

# **Internet Anwendungen unter z/OS und OS/390**

**Dr. rer. nat. Paul Herrmannn  
Prof. Dr.rer.nat. Udo Keschull  
Prof. Dr.-Ing. Wilhelm G. Spruth**

**WS 2004/2005**

**Teil 3**

**z/OS Betriebssystem**

## S/390 Betriebssysteme

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

Alle S/390 Betriebssysteme sind Server Betriebssysteme, optimiert für den Multi-User Betrieb

es 0521z ww6

wgs 09-99

## S/390 Betriebssysteme Lizenzen weltweit

OS/390, MVS	13 500
VSE	12 000
TPF	300
UTS 4 Amdahl Unix SVR4	300

### Gastbetriebssysteme

VM	10 000
Linux	

Die meisten der VM Installationen laufen auf Rechnern, auf denen VSE oder OS/390 als Haupt-Betriebssystem installiert ist.

es 0120 ww6

wgs 06-99

# **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**

**bis zu 10 000 Transaktionen / s**

es 0517z ww6

wgs 09-99

## **z/OS und OS/390**

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

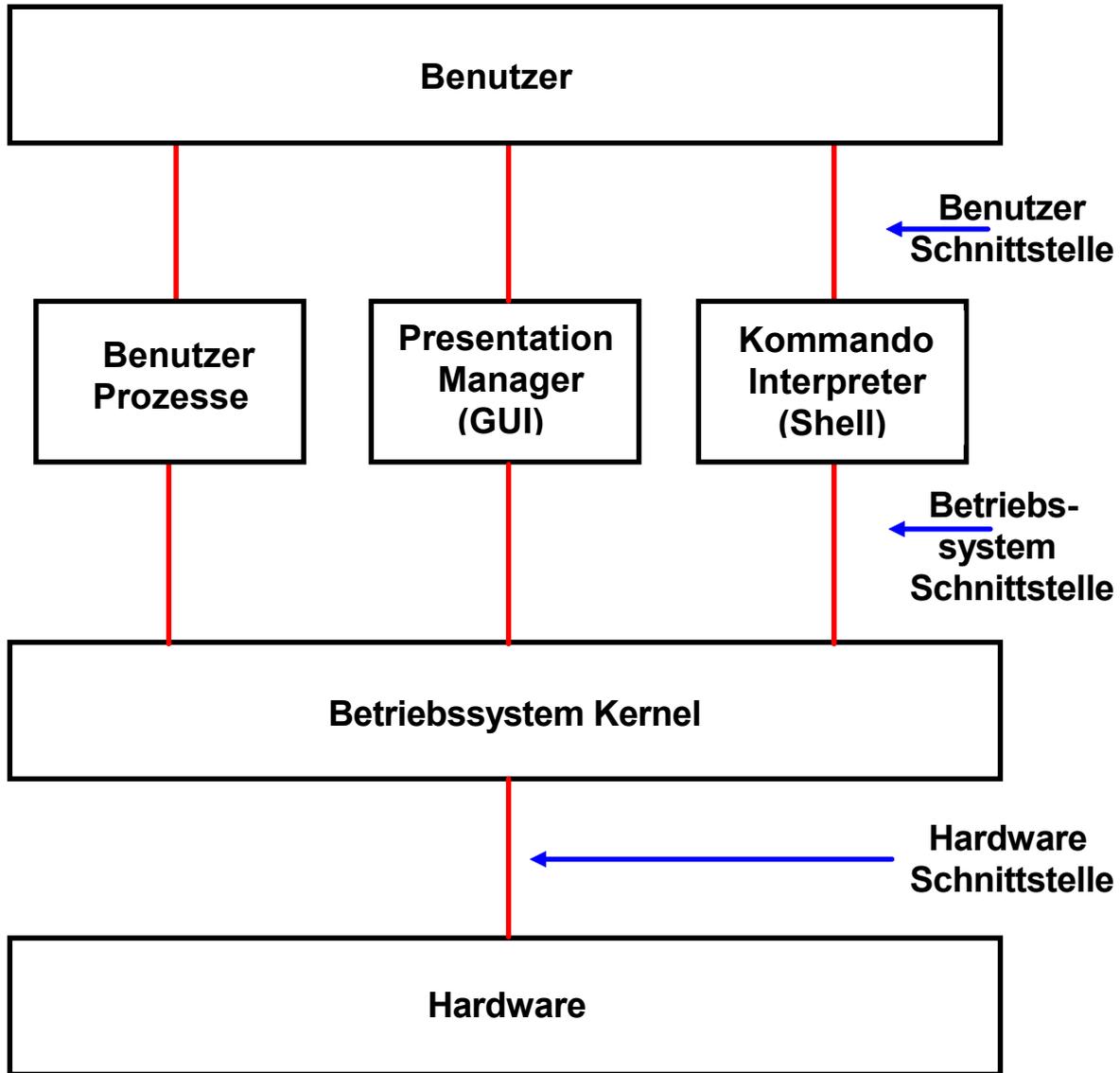
**Spätere Namenswechsel OS/360 → MFT → MVT → MVS**

**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**

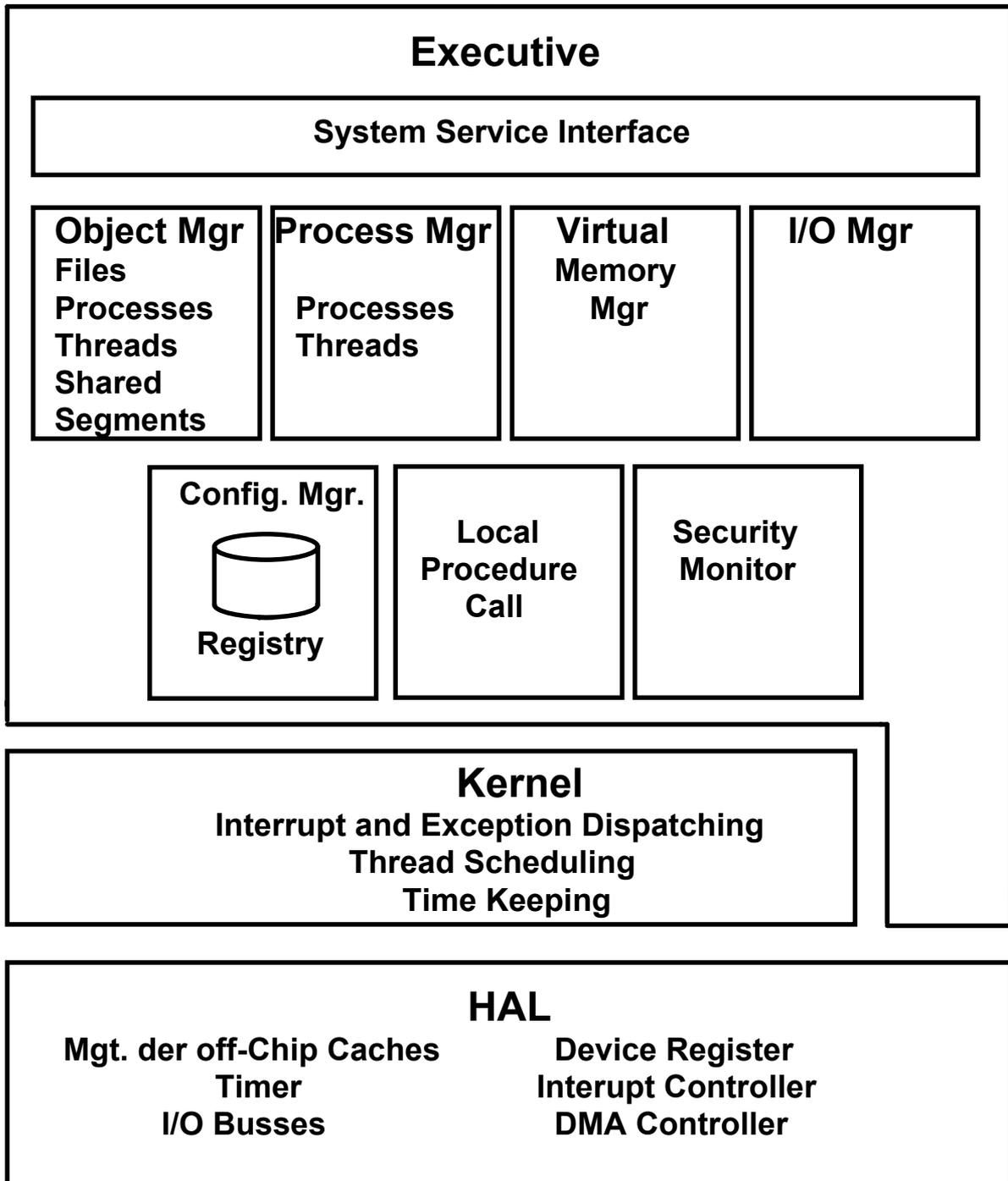
es 0555 ww6

wgs 11-00



## Schichtenmodell der Rechnerarchitektur

## Executive System Service Calls (Exceptions)



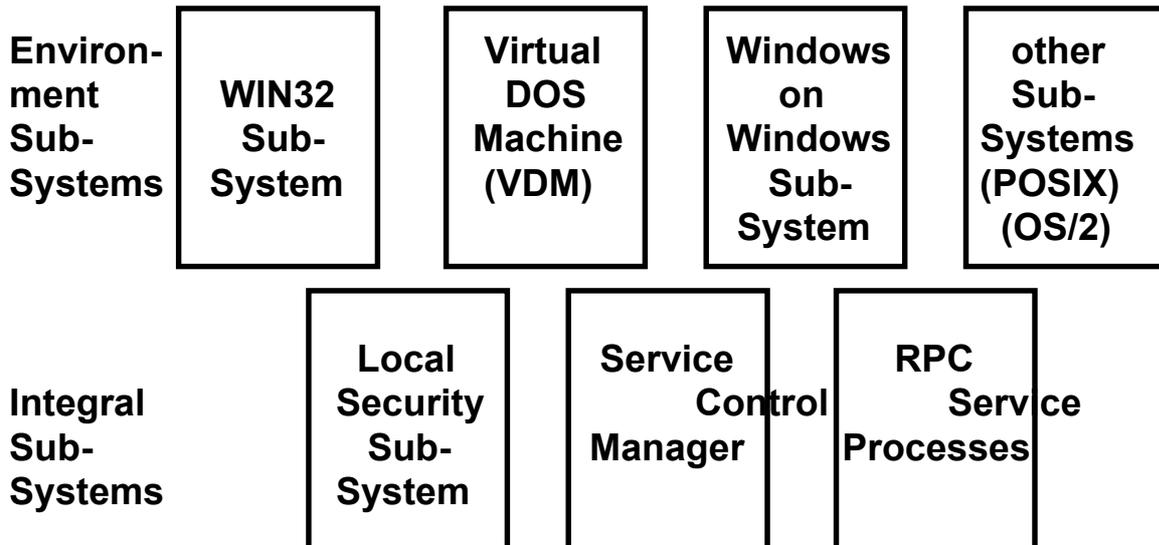
## Windows NT Überwacher Kernel Mode Funktionen

## Anwendungen

### Ring 3

---

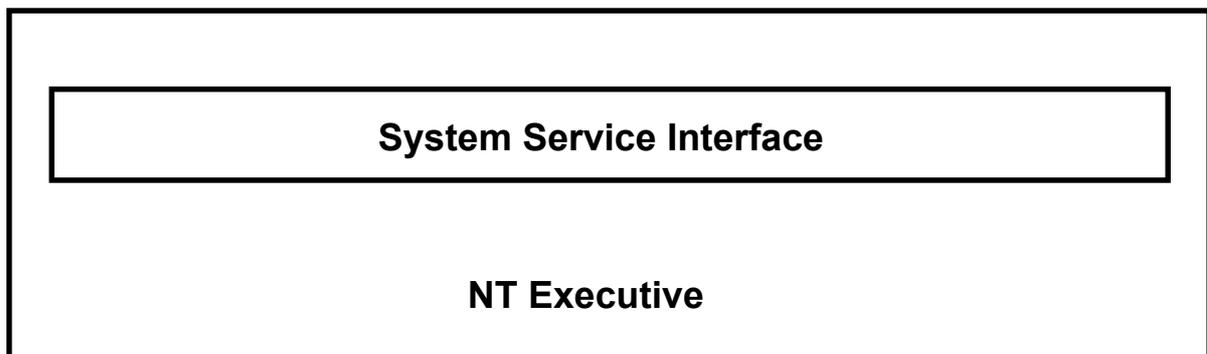
### Ring 1,2



### Ring 1,2

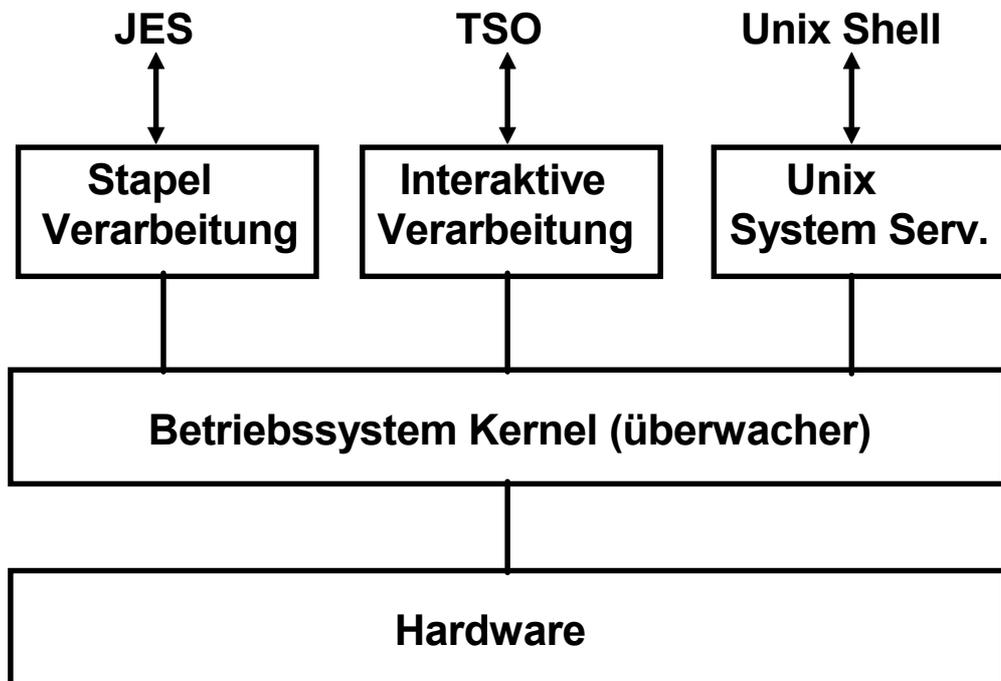
---

### Ring 0



## Windows NT Überwacher Erweiterte Systemfunktionen

Virtual DOS Machine verarbeitet 16 Bit Anwendungen  
Windows on Windows Subsystem verarbeitet 16 Bit Windows Apps.

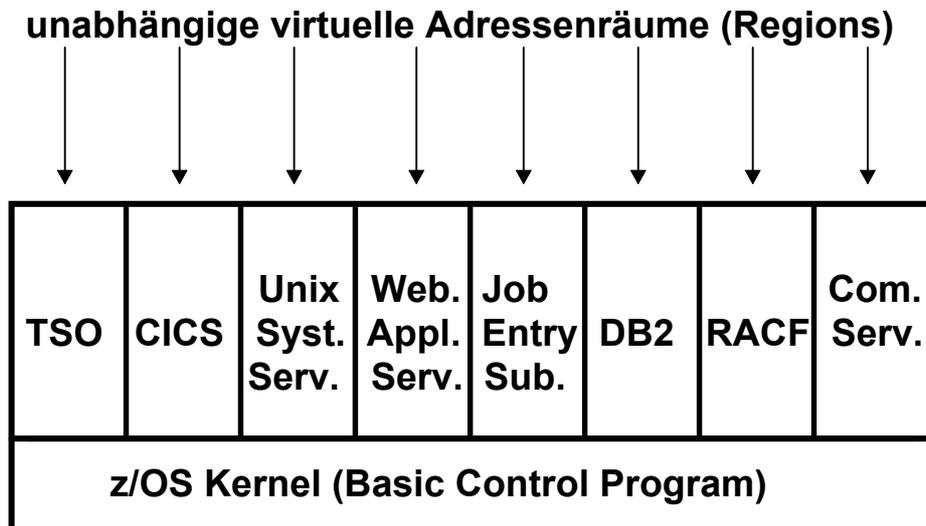


## OS/390 Grundstruktur

Die drei wichtigsten Subsysteme:

**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**

## OS/390 Grundstruktur



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. 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

## **OS/390 Prozessverwaltung**

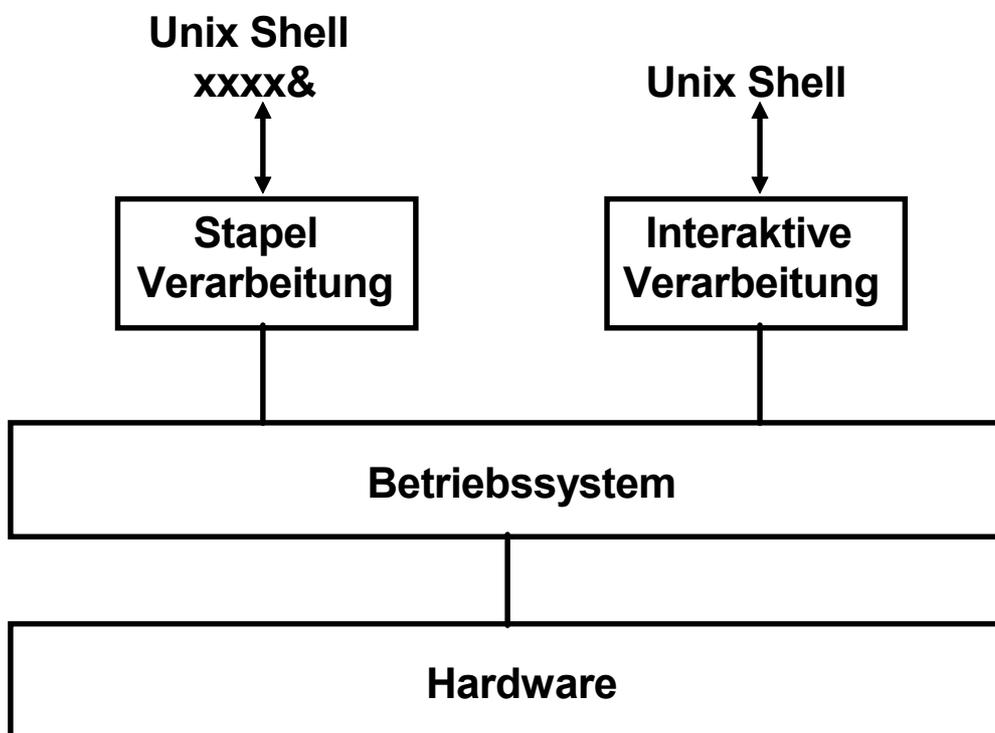
**Ein OS/390 Prozess besteht aus mehreren Arbeitseinheiten, die als „Tasks“ bezeichnet werden. Eine „Main Task“ generiert Subtasks und entspricht in etwa einem Thread in Unix oder NT. Wenn ein Programm gestartet wird, erstellt OS/390 hierfür einen Main Task Control Block (TCB). Das Programm kann weitere Subtasks erstellen mit Hilfe des ATTACH System Aufrufes.**

**Da der ATTACH Overhead relativ groß ist, implementieren zeitkritische Subsysteme wie z.B. CICS ihr eigenes Subtasking. Im Überwachermodus existieren zusätzlich SMB's und SRB's. Windows kennt „Fibers“ (light weight threads, die durch die Anwendung gescheduled werden). Die CICS Portierung auf Windows benutzt Fibers.**

**Ein Stapelverarbeitungsauftrag interagiert während seiner Ausführung nicht mit dem Benutzer.**

(Er kann während der Ausführung temporär unterbrochen und wieder aufgenommen werden, jenachdem ob Aufträge mit höherer Priorität die Ressourcen dringender benötigen).

**Bei einer interaktiven Verarbeitung monitored der Benutzer die Ausführung und wartet auf die Ergebnisse.**



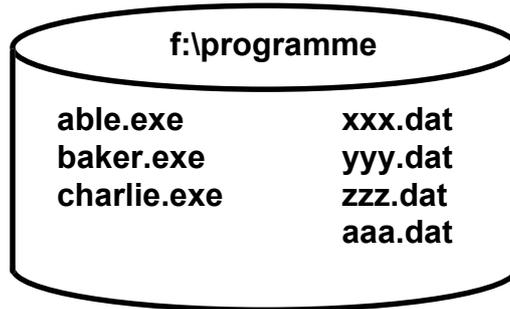
## **Unix Grundstruktur**

**Die Stapelverarbeitung ist ein Sonderfall der interaktiven Verarbeitung. In der Shellsprache werden Batch-Aufträge durch ein nachgestelltes „&“ gekennzeichnet.**

## Beispiel eines DOS 6.22 Jobs

auftrag.bat

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



able

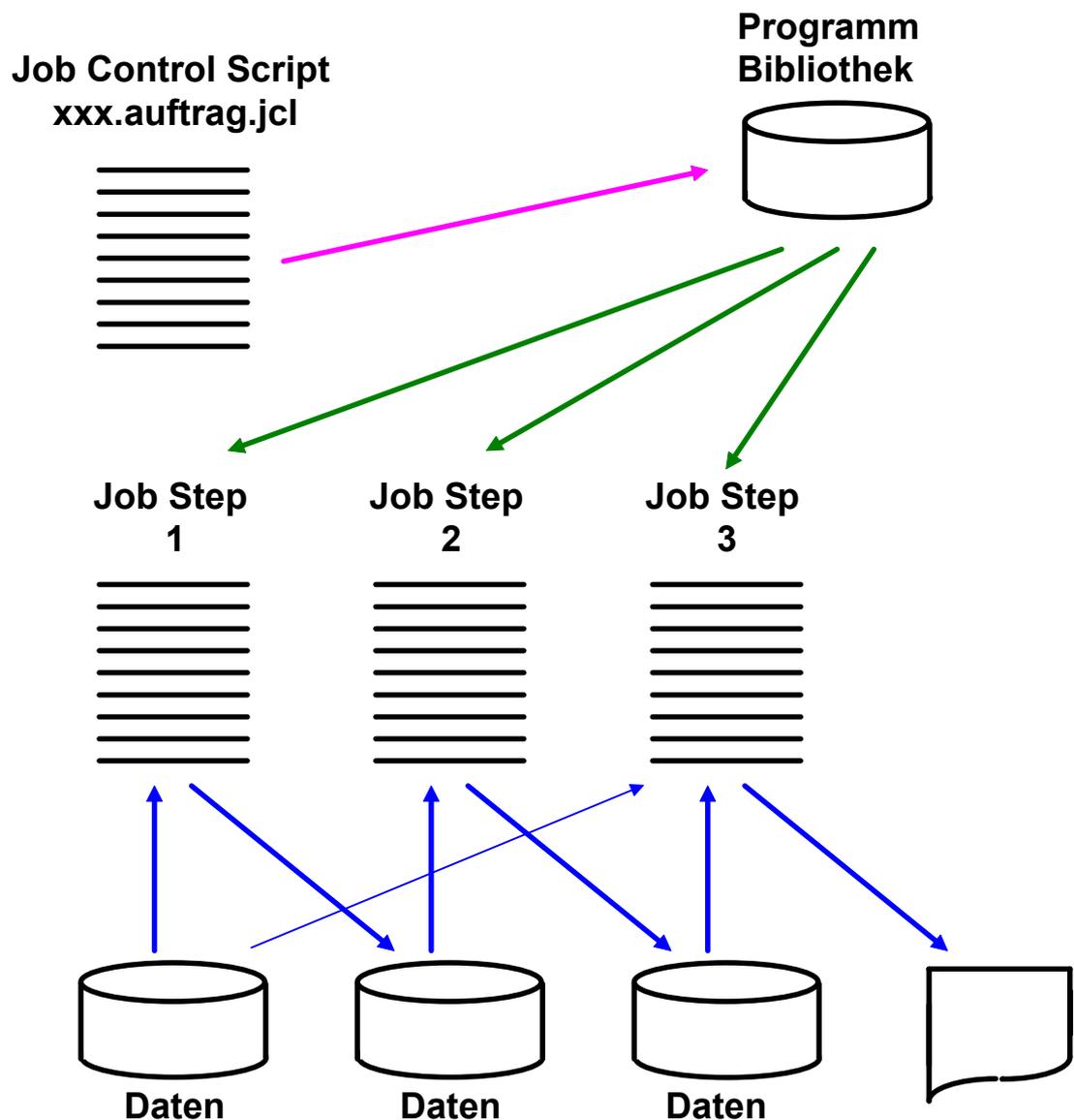
```
=====  
=====  
=====  
read xxx.dat  
=====  
write yyy.dat  
=====  
end
```

baker

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

charlie

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



## Konzept eines OS/390 Jobs

Da das JCL Programm die verwendeten Dateien angibt, ist ein „late Binding“ der verwendeten Dateien 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

Die Ablaufsteuerung für die einzelnen Job Steps innerhalb eines Jobs erfolgt durch ein Job Script, unter OS/390 als „Job Control Programm“ bezeichnet.

Ein Job Script ist in einer Script Sprache geschrieben. Unter Unix sind dies häufig Perl oder Tcl/Tk; unter Windows sind es z.B. VBScript, JavaScript oder Rexx. Unter OS/390 sind es meistens Rexx oder JCL.

Rexx ist auch außerhalb S/390 weit verbreitet und auf zahlreichen Plattformaen verfügbar; „it is just another Script Language“. Dagegen ist JCL (Job Control Language) nur unter OS/390 verfügbar, sehr ungewöhnlich, trotzdem aber sehr populär.

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.

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), diese 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

## Die wichtigsten z/OS Sprachen

- Cobol
- PL/1
- Assembler
- Fortran
- C/C++
- Java

## Skripte

- Shellskripte unter Unix
- BAT-Files unter DOS\*
- JCL und REXX unter z/OS

## REXX

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

### REXX Programmierbeispiel

```
/* 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 '!'
```

```
000100 //SPRUTHC JOB ( ), CLASS=A, MSGCLASS=H, MSGLEVEL= (1, 1) ,
000200 //          NOTIFY=&SYSUID, TIME=1440
000300 //PROCLIB JCLLIB ORDER=CBC.SCBCPRC
000400 //CCL      EXEC PROC=EDCCB,
000500 //          INFILE= ' SPRUTH.TEST.C (HELLO1) ' ,
000600 //          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 den Anfang einer Prozedur an

**DD Statement** bezeichnet die zu benutzenden Dateien

**/\* Statement** markiert das Ende von Daten, die in die JCL Statements eingeführt werden



# Job Scheduler Job Entry Subsystem

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

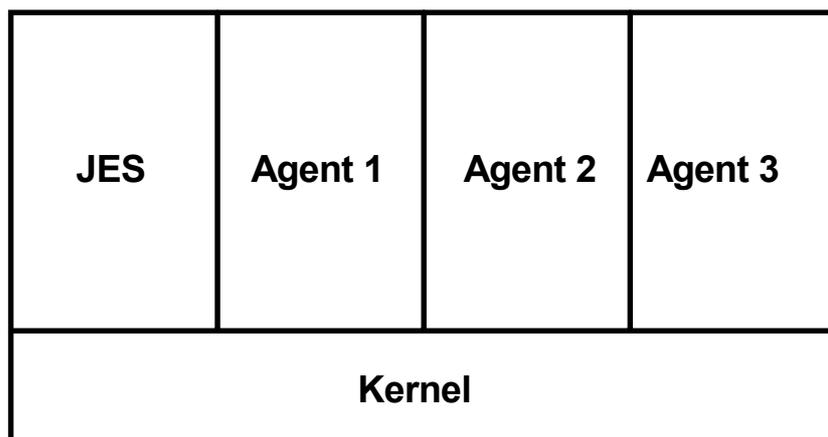
Der Master (Job entry subsystem, entweder JES2 oder JES3)

- 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, Durchsatzoptimierung sowie die Zuordnung oder Sperrung von Ressourcen ist die Aufgabe eines Job Steuerungssystems.**

**Unter Windows ist dies der „Windows Scripting Host“. Unter Unix existieren verschiedene Software Produkte, z.B. S+ batch, fbatch 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.**

# Server Zugriff

Unterschied zwischen Einzelplatzrechner  
und Client/Server Betriebssystemen.

Windows, Linux und Unix werden für beides eingesetzt. OS/390 und z/OS sind 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
OS/390 Server	3270 Client
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



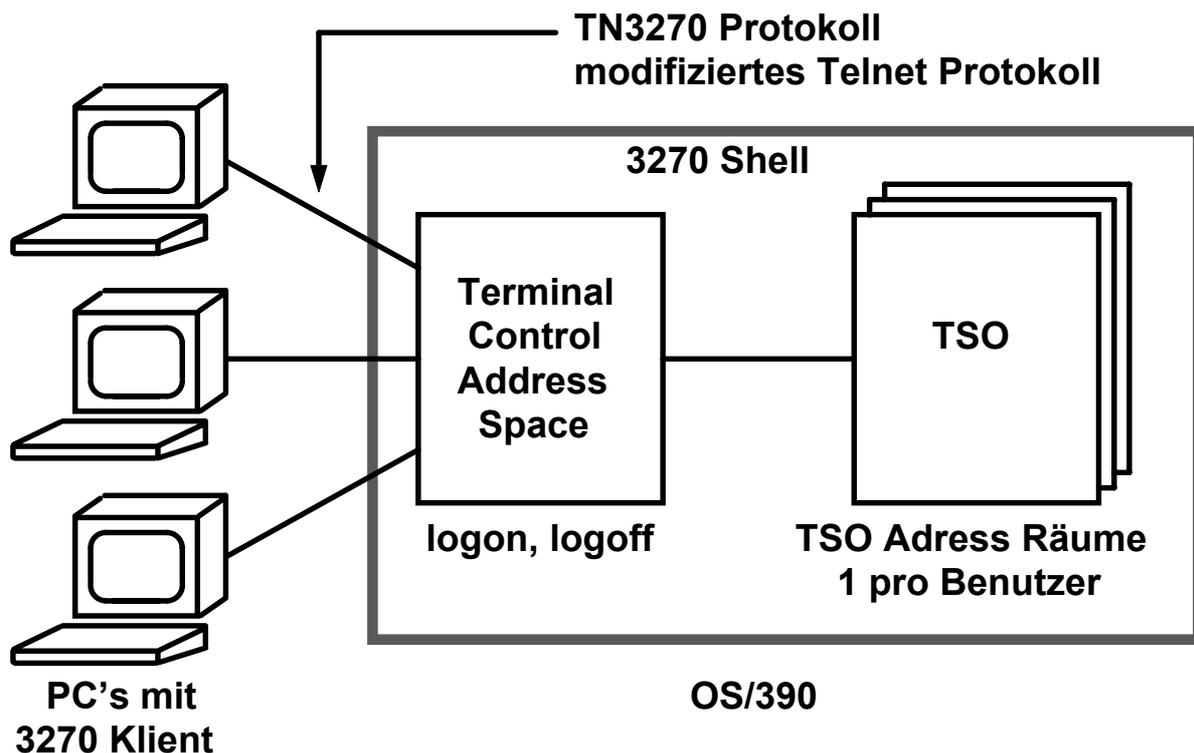
# TSO (Time Sharing Option)

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

Alternative für die Software Entwicklung: Nutzung des VM Betriebssystems in einer separaten LPAR

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.



## Time Sharing Option (TSO) Operation

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.

CUSTOMPAC MASTER APPLICATION MENU

OPTION ==>

SCROLL ==> PAGE

- IS ISMF - Interactive Storage Management Facility
- P PDF - ISPF/Program Development Facility
- ATC ATC - Application Testing Collection
- ART ARTT - Automated Regression Testing Tool
- DB2 DB2 - Perform DATABASE 2 interactive functions
- QMF QMF - QMF Query Management Facility
- C CPSM - CICSplex/SM
- M MQ - MQSeries
- IP IPCS - Interactive Problem Control Facility
- OS SUPPORT - OS/390 ISPF System Support Options
- OU USER - OS/390 ISPF User Options
- SM SMP/E - SMP/E Dialogs
- SD SDSF - System Display and Search Facility
- R RACF - Resource Access Control Facility
- DI DITTO - Data Interfile Transfer, Testing and Operations
- HC HCD - Hardware Configuration Definition
- S SORT - DF/SORT Dialogs
- BMR BMR READ - BookManager Read (Read Online Documentation)

F1=HELP      F2=SPLIT      F3=END      F4=RETURN      F5=RFIND      F6=RCHANGE  
 F7=UP      F8=DOWN      F9=SWAP      F10=LEFT      F11=RIGHT      F12=RETRIEVE

Menu Utilities Compilers Options Status Help

#####  
 ISPF Primary Option Menu

0	Settings	Terminal and user parameters	User ID . . : SPRUTH
1	View	Display source data or listings	Time. . . : 18:20
2	Edit	Create or change source data	Terminal. : 3278
3	Utilities	Perform utility functions	Screen. . : 1
4	Foreground	Interactive language processing	Language. : ENGLISH
5	Batch	Submit job for language processing	Appl ID . : PDF
6	Command	Enter TSO or Workstation commands	TSO logon : IKJACCNT
7	Dialog Test	Perform dialog testing	TSO prefix: SPRUTH
8	LM Facility	Library administrator functions	System ID : DAVI
9	IBM Products	IBM program development products	MVS acct. : ACCT#
			Release . . : ISPF 4.5

#####N r  
 e Licensed Materials - Property of IBM e  
 e 5647-A01 (C) Copyright IBM Corp. 1980, 1997. e  
 e All rights reserved. e  
 e US Government Users Restricted Rights - e s  
 e Use, duplication or disclosure restricted e  
 e by GSA ADP Schedule Contract with IBM Corp. e  
 D#####M

Option ==>

F1=Help      F3=Exit      F10=Actions      F12=Cancel

Detaillierte Screen by Screen Tutorials sind zu finden unter

<http://jedi.informatik.uni-leipzig.de>

# **Dateisystem**

**Ein Dateisystem verbirgt die Eigenschaften der physikalischen Datenträger (Plattenspeicher, CD, evtl. Bandlaufwerke) weitestgehend vor dem Anwendungsprogrammierer.**

## **Unix, NT**

**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, OS/390**

**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.**

# Konzepte: Datenorganisation

---

## ► UNIX/NT

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



- Fixed Block Architecture

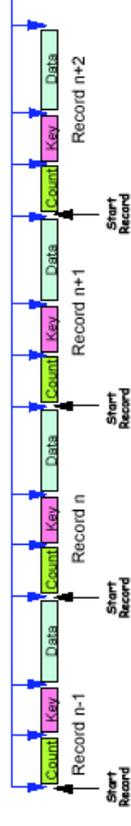
## ▪ Zugriffsmethode: "raw"

- Implementiert bei Datenbankanwendungen ausserhalb des Dateisystems

## ► S/390

- Record orientierter Zugriff
- Zugriffsmethode: GET recordid, buffer
  - Teil des Betriebssystem
  - sequentiell, 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"

# Data sets and files in z/OS and OS/390

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

**z/OS and OS/390 verwenden zwei vollständig unterschiedliche Verfahren um Daten auf Plattenspeichern anzuordnen;**

- **Traditional MVS Data sets and**
- **Unix files.**

# **DFSMS Klassen**

**Beim Neuanlegen einer Datei müssen angegeben werden:**

## **Data Class**

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

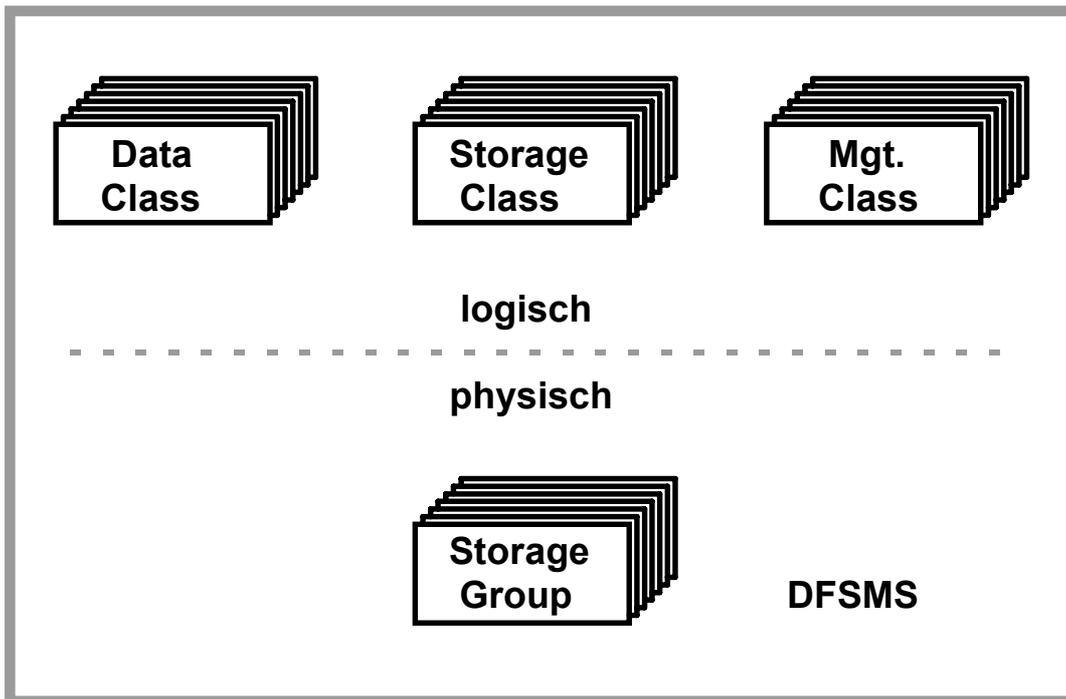
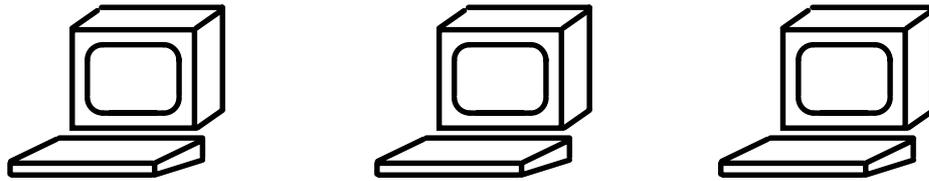
## **Storage Class**

**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)**



Physische Plattenspeicher (Volumes)

## Data Facility Storage Management System

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

# **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**

# **DB2 relationale Datenbank**

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

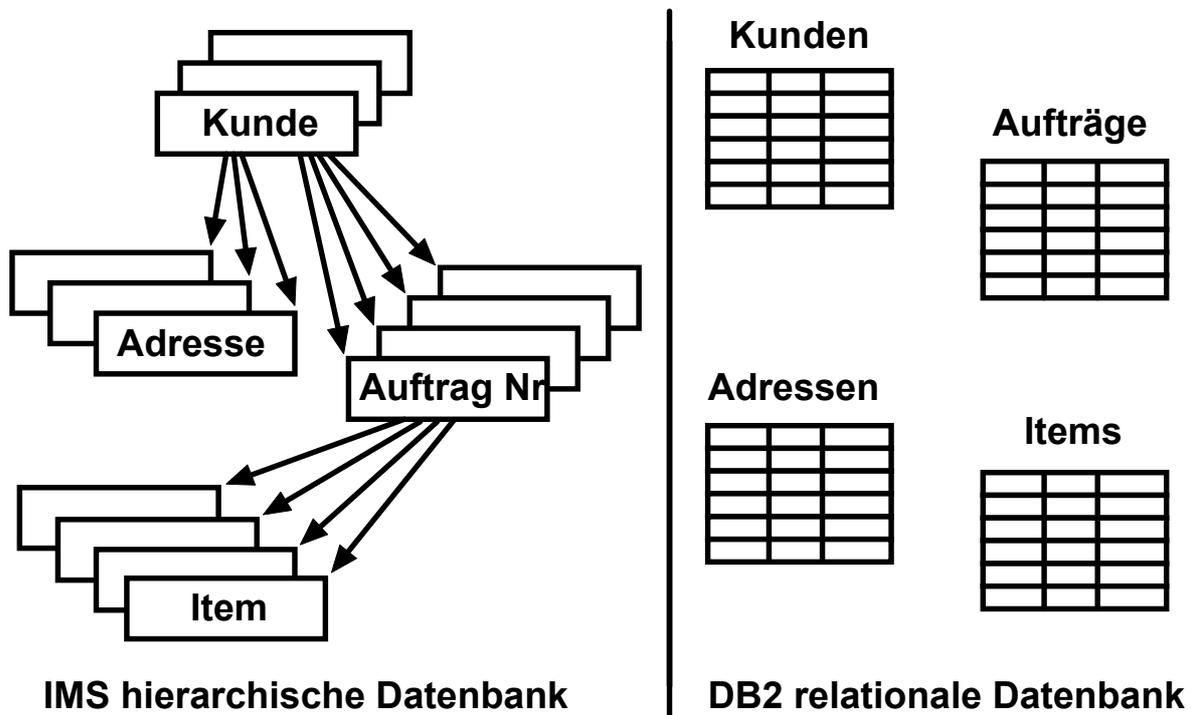
**Identische Implementierung für alle UNIX-, Linux-, OS/2-, und Windows-Betriebssysteme.**

**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.**

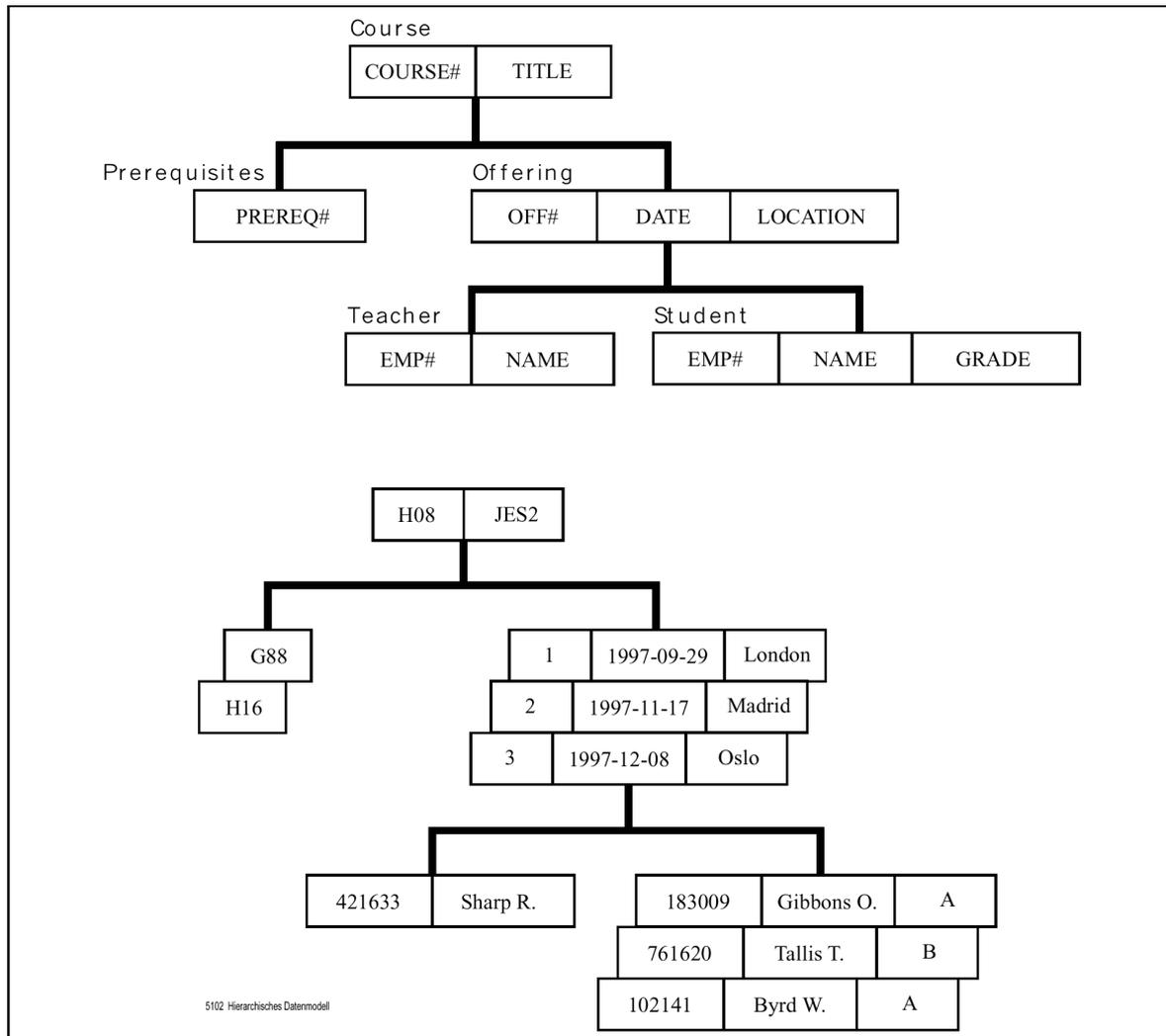
**Der Zugriff kann mit Hilfe von SQL-Statements erfolgen, die in einem Anwendungsprogramm eingebettet sind. Alternativ existieren eine Reihe spezifischer SQL-Klienten-Anwendungen.**



## 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.**



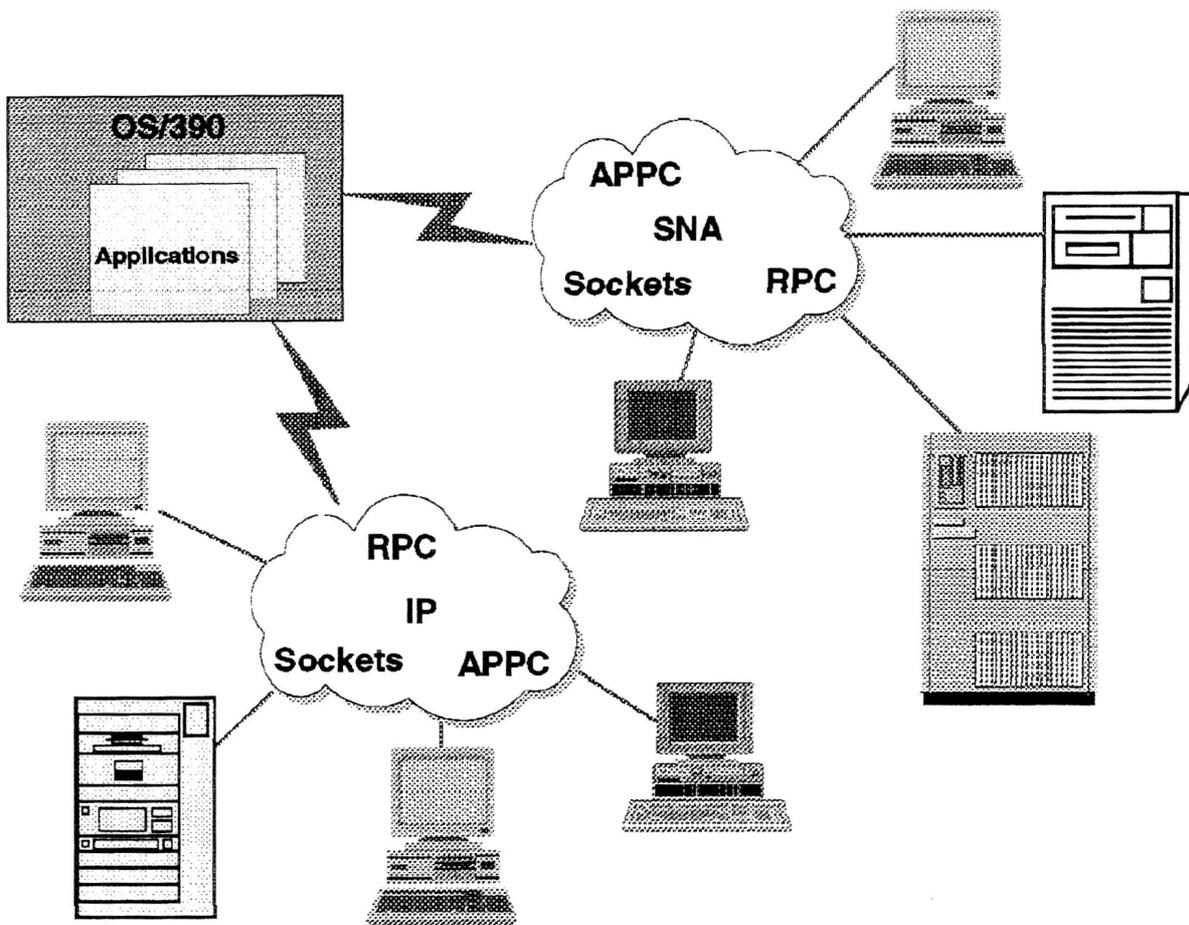
## IMS Datenbanksystem

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.

Das obige 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.



## OS/390 Communication Server

**Der OS/390 Communication Server ist ein eigenständiges Subsystem in einem eigenen virtuellen Adressenraum. Er implementiert die TCP/IP und SNA Netzwerk Architektur Stacks.**

## **z/OS „SecureWay“ Security Server**

- **RACF**
- **LDAP Server - Secure Directory Server**
- **Kerberos Network Authentication Service, einschliesslich Kryptographie und Firewall Unterstützung**

### **RACF (Resource Access Control Facility)**

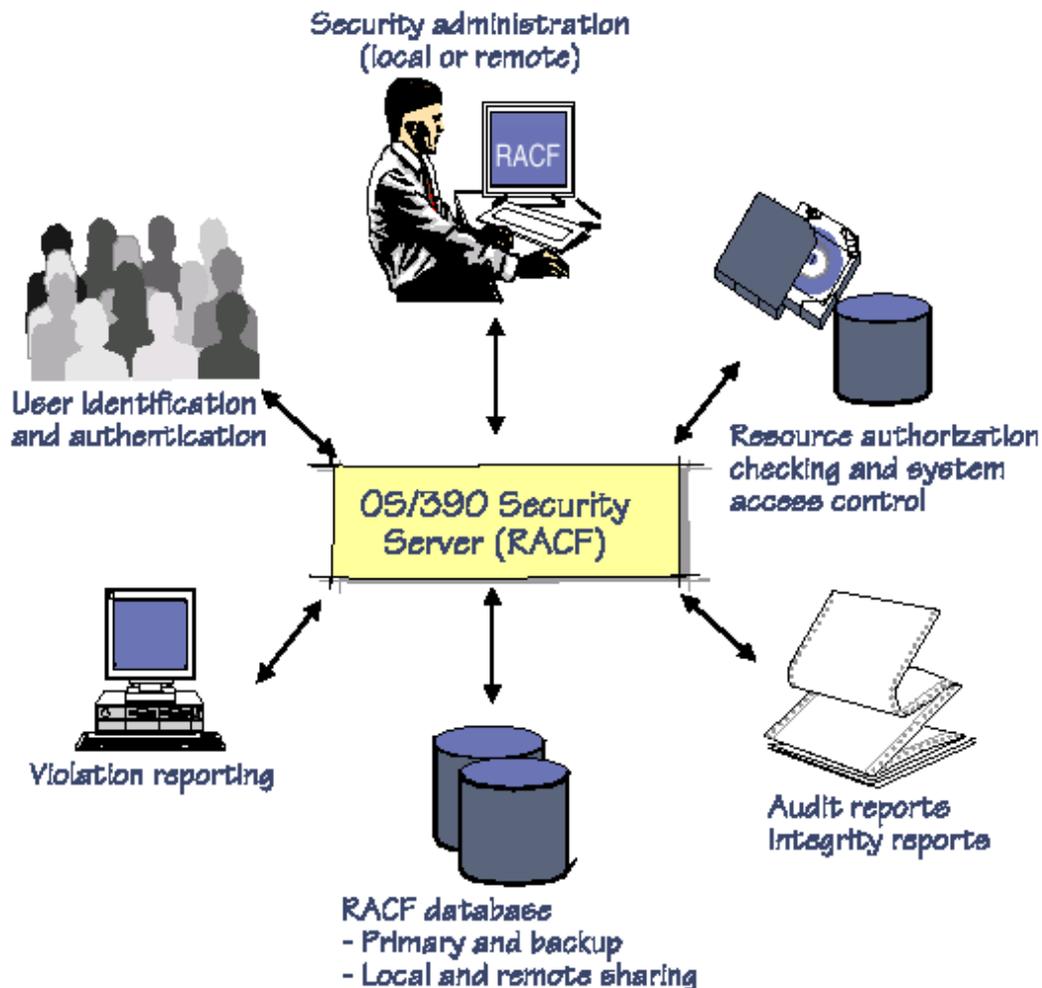
**RACF entscheidet,**

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

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

**Kryptographischer Koprozessor**

# z/OS Security Server



RACF provides the ability to:

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

Literatur : IBM Form No. GC28-1912-06

# RACF

RACF benutzt das Konzept von zu schützenden „Ressourcen“.

Ressourcen 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 Berechtigungen vorhanden sind. Die Profile werden in einer Systemdatenbank, der RACF Profildatenbank, abgespeichert.

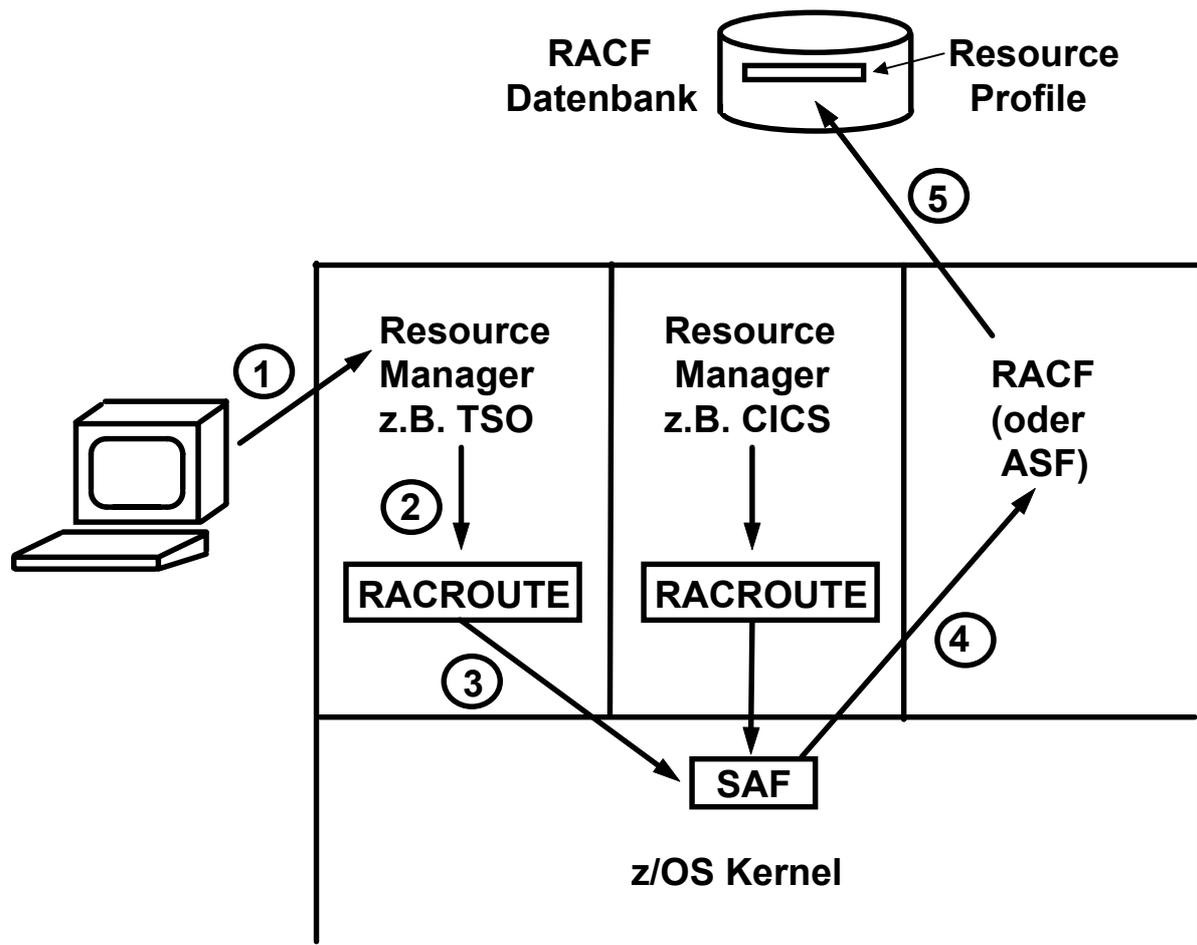
Beispiel: Ein interaktiver Benutzer logged sich ein. Der Login Prozess überprüft, ob die Login Berechtigung besteht. Er überprüft weiterhin, ob das Terminal, von dem der Benutzer sich einwählt, eine Zugangsberechtigung hat. Hierzu ruft der Login Prozess RACF auf, welches die entsprechenden Profile in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



## RACF Arbeitsweise

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.



## Supervisor Calls

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

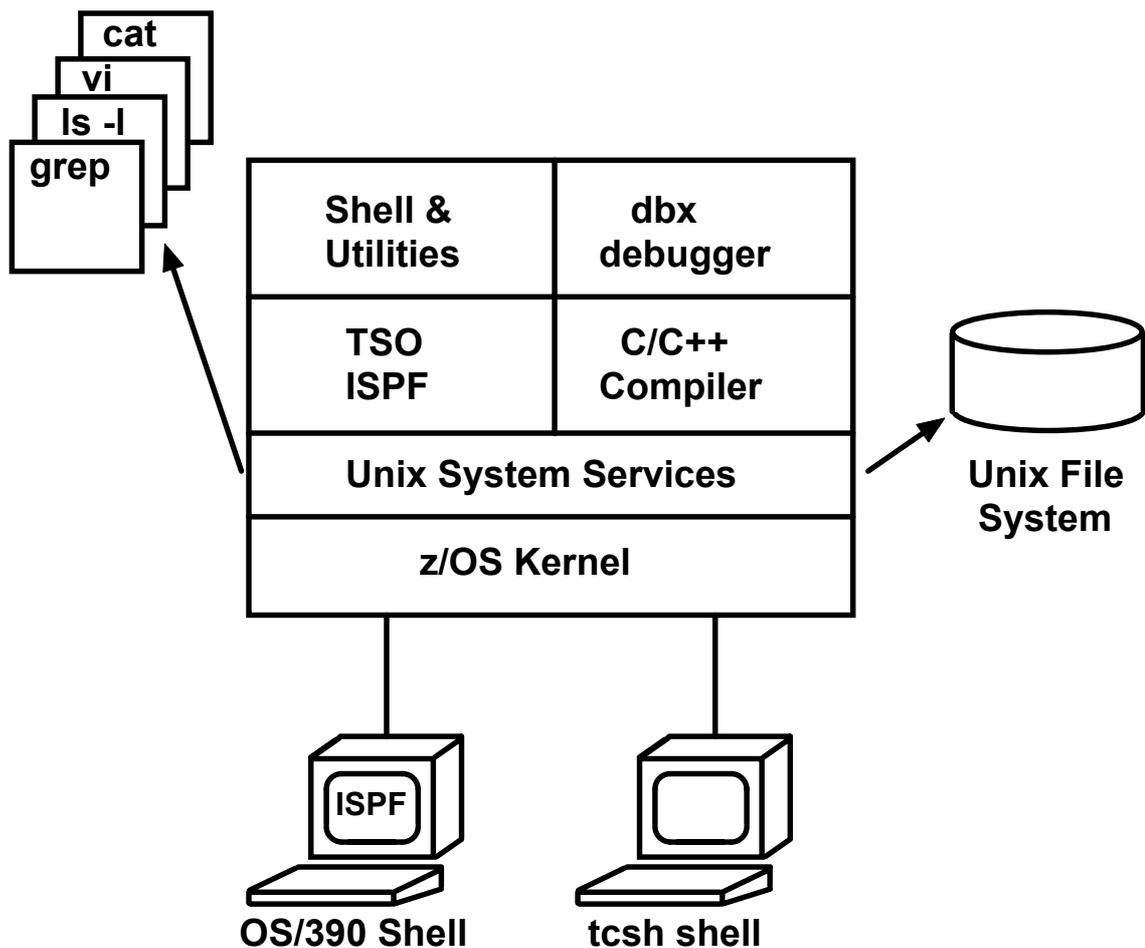
**Supervisor Calls im weiteren Sinne sind Library Routinen, die eine Dienstleistung des z/OS Kernels in anspruch nehmen.**

**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.**

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

**Beispiele sind:**

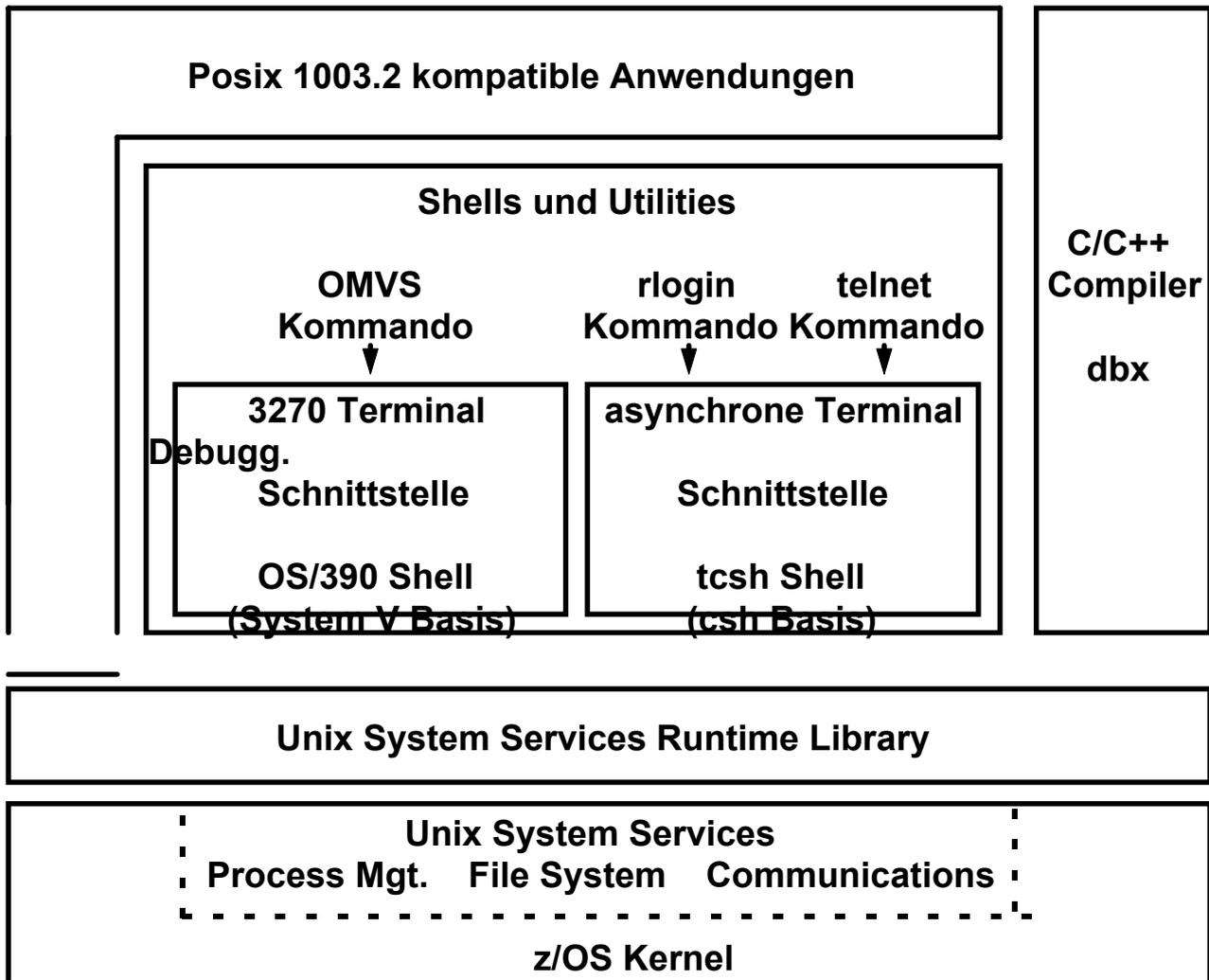
<b>GETMAIN</b>	<b>SVC 10</b>	<b>Anforderung von Virtual Storage</b>
<b>OPEN</b>	<b>SVC 19</b>	<b>Öffnen eines Data Sets</b>
<b>EXCP</b>	<b>SVC 0</b>	<b>Lesen oder Schreiben von Daten</b>
<b>WAIT</b>	<b>SVC 19</b>	<b>Warten auf ein Ereignis, z.B. Abschluß einer Lese Operation</b>



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

**Die OS/390 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)

z/OS Unix System Services Kernel Funktion läuft in einem eigenen virtuellen Adressenraum, der als Teil des IPL 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 tcsch Shell wird normalerweise über rlogin aufgerufen.

Es ist nicht unüblich, auf dem gleichen S/390 Rechner Unix System Services und zLinux parallel zu betreiben.

es 0565 ww6

wgs 07-01

(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 >

==>

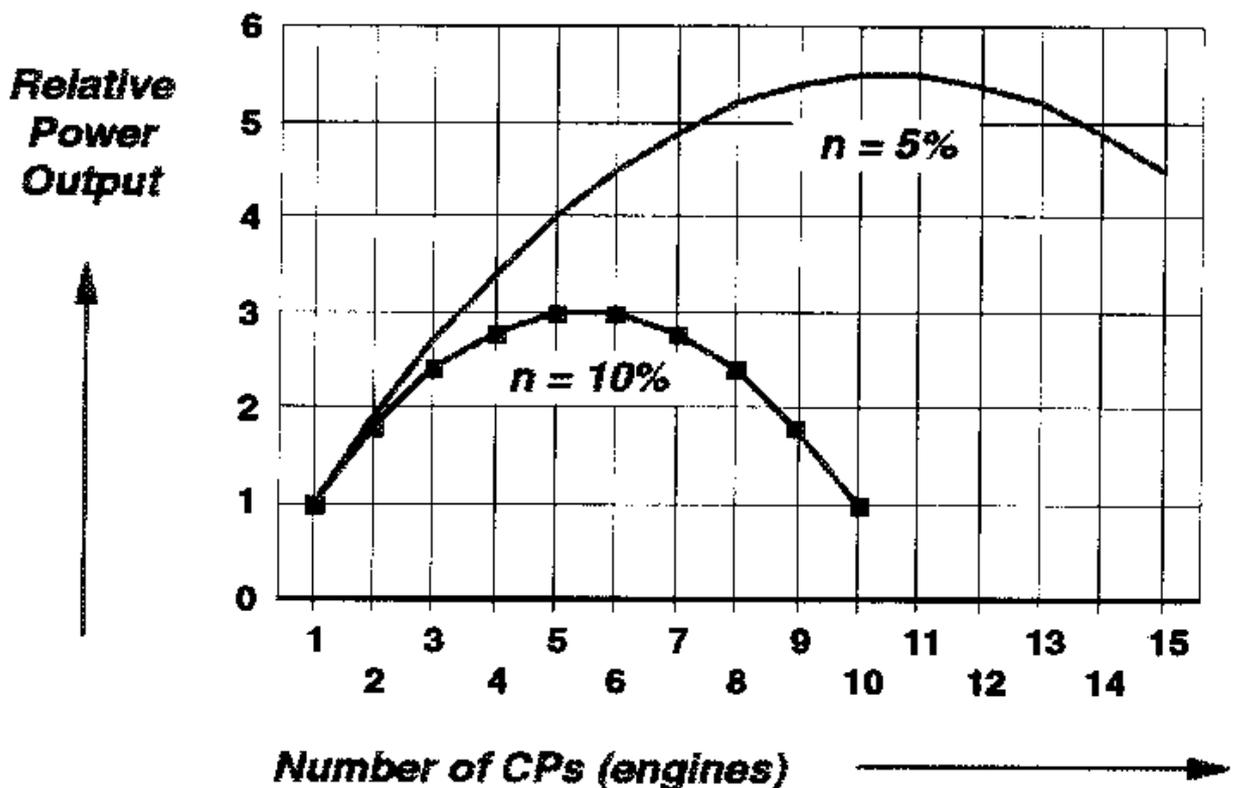
ESC=¢	1=Help	2=SubCmd	3=HlpRetrn	4=Top	5=Bottom	6=TSO	INPUT
	7=BackScr	8=Scroll	9=NextSess	10=Refresh	11=FwdRetr	12=Retrieve	

## z/OS Unix System Services Shell

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

**Die OS/390 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.**



## Leistungsverhalten eines Symmetrischen Multiprocessors (SMP)

Angenommen, ein Zweifach Prozessor leistet das Zweifache minus  $n\%$  eines Einfach Prozessors. Für  $n = 10\%$  ist es kaum sinnvoll, mehr als 4 Prozessoren einzusetzen. Für  $n = 5\%$  sind es 8 Prozessoren,

Bei einem z900 Rechner ist  $n = 2\%$ . Es ist sinnvoll, einen SMP mit 16 Prozessoren einzusetzen.

Die Gründe für den Leistungsabfall sind Zugriffskonflikte bei der Hardware und Zugriffskonflikte auf Komponenten des Überwachers. Die Letzteren überwiegen.

