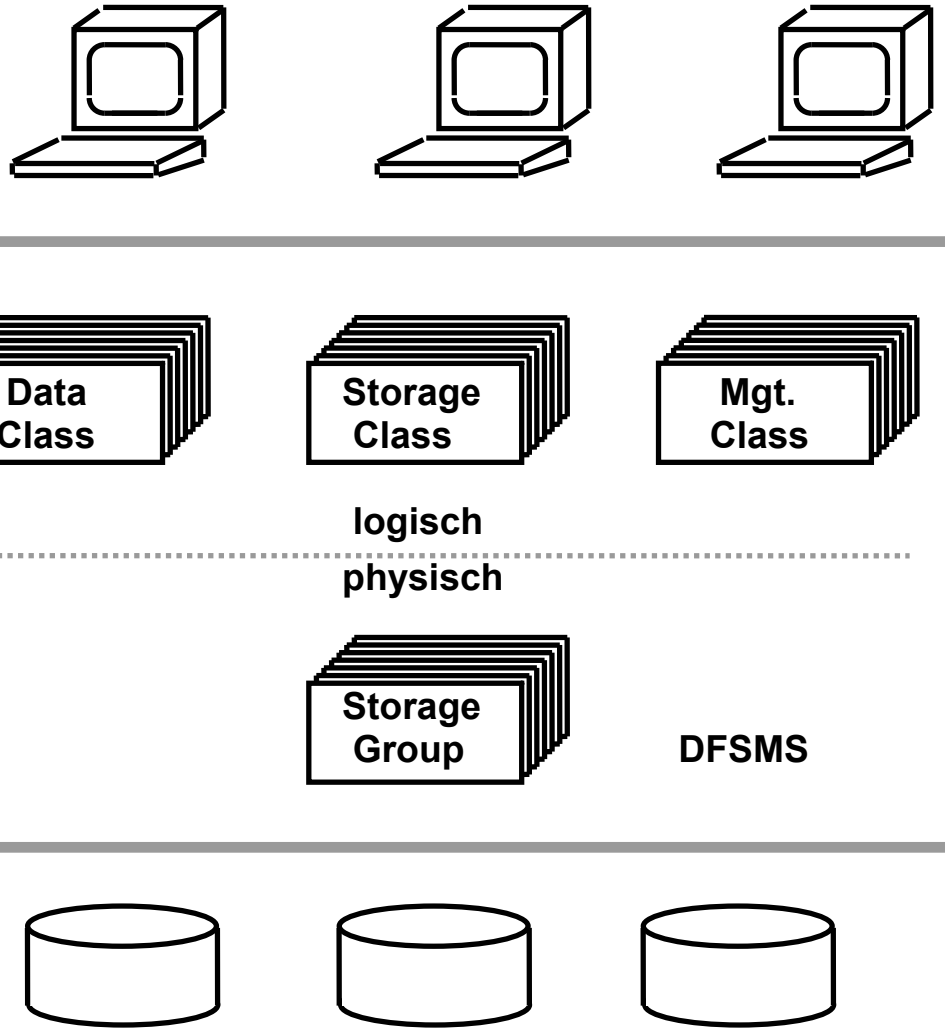
Storage Class

Performance Angaben wie Antwortzeit (ms), Nutzung von Read/Write Caching, Verwendung der Dual Copy Funktion

Management Class

Migration nach x Tagen, Anzahl Backup Versionen, schrittweiser Abbau der Backup Versionen, Löschen der Daten

DFSMS bewirkt die Zuordnung der Datei zu einer Storage Group, einer Gruppe von Plattenspeichern (Volumes)



Der Benutzer sieht nur die logische Sicht der Daten:

- **vorbereitete Datenmodelle in den Datenklassen**
- **in den Storageklassen festgelegte Serviceanforderungen**
- **Management Kriterien für die Auslagerung der Daten**

Data Facility Storage Management System

Lebenszyklus einer Datei
System managed Storage (SMS)

Anlegen der Datei durch den Benutzer

Benutzung (Schreiben und/oder Lesen der Daten)

Sicherungskopien anlegen

Platzverwaltung (freigeben / erweitern / komprimieren)

Auslagern von inaktiven Dateien, sowie Wiederbenutzung

Ausmustern von Sicherungskopien

Wiederherstellung von Dateien

Löschen der Datei

Unix System Services

Supervisor Calls

System z SVC's (Supervisor Calls) sind das Äquivalent zu den Unix System Calls.

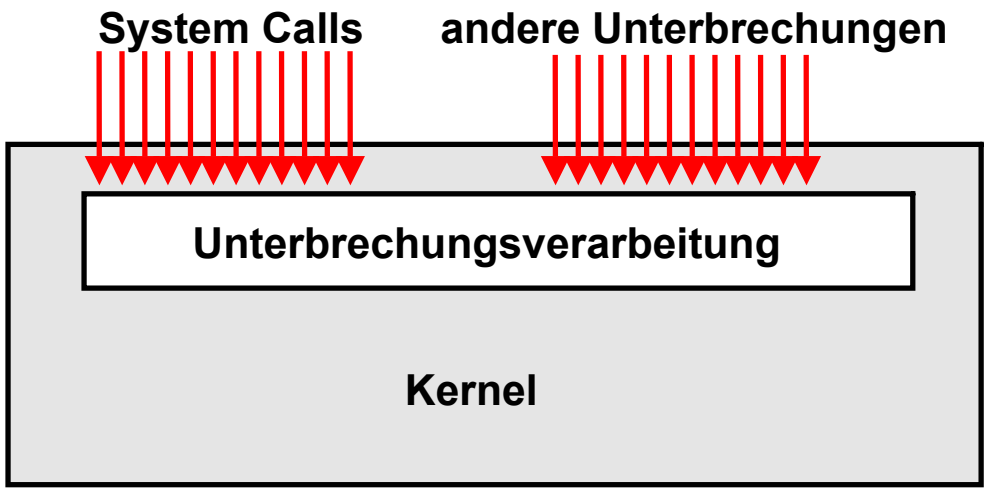
Supervisor Calls im weiteren Sinne sind Library Routinen, die eine Dienstleistung des Kernels in anspruch nehmen.

Supervisor Calls im engeren Sinne sind Maschinenbefehle, die über einen Übergang vom User Mode (Problem Status) zum Kernel Mode (Supervisor Status) einen Interrupt Handler des Kernels aufrufen. Bei der IA32 (Pentium) Architektur sind dies die INT und CALLGATE Maschinenbefehle.

Ein SVC Maschinenbefehl enthält einen 8 Bit Identifier, welcher die Art des Supervisor Calls identifiziert.

Beispiele sind:

GETMAIN SVC 10 Anforderung von Virtual Storage
OPEN SVC 19 Öffnen eines Data Sets
EXCP SVC 0 Lesen oder Schreiben von Daten
WAIT SVC 19 Warten auf ein Ereignis, z.B. Abschluß einer Lese Operation



Die Kernel aller Betriebssysteme haben de facto identische Funktionen, z. B.

- **Unterbrechungsverarbeitung**
- **Prozessmanagement**
- **Scheduling/Dispatching**
- **Ein/Ausgabe Steuerung**
- **Virtuelle Speicherverwaltung**
- **Dateimanagement**

Ein Unix Kernel unterscheidet sich von einem Windows Kernel durch die Syntax und Semantik der unterstützten System Calls, seine Shells sowie durch die unterstützen Dateisysteme.

Linux, Solaris, HP-UX und AIX haben unterschiedliche Kernel, aber (nahezu) identische System Calls..

Unix System Services (USS)

Die Unix System Services (USS) des z/OS Betriebssystems sind eine Erweiterung des z/OS Kernels um 1100 Unix System Calls, zwei Unix Shells und zwei Unix Datei-Systeme.

Damit wird aus z/OS ein Unix Betriebssystem.

z/OS vs. Unix

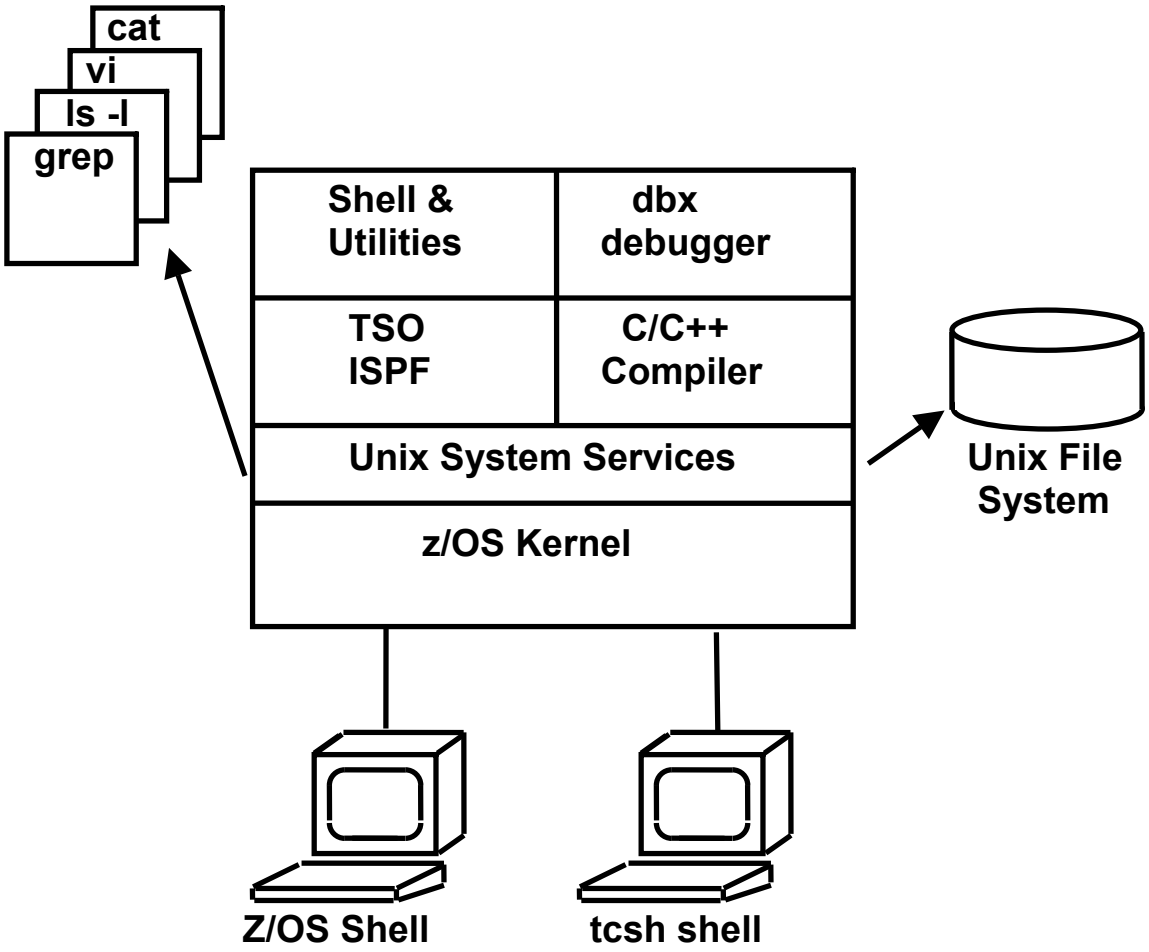
z/OS **mehrere 1000 E/A Operationen / s**

Unix **mehrere 100 E/A Operationen / s**

Native Unix Betriebssysteme für System z

- **Amdahl UTS (Universal Time Sharing System)**
  Marktführer, < 300 Installationen
- **Hitachi HI-OSF/1-M**
- **IBM AIX/ESA**
- **z/OS Unix System Services**

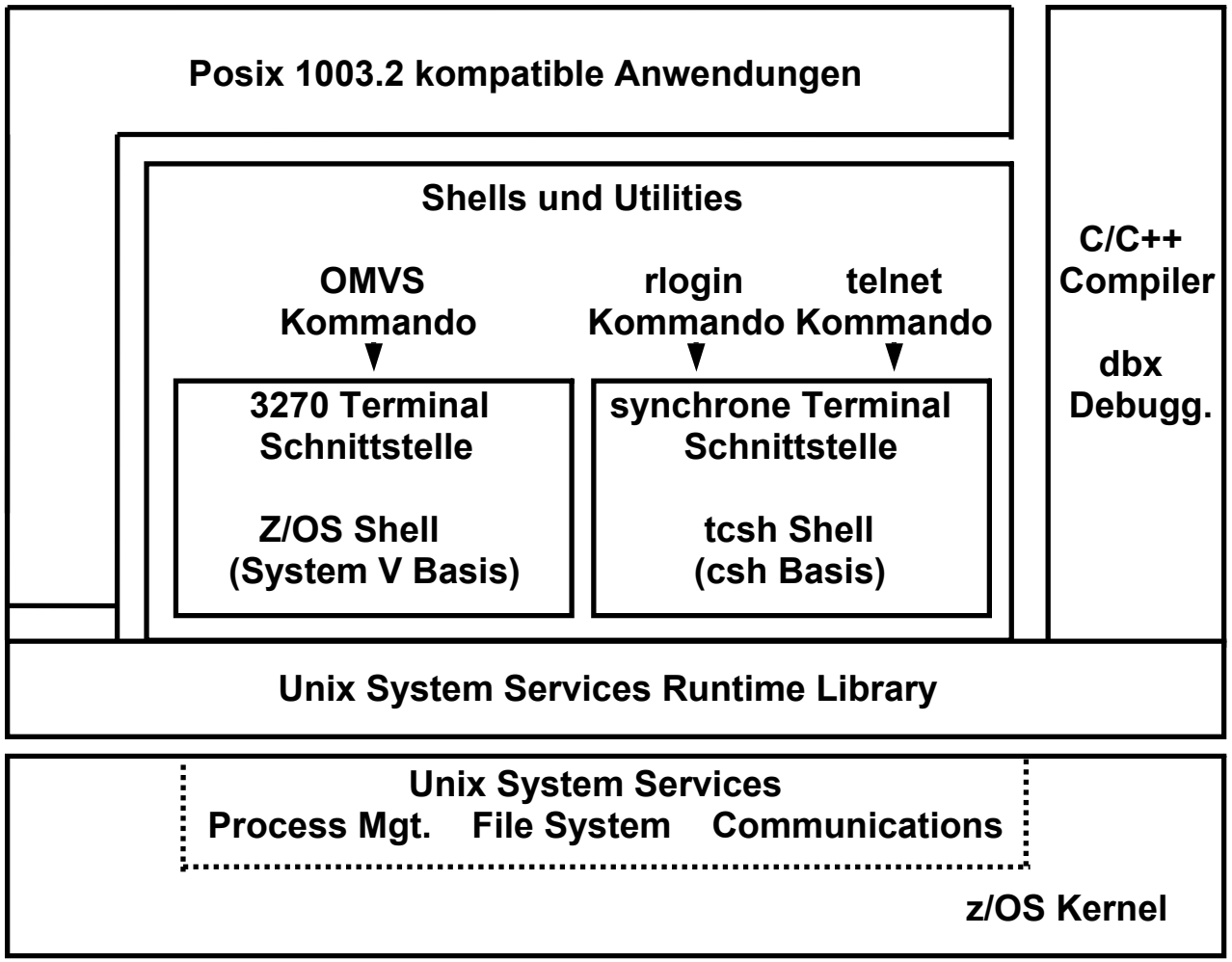
z/OS Unix System Services wurde früher als Open Edition MVS bezeichnet, 1100 Unix API´s



z/OS Unix System Services verfügt über zwei unterschiedliche Shells.

Die Z/OS Shell gleicht der Unix System V Shell mit einigen zusätzlichen Eigenschaften der Korn Shell. Sie wird meistens von TSO aus über das OMVS Kommando aufgerufen und benutzt den ISPF Editor.

Die tcsh Shell ist kompatibel mit der csh Shell, der Berkley Unix C Shell. Sie wird über rlogin oder telnet aufgerufen und verwendet den vi Editor.



Unix System Services (USS)

Die Unix System Services Kernel Funktion läuft in einem eigenen virtuellen Adressenraum, der als Teil des IPL (Boot) Vorgangs hochgefahren wird. Er wird wie jeder Unix Kernel über eine API aufgerufen, die aus C/C++ Function Calls besteht.

Das Byte-orientierte hierarchische File System arbeitet wie jedes Unix File System. Es wird in z/OS Data Sets abgebildet. Alle Files sind dem Unix System Services Kernel zugeordnet, und alle Ein-/Ausgabe Operationen bewirken Calls in den Kernel.

Die tsch Shell wird normalerweise über rlogin aufgerufen.

Es ist nicht unüblich, auf dem gleichen z/OS Rechner Unix System Services und zLinux parallel zu betreiben.

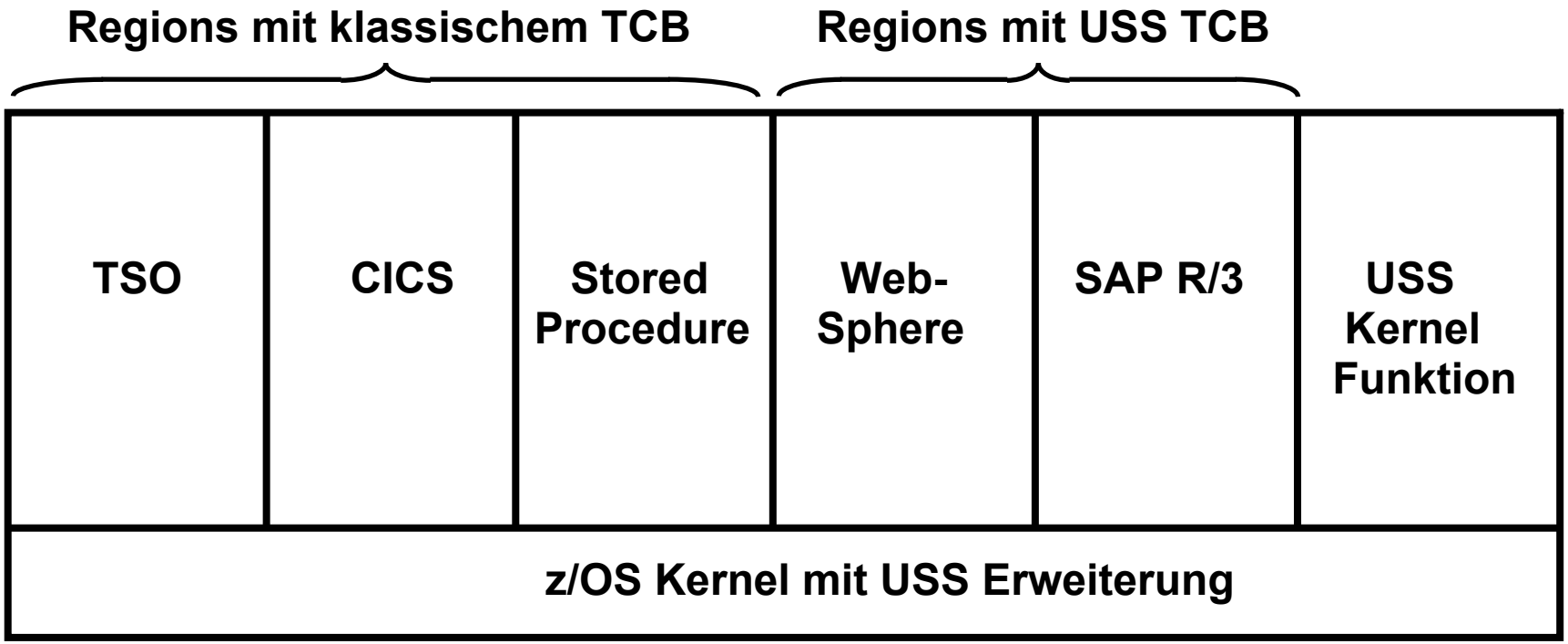
```
(C) Copyright Software Development Group, University of Waterloo, 1989.

All Rights Reserved.

U.S. Government users - RESTRICTED RIGHTS - Use, Duplication, or
Disclosure restricted by GSA-ADP schedule contract with IBM Corp.

IBM is a registered trademark of the IBM Corp.

SPRUTH : /u/spruth >ls -als
total 96
  16 drwxr-xr-x   4 BPXOINIT SYS1        8192 Oct  3 23:40 .
  16 drwxrwxrwx  120 BPXOINIT SYS1       8192 Sep 30 17:32 ..
   8 -rwx------   1 BPXOINIT SYS1         365 Mar 25  2002 .profile
   8 -rw-------   1 BPXOINIT SYS1        2347 Oct  3 23:42 .sh_history
   8 -rw-r-----   1 BPXOINIT SYS1        3715 Jul  7  2001 index.htm
   8 -rw-r-----   1 BPXOINIT SYS1        2806 Jul  7  2001 links01.htm
  16 drwxr-x---   2 BPXOINIT SYS1        8192 Mar 28  2002 sm390
  16 drwxr-xr-x   6 BPXOINIT SYS1        8192 Apr 12 20:06 was_samples
SPRUTH : /u/spruth >
 ===>
                                                                    INPUT
ESC=¢   1=Help      2=SubCmd    3=HlpRetrn  4=Top       5=Bottom    6=TSO
        7=BackScr   8=Scroll    9=NextSess 10=Refresh  11=FwdRetr  12=Retrieve
```

z/OS Unix System Services Shell

z/OS Unix System Services verfügt über zwei unterschiedliche Shells.

Die z/OS Shell gleicht der Unix System V Shell mit einigen zusätzlichen Eigenschaften der Korn Shell. Sie wird meistens von TSO aus über das OMVS Kommando aufgerufen und benutzt den ISPF Editor.

Die tcsh Shell ist kompatibel mit der csh Shell, der Berkley Unix C Shell. Sie wird über rlogin oder telnet aufgerufen und verwendet den vi Editor.



Regions, welche Unix System Services einsetzen benutzen einen modifizierten (erweiterten) TCB

Die Unix System Services Kernel Funktion läuft in einem eigenen virtuellen Adressenraum, der als Teil des IPL (Boot) Vorgangs hochgefahren wird.

Datenbanken

DB2 relationale Datenbank

***DB2 Universal Database (UDB)*, ist das z/OS (objekt-) relationale Datenbank-Produkt.**

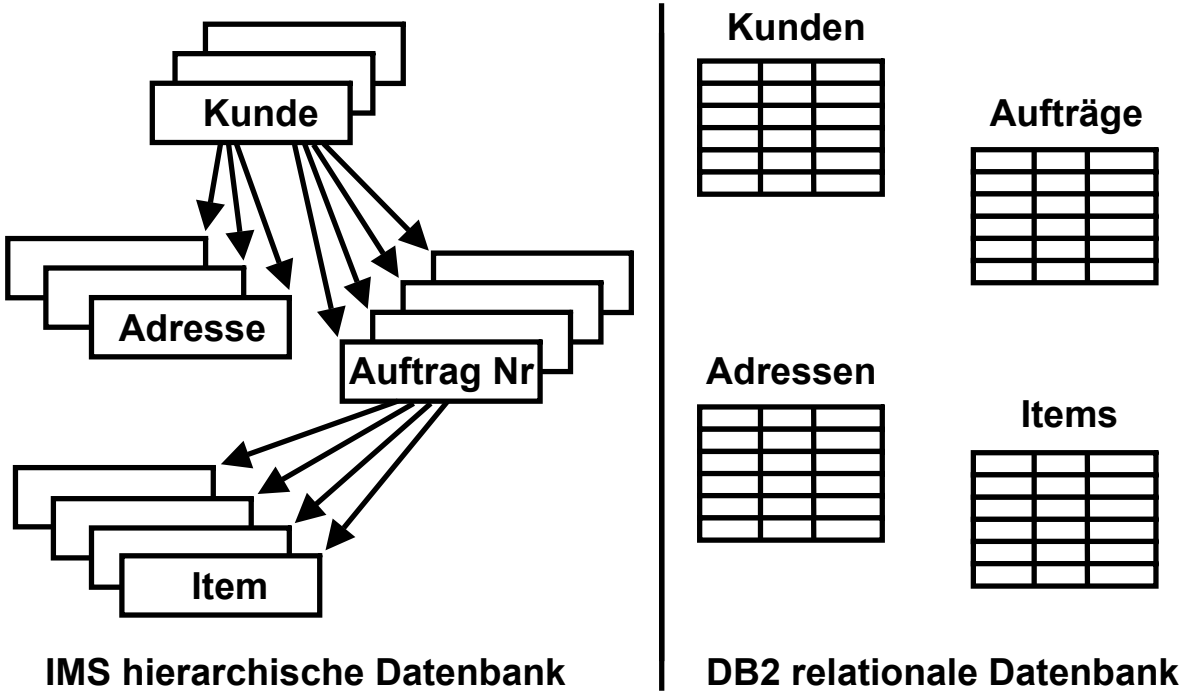
Identische Implementierung für alle UNIX-, Linux-, OS/2-, und Windows-Betriebssysteme.

Eine zweite getrennte Implementierung mit dem gleichen Namen und erweitertem Funktionsumfang ist für das z/OS-Betriebssystem verfügbar, Obwohl es sich um zwei getrennte Implementierungen handelt, ist die Kompatibilität sehr gut.

Neben Oracle- und Microsoft-SQL ist DB2 eines der drei führenden relationalen Datenbankprodukte. Unter z/OS ist es neben IMS die am häufigsten eingesetzte Datenbank. Andere populäre z/OS-Datenbanksysteme sind IDMS der Fa. Computer Associates, Oracle sowie Adabas der Fa. Software AG.

DB2 ist eine Server-Anwendung, die grundsätzlich in einem getrennten Adressraum läuft. Wie bei allen Server-Anwendungen ist ein Klient erforderlich, um auf einen DB2-Server zuzugreifen. Der Klient kann ein Anwendungsprogramm auf dem gleichen Rechner sein, oder auf einem getrennten Rechner laufen.

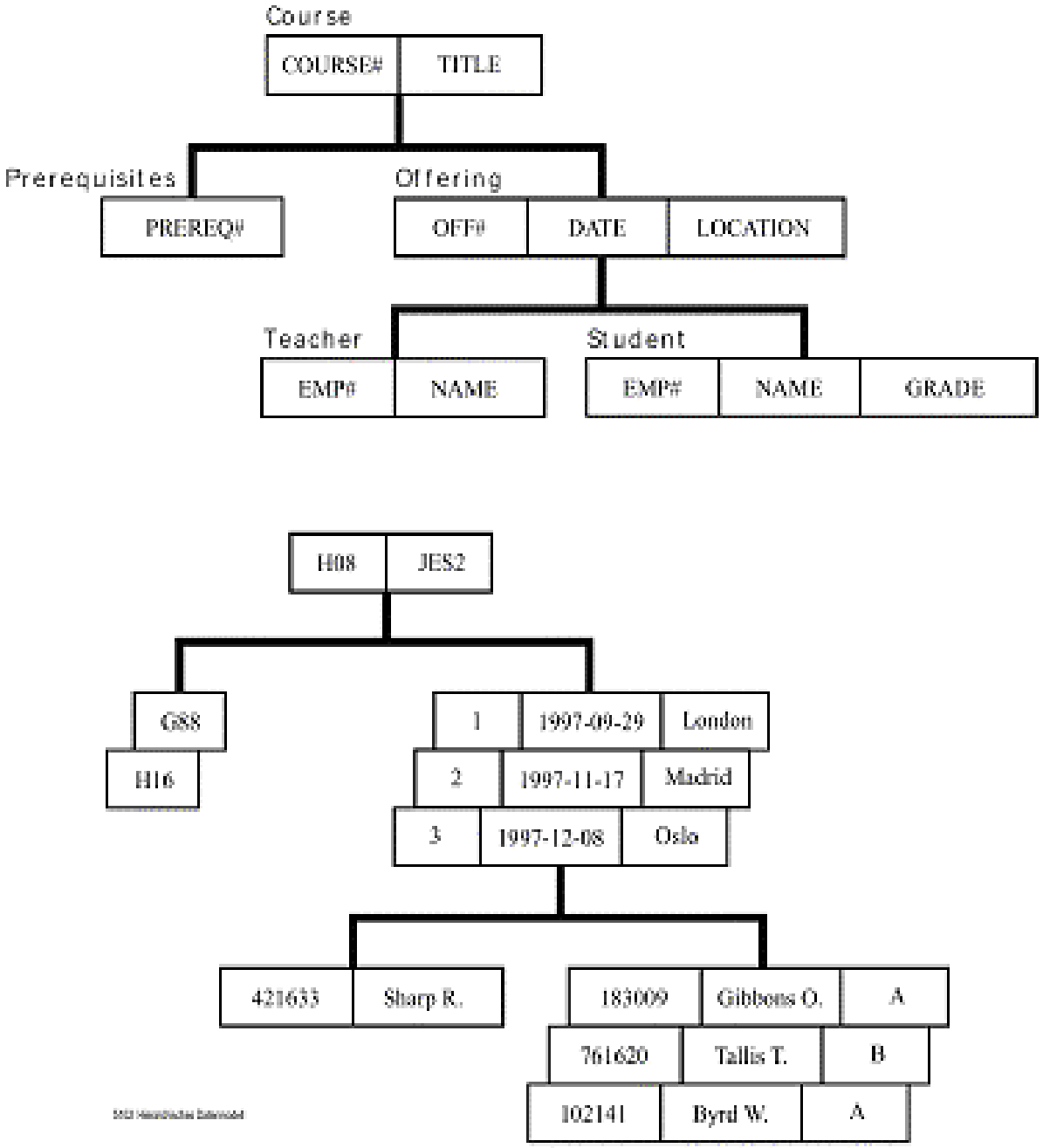
Der Zugriff kann mit Hilfe von SQL-Statements erfolgen, die in einem Anwendungsprogramm eingebettet sind. Alternativ existieren eine Reihe spezifischer SQL-Klienten-Anwendungen.



IMS Datenbanksystem

IMS ist im Gegensatz zu DB2 ein nicht-relationales, hierarchisches Datenbanksystem. IMS ermöglicht höhere Transaktionsraten als DB2.

Der CICS Transaktionsmonitor kann auf IMS Daten Zugreifen. Daneben verfügt IMS über einen eigenen Transaktionsmonitor IMS DB/DC.



IMS Datenbank-system

Das Beispiel zeigt im oberen Teil einen Baum-Typen und im unteren Teil eine Ausprägung dieses Baum-Typs, einen Kurs, seine Termine und seine dazugehörigen Buchungen.

IMS Datenbanksystem

IMS besteht aus zwei Komponenten, dem Database Manager (IMS DM bzw. DB) und dem Transaction Manager (IMS TM).

Das hierarchische Datenmodell des IMS besteht aus einer geordneten Menge von Bäumen, genauer aus Ausprägungen von Bäumen eines bestimmten Typs. Jeder Baum-Typ enthält eine Basis (Root-Segment) und keinen, einen oder mehrere Unterbaum-Typen. Der Unterbaum-Typ seinerseits enthält wiederum eine Basis und ggf. Unterbäume. Die Basis ist jeweils ein logischer Satz (Segment) bestehend aus einem oder mehreren Feldern (Fields).

IMS gestattet die physische und logische Anordnung von Segmenten zu entsprechenden physischen und logischen Datenbanken.

RACF

z/OS “SecureWay” Security Server

- **LDAP Server - Secure Directory Server**

- **Kerberos Network Authentication Service, einschliesslich Kryptographie und Firewall Unterstützung. Hierzu Hardware Unterstützung:**
  - **Zusätzliche Maschinenbefehle für kryptographische Operationen**
  - **Kryptographie Assist-Prozessoren**

- **RACF**

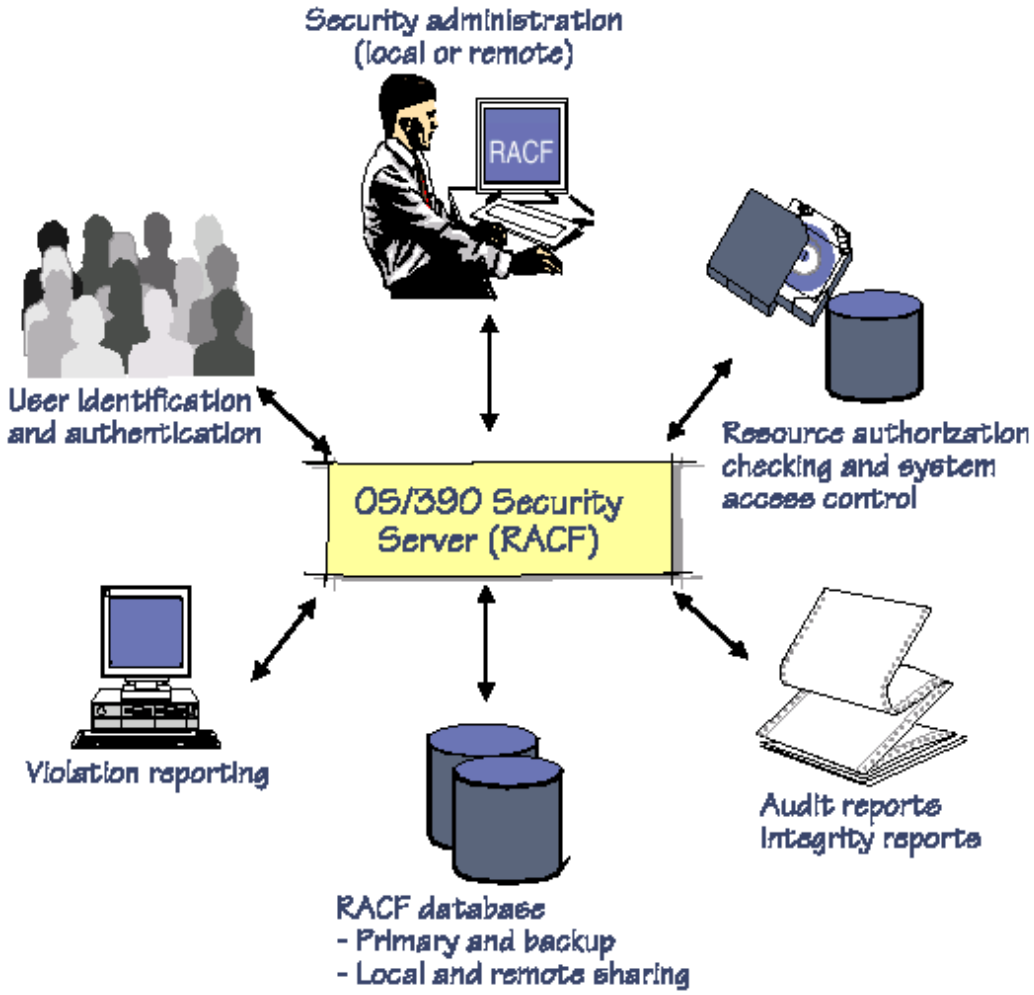
RACF
Resource Access Control Facility

RACF entscheidet,

- **ob der Benutzer für RACF definiert wird**
- **ob der Benutzer ein gültiges Password , PassTicket, Operator Identification Card und einen gültigen Gruppen-Namen verwendet.**
- **ob der Benutzer das System an diesem Tag und zu dieser Tageszeit benutzen darf**
- **ob der Benutzer authorisiert ist, auf das Terminal (Tag und Zeit) zuzugreifen**
- **ob der Benutzer authorisiert ist, auf die Anwendung zuzugreifen**
- **ob der Benutzer authorisiert ist, auf spezifische Daten zuzugreifen**

zSeries bzw. S/390 ist die einzige Rechnerarchitektur auf dem Markt, die über Hardware Speicherschutzschlüssel verfügt.

RACF



RACF bewirkt:

- **Identifizierung und Authentifizierung von Benutzern**
- **Benutzer Authorisierung für Zugriff auf geschützte Resourcen**
- **Logging und Berichte über unauthorisierte Zugriffe**
- **Überwacht die Art, wie auf Resourcen zugegriffen wird**
- **Anwendungen können RACF Macros benutzen**
- **Audit Trail**

Literatur : IBM Form No. GC28-1912-06

RACF

RACF benutzt das Konzept von zu schützenden„Resourcen“.

Resourcen werden in „Klassen“ aufgeteilt. Beispiele für Klassen sind:

- **Benutzer**
- **Dateien**
- **CICS Transaktionen**
- **Datenbank Rechte**
- **Terminals**

Jedem Element einer Klasse wird ein „Profil“ zugeordnet. Das Profil besagt, welche sicherheitsrelevanten Berechtigingen vorhanden sind. Die Profile werden in einer Systemdatenbank, der RACF Profildatenbank, abgespeichert.

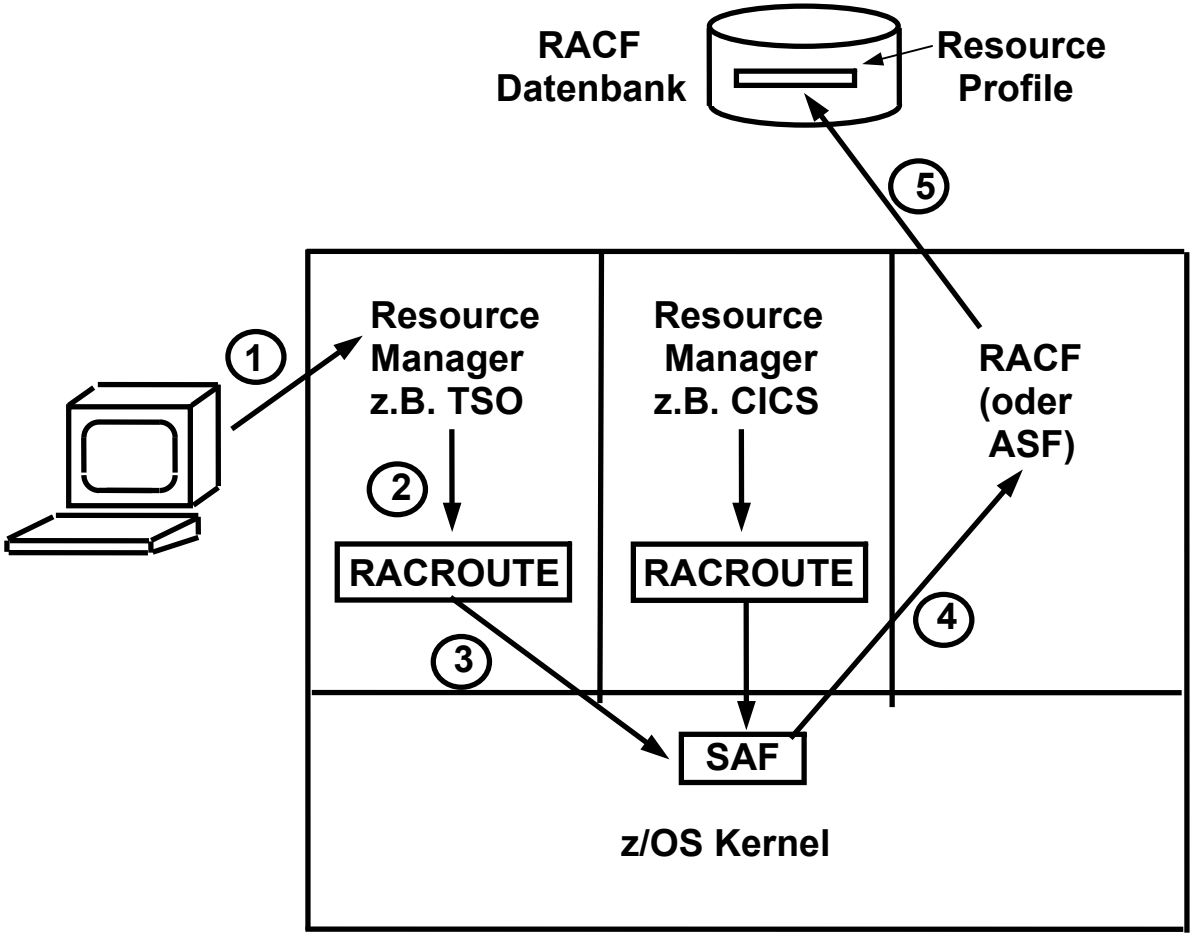
Beispiel: Ein interaktiver Benutzer logged sich ein. Der Login Prozess überprüft, ob die Login Berechtigung besteht. Er überprüft weiterhin, ob das Terminal, von dem der Benutzer sich einwählt, eine Zugangsberechtigung hat. Hierzu ruft der Login Prozess RACF auf, welches die entsprechenden Profile in seiner Datenbank konsultiert.

Anschließend ruft der Benutzer ein Programm auf, welches auf eine Datei zugreift. Die OPEN Routine ruft RACF auf, welches das Profil der Datei bezüglich der Zugriffsrechte befragt.

Benutzer Profile enthalten „Capabilities“. Datei Profile enthalten „Access Control Listen“.

Zugriffsrechte können von spezifischen granularen Bedingungen abhängig gemacht werden; z.B. der Zugriff auf eine Datenbank kann von der Nutzung eines spezifischen Anwendungsprogramms abhängig gemacht werden, oder auf bestimmte Tageszeiten begrenzt sein.

Problem: Wartung der Profile.

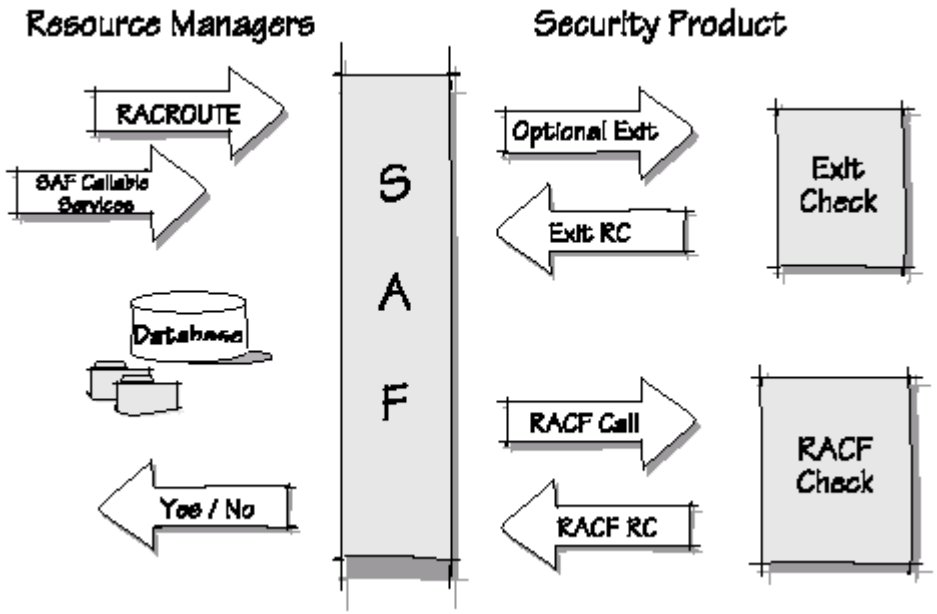


1. Ein Benutzer greift auf eine Resource über einen Resource Manager zu, z.B. TSO
2. Der Resource Manager benutzt einen System Call „RACROUTE“ um auf die Security Access Facility (SAF) des z/OS Kernels zuzugreifen. SAF ist eine zentrale Anlaufstelle für alle sicherheitsrelevanten Aufgaben.
3. SAF ruft ein Zugriffskontrolle Subsystem auf. Dies ist normalerweise RACF. (Eine Alternative ist die Access Control Facility der Fa. Computer Associates, die ähnlich arbeitet).
4. RACF greift auf einen „Profile“ Datensatz in seiner eigenen RACF Datenbank zu und überprüft die Zugangsberechtigungen
5. Das Ergebnis teilt RACF dem anfragenden Resource Manager mit.

RACF Arbeitsweise

RACF

Z/OS spezifiziert kritische Events innerhalb des Betriebssystems als sicherheitssensitive Verzweigungen. Die MVS Security Authorisation Facility (SAF) des z/OS Kernels bewirkt an diesen Stellen den Aufruf einer Security Engine, eines Prozesses, der im Benutzer Status läuft.

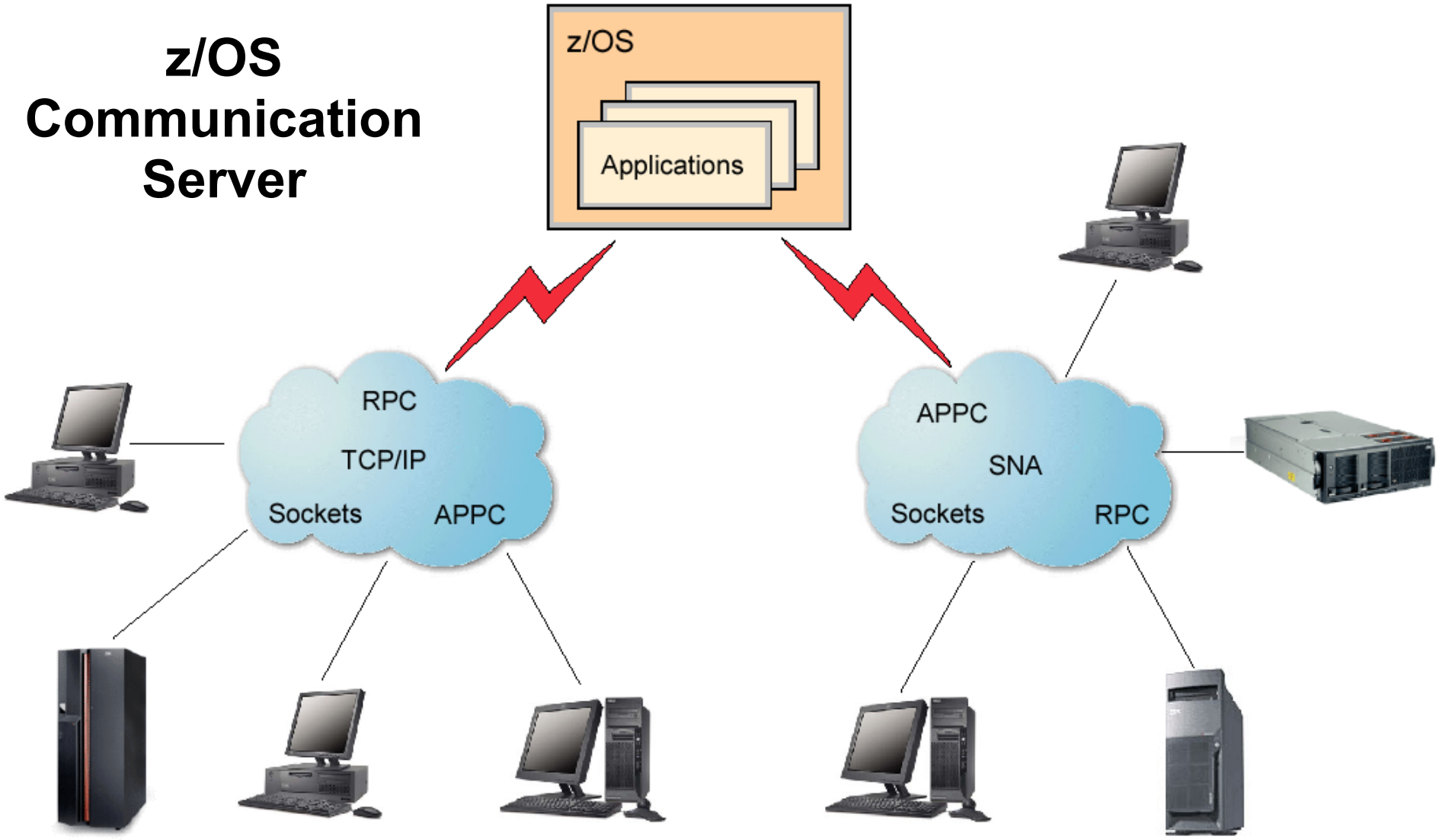


In z/OS ist diese Security Engine häufig RACF; alternative z/OS Security Engines sind ACF/2 oder TopSecret der Firma Computer Associates.

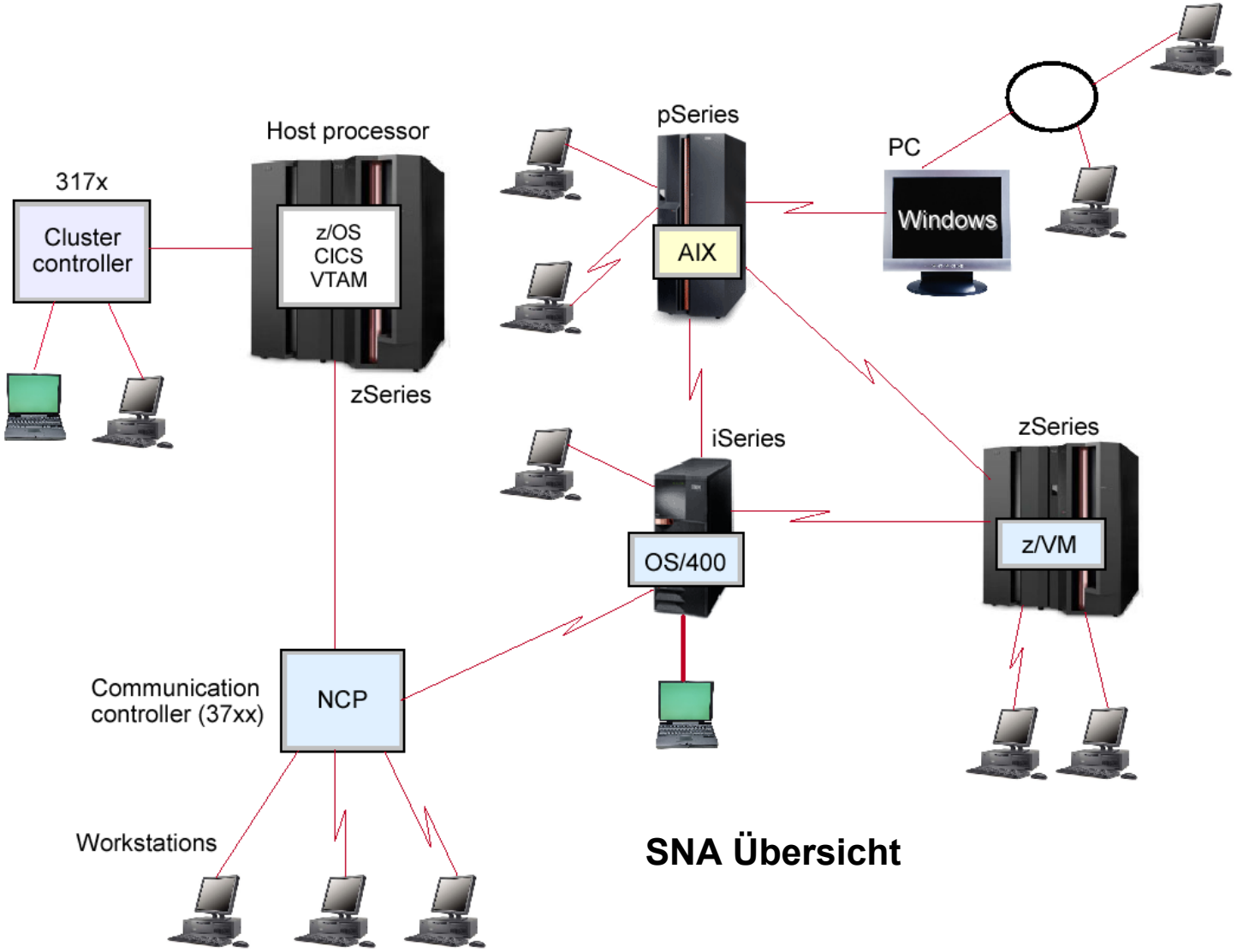
Zugriffsrechte können von spezifischen granularen Bedingungen abhängig gemacht werden; z.B. der Zugriff auf eine Datenbank kann von der Nutzung eines spezifischen Anwendungsprogramms abhängig gemacht werden, oder auf bestimmte Tageszeiten begrenzt sein.

SAF übergibt der externen Security Engine die pertinente Information. Die Security Engine kann dann auf der Basis von „Profilen“ über die Zugriffsrechte entscheiden.

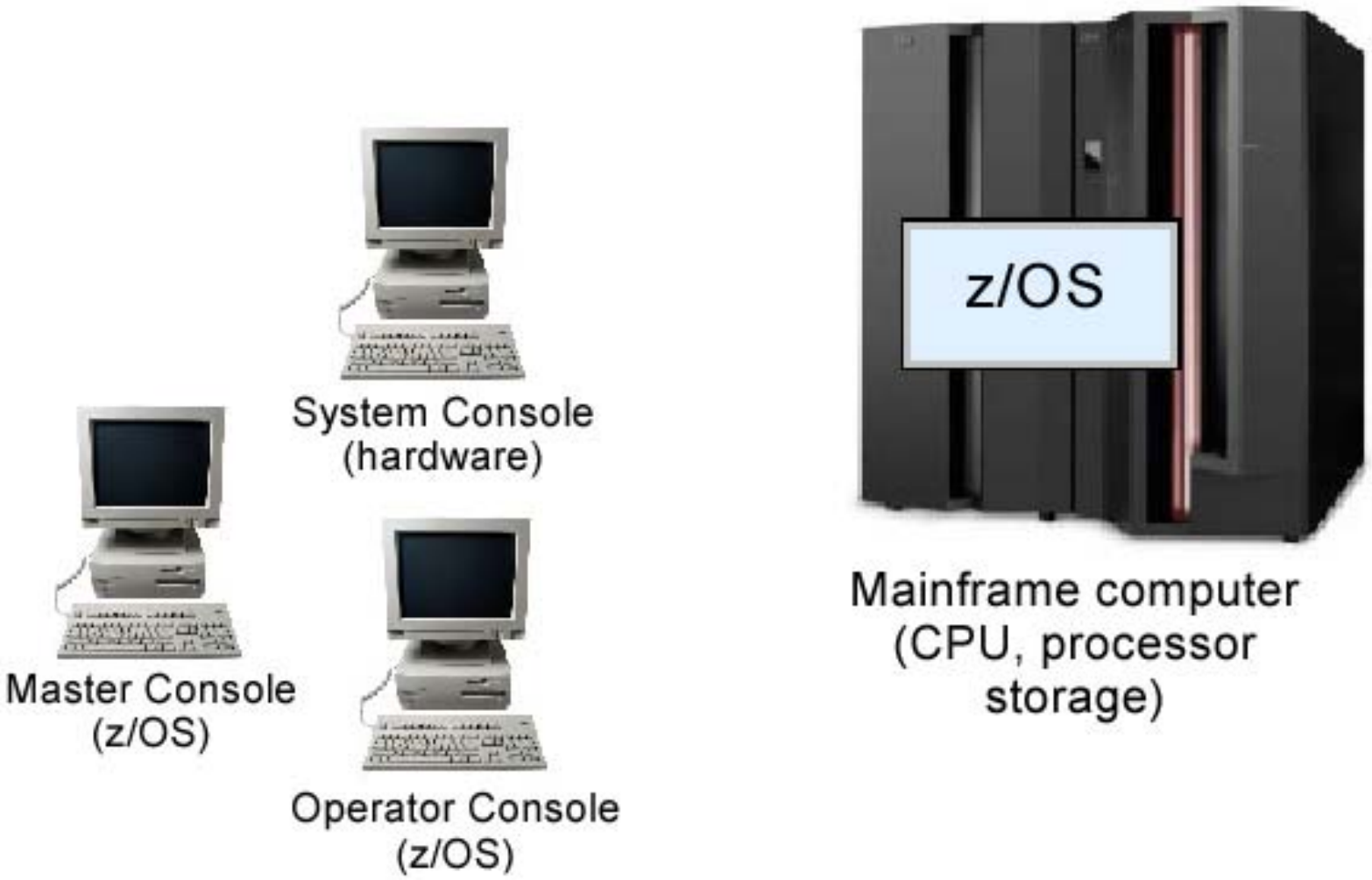
z/OS Communication Server



Der z/OS Communication Server ist ein eigenständiges Subsystem in einem eigenen virtuellen Adressenraum. Er implementiert die TCP/IP und SNA Netzwerk Architektur Stacks.



SNA Übersicht



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