

Internet Anwendungen unter OS/390

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

WS 2001/2002

Teil 3

OS/390 Betriebssystem

S/390 Betriebssysteme

VSE	IBM	mittelgroße Installationen
VM	IBM	Software Entwicklung, Unix Alternative
OS/390 (MVS)	IBM	große Installationen
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 (außer Linux) sind Server Betriebssysteme, optimiert für den Multi-User Betrieb

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.

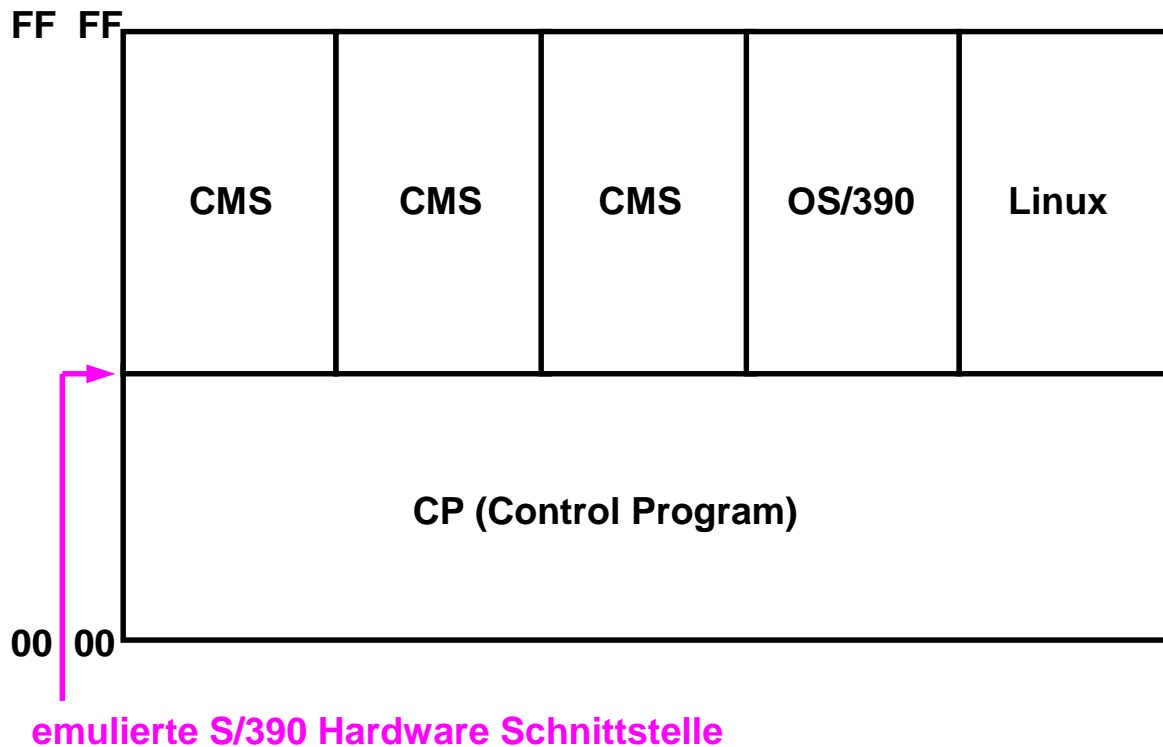
TPF

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 5000 Transaktionen / s



VM/ESA Betriebssystem

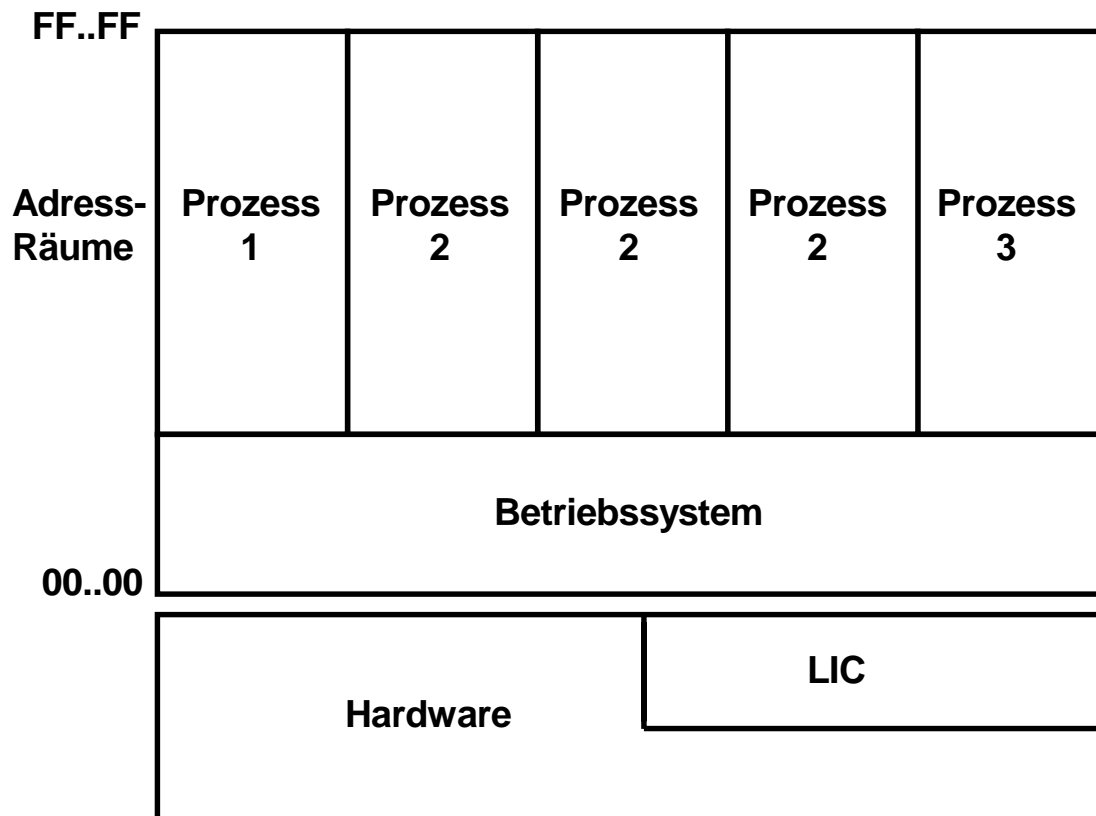
CP läuft im Überwacherstatus.

CMS und alle anderen Gast Betriebssysteme (einschließlich ihrer Kernel Funktionen) laufen im Problemstatus. Privilegierte Maschinenbefehle (z.B. E/A) werden von CP abgefangen und interpretativ abgearbeitet.

Volle S/390 Kompatibilität für alle Gast Betriebssysteme. Geringer Performance Verlust (< 5 %).

CMS (Conversational Monitor Program) ist ein besonders für die Software Entwicklung ausgelegtes Einzelplatz Betriebssystem. Für 1000 gleichzeitige CMS Benutzer werden 1000 CMS Instanzen angelegt. Ähnliches ist mit Linux/390 möglich.

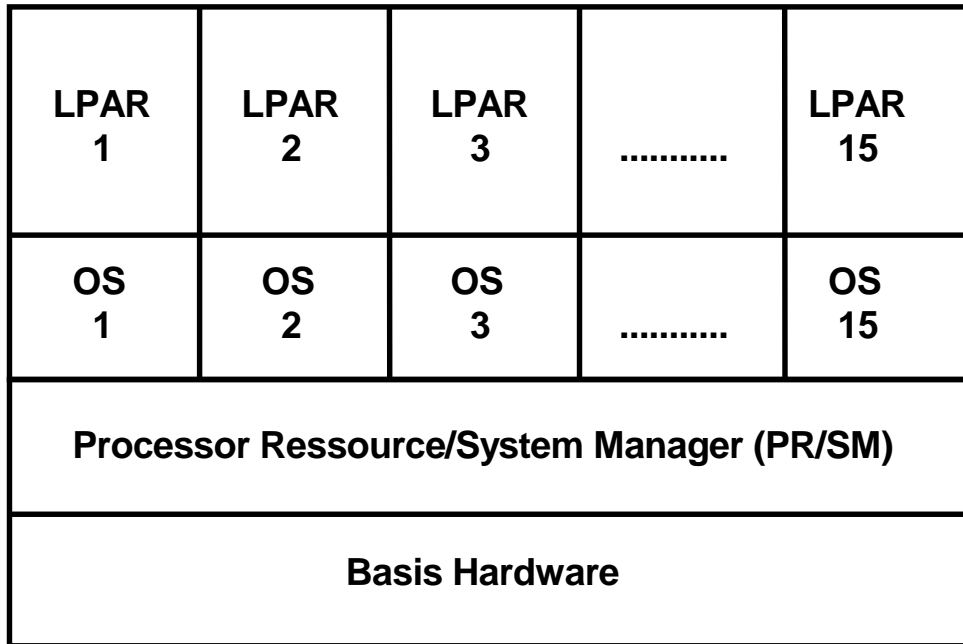
Plattenspeicherplatz wird allen Gastbetriebssystemen in der Form virtueller „Minidisks“ statisch zugeordnet. Hauptspeicherplatz wird dynamisch verwaltet.



S/390 Struktur

LIC (Licensed Internal Code) ist Maschinencode, der von der CPU ausgeführt wird, aber in einem separaten Speicher, außerhalb des architekturmäßig zugänglichen realen Hauptspeichers untergebracht und ausgeführt wird.

Häufig bestehen Teile aus regulären S/390 Maschinenbefehlen



LPAR Logical Partition
OS Operating System

IBM PR/SM und LPAR

PR/SM ist eine Hardware (Microcode) Einrichtung, welche die Partitionierung eines physikalischen Rechners in mehrere logische Rechner (LPAR. Logical Partition) ermöglicht. Jeder logische Rechner hat sein eigenes Betriebssystem, seinen eigenen unabhängigen realen Hauptspeicherbereich und seine eigenen Kanäle und I/O Geräte. Gemeinsame Nutzung von Krypto Coprozessoren und I/O Geräten durch mehrere LPAR´s ist möglich (EMIF).

PR/SM schedules die einzelnen Betriebssysteme auf die einzelnen CPU´s eines SMP.

Mit PR/SM vergleichbare Möglichkeiten sind unter dem VM/390 Betriebssystem vorhanden.

S/390 Rechner anderer Hersteller verfügen über ähnliche Einrichtungen: Hitachi MLPF, Amdahl Multiple Domain Facility.

Für die Intel Architektur bietet die Firma vmware ein Softwareprodukt mit (nicht ganz) vergleichbarer Funktionalität.

PR/SM and LPAR

LPARs have been certified by the US government to have equivalent security characteristics as physically separate machines.

z900 features the "Virtual Image Facility". This is a stripped down VM system hypervisor layer, easing system administration, and allowing > 16 LPARs, e.g. for Linux. Virtual memory allocation to guest machines, operates native or under LPAR. It provides complete workload isolation and eliminates network delays.

es 0216 ww6

wgs 06-00

OS/390

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

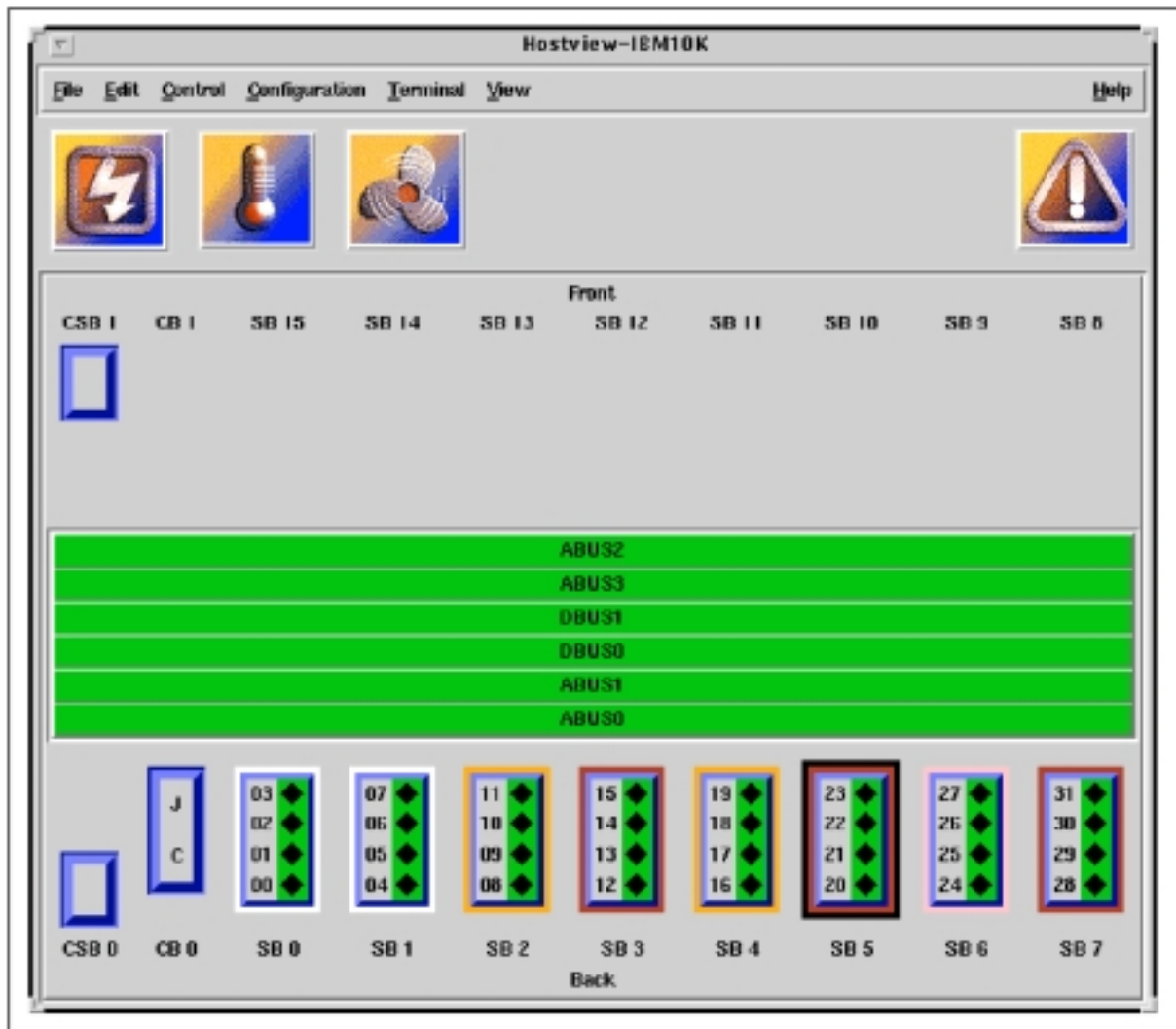
Spätere Namenswechsel OS/360 → MFT → MVT → MVS

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

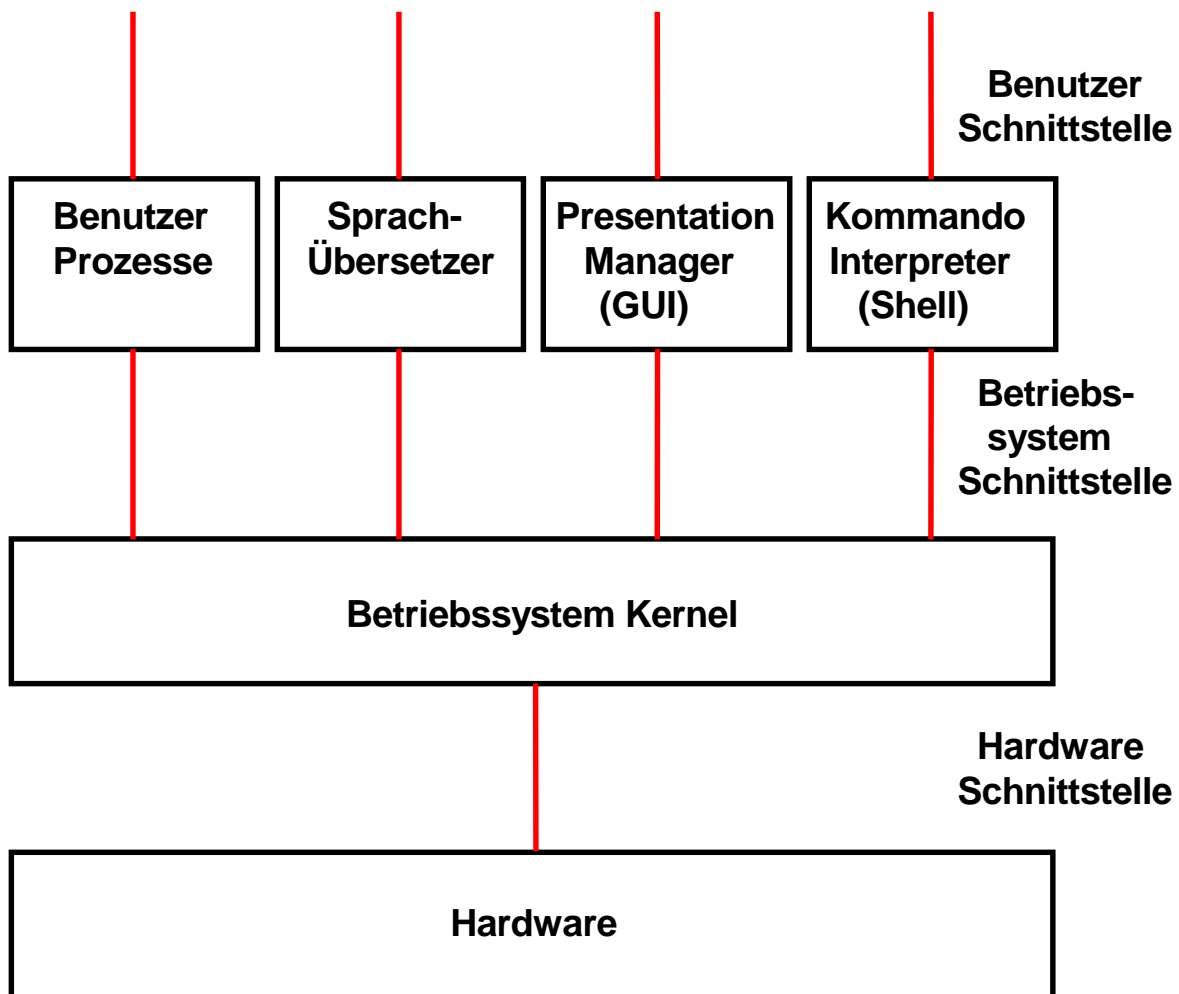
wgs 11-00



Sun E 10 000 Administrator Konsole

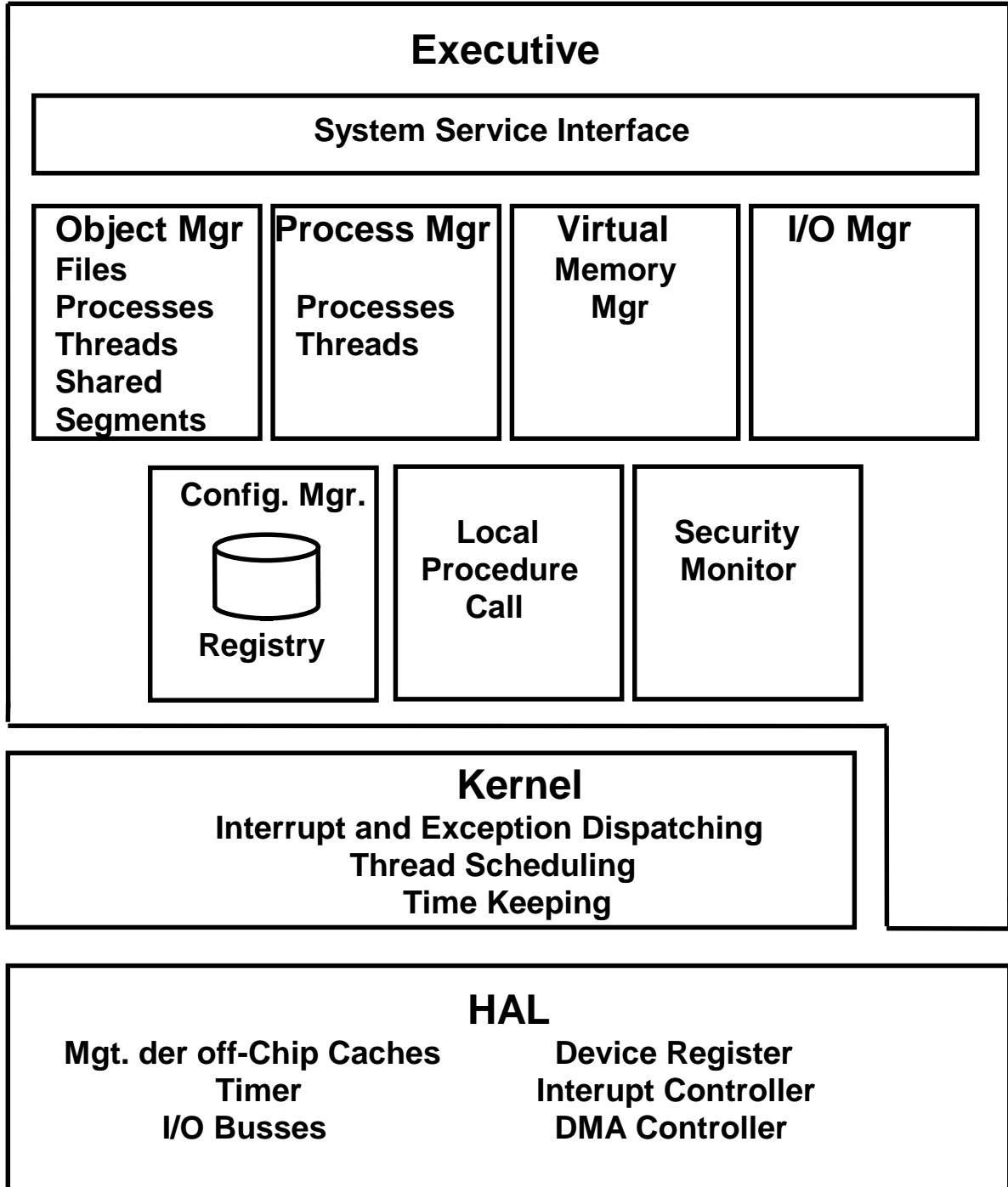
Dargestellt sind 8 Prozessor Boards SB0 .. SB7.

Cluster: SB0, SB1
 SP2, SB4
 SP3, SB7



Schichtenmodell der Rechnerarchitektur

Executive System Service Calls (Exceptions)

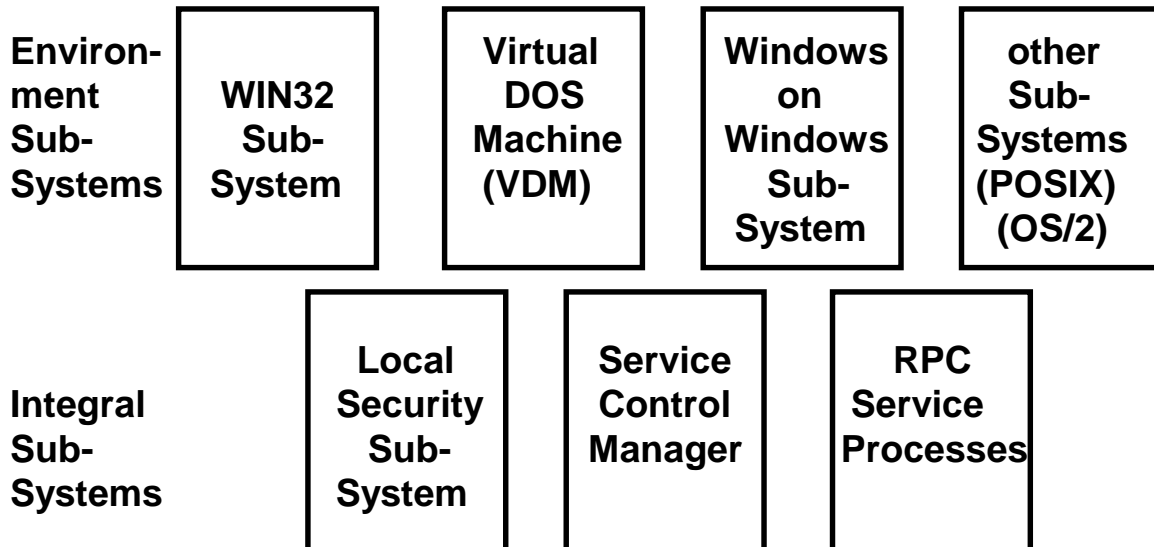


Windows NT Überwacher Kernel Mode Funktionen

Anwendungen

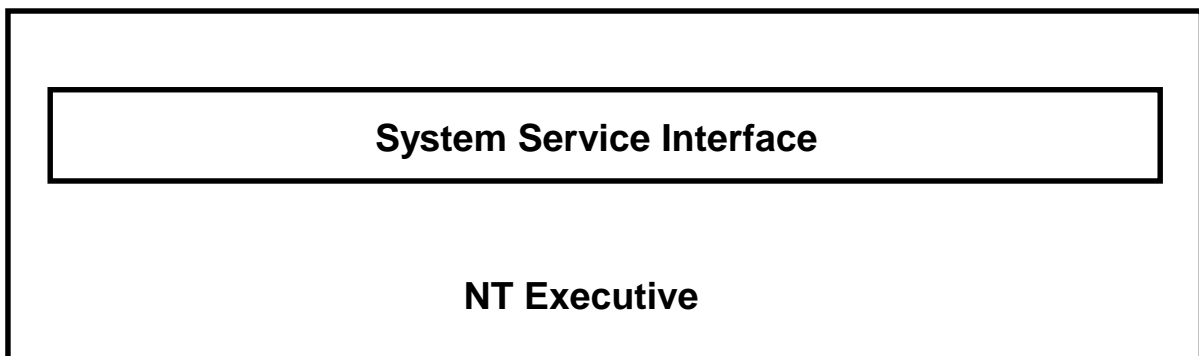
Ring 3

Ring 1,2



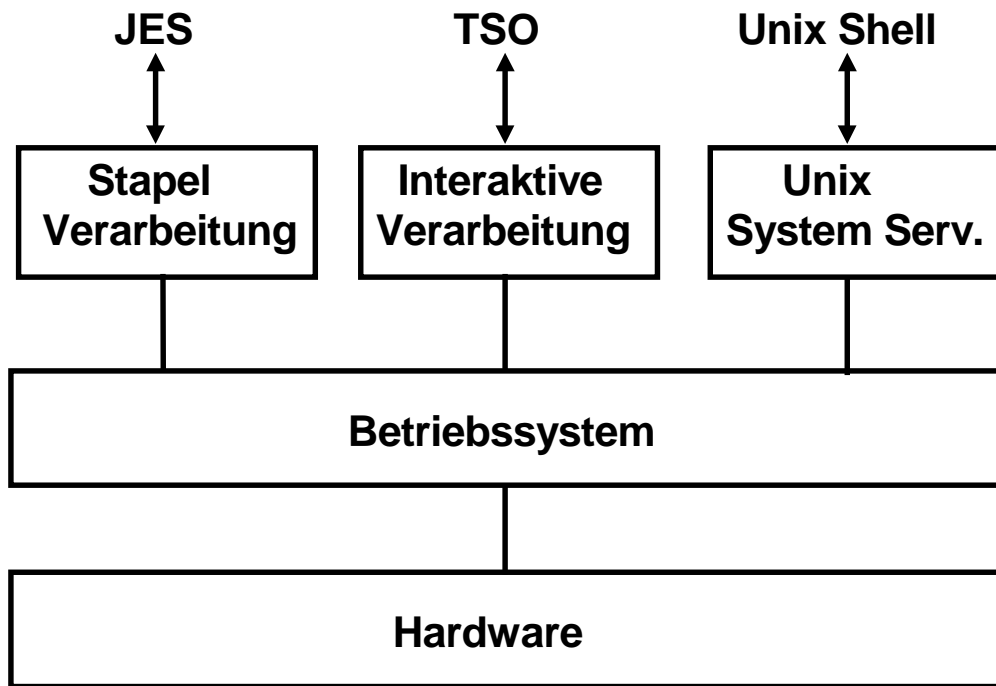
Ring 1,2

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 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

es 0518z ww6

wgs 09-99

OS/390 Prozessverwaltung

Ein OS/390 Prozess besteht aus mehreren Arbeitseinheiten, die als „Tasks“ bezeichnet werden. Eine Task 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.

es 0542 ww6

wgs 09-99

Server Zugriff

Unterschied zwischen Einzelplatzrechner und Client/Server Betriebssystemen. NT und Unix werden für beides eingesetzt. OS/390 ist ein reinrassiges Server Betriebssystem. Andere Beispiele: Tandem Pathway, DEC Vax.

Für einen Server Zugriff braucht der Klient Software. Möglichkeiten für selbstgeschriebene Anwendungen:

Sockets, RPC, Corba, DCOM, RMI

Unintelligente Klienten:

NT	Citrix
Unix	Telnet, X-Window
OS/390	3270
Vax	VT 100

Anwendungsspezifische Klienten:

SAP R/3	SAPGUI
WWW	Browser

TSO („Time Sharing Option“)

Vergleichbar mit dem ähnlichen UNIX Time Sharing

Hauptsächliche Anwendung: Software Entwicklung und Test

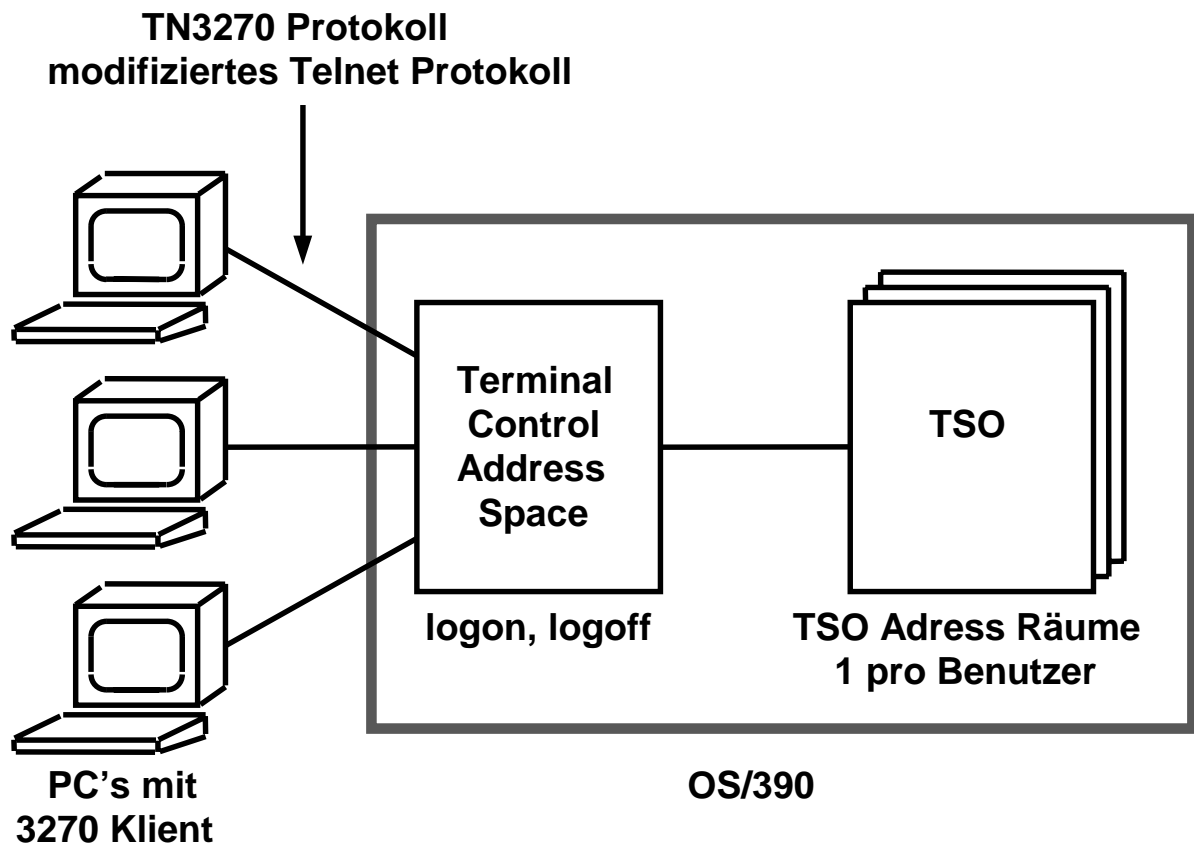
Alternative für die Software Entwicklung: Nutzung des VM Betriebssystem in einer separaten LPAR

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.

Typische Vorgehensweise:

- **Benutzer am TSO Bildschirm erstellt Anwendung und kompiliert sie entweder interaktiv (im Vordergrund, unter TSO), oder im Background (unter JCL).**
- **Benutzer führt die Anwendung entweder interaktiv aus (unter TSO), oder schreibt JCL Code und übergibt Anwendung an JES für Ausführung im Stapel.**
- **Im letzteren Fall schedules JES die Anwendung und stellt Ergebnisse in JES SPOOL File (JOB Output Datasets).**



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.

TCPIP MSG10 ==> SOURCE DATA SET = SYS1.LOCAL.VTAMLST (USSTCPIP)

08/01/01

W E L C O M E T O

17:17:08

```
SSSSSS // 3333333 9999999 0000000
SS // 33 33 99 99 00 00
SS // 33 99 99 00 00
SSSS // 33333 9999999 00 00
SS // 33 99 00 00
SS // 33 33 99 99 00 00
SSSSSS // 3333333 9999999 0000000
```

YOUR TERMINAL NAME IS : SCOTCP05

YOUR IP ADDRESS IS : 217.081.157.057

APPLICATION DEVELOPMENT SYSTEM

OS/390 RELEASE 2.7.0

==> ENTER "L " FOLLOWED BY THE APPLID YOU WISH TO LOGON TO. EXAMPLE "L TSO"
FOR TSO/E OR "L C001" FOR THE CICSC001 CICS APPLICATION.

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 : IKJACNT
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

Detallierte Screen by Screen Tutorials sind zu finden unter

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

Die wichtigsten OS/390 Sprachen

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

Skripte

Shellskripte unter Unix

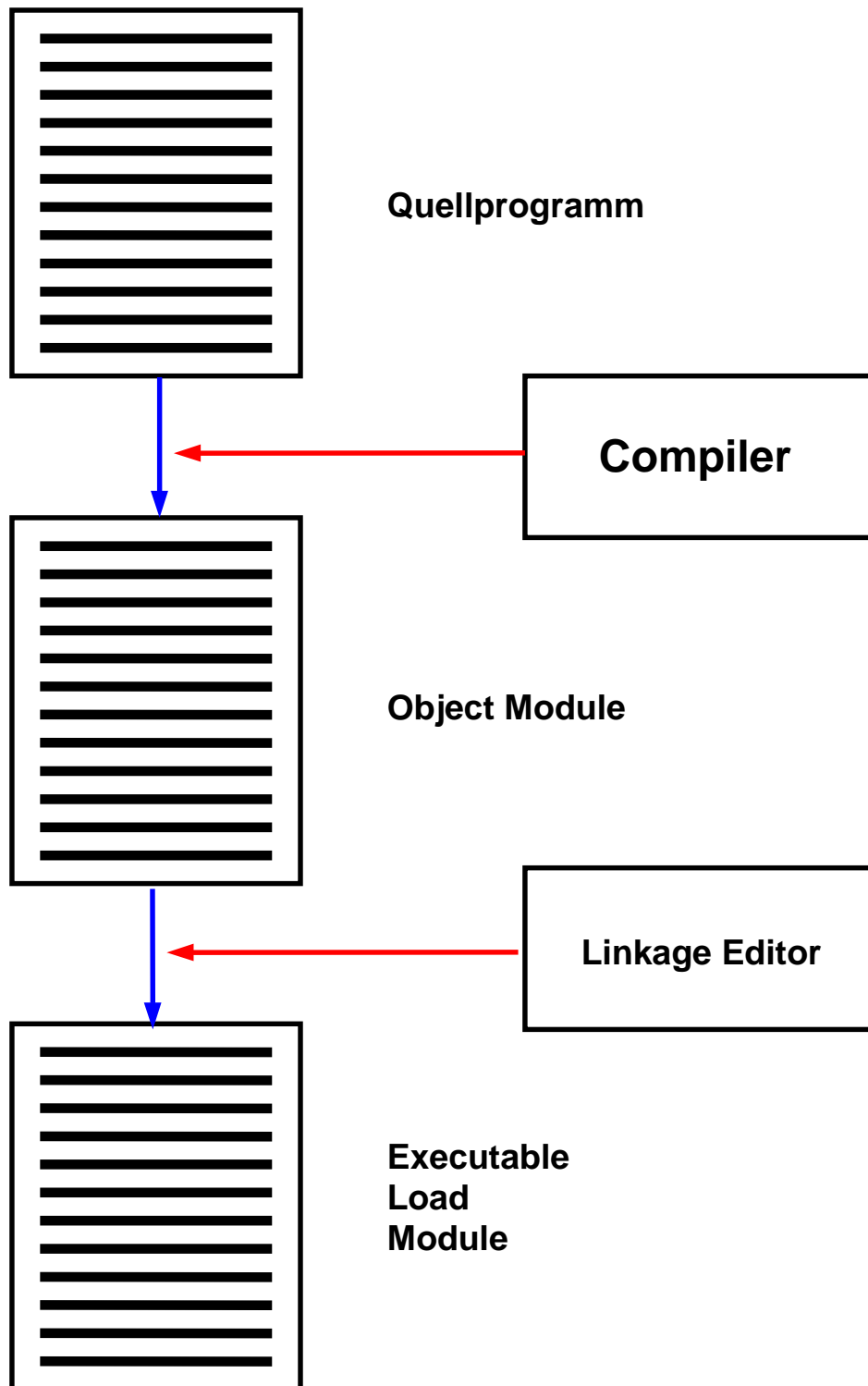
BAT-Files unter DOS

REXX unter OS/390

REXX

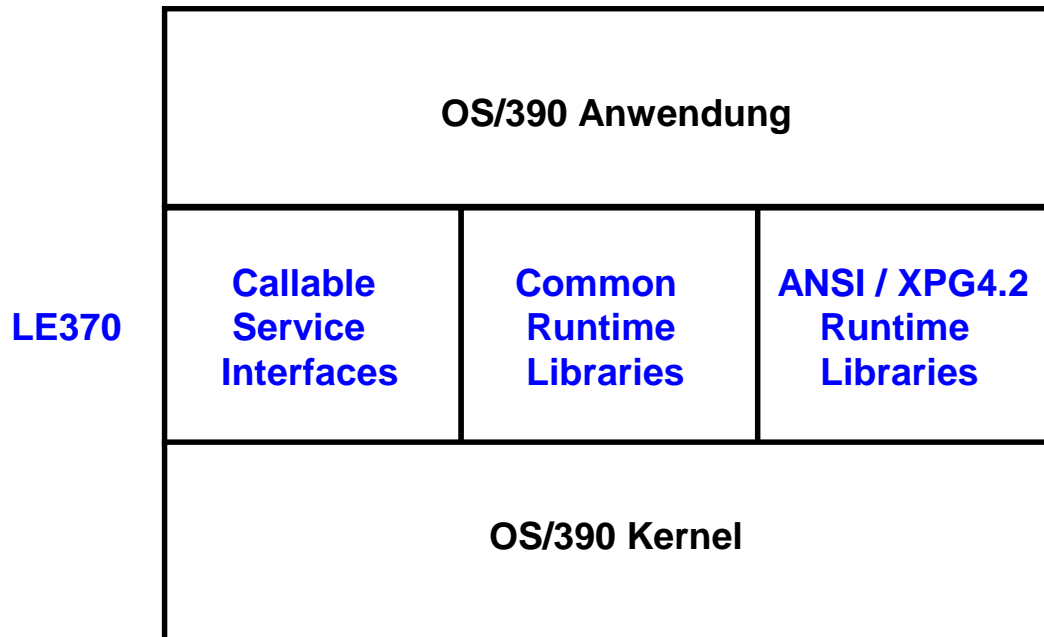
Von IBM entwickelte interpretative Scriptsprache, verfügbar für alle IBM Betriebssysteme, vergleichbar zu Pearl oder Tcl/tk. Vollwertige Programmiersprache.

Weit verbreitet bei den Benutzern von S/390 Rechnern.



Programme und Load Module

Ein JCL Script kann eingesetzt werden, um Compile und Link Edit in einem Schritt durchzuführen



LE370

LE370 ist eine von mehreren Sprachen (C/C++, Cobol, PL/1) gemeinsame genutzte Laufzeitumgebung. Es enthält:

Basis Funktionen für

- **den Start oder Stop von Programmen**
- **Aufrufe von Programmen untereinander, die in unterschiedlichen Sprachen geschrieben sind**
- **Behandlung von Fehlermeldungen**

Library Funktionen wie z.B.

- **mathematische Funktionen**
- **Datum und Zeitdienste**

Code Page, Locale

Ein „Character Set“ ist eine Menge von alpha-numerischen Zeichen

Eine „Code Page“ ist eine Zuordnung von hexadezimalen Werten zu den Zeichen eines Charakter Sets.

Eine „Locale“ beschreibt eine kulturelle Umgebung. Hierzu gehören neben einer Code Page Spezifikation:

- Sortierfolge
- Datum Konventionen
- Uhrzeit Konventionen
- Darstellung numerischer Daten (z.B. Dezimal Komma, Dezimalpunkt)
- Darstellung von Geldbeträgen

Wichtige OS/390 Code Pages sind:

- 273 Deutschland
- 1141 Deutschland Euro
- 037 USA
- 1047 USA

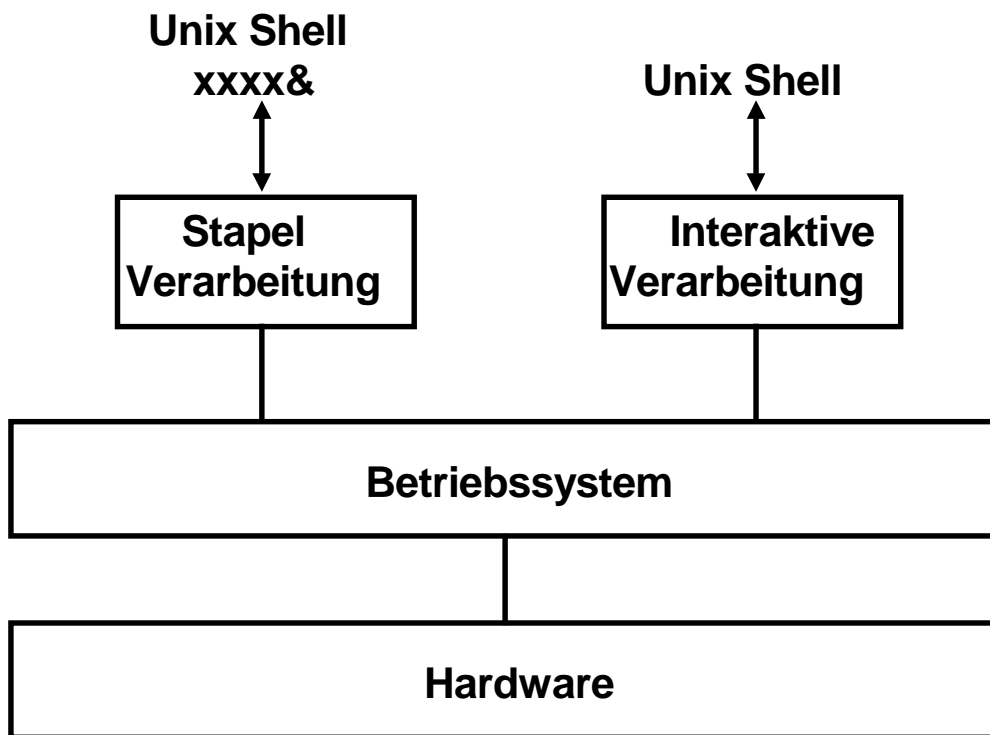
Für Entwicklungsaufgaben in C/C++ empfiehlt sich OS/390 Code Page 1047. Ein bekanntes Problem ist die unterschiedliche Darstellung der rechteckigen Klammern in Code Page 037 (normalerweise von OS/390 verwendet) und Code Page 1047.

Ein 3270 Klient (Emulator) konvertiert automatisch ASCII in EBCDIC. Die OS/390 Code Page läßt sich in der Regel im 3270 Klienten einstellen.

Unix File System	OS/390 Partitioned Data Set
Root /	Master Catalog
User Directory /u/maier	User Prefix MAIER
Subdirectory /u/maier/test1/c	Partitioned Data Set MAIER.TEST1.C
File /u/maier/test1/c/pgma	Member MAIER.TEST1.C(PGMA)

Vergleich Unix Hierarchisches File System mit OS/390 Partitioned Data Set

Unix Files	adressieren einzelne Bytes Plattenspeicherplatz wird automatisch zugeordnet (allocated)
OS/390 Files	adressieren Records Plattenspeicherplatz wird von Hand zugeordnet (allocated)



Unix Grundstruktur

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

Stapelverarbeitung vs. Interaktive Verarbeitung Batch Processing vs. Interactive Processing

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 werden

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

Transaktionsverarbeitung ist ein Sonderfall der Interaktiven Verarbeitung, gekennzeichnet durch

- **run-to-completion im Gegensatz zu preemptive Processing**
- **ACID Properties (Atomicity, Consistency, Isolation, Durability)**

Geldausgabeautomaten oder Flugplatzreservierungssysteme sind typische Beispiele für eine Transaktionsverarbeitung

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 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.

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

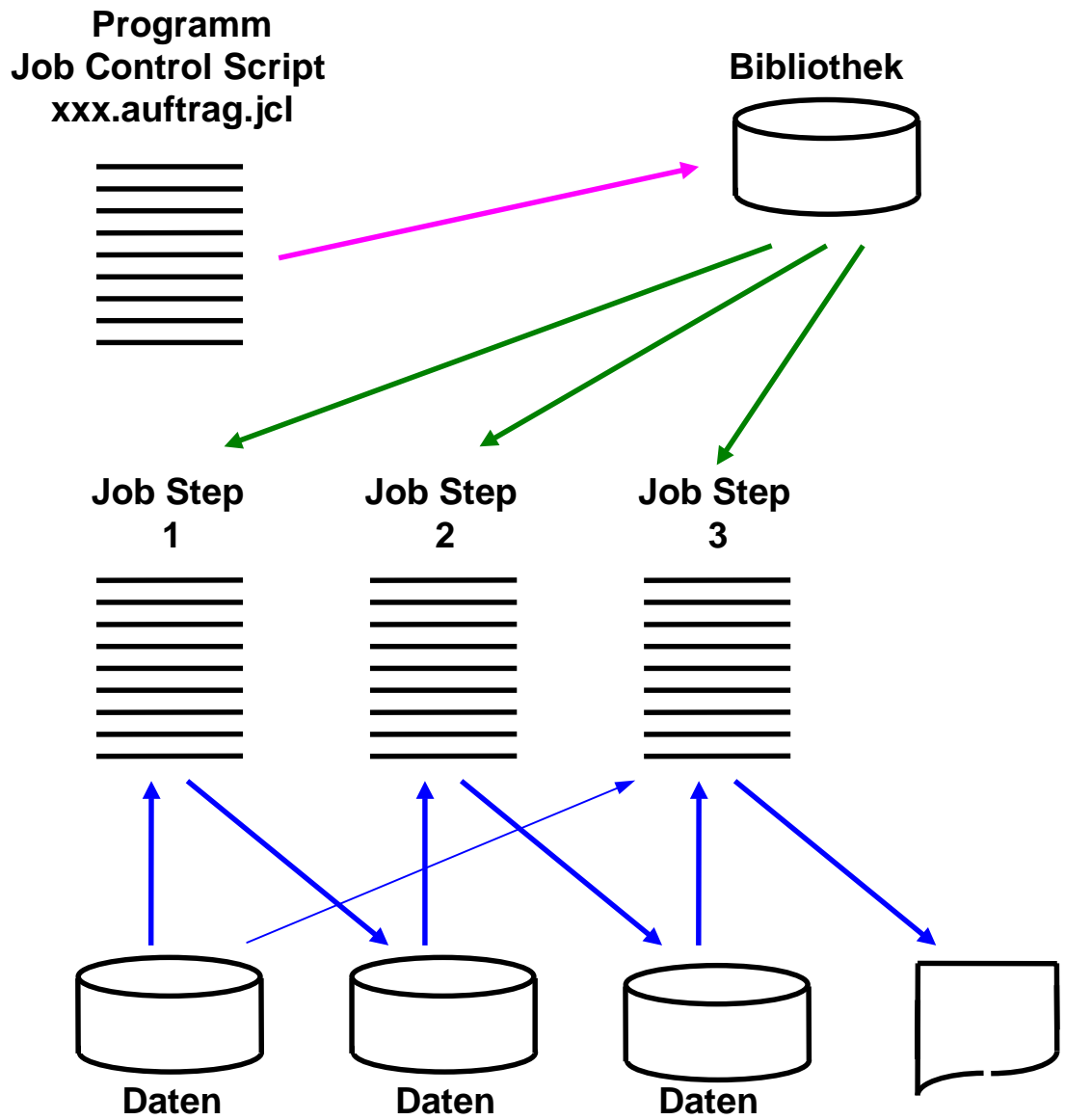
Voraussetzung für die Ausführung eines Jobs ist, daß benutzte Programme und Dateien (Data Sets, Files) bereits existieren.

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)**
- 3. Submission**

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.



Konzept eines OS/390 Jobs

```

//LGRH JOB (OS33,0222),SPRUTH,CLASS=G,PRTY=6
//STEP1 EXEC PLILFCLG
//PLIL.SYSIN DD *
LGRH : PROCEDURE OPTIONS (MAIN);
      DECLARE LAGRBST FILE RECORD SEQUENTIAL BUFFERED;
      DECLARE 1 EABRCH
            2 KENNZEICHEN,
            3 TEILNR      PICTURE 'XXXXX',
            3 BEZEICHNUNG CHARACTER (20),
            2 BESTAND,
            3 VORHANDEN   PICTURE '99999',
            3 BESTELLT   PICTURE '99999',
            2 PREIS,
            3 EINKAUF     PICTURE '999V99',
            3 VERKAUF     PICTURE '999V99',
            2 DUMMY CHAR(35);
      OPEN FILE (LAGRBST) OUTPUT;
      ON ENDFILE (LAGRBST) GO TO Y100;
Y200 : GET FILE (INPUT) EDIT (EABRCH) (P'XXXXX',A(20),F(5),
                                       F(5),F(5,2),F(5,2),A(35));
      WRITE FILE (LAGRBST) FROM (EABRCH);
      GO TO Y200;
Y100 : CLOSE FILE (LAGRBST);
      END LGRH;

/*
//GO.LAGRBST DD UNIT=TAPE9,DSN=WGS.LAGRBST,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),
// DISP=(NEW,KEEP),LABEL=(1,SL),VOL=SER=006549
//GO.INPUT DD *
A4711      MESSLEHRE 300 100 450 750
A1354      LOETZINNROLLE 3000 575 80 150
A2571      BOWDENZUG 160 20 550 820
A1113      GETRIEBE 6 1 7500 9800
A0104      KUGELLAGER 105 10 1300 1850
A3295      MESSZYLINDER 50 5 660 900
A5301      OELKANISTER 1700 200 350 490
A4009      GEWINDESCHNEIDER 15 0 6000 7500
A0008      BALKENWAAGE 80 10 1500 2100
A3193      WASSERWAAGE 30 20 2500 3300
A6387      SCHWEISSAPPARAT 3 08000095000
/*

```

PL/1 Programmierbeispiel

Job Entry Subsystem (JES)

Die Übergabe (Submission) eines Stapelverarbeitungs-Jobs an das Betriebssystem erfolgt durch einen Benutzer; bei einem Großrechner durch einen spezifisch hierfür abgestellten Bediener (Operator).

Da im Laufe eines Tages hunderte oder tausende von Jobs bewältigt werden müssen, wird der Vorgang der 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.**

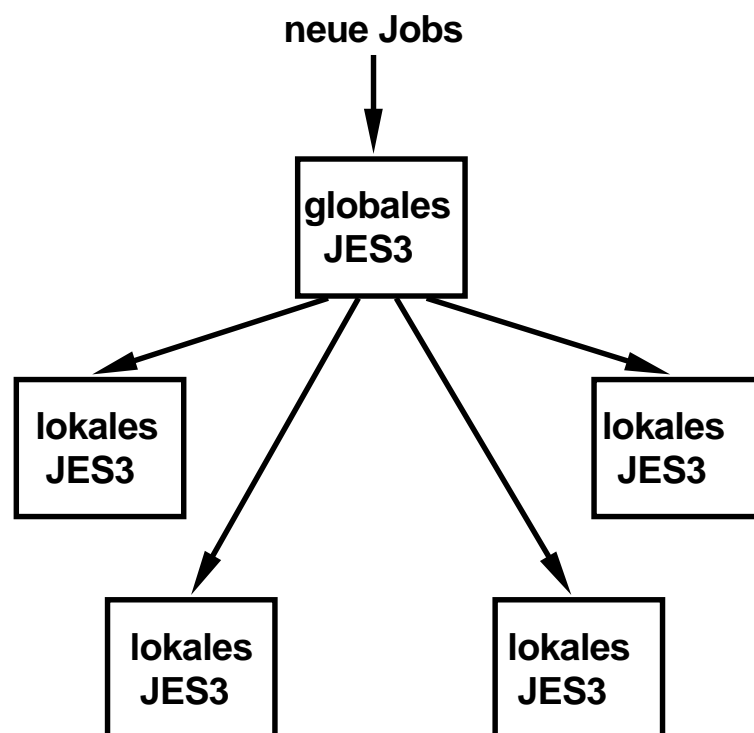
Jobs werden an das Betriebssystem über JES gesteuerte Queue Server (Initiator) übergeben. Jeder Initiator bedient einen einzigen virtuellen Adressenraum, in dem er auch selbst untergebracht ist..

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

JES2 und JES3

Aus historischen Gründen existieren zwei leicht unterschiedliche JES Varianten: JES2 und JES3. In einer gegebenen OS/390 Installation wird jeweils nur eine der beiden Varianten eingesetzt.

In großen SYSPLEX Konfigurationen ist JES3 die bevorzugte Komponente.

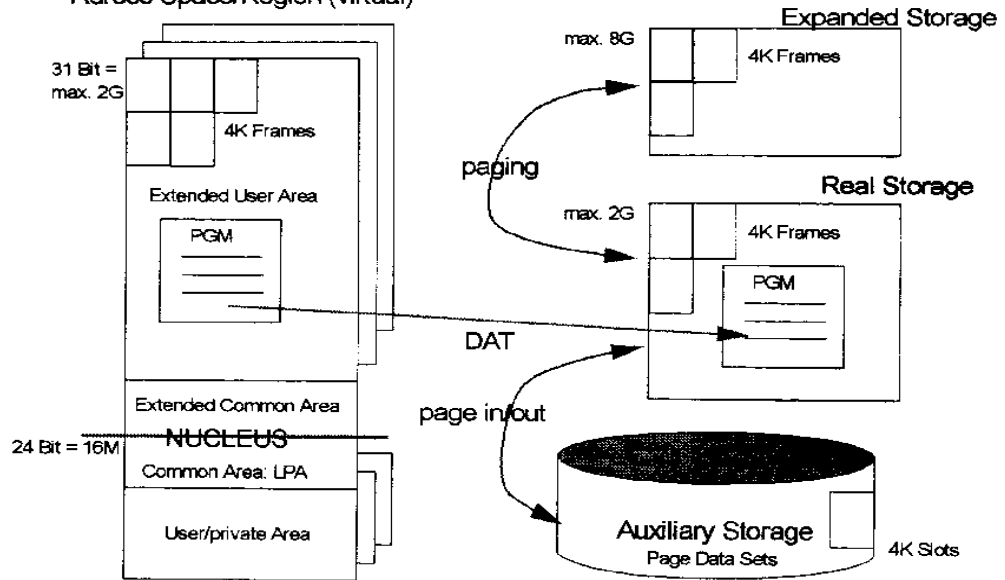


Mit Hilfe einer zentralen Ressource Verwaltung und Zuordnung kann ein „Single System Image“ erstellt werden, welches die verteilte Struktur des SYSPLEX vor dem Benutzer weitgehend verbirgt.

Anwendungsprogrammierer können mit Hilfe von JCL ähnlichen Befehlen in ihrem Code festlegen, daß der Job zu vorgegebenen Zeitpunkten (z.B. täglich, monatlich oder jährlich) laufen soll.

OS/390 Memory Management

Address Space/Region (virtual)



Es 0420 ww97

wgs 08-00

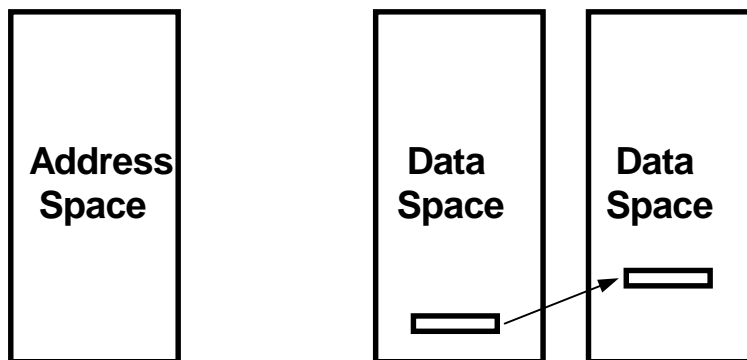
Es 0420 ww97

wgs 08-00

OS/390 Adressenräume

Virtuelle Adressräume werden als „Spaces“ oder „Regions“ bezeichnet. Die Größe war bisher auf 2 Gbyte begrenzt (31 Bit). Die X3 Serie erweitert dies auf 63 Bit.

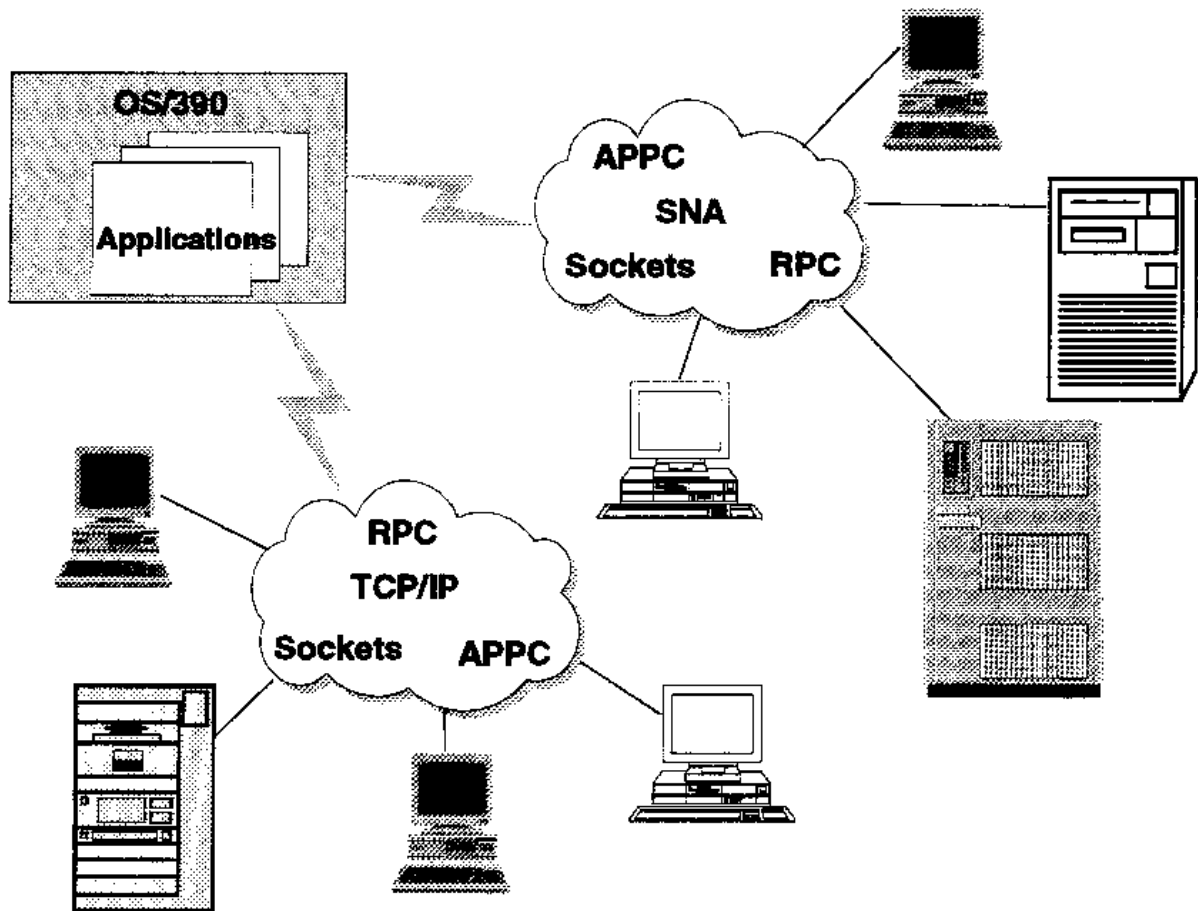
Ein Prozeß kann mehrere 2 Gbyte Spaces gleichzeitig verwalten. Es wird zwischen Address Spaces und Data Spaces unterschieden.



Programmcode befindet sich in einem Address Space. Ein Prozess kann auf die ihm zugeordneten Data spaces gleichzeitig zugreifen, z. B. mit einem MOVE Maschinenbefehl.

Multitasking innerhalb eines Address Spaces ist möglich. Prozesse und Threads werden durch Task Control Blöcke (TCB) und Service Request Blöcke (SRB) verwaltet.

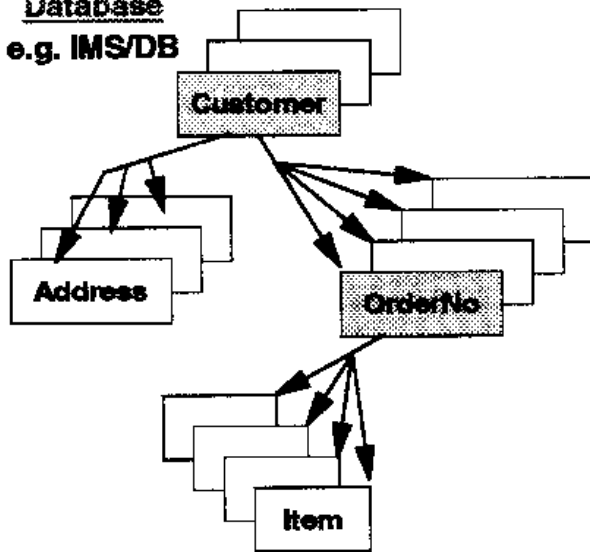
Der Inhalt eines Data Spaces geht verloren, wenn der Prozess, der diesen Data Space besitzt, terminiert wird.



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.

Hierarchical Database
e.g. IMS/DB



Good for higher volume, lower complexity data

Customer Table

Address Table

Relational Database
e.g. DB2

Orders Table

Items Table

Good for lower volume, higher complexity data

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.

OS/390 Sicherheit

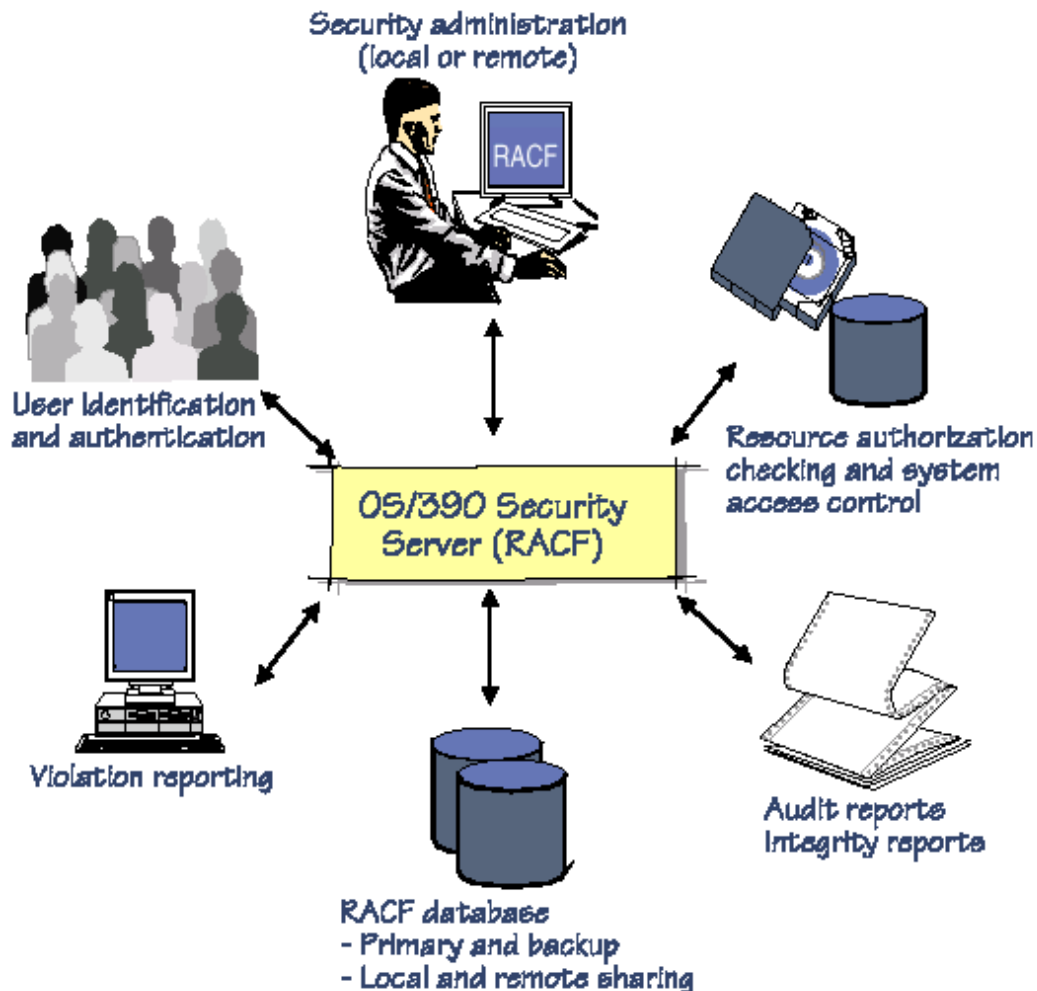
Unter OS/390 kann ein Programm nur auf Bereiche zugreifen, für die es eine explizite Zugriffsauthorisierung erhalten hat, die durch den Überwacher des Betriebssystems vergeben und überwacht wird.

Der Default Setup unter OS/390 gestattet einem Programm keinen Zugriff auf Bereiche, die anderen Programmen zugeordnet sind. Um hierüber hinweg zu kommen, bedarf es einer expliziten Authorisierung durch Sicherheitsadministratoren oder Systemprogrammierer.

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

Kryptographischer Koprozessor

OS/390 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

Literatur : IBM GC28-1912-06

RACF (1)

Resource Access Control Facility

OS/390 spezifiziert kritische Events innerhalb des Betriebssystems als sicherheitssensitive Verzweigungen. Die MVS Security Authorisation Facility (SAF) des OS/390 Kernels bewirkt an diesen Stellen den Aufruf einer Security Engine, eines Prozesses, der im Benutzer Status läuft.

In OS/390 ist diese Security Engine häufig RACF; alternative OS/390 Security Engines sind ACF/2 oder TopSecret der Firma Computer Associates. Neben einer Kontrolle von Events (ehe sie wirksam werden) wird ein umfassender Audit Trail erzeugt.

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.

Problem: Maintenance der Profile.

RACF

Resource Access Control Facility

RACF benutzt das Konzept von „Ressourcen“, die geschützt werden sollen.

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

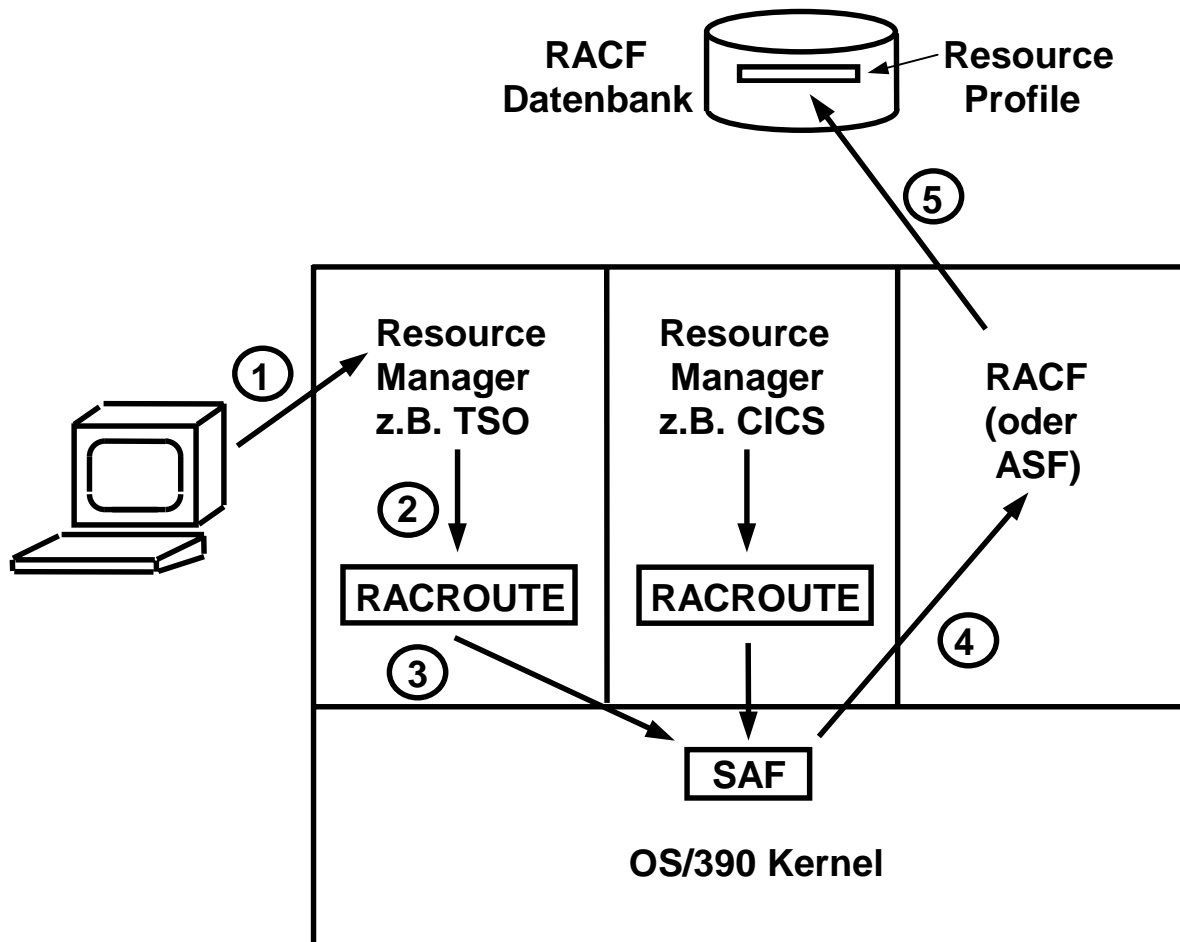
- **Benutzer**
- **Dateien**
- **CICS Transaktionen**
- **Datenbank Rechte**
- **Terminals**

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

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.

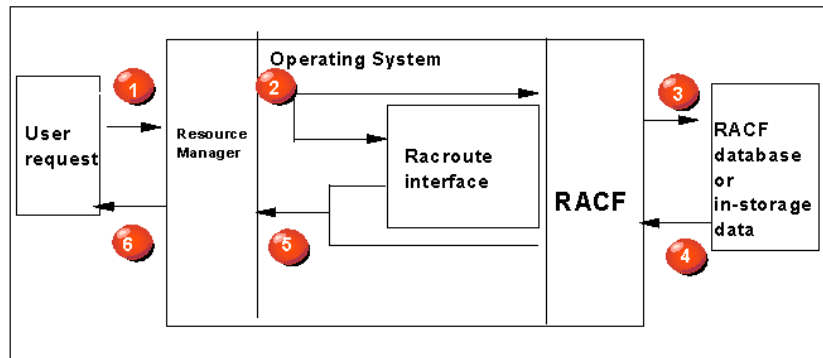
Benutzer Profile enthalten „Capabilities“. Datei Profile enthalten „Access Control Listen“.



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 OS/390 Kernels zuzugreifen
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

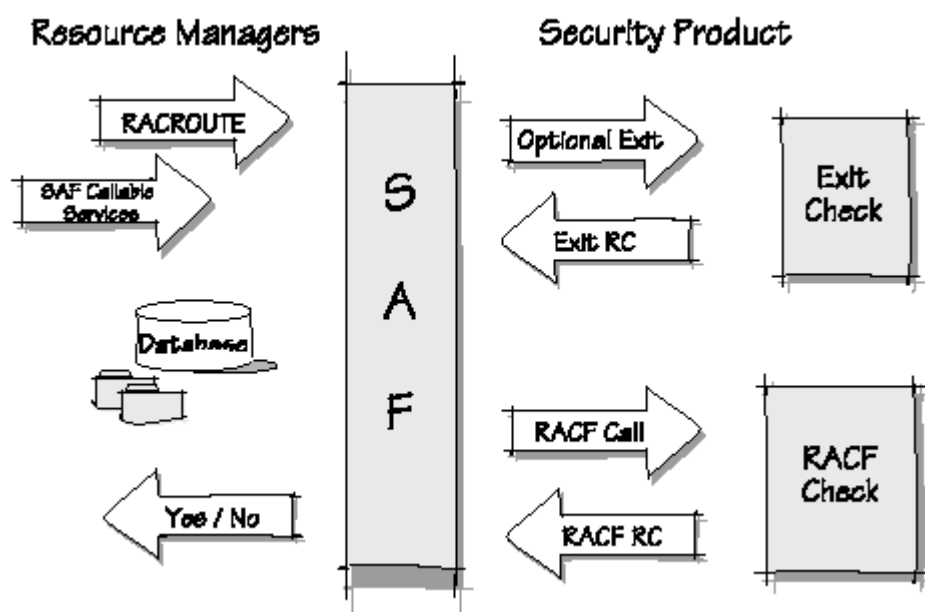


After a user enters a command to the system to access the resource, a system resource manager (such as data management) processes the request. The resource manager, based on what RACF indicates, either grants or denies the request:

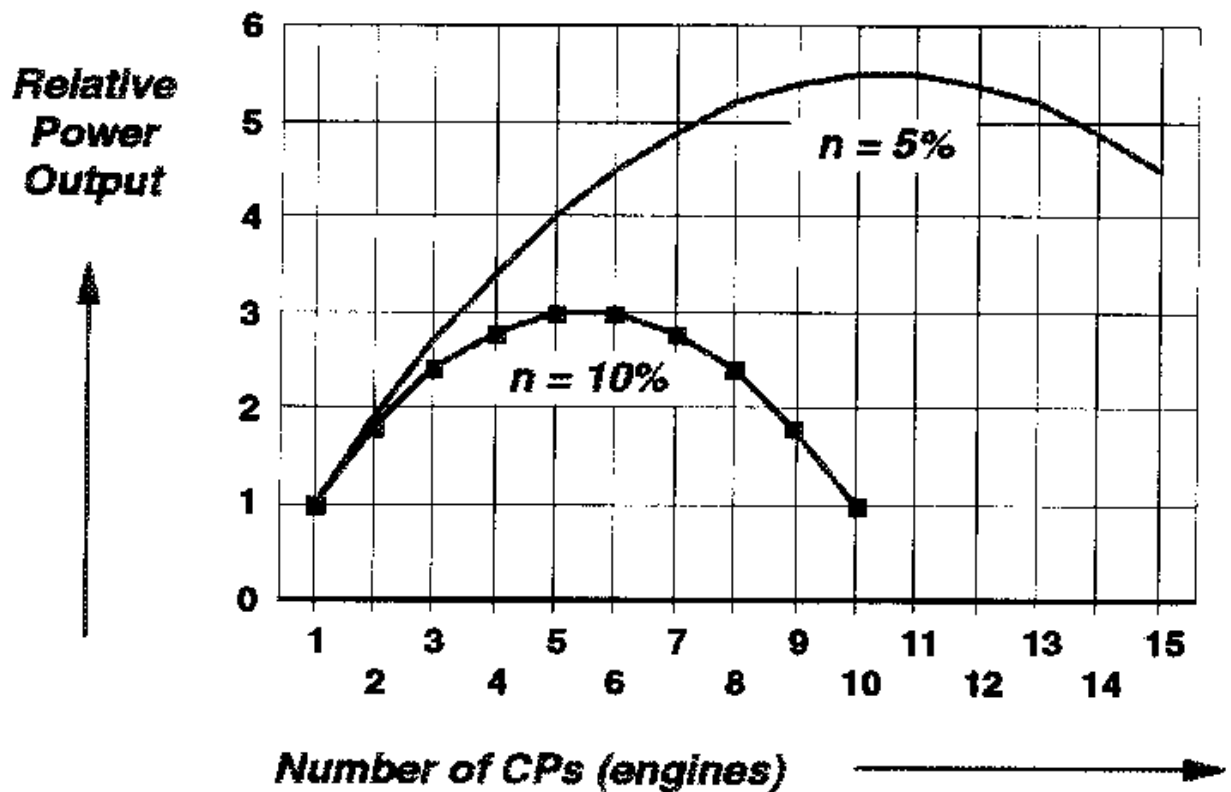
- 1. A user requests access to a resource using a resource manager (for example, TSO/E).**
- 2. The resource manager issues a RACF request to see if the user can access the resource.**
- 3. RACF refers to the RACF database or in-storage data and.. .checks the appropriate resource profile..**
- 4. Based on the information in the profile...**
- 5. RACF passes the status of the request to the resource manager.**
- 6. The resource manager grants (or denies) the request.**

System Authorization Facility

The system authorization facility (SAF) is part of the operating system and conditionally directs control to RACF, if RACF is present, or to a user-supplied processing routine, or both, when receiving a request from a resource manager.]



SAF does not require any other product as a prerequisite, but overall system security functions are greatly enhanced and complemented if it is used concurrently with RACF. The key element in SAF is the SAF router. The SAF router is always present, even when RACF is not present. The SAF router is a system service that provides a common focal point for all products providing resource control. This focal point encourages the use of common control functions shared across products and across systems. The resource-managing components and subsystems call the OS/390 router as part of certain decision-making functions in their processing, such as access-control checking and authorization-related checking. These functions are called control points.



Leistungsverhalten eines Symmetric 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.

OS/390 Multiprocessing Faktoren (MPF)

Jahr	Prozessor Modell	Anzahl CPU's	Leistungsverhältnis 2-W- SMP zu Einzelrechner
1969	S/360	2	1,3 .. 1,4
1973	S/370	2	1,5 .. 1,8
1977	3033	2	1,7 .. 1,9
1981	3081	2	1,8 .. 1,96
1983	3084	4	1,8 .. 1,96
1985	3090	4	1,79 .. 1,94
1987	3090 E	6	1,81 .. 1,97
1989	3090 J	6	1,84 .. 2,0
1991	9021 900	6	1,78 .. 1,93
1994	9021 9x2	10	1,80 .. 1,93
1995	9672 R2	7	1,85 .. 1,99
1996	9672 G3	10	1,90 .. 2,0
1997	9672 G4	10	1,82 .. 1,96
1998	9672 G5	10	1,83 .. 1,96
1999	9672 G6	12	
2000	z900	16	
2002			

(S/390) MIPS
Million Instructions Per Second

Performance Benchmark für S/390 Rechner

Ausführungszeit für eine Mischung von Maschinenbefehlen

Reine CPU Leistung, keine Ein/Ausgabe

Proprietärer IBM Standard

**verfügbar seit 1965, ständig erweitert und angepaßt
an realistischen Anwendungsprofilen orientiert
anwendbar für Rechner unterschiedlicher Hersteller**

Berücksichtigt

**Häufigkeit der einzelnen Maschinenbefehle
Cache- und Hauptspeicherzugriffszeiten
Bus Latency/Contention
Cache Misses und Cache-Line reload
Memory Refresh**

OS/390 vs. Unix

OS/390

mehrere 1000 E/A Operationen / s

Unix

mehrere 100 E/A Operationen / s

Native Unix Betriebssysteme für S/390

**Amdahl UTS (Universal Time Sharing System)
Marktführer, < 300 Installationen**

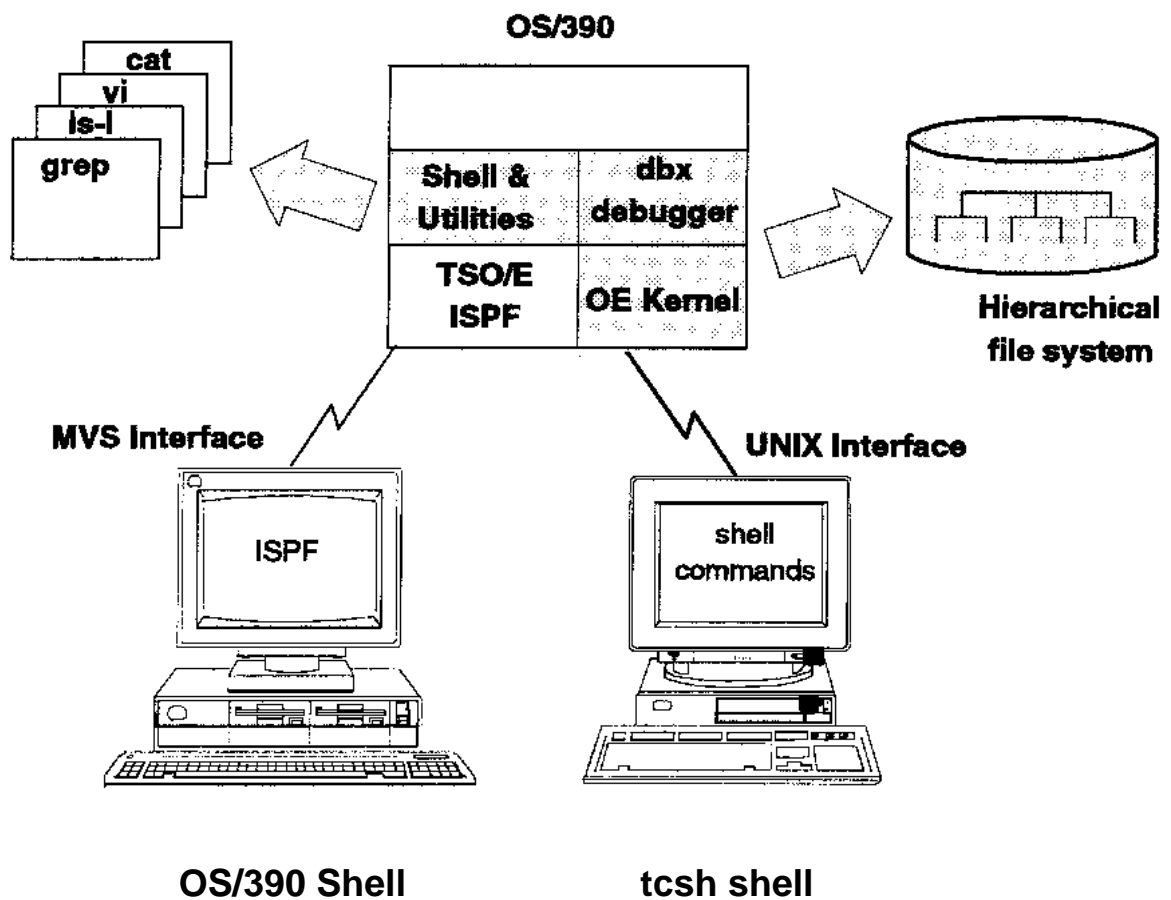
Hitachi HI-OSF/1-M

IBM AIX/ESA

OS/390 Unix System Services

früher als Open Edition MVS bezeichnet

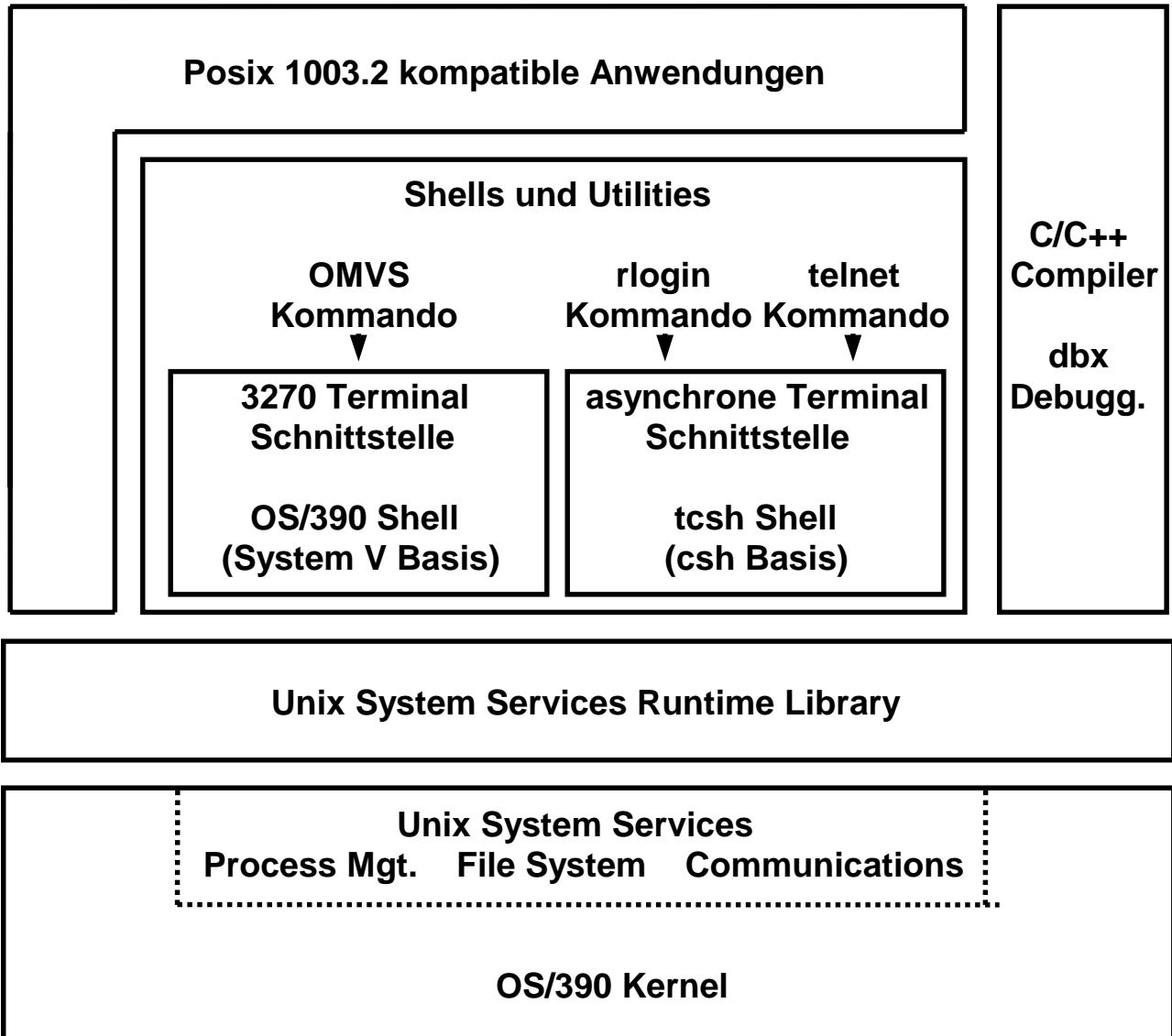
1100 Unix API's



OS/390 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

Supervisor Calls

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

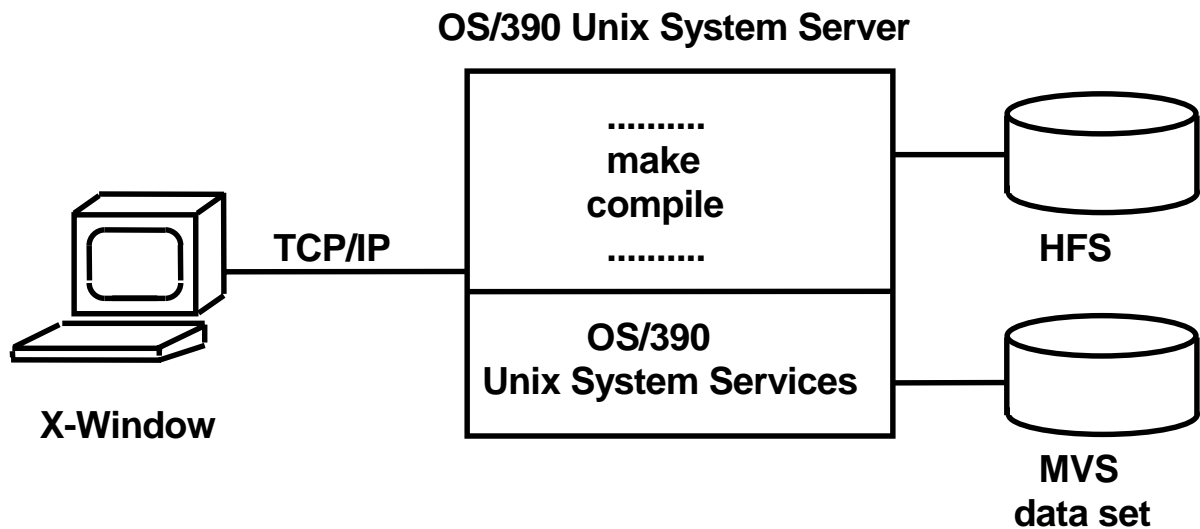
Supervisor Calls im weiteren Sinne sind Library Routinen, die eine Dienstleistung des OS/390 Kernels in anspruch nehmen.

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:

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



Compiling with c89

Application source code in HFS files or OS/390 data sets can be compiled and link-edited from within the shell using the Unix System Services c89 utility. The syntax is:

```
c89 [-options ...] [file.c ...] [file.a ...] [file.o ...] [-l libname]
```

where:

file.c	source file
file.o	object file
file.a	archive file
libname	archive library

Makefiles may be used to maintain Unix System Services application source and object files automatically when updating individual modules. The make utility runs c89.

The X Window API allows to write applications in the OpenEdition MVS environment that can be displayed on X11 servers on a TCP/IP based network. It provides that application with graphics capabilities as defined by the X Window System protocol.

Pitfalls

The C language allows to code the following:

```
char z;  
z = '7' ;  
if (47 < z && z < 58) { /* is numeric */  
    some code here  
}
```

What the code does is to check if the binary representation of `z` lies in the region between decimal 47 and 58. If you look up an ASCII table you will find that the numbers 0-9 are located at these code points. This code will compile immediately without any error on every machine.

What happens if such a piece of code is ported to the OpenEdition environment and translated to EBCDIC? It will definitely fail. Thoughtful testing is required even if the source code compiles immediately.

The OS/390 C compiler provides services which makes such a piece of code portable across ASCII and EBCDIC platforms

A similar problem occurs because the ASCII characters are stored contiguously in the code table, where the EBCDIC characters are not. The following do-loop runs nicely and makes sense in an ASCII environment.

```
for (i='A';i<='Z';i++) &lrb. do something here }
```

As the alphabet in an EBCDIC environment is not stored contiguously this piece of code is not directly portable to OpenEdition

Data Storage in the Hierarchical File System

The implementation of the Hierarchical File System is transparent to the type of data to be stored. Data may be text or binary; the implementation does not care about this. However, the user may need to know what kind of data he is working with when exchanging data with other systems:

Text Data

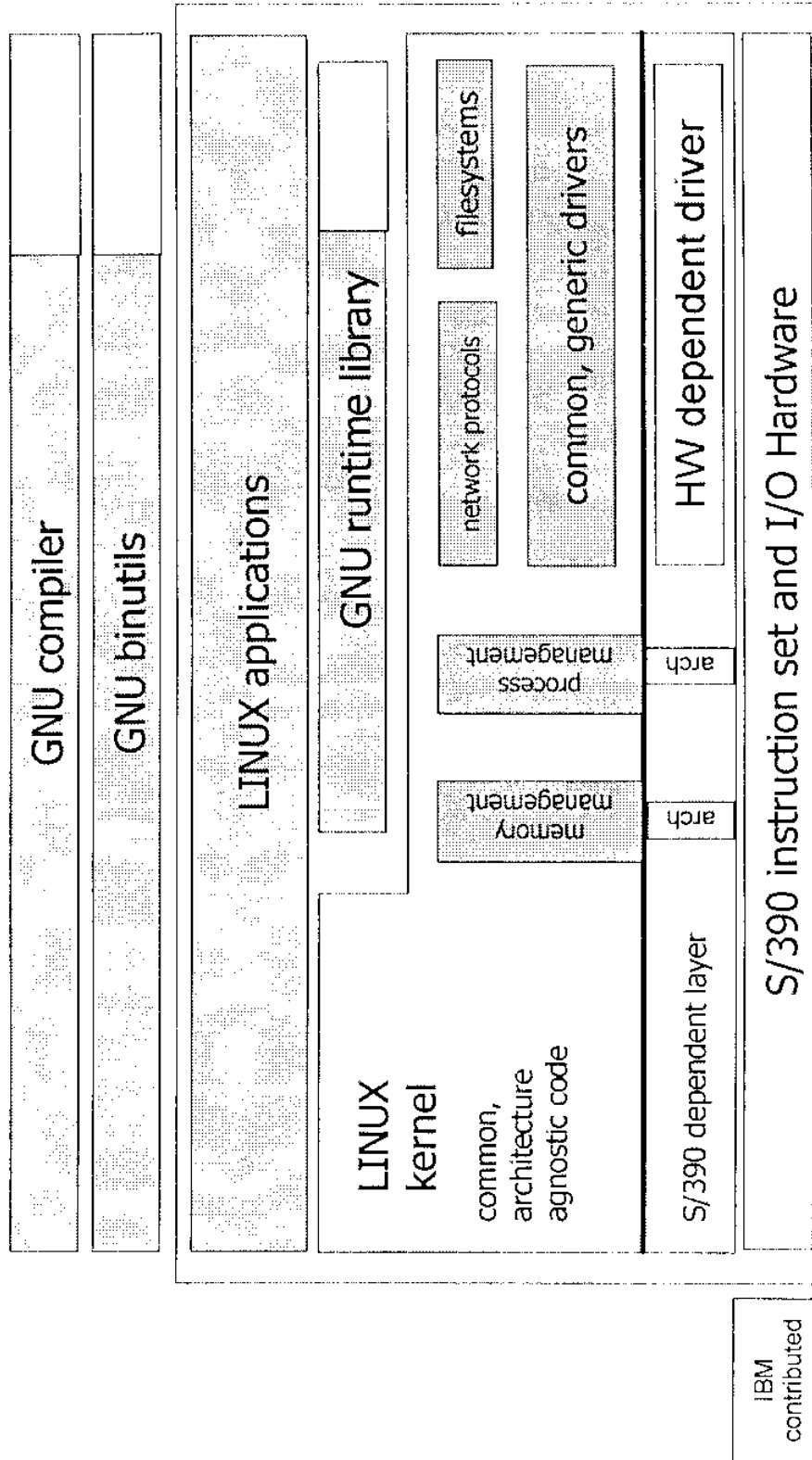
In a client/server environment where OS/390 participates, you have to consider how to do ASCII-EBCDIC translation. At first this might seem to be only a matter of two translation tables, but it goes much further than that. As soon as you have to deal with different countries, you have to handle translations using the appropriate code page. Unix System Services (USS) provides in the shell environment a command (ICONV), and for programs coded in C a conversion routine iconv(). Both allow you to convert according to the code pages that are applicable.

Binary Data

Binary data is much more complicated to handle than text data. Integer variables may be stored differently according to the byte order in storage (little endian (PC) versus big endian (S/390)). Different representations of floating point variables exist. There is no single simple solution for handling such differences. Each situation has to be analyzed to find a suitable solution.

In an ONC/RPC application as well as in a DCE/RPC application, the programming interface layer takes care of these translations and conversions.

Linux for S/390 structure



S/390 Linux

Entwicklung TSO	Lohn/Gehalt Stapel	SAP/R3 USS	Linux Anwendung
OS/390 LPAR # 1			Linux LPAR # 2
PR/SM			
Hardware			

Unix System Services (USS)

**OS/390 Nutzung: Sysplex, WLM, E/A
aber begrenzter Funktionsumfang**

Linux

Reichhaltige Anwendungsumgebung

CMS	CMS	OS/390	Linux	Linux	Linux
VM - CP					

S/390 Linux

David Boyes, System Administrator bei der Data Center Design und Testing Firma Dimension Enterprises in Herndon, Va., (www.dimension.net) berichtet von einem Experiment, bei dem er gleichzeitig 41400 Linux Instanzen auf einer logischen Partition eines S/390 Rechners unter dem VM/390 Betriebssystem hat laufen lassen.

http://opensourceit.earthweb.com/tools/000327_mainframe.html,
oder <http://www.linuxplanet.com/linuxplanet/reports/1532/1/>)

Telia, Scandnaviens größter Telekommunikations und Internet Service Provider, ersetzt seine 70 existierenden Web Hosting Unix Server mit einem einzigen IBM S/390 ZZ 7 Enterprise Server. Dieser wird 1500 virtuelle Internet Linux Server gleichzeitig hosten. Jeder virtuelle Server agiert als ein Web Server für individuelle Telia-hosted Geschäftskunden.

<http://www.hoise.com/primeur/01/articles/monthly/AE-PR-01-01-20.html>
es 0231 ww6

wgs 06-01

Linux bei der Banco Mercantil

Banco Mercantil ist eine der größten Banken in Venezuela mit 375 Zweigstellen, sowie weiteren Zweigstellen in den USA und anderen Ländern.

Banco Mercantil hat 30 existierende NT Server auf eine Linux LPAR ihres S/390 G6 Servers portiert. Es wird die SuSE S/390 Linux Distribution eingesetzt.

Die bisherigen NT Server arbeiten als File Server, Internet Domain Name Server und als LDAP Sserver.

Der Banco Mercantil S/390 Server benutzt drei logische Partitionen (LPARs) -- eine für Linux für S/390, eine für Domino für S/390, and eine für reguläre OS/390 Anwendungen. Es wird die Virtual Image Facility für Linux eingesetzt.

http://www-1.ibm.com/servers/eserver/zseries/news/pressreleases/2001/banco_linux_mainframe_050301.html
es 0574 ww6

wgs 07-01

Web articles

- "It's Official: IBM Announces Linux for the S/390" by Scott Courtney
<http://www.linuxplanet.com/linuxplanet/reports/1848/1/>
- "S/390: The Linux Dream Machine" by Scott Courtney
<http://www.linuxplanet.com/linuxplanet/reports/1532/1/>
- "Linux on Big Iron" by Daisy Whitney
http://www.opensourcet.com/tools/000327_mainframe.html
- "Has the Linux 'dream machine' arrived?" by Mary Jo Foley
<http://www.zdnet.com/zdnn/stories/news/0,4586,2491800,00.html>
- "Linux livin' large on mainframe" by Marc Songini
<http://www.nwfusion.com/news/2000/0403linuxibm.html>
- "Linux On A Mainframe Isn't A Joke"
<http://www.zdnet.com/sr/stories/column/0,4712,2550434,00.html>
- "IBM Unveils Way to Make Mainframes Act Like Hundreds of Smaller Servers" from Bloomberg (Technology News)
<http://www.bloomberg.com/bbn/technology.html>
- "IBM kicks off Linux developer program for its S/390 mainframes"
<http://www.infoworld.com/articles/pi/xml/00/03/31/000331piibmlinux.xml>
- "IBM Goes Countercultural With Linux" from the NY Times
<http://www.nytimes.com/library/tech/00/03/biztech/articles/20soft.html>

The iron penguin series by Neale Ferguson

- "Part 1: Linux takes to big iron within virtual machines"
<http://www.linuxworld.com/lw-2000-06/lw-06-S390-1.html>
- "Part 2: Building the iron penguin"
<http://www.linuxworld.com/lw-2000-06/lw-06-S390-2.html>
- "Part 3: Linux for S/390 in action"
<http://www.linuxworld.com/lw-2000-06/lw-06-S390-3.html>

Related web sites

IBM Linux

<http://www.ibm.com/linux/>

IBM Linux for S/390

<http://www.ibm.com/s390/linux/>

IBM VM/ESA and Linux Resources site

<http://www.ibm.com/s390/vm/linux/>

Marist College Linux for S/390 download site

<http://linux390.marist.edu/>

Linux for S/390 discussion group

Send a note to listserv@marist.edu, the body should contain

SUBSCRIBE LINUX-390

<http://linux.s390.org/>