

3. OS/390-Betriebssystem

3.1 Einführung

Ein Betriebssystem stellt dem Nutzer einer Rechenanlage eine Schnittstelle (Architektur) zur Verfügung, die leichter benutzbar ist als die direkten Zugriffe zur Hardware. Weiterhin besteht die Aufgabe des Betriebssystems darin, die Betriebsmittel (Ressourcen) wie Hauptspeicher- und Plattenspeicherplatz, Zugriff zu den E/A-Geräten und CPU-Zeit für eine bestimmte Anzahl von Benutzern und Prozessen zu verwalten. Damit das Betriebssystem, das sich aus dem Überwacher und weiteren Systemprogrammen zusammensetzt, diese Aufgabe erfüllen kann, wird es zusätzlich von den spezifischen Hardware-Einrichtungen unterstützt.

Für die Rechner der /390-Architektur existieren unterschiedliche Betriebssysteme, die in Abhängigkeit von der Größe der Installation implementiert werden. Dazu zählen die S/390-Betriebssysteme VSE, VM, OS/390 und TPF, die ausschließlich Produkt-Entwicklungen der Firma IBM darstellen. Das VSE-Betriebssystem hat heute keine besondere Bedeutung mehr. Es wurde im IBM-Entwicklungs-Labor in Böblingen aus der Taufe gehoben und läuft historisch bedingt momentan noch in direkter Konkurrenz zu Unix auf mittelgroßen /390-Installationen mit Erfolg. Insgesamt beläuft sich die Zahl der VSE-Lizenzen weltweit auf ca. 12.000.

Das Betriebssystem VM (Virtuell Machines) setzt sich aus einem Kernel (Control Program, CP) und darauf aufsetzenden, S/390-kompatiblen Nutzer-Betriebssystemen zusammen. Letztere werden von den Betriebssystemen CMS (Conversational Monitor Program), OS/390 und Linux implementiert (Performance-Verlust: < 5%). Das Control Program läuft im Überwacherstatus, während die Nutzer-Betriebssysteme einschließlich ihrer Kernel-Funktionen nur im Problemstatus arbeiten können. Privilegierte Maschinenbefehle (z.B. Ein-/Ausgabe) werden vom CP abgefangen und interpretativ abgearbeitet.

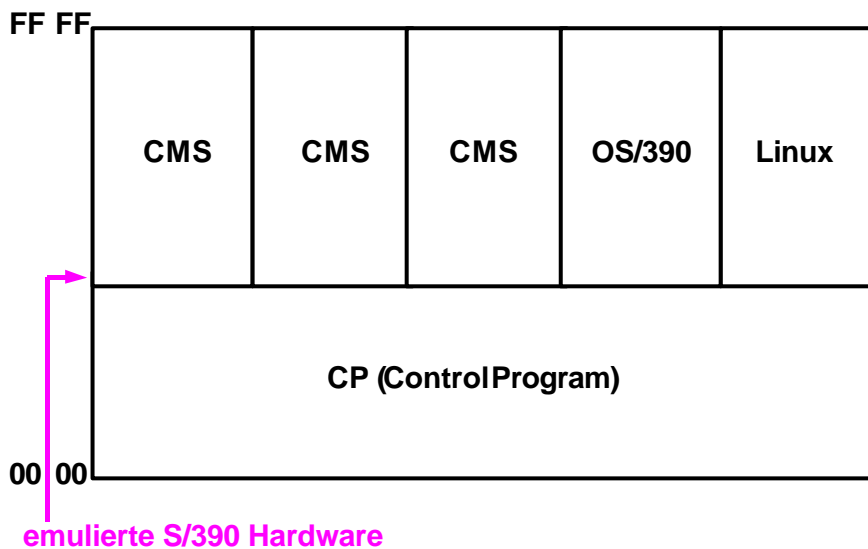


Abbildung 1: VM/ESA Betriebssystem

Das CMS-Betriebssystem ist ein besonders für die Software-Entwicklung ausgelegtes Einzelplatz-Betriebssystem. Für jeden CMS-Benutzer wird jeweils eine CMS-Instanz angelegt. Dasselbe gilt auch für Linux. Plattenspeicherplatz wird allen Nutzer-Betriebssystemen in Form virtueller "Minidisks" zugeordnet. Dagegen erfolgt die Verwaltung des Hauptspeicherplatzes dynamisch. Das VM-Betriebssystem gestattet es, wahlweise von einem Nutzer-Betriebssystem in ein anderes umzuschalten. Dieser Umschaltprozess erfolgt im Mikrosekunden-Bereich. Inzwischen existiert auf dem PC-Sektor ein Software-Produkt (VMware) mit ähnlichen Eigenschaften, d.h. es kann z.B. von NT auf Linux oder umgekehrt gewechselt werden, ohne den PC herunter- und anschließend wieder hochzufahren.

Für fast alle Großinstallationen von /390-Rechnern wird das OS/390 (MVS)-Betriebssystem implementiert. Es verwaltet alle Programme, startet ihre Ausführung und stellt die notwendigen Ressourcen und Dienstleistungen wie Drucker, Hauptspeicher und Plattenspeicher zur Verfügung. Momentan laufen ca. 13.500 Rechner unter dem Betriebssystem OS/390.

Eine Sonderstellung unter den S/390-Betriebssystemen nimmt das TPF (Transaction Processing Facility)-Betriebssystem ein. Es wurde speziell als Platzreservierungssystem für die Fluggesellschaft American Airlines

entwickelt und wird heute noch neben Registrierungen für Hotels (35.000), Reisebüros (30.000), Mietwagenfirmen (50) und Eisenbahngesellschaften vorrangig in Geldausgabe-Automaten eingesetzt. TPF unterscheidet nicht zwischen Kernel- und Problemstatus, d.h. sämtliche Anwendungen laufen aus Performance-Gründen im Kernelstatus ab.

Weitere S/390-Betriebssysteme wurden von den Firmen Amdahl (UTS 4, based on System V, Release 4) und Hitachi (OSF/1-M, Open System Foundation Unix) entwickelt. Bei allen S/390-Betriebssystemen (außer Linux) handelt es sich Server-Betriebssysteme, die für den Multi-User-Betrieb optimiert sind.

Für die S/390-Architektur existiert außerdem ein aus Benutzersicht "reines" Linux-Betriebssystem. Letzteres unterstützt die S/390-Prozessor-Architektur und -umgebungsspezifische Geräte. Linux und OS/390 können bei entsprechender Hauptspeicher-Größe (512 MByte) parallel benutzt werden.

3.2 PR/SM und LPAR

Die **Processor Resource/System Manager (PR/SM) Facility** ist ein Hardware-Standard in den modernen IBM-Server-Architekturen (S/390 Multiprise 2000, 9672 Parallel Enterprise Server, 9672 Parallel Transaction Server, 9674 Coupling Facility). PR/SM erlaubt die logische Partitionierung des **Central Processor Complex (CPC)** und unterstützt im Betriebssystem VM mehrfache Nutzer-Betriebssysteme. Mit Hilfe dieser Hardware (Mikrocode)-Einrichtung kann der physikalische Rechner in mehrere logische Rechner (**LPAR, Logical PARTition**) unterteilt werden. Jeder logische Rechner besitzt sein eigenes Betriebssystem, seinen eigenen unabhängigen realen Hauptspeicherbereich sowie eigene Kanäle und Ein-/Ausgabe-Geräte. Die gemeinsame Nutzung von Krypto-Coprozessoren und Ein-/Ausgabe-Geräten durch mehrere LPARs ist möglich (EMIF).

Im LPAR-Mode werden die Ressourcen des CPC (Central Processor Complex) auf mehrere Steuerprogramme aufgeteilt. Diese Steuerprogramme können auf demselben CPC simultan laufen. Jedes Steuerprogramm muss die Ressourcen benutzen, die zu der logischen Partition, in der es läuft, definiert sind.

Man kann eine **Logic Partition (LP)** so definieren, dass sie folgende Einheiten enthält:

- Ein oder mehrere **CPs** (CP: **C**entral **P**rocessor)
- **Central Storage** (Hauptspeicher)
- **Channel Pathes**
- **Optionaler Expanded Storage**
- **Asynchronous Data Mover (ADM) Facility**
- **Integrated Coupling Migration Facility (ICMF)**

Es ist außerdem möglich, eine LP zu einer Coupling Facility zu definieren. In dieser LP läuft der Coupling Facility Control Code.

LPs besitzen folgende Eigenschaften:

- Es können maximal 10 LPs definiert werden
- LPs sind nutzbar in ESA/390-, ESA/390 TPF-, S/370- oder Coupling Facility Mode
- Der Speicher für jede LP ist isoliert. Central Storage und Expanded Storage können durch LPs nicht gemeinsam (shared) benutzt werden.
- Wenn die dynamische Speicher-Rekonfiguration verwendet wird, kann eine LP Speicher abgeben oder erhalten. Dieser erhaltene Speicher muss von anderen LPs abgegeben werden.
- Alle Channel Path's (ausgenommen CFR und ISD Channel Paths) sind als rekonfigurierbar definierbar. Channel Path's werden den LPs zugewiesen. Es ist möglich, rekonfigurierbare Channel Path's zwischen den LPs zu bewegen, indem Tasks benutzt werden, die entweder in der Hardware Management Console oder in der Support Element Console verfügbar sind. Wenn das Steuerprogramm, das in den LPs läuft, die physikalische Channel Path Rekonfiguration unterstützt, können die Channel Path's zwischen den LPs durch Steuerprogramm-Kommandos bewegt werden, ohne das Steuerprogramm zu unterbrechen.
- **EMIF (ESCON Multiple Image Facility)** erlaubt, Channel Paths von 2 oder mehr LPs zur selben Zeit zu nutzen. Nur CTC, CNC, CFS, OSA und ISD Channel Paths sind gemeinsam (shared) verwendbar.
- LPs können per Definition so viele CPs besitzen, wie installiert sind. CPs können den LPs zugeordnet werden oder durch sie geshared werden. CPs, die man definiert als einem LP zugeordnet, sind nicht verfügbar, um für andere aktive LPs zu arbeiten. Die Ressourcen von gemeinsam benutzten CPs werden zu aktiven LPs zugewiesen, wie sie benötigt werden. Man kann CP-Ressourcen begrenzen, wenn es erforderlich ist.

- Ein Mix von gemeinsam benutzten und zugeordneten (dedizierten) CPs für eine einzelne LP ist nicht möglich. CPs für eine LP sind entweder alle zugeordnet oder alle shared. Jedoch kann ein Mix aus LPs mit shared CPs und LPs mit zugeordneten CPs definiert und parallel aktiviert werden.

Aus Sicherheitsgründen werden folgende Maßnahmen getroffen:

- Reservieren rekonfigurierbarer Channel Paths für die exklusive Nutzung von einem LP
- Begrenzung der Autorität von einer LP, um irgendeine IOCDs in der Konfiguration zu lesen oder zu schreiben und Begrenzung der Autorität von einer LP, um die I/O-Konfiguration dynamisch zu ändern.
- Begrenzen der Autorität einer LP, um die CPU-Nutzungswerte für alle LPs in der Konfiguration abzurufen.
- Begrenzen der Autorität einer LP, um bestimmte Steuerprogramm-Anweisungen, die andere LPs beeinflussen, zu untersuchen.

3.3 OS/390-Betriebssystem

Das Betriebssystem OS/390 unterscheidet sich in seiner Basis-Struktur (Abbildung 2) grundsätzlich nicht von einem beliebig anderen Betriebssystem. Das herkömmliche 3-Schichtenmodell einer modernen Rechnerarchitektur besteht aus: Hardware, Betriebssystem, Benutzer-Prozesse. Zwischen dem eigentlichen OS/390-Betriebssystem und den Nutzer-Prozessen werden ähnlich wie z.B. bei Windows NT verschiedene Subsysteme eingeschoben. Letztere bilden im OS/390: Job Entry Subsystem (JES) für den Hintergrund-Betrieb (Stapelverarbeitung), Time Sharing Option (TSO) für den Vordergrund-Betrieb (interaktiv) und die Unix System Services (Posix kompatibles Unix-Subsystem). Zu diesen speziell-nutzerspezifischen kommen im OS/390 noch eine ganze Reihe anderer Subsysteme hinzu (Abbildung 3).

OS/390 stellt ein sehr komplexes Betriebssystem mit einer Vielzahl von Software-Komponenten dar. Neben Scheduling-, Dispatching- und Control-Funktionen verfügt OS/390 über mehr als 70 Software-Pakete. Dazu gehören u.a.:

- WebSphere Application Server
- Unix Services einschließlich Shell, Utilities und Debugger
- eNetwork Communication Services mit Unterstützung von TCP/IP und SNA/APPN
- Distributed Computing Services integriert Distributed Computing Environment (DCE), Network File System (NFS), Distributed File Service (DFS) und File Transfer Protocol (FTP)
- LAN und Print Services für Clients, um die OS/390-Ressourcen effizient zu nutzen und LAN-Server miteinander verbinden zu können
- Sicherheits-Infrastruktur mit Basis-Integration, um sicher zu stellen, dass alle Ressourcen geschützt werden
- Laufzeit-Sprach-Unterstützung für C/C++, COBOL, Object Orientiertes COBOL und PL/1
- C/C++ Open Class Library

Weitere optionale Komponenten beinhalten Sprach-Compiler für COBOL, PL/1, C/C++, Java Development Kit (JDK) und Java Virtual Machine (JVM).

Der OS/390-Security Server liefert neben dem Firewall und dem Lightweight Directory Access Protocol (LDAP) Unterstützung für Resource Access Control Facility (RACF) und DCE-Security.

Zwei Manager unterstützen Hochleistungs-Transaktionsverarbeitung:

- Customer Information Control System (CICS) Transaktions-Server
- Information Management System (IMS) Transaktions-Manager

Bezüglich der wichtigsten nutzerrelevanten Subsysteme existiert zwischen OS/390 und Windows NT ein wesentlicher Unterschied. Dieser besteht darin, dass im OS/390 diese Subsysteme vollkommen voneinander unabhängig sind, während im Betriebssystem NT die betreffenden Subsysteme im Windows 32-Subsystem integriert werden. Die Kapselung dieser Subsysteme bei der Firma Microsoft verursacht sowohl Performance- als auch Zuverlässigkeits-Probleme. Ein weiterer Unterschied zwischen den beiden betrachteten Betriebssystemen liegt in der Nutzung der System-Call-Schnittstelle, die es dem Systemprogrammierer erlaubt,

im Kernel-Status zu arbeiten. Im OS/390 besteht die Möglichkeit, über die sogenannte Supervisor-Call-Schnittstelle zusätzliche Kernelfunktionen zu programmieren. Die Firma Microsoft dagegen hält die System-Call-Schnittstelle streng geheim, so dass so gut wie keine Chancen bestehen, in den Kernel-Status zu gelangen.

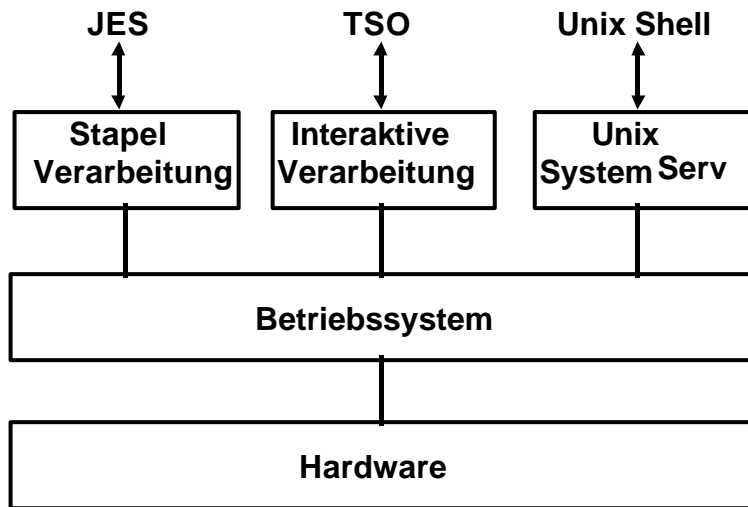


Abbildung 2: OS/390 Grundstruktur

Von den angegebenen Subsystemen hat JES seinen Ursprung in der Stapelverarbeitung der S/360-Architektur. Das System/360 war als reines Stapelverarbeitungs-Betriebssystem konzipiert worden und kannte keinerlei interaktive Arbeitsweise.

OS/390

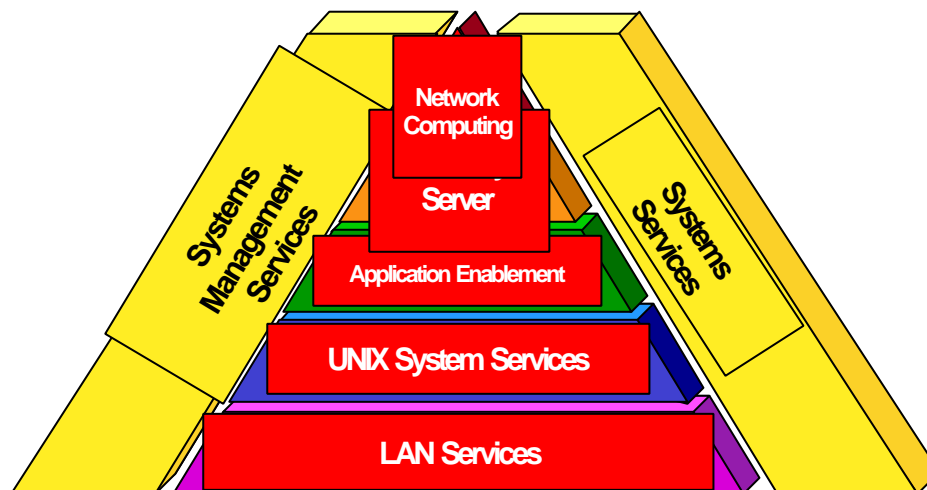


Abbildung 3: OS/390-Gesamt-Übersicht

Die sogenannten Jobs wurden als Lochkarten-Stapel über einen Lochkarten-Leser in die Zentraleinheit eingelesen und mehrere Jobs konnten nur rein seriell abgearbeitet werden. Jeder Job besteht aus einer bestimmten Anzahl von Steuerkarten, die in der Regel am Anfang und am Ende des Lochkarten-Stapels platziert werden. Dazwischen liegen dann die Daten-Karten (Anwender-Programm). Jede Programmzeile entspricht dabei einer Lochkarte (80 Spalten). Stapelverarbeitungs-Prozesse werden heute noch neben den interaktiven Prozessen verwendet. Erstere implementieren langläufige Prozesse von einigen Stunden bis zu mehreren Tagen. Ein Stapelverarbeitungs-Auftrag interagiert während seiner Ausführung nicht mit dem Nutzer. Der Job kann

während der Ausführung temporär unterbrochen und danach wieder ausgeführt werden, je nachdem, ob Aufträge mit höherer Priorität die vorhandenen Ressourcen dringender benötigen. Das Konzept eines OS/390-Jobs zeigt die Abbildung 4.

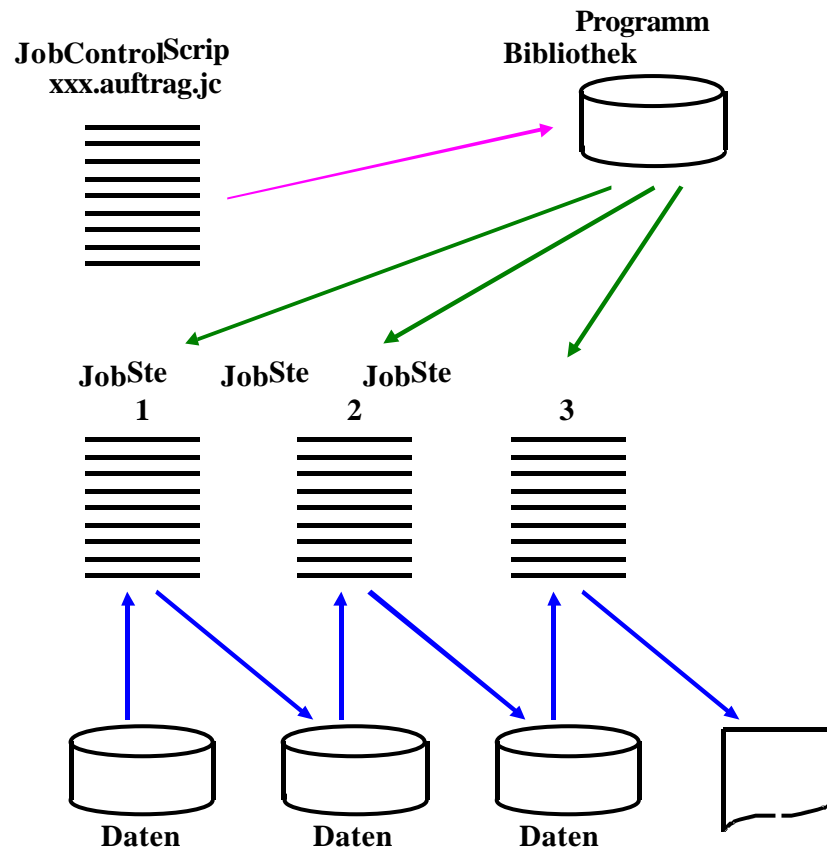


Abbildung 4: Konzept eines OS/390 Jobs

Ein OS/390-Job kann prinzipiell aus mehreren Job-Steps bestehen. Jeder Schritt verfügt über seine eigenen Control- und Daten-Anweisungen. Diese implementieren eine Menge von prozeduralen Befehlen und werden in der "Job Control Language" (JCL) erstellt. Diese Sprache hat ihren Ursprung in der Zeit der Lochkarten. Jeder JCL-Befehl muss immer in Großbuchstaben notiert werden, beginnt mit den beiden Zeichen "//" in der ersten Spalte und kann maximal bis Spalte 71 geschrieben werden. Ein Beispiel für einen JCL-Befehl lautet:

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

Die Ausdrücke RECFM, FB, LRECL, BLKSIZE sind Schlüsselworte der JCL-Sprache. Der obige Befehl sagt aus: Die hiermit angesprochene Datei (bzw. ihr Data Control Block, DCB) besitzt einen Fixed Block (FB), der über RECord ForMat (RECFM) zugewiesen wird. Alle Datensätze haben die gleiche Länge, mittels Logical RECord Length (LRECL) mit 80 Byte definiert. Weiterhin werden für die Übertragung vom/zum Hauptspeicher jeweils 5 Datensätze zu einem Block (BLKSIZE) von 400 Bytes zusammengefasst.

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

Für die Erstellung und Eingabe (Submission) eines Jobs sind folgende Schritte notwendig:

- Zuordnung einer Datei, die das Job Control-Programm enthalten soll
- Editieren und Abspeichern der JCL-Datei
- Submission

Da das JCL-Programm die verwendeten Dateien angibt, ist ein "Late Binding" der verwendeten Dateien an die auszuführenden Programme möglich.

Als Beispiel eines JCL-Jobs dient der folgende, der notwendig ist, um ein Sortierprogramm aufzurufen:

```
//ED1SORT JOB (EDUC,19-30), HPOTTER,NOTIFY=ED1,CLASS=A,REGION=4096K, *
//          MSGCLASS=H,USER=ED1
//SORTSTEP EXEC SORTV,SOUT=H
//SORTIN   DD DISP=SHR,DSN=ED1.SORTIN.DATA
//SORTOUT  DD DSN=ED1.SORTOUT.DATA,DISP=(,CATLG),VOL=SER=LST028,
//          SPACE=(TRK, (1,1)) ,UNIT=3390
//SYSIN    DD *
           SORT FIELDS=(20,3,CH,A)
```

Dabei werden die Daten aus dem Dataset ED1.SORTIN.DATA aufsteigend sortiert. Die sortierten Daten sollen in einen neu anzulegenden Dataset ED1.SORTOUT.DATA geschrieben werden. Der Sortierbegriff steht in jedem Record in der Spalte 20 mit der Länge 3.

Die zum Sortieren notwendige JCL wird als gegeben vorausgesetzt, d.h. sie befindet sich in einer definierten Prozedur SORTV. Neben dem EXEC-Record müssen dann mit den DD-Namen SORTIN und SORTOUT der Ein- und Ausgabe-Dataset angegeben werden. Die Sortierungs-Merkmale werden über SYSIN angegeben, sie stehen in keinem Dataset, sondern folgen als Record direkt dem SYSIN-Record.

Prinzipiell besteht ein JCL-Job aus 3 unterschiedlichen Statements: JOB-, EXEC- und DD-Statements.

JOB-Statements bezeichnen die auszuführende Arbeit, legen die Prioritäten fest, verlangen bestimmte Systemmittel und können den Nutzer angeben.

EXEC-Statements definieren das Verarbeitungsprogramm, fordern Systemmittel an, legen die Reihenfolge der Programmausführung fest und können einem Programm Parameter übergeben.

DD-Statements legen die Dateien fest und definieren die Datenträger, peripheren Einheiten, Dateimerkmale und Dateidisposition.

Das Time Sharing Option (TSO) implementiert ein interaktives Teilnehmersystem, das dem Nutzer die Möglichkeit bietet, in die Verarbeitung seines Prozesses einzugreifen. Es setzt auf das multiprogrammierte Betriebssystem auf und vermittelt dem Anwender während dessen Sitzung die Illusion einer alleinigen Benutzung der Zentraleinheit. In Wirklichkeit wird letztere nach einem spezifischen Algorithmus (z.B. Round Robin) für kurze Zeitabschnitte an jeweils einen TSO-Teilnehmer vergeben. Nach dem Prozess-Modell in der Abbildung ... kann ein Prozess unterschiedliche Zustände annehmen.

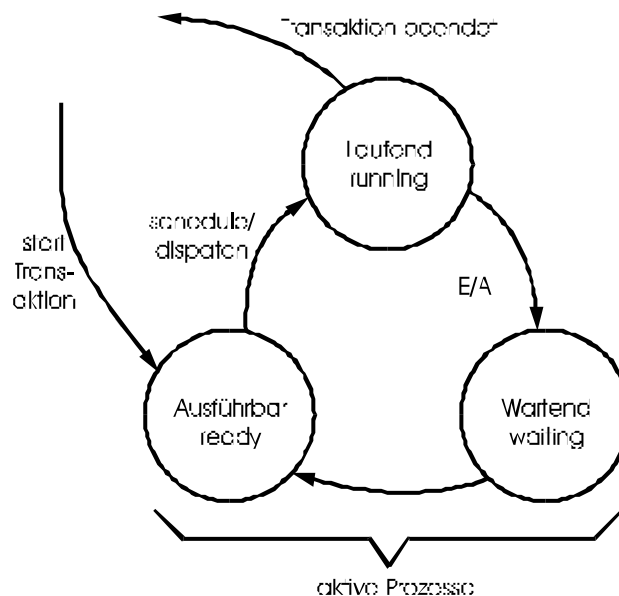


Abbildung 5: Prozess-Modell

Nach dem Eintritt des Nutzers in das TSO (erfolgreiches login) beginnt die Lebensdauer eines Prozesses. Nach einer bestimmten Zeit antwortet das System mit der Ausschrift "ready (bereit)" am Bildschirm. In diesem Moment befindet sich der Prozess im Zustand "ausführbar". Dieser Zustand geht in "laufend" über, wenn der Nutzer über ein entsprechendes TSO-Kommando (z.B Aufruf des Objekt-Moduls) sein übersetztes Programm ausführt. Diese Ausführung kann aber bei mehr als einem Nutzer des Systems endlich oft unterbrochen werden,

um evtl. den Prozess eines anderen TSO-Nutzers auszuführen. Ein laufender Prozess wird in den Zustand "wartend" gesetzt, wenn er z.B. Daten ein- oder ausgibt. In dieser Zeit erhält ein anderer Prozess die Zentraleinheit zugeteilt, d.h. dieser Prozess gelangt aus dem Zustand "ausführbar" in den Zustand "laufend". Die Lebensdauer eines Prozesses endet nach dem Abmelden des Nutzers aus dem System (erfolgreiches logoff).

Dem TSO ist ein weiteres System aufgesetzt, das die Produktivität des Nutzers im interaktiven Betrieb weiter erhöht. Der Anwender hat die Möglichkeit, aus dem TSO in das Interactive System Productivity Facility (ISPF) zu wechseln (s. Anhang). ISPF war das erste Produkt im TSO, das im sogenannten Full-Screen-Mode arbeitet und damit alle Möglichkeiten eines Bildschirm-Terminals ausnutzt. Dagegen arbeitet TSO immer noch im Line-Mode, d.h. zeilenweises Ein- und Ausgeben der Daten.

Ursprünglich gab es zwei zusammengehörige Produkte: ISPF und ISPF/Program Development Facility (PDF). Beide wurden ab der Version 4 zu einem Produkt ISPF zusammengefasst.

Die Hauptfunktionen im ISPF sind:

- Full-Screen-Editieren mit der Möglichkeit, mehrere Eingaben und Änderungen auf dem Bildschirm mit nur einer Kommunikation zum System durchzuführen
- Scrolling, d.h. verschieben des Bildschirms vor einer Liste oder einem Dataset in jeder Richtung
- Split-Screen erlaubt die Aufteilung des Bildschirms in zwei (oder mehr) voneinander unabhängige Teile
- Utilities, Funktionen zum Erstellen und Verarbeiten von Datasets
- Programmier-Unterstützung durch Aufruf von Compilern, Assembler, Linkage Editor usw.
- Unterstützung in der Entwicklung von Text-Dokumenten durch eine Verbindung zum Programm-Produkt-BookMaster
- Direkte Tutorial-Unterstützung zur Einführung, Referenz und für den Fehlerfall
- Browse- und Edit-Service, der von anderen Anwendungen aufgerufen werden kann

Die Arbeit im ISPF wird komplett Menu-geführt, d.h. alle Funktionen werden über Panels oder Menus aufgerufen und gesteuert. Der Nutzer gelangt durch Eingabe eines oder mehrerer der angebotenen Menu-Ziffern in das ausgewählte Submenu. Ausgangspunkt im ISPF bildet generell das Primary Option Menu (Abbildung 6).

```

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. . . : 21:00
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
-----
| Licensed Materials - Property of IBM      |
| 5647-A01 (C) Copyright IBM Corp. 1980, 1997. |
| All rights reserved.                    |
| US Government Users Restricted Rights -   | s
| Use, duplication or disclosure restricted |
| by GSA ADP Schedule Contract with IBM Corp. |
-----
Option ==> 3
-----
F1=Help      F3=Exit      F10=Actions  F12=Cancel
-----
23/015

```

Abbildung 6: Primary Option Menu im ISPF

Das MVS-Betriebssystem stellt eine spezielle Systemkomponente, die für die Steuerung und Ablaufkontrolle aller Jobs einschließlich aller TSO-Sitzungen, die aus der Sicht des Systems ebenfalls Jobs darstellen, zuständig

ist, zur Verfügung. Diese Systemkomponente heißt Job Entry Subsystem (JES). Es existieren davon zwei Varianten: JES2 und JES3.

Beide haben die gleiche Hauptaufgabe, sie unterscheiden sich dabei allerdings in einzelnen Funktionen. Es hängt wesentlich von der Gesamtkonzeption eines Rechenzentrums ab, welches dieser beiden Subsysteme eingesetzt wird. Beide Subsysteme sind in der Lage, die Jobverarbeitung auf einem einzelnen Rechner zu steuern. Der wesentliche Unterschied zwischen JES2 und JES3 liegt in der Kontrolle einer System-Konfiguration, die aus mehreren Rechnern besteht. Ein Einzel-Prozessor wie auch ein Multi-Prozessor (tightly coupled) wird in der Regel durch JES2 gesteuert. Bei einem Multi-Prozessor bilden zwei Rechner eine Einheit, und ein einzelner Job kann zeitweise auf dem einem und zeitweise auf dem zweiten Rechner laufen, ohne dass vom Benutzer oder Operator darauf Einfluss genommen werden kann. Generell wäre auch eine Steuerung durch JES3 möglich, aufgrund der höheren Komplexität wird aber darauf verzichtet. Die Mehrrechner-Architektur (Cluster, loosely coupled) steht unter Kontrolle von JES3. Mehrere Rechner arbeiten gemeinsam, und jeder Job wird von JES3 einem Rechner zur Verarbeitung zugeordnet.

Die Ablaufkontrolle aller Jobs durch JES gliedert sich in drei Phasen:

1. Preprocessing

- Lesen und Interpretieren der JCL-Records
- Bereitstellen der notwendigen Daten (z.B. durch Information an den Consol-Operator, eine Platte oder ein Band zu montieren). Bei evtl. Fehlern (z.B. Syntaxfehler in den JCL-Records oder Ansprechen von Datasets, die im System unbekannt sind) endet der Job bereits in dieser Phase mit einem JCL-Fehler.

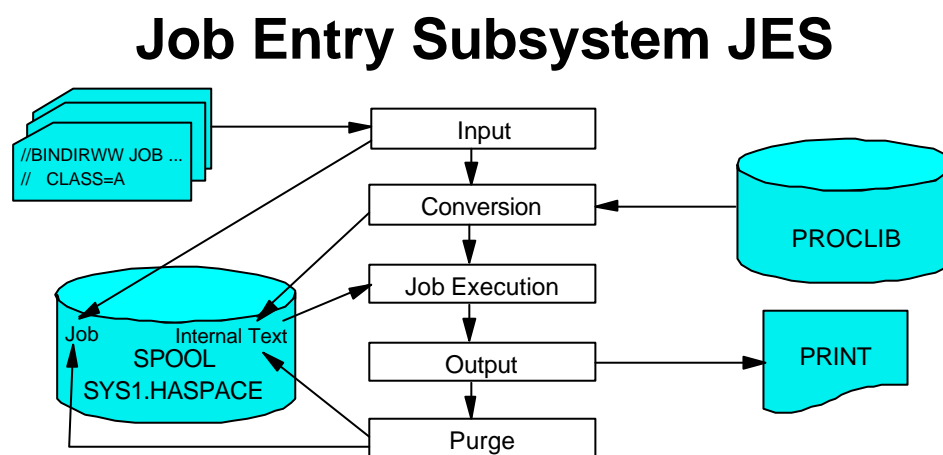


Abbildung 7: Arbeitsweise des Job Entry Subsystems (JES)

2. Processing

In dieser Phase werden die einzelnen Steps eines Jobs ausgeführt.

3. Postprocessing

Die durch den Job erstellten Ausgabe-Daten werden entweder gedruckt (entsprechend der Output-Klasse) oder dem TSO-Benutzer zur Weiterverarbeitung zur Verfügung gestellt. Datasets, die unter alleiniger Kontrolle des Jobs waren (DISP=NEW/OLD/MOD), werden für andere Jobs freigegeben. Nach Abschluss aller dieser Aktivitäten wird der Job durch die PURGE-Phase endgültig das System verlassen.

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

Der Benutzer, zu dem der laufende Job gehört, erhält ein Protokoll, dessen Format im JES2 sich von dem im JES3 unterscheidet. Zusätzlich kann der Systemprogrammierer die spezielle Protokoll-Struktur beeinflussen. In der Regel werden in dem Protokoll folgende Daten ausgegeben:

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

Die OS/390-Unix-System-Services bieten Unterstützung für zwei Open System Interfaces:

- Application Programming Interface (API)
- Interactive Shell Interface

Mit Hilfe des API sind Programme in einer beliebigen Umgebung lauffähig. Dazu zählen Batch-Jobs und Jobs, die durch einen TSO-Nutzer über das Submit-Kommando im Hintergrund abgearbeitet werden sowie andere MVS-Tasks. Eine gestartete MVS-Task in OS/390 hat Ähnlichkeit mit einem Unix-Deamon. Diese Programme können folgende Dienste in Anspruch nehmen:

- MVS-Services
- OS/390-Unix-System-Services
- MVS und OS/390-Unix-System-Services

Die Implementierung der Unix Services stellt keinen "Port" einer Unix-Implementierung dar, sondern bildet die Integration der IEEE Portable Operation System Interface-Funktion in den OS/390-Kernel. Diese System-Facility wird als "Open MVS" oder kurz "OMVS" bezeichnet.

Die Unix System Services Shell ist ein Kommando-Prozessor. Letzterer kann Shell-Kommandos oder Utilities aufrufen, die wieder spezielle Dienste vom System anfordern. Der Nutzer ist in der Lage, über die Shell-Programmiersprache Shell-Scripte zu generieren. Diese und C-Programme können sowohl im Vordergrund, im Hintergrund als auch im Batch-Betrieb abgearbeitet werden. Die Shell erlaubt einem Unix-Nutzer, sich im OS/390-System anzumelden (login) und Tasks genauso zu bilden wie in irgendeinem anderen Unix-System, d.h. sie kann als ein Unix-API auf dem OS/390-Betriebssystem mit integrierten POSIX-Diensten betrachtet werden. Für den ISPF-Nutzer wird nach der Eingabe des Kommandos "TSO OMVS" auf der Kommando-Zeile die Shell initialisiert. Nach Erscheinen des Shell-Prompts (#) können alle vom Unix her bekannten Kommandos verwendet werden.

Anstatt TSO- oder Shell-Kommandos kann auch die OpenMVS-ISPF-Shell oder IShell benutzt werden. Letztere stellt ein ISPF Panel-Interface dar, das folgende Tasks ausführen kann:

- Anzeige aller verbundenen File-Systeme
- Anzeige der Attribute eines verbundenen File-Systems (Anzahl der Blöcke, benutzte Blöcke, DDNAME)

Im Superuser-Status oder mit RACF-SPECIAL-Attribut oder beiden können über die ISPF-Shell weitere Tasks gestartet werden. Dazu gehören u.a.:

- Setup des Root-File-Systems
- Erzeugen von speziellen Character-Files
- Anhängen (mount) eines File-Systems
- Abhängen (unmount) eines File-Systems
- Reset eines Pending Unmount
- Ändern der Attribute eines OS/390-Unix-Nutzer
- Anzeigen der Nutzer und Sortieren nach den Namen, UID, GID
- Drucken der Nutzer-Liste
- Setup der OS/390-Unix-Nutzer
- Setup der OS/390-Unix-Gruppen

Um in die IShell zu gelangen, wird im ISPF Primary Option Menu die Option 6 ausgewählt. Damit erscheint die ISPF Command Shell. Durch geeignetes Platzieren des Cursors und mehrfaches Betätigen (3x) der Enter-Taste wird die Directory List erreicht. Von hier aus können verschiedene Tasks im Hierarchical File System (HFS)

gestartet werden (z.B. list, browse, edit, copy, delete von Files und Directories). Das HFS unterstützt eine Directory-Struktur, die analog ist zu anderen Unix-Plattformen. Es wird in sogenannten Containern implementiert. Diese stellen ganz normale MVS-Datasets dar. Ein HFS-Container besitzt einen MVS-Dataset-Namen mit einem DSNTYPE des HFS. Der Container-Dataset kann mit Hilfe von traditionellen DFSMS/MVS-Storage-Management Tools (DFHSM oder DFDSS) bearbeitet werden. Die Entscheidung, welche Teile des gesamten HFS in welchen Containern implementiert werden, liegt beim Nutzer. Letzterer kann das ganze HFS in einem Container unterbringen oder Teile davon über mehrere Container verteilen. Die Implementierung des HFS ist transparent zu dem zu speichernden Datentyp (Text-, Binär-Daten).

3.4 Speicherverwaltung unter OS/390

Die S/390-Architektur verfügt über 3 verschiedene Speicher-Ebenen, die vom Betriebssystem verwaltet werden: Hauptspeicher (maximal 2 Gbyte), Erweiterungsspeicher (maximal 16 Tbyte), Page Data Sets auf den Plattenspeichern (Abbildung 8).

Der reale Hauptspeicher im Betriebssystem OS/390 wird in Page Frames (Rahmen) zu je 4096 Bytes aufgeteilt. Programme im OS/390 arbeiten nicht mit realen sondern mit virtuellen Hauptspeicher-Adressen. Das Betriebssystem verwaltet den Hauptspeicher nach dem Demand-Prinzip: Der Real Storage Manager weist die Rahmen von der Available Frame Queue (AFQ) dann zu, wenn diese gebraucht werden. Die Zuweisung erfolgt aber nur dann, wenn die Länge der AFQ unter einen bestimmten Schwellenwert abfällt oder der Bedarf an Rahmen nicht durch die aktuelle AFQ abgedeckt werden kann.

In der S/370-Architektur war die obere Grenze sowohl für die virtuellen als auch für die realen Hauptspeicher-Adressen auf 16 MByte begrenzt. Diese Schwierigkeit wurde mit dem "Extended Addressing" Feature gelöst, indem ein reserviertes Bit im Eintrag der Adress-Übersetzungstabelle benutzt wird, um die Größe des realen Hauptspeichers auf das Doppelte (32 MByte) zu erhöhen. Im System/370 Extended Architecture und Enterprise System Architecture (entsprechend MVS/XA und MVS/ESA) wurde der reale Hauptspeicher auf 2 Gbyte erweitert, d.h. es wird mit 31 Bit-Adressen gearbeitet. Die neuen Z-Architekturen verfügen dagegen schon über 63 Bit-Adressen. Die XA-Kompatibilität erfordert, dass S/370-Channel-Programme unverändert im XA-Modus laufen. S/360- und S/370-Channel Command Words (CCWs) sind aber auf reale 24 Bit-Adressen begrenzt. XA, ESA und OS/390 verfügen aus diesem Grund über zwei unterschiedliche CCW-Formate: Format 0 als das ursprüngliche 24 Bit-CCW und das Format 1 als die neue 31 Bit-Version.

OS/390 Memory Management

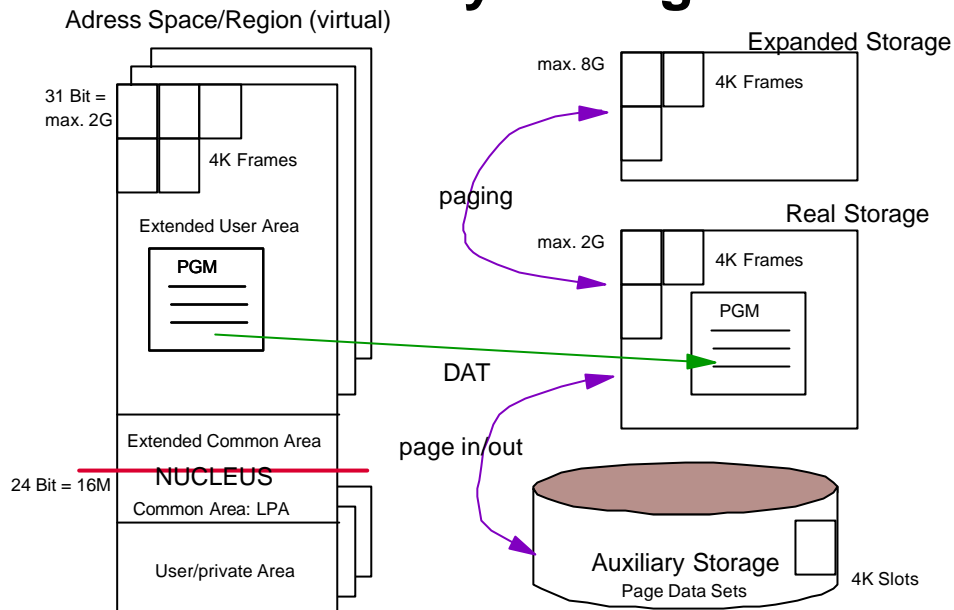


Abbildung 8: Speicherverwaltung unter OS/390

Eine Methode zur Unterbringung nichtaktiver Seiten, die ohne Reaktivierungs-Verzögerung des zusätzlichen Speichers oder Zusatzkosten des realen Speichers auskommt, wurde für die 3090-Produkt-Linie vorgeschlagen und in der CPU der Enterprise System Architektur implementiert. Diese Methode heißt Expanded Storage (ES). In Systemen mit ES wird der Hauptspeicher in zwei Teile geteilt: Central Storage und Expanded Storage. Central Storage bezeichnet den realen Hauptspeicher. Expanded Storage ist dem Ein-/Ausgabe-Subsystem oder den normalen CPU-Befehlen nicht zugänglich und nur in Einheiten von Pages (nicht Byte-weise), d.h. 4k-weise adressierbar. Der Expanded Storage hat gegenüber dem Central Storage einen Vorteil, der darin besteht, dass der ES kostengünstiger ist als der eigentliche Hauptspeicher, da nur große Blöcke adressiert werden und dadurch ein Teil der Adressierungslogik eingespart wird. Praktische Untersuchungen haben weiterhin gezeigt, dass beim Einsatz gleich großer Central Storage- bzw. Expanded Storage-Erweiterung in der /390-Architektur sich die Performance bei der ES- gegenüber der CS-Erweiterung verbessert.

In dem Maß, wie sich der Bedarf an Hauptspeicher erhöht, verwalten die Memory Management Algorithmen den Hauptspeicherplatz entsprechend der Workload-Priorität und dem momentanen Memory Reference-Schema, d.h. Seiten, auf die eine bestimmte Zeit nicht zugegriffen wurde, oder deren Speicherplatz für höher priorisierte Tasks (Prozesse) benötigt wird, werden zum ES oder zu Page Datasets ausgelagert.

OS/390 liefert zwei Erweiterungen zum normalen Paging. Wenn eine Menge von Seiten ausgelagert werden soll, so werden diese zusammengefasst und als Block herausgeschrieben (Block Paging). Das Betriebssystem erkennt, ob sich die Arbeit eines TSO-Nutzers oder eines Batch-Jobs im Wartezustand befindet. In diesem Fall erfolgt das Auslagern aller dazugehörigen Seiten innerhalb einer Operation.

3.5 OS/390-Sicherheit

Die Sicherheit ist ein integraler Bestandteil des OS/390-Betriebssystems. Alle Ressourcen sind standardmäßig geschützt. Die Sicherheits-Administration besteht vorrangig in der Identifikation der autorisierten Nutzer oder Nutzer-Gruppen und der Zugriffserlaubnis zu den Ressourcen.

Der OS/390-Sicherheits-Server hat folgende Aufgaben:

- Identifiziert Nutzer und gibt ihnen ein eindeutiges Sicherheits-Profil, das die von ihnen autorisiert benutzten Ressourcen einschließt

- Nutzer-Bestätigung, die mit Hilfe von Kennworten, PassTickets, DCE-Berechtigungen oder digitalen Zertifikaten realisiert werden
- Berechtigt Nutzer durch Erteilen einer angemessenen Authorisierungsstufe, auf geschützte Ressourcen zuzugreifen
- Protokollieren des Zugriffs zu geschützten Ressourcen einschließlich Meldungen von unberechtigten Zugriffsversuchen

Wenn sich der Nutzer beim System anmeldet, wird ein ACcess Environment Element (ACEE) erzeugt. Letzteres beschreibt den Nutzer mit eindeutigem ID, die gültige Gruppe, Nutzer-Attribute und Gruppen-Authorisierung. Dieses Element begleitet den Nutzer durch das gesamte System und stellt sicher, dass die Arbeit im Interesse des Nutzers auf seinem "Konto" verwahrt wird.

Es existieren 4 unterschiedliche Sicherheits-Mangement-Facilities unter OS/390:

- Resource Access Control Facility (RACF)
- DCE Security Server
- Firewall-Architektur
- Lightweight Directory Access Protocol Server (LDAP-Server)

Alle Sicherheits-Anforderungen passieren den OS/390-Operationssystem-Sicherheits-Service, bekannt als System Authorization Facility (SAF). Der Aufruf kann je nach Art der Implementierung direkt an RACF oder einen Nutzer-spezifischen Security-Manager oder an beide gerichtet werden.

Die Resource Access Control Facility (RACF) implementiert eine Komponente des OS/390-Sicherheits-Servers. Sie überprüft Benutzer-Identitäten, kontrolliert Nutzeranforderungen bezüglich der Ressourcen-Zugriffe, protokolliert diese Zugriffsanforderungen und liefert dem Administrator eine Schnittstelle für den Zugriff auf den Inhalt der RACF-Datenbank, d.h. RACF prüft sämtliche Aktivitäten eines Anwenders unter dem OS/390-Betriebssystem.

RACF bietet optimale Sicherheit durch:

- Flexible Zugriffs-Steuerung auf geschützte Ressourcen
- Schutz von Installations-definierten Ressourcen
- Fähigkeit, Informationen für andere Produkte zu speichern
- Auswahl von zentralisierter oder dezentralisierter Profile-Steuerung
- ISPF-Panel-Interface
- Transparenz für die End-Nutzer
- Ausgänge für Installations-Routinen

Die Abbildung 9 zeigt die Wechselwirkung von RACF mit dem Operationssystem bei Zugriffen auf geschützte Ressourcen. Das Betriebssystem interagiert mit dem RACF auf ähnliche Art bei der Nutzer-Identifikation und -Überprüfung.

Während des Authorisierungs-Checks prüft RACF das Ressourcen-Profil, um sicherzustellen, dass auf die Ressource in der geforderten Art und Weise zugegriffen werden kann und der Nutzer die geeignete Berechtigung für den Ressourcen-Zugriff besitzt. Die notwendigen Benutzer-/Ressourcen-Bedingungen müssen übereinstimmen, bevor der Ressourcen-Manager den Zugriff auf eine geschützte Ressource erlaubt. Folgende Schritte sind erforderlich:

1. Ein Benutzer fordert den Zugriff auf eine Ressource mit Hilfe eines Ressourcen-Managers.
2. Der Ressourcen-Manager gibt eine RACROUTE-Anforderung aus, um zu prüfen, ob der Nutzer auf die Ressource zugreifen darf.
3. RACF konsultiert die RACF-Datenbank und durchsucht die eingespeicherten Daten.
4. RACF erhält die Daten für das Profile zurück.
5. Aufgrund der Information im dem Profil übermittelt RACF den resultierenden Statuscode für die Anforderung an den Resource-Manager.
6. Der Ressourcen-Manager gewährt oder verbietet die Anforderung.

RACF

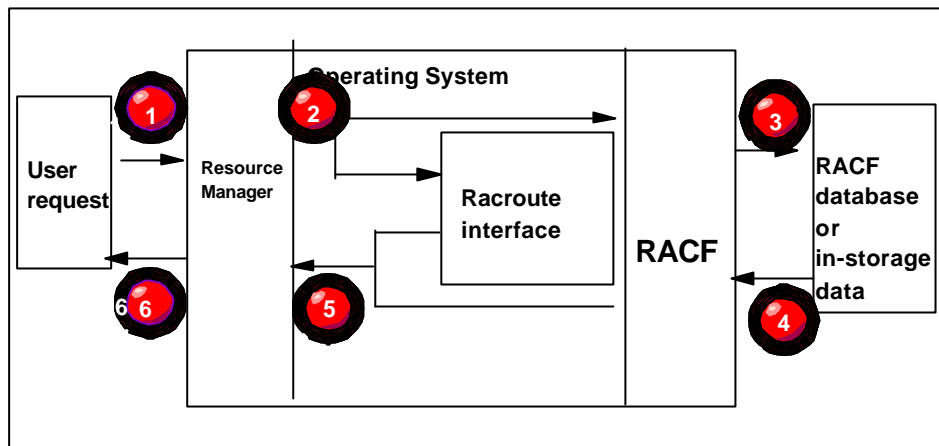


Abbildung 9: Wechselwirkung zwischen RACF und Betriebssystem

Nicht RACF entscheidet über die Nutzer-Anforderung sondern sie gibt an den Ressource-Manager den Status-Code zurück, und der Ressource-Manager trifft die Entscheidung. RACF generiert einen von 4 möglichen Status-Codes: "Has the right", "No access", "Don't know" und "Not working".

Eine weitere Sicherheits-Komponente des Betriebssystems bildet die System Authorization Facility (SAF). Letztere leitet die Steuerung, wenn eine Anfrage vom Ressource-Manager eintrifft, entweder an RACF oder an eine Nutzer-spezifische Verarbeitungs-Routine oder an beide weiter, je nachdem, ob RACF installiert ist oder nicht. RACF und SAF zusammen verbessern bzw. ergänzen die Sicherheitsfunktionen von OS/390 erheblich. Das Schlüsselement in SAF ist der SAF-Router. Dieser ist immer im Betriebssystem aktiv, selbst dann, wenn RACF nicht implementiert ist. Der SAF-Router stellt einen Systemdienst dar und beteiligt sich u.a. an der Ressourcen-Steuerung. Die Komponenten des Ressource-Managers und Subsysteme rufen den Router als Teil bestimmter Entscheidungs-trächtiger Funktionen in ihrer Verarbeitung auf wie z.B. Check von Zugriffs-Steuerung und Autorisierung. Diese Funktionen heißen "Control-Points".

RACF-Verwaltungs-Funktionen verfügen über ISPF-Entry-Panels und damit verbundene Help-Panels. Diese erleichtern die Eingabe von RACF-Kommandos und ihrer Optionen.

RACF identifiziert jeden Nutzer, wenn dieser sich beim System anmeldet. Dabei wird eine Nutzer-Identifikation angefordert. RACF stellt anschließend durch die Anforderung des Kennwortes und dessen Überprüfung sicher, daß der Nutzer tatsächlich autorisiert ist. Jede RACF-Nutzer-ID hat ein eindeutiges Kennwort. Es existieren auch Alternativen zu Kennwörtern, die von RACF verwendet werden können, um Benutzer zu identifizieren. RACF gestattet es z.B., in Client-Maschinen einer Client-Server-Umgebung ein PassTicket statt eines Kennwortes zu verwenden. Ein PassTicket kann von RACF oder einer anderen autorisierten Funktion generiert werden. Es kann auch eine Operator Identification Card (OIDCARD) anstatt oder zusätzlich zu dem Kennwort während einer Terminal-Sitzung benutzt werden. OS/390-Unix-Benutzer werden auch mit numerischen Nutzer-IDs (UIDs) identifiziert. Die Identifikation von OS/390-Unix-Gruppen erfolgt mittels numerischer Gruppen-IDs (GIDs). In einer Client/Server-Umgebung kann RACF eine RACF-Benutzer-ID durch Extrahieren der Information aus dem digitalen Zertifikat identifizieren. Ein digitales Zertifikat oder digitale ID enthält Informationen, die den Client eindeutig identifizieren. Der Lotus Domino Go Webserver bestätigt einen Client mit Hilfe des Client-Zertifikats und des Secure Socket Layer (SSL)-Protokolls. Der Domino Go-Webserver überträgt das digitale Zertifikat des Client zur Bestätigung zum OS/390-Unix. OS/390-Unix leitet es weiter zu RACF. Dies bedeutet, daß RACF-Nutzer-ID und Kennwort von jedem Client nicht verwendet werden müssen, um auf geschützte Web-Seiten zuzugreifen.

RACF integriert folgende Komponenten:

- RACF-Datenbank
- RACF-User- und Group Management-Konzepte
- RACF-User-Attribute
- RACF-Segmente
- Verwalter-RACF-Gruppen

Alle Informationen über Mainframe-Nutzer, -Gruppen, -Dateien und andere Ressourcen sind in der RACF-Datenbank abgespeichert. Die Records in der Datenbank, die all diese Objekte beschreiben, werden Profiles genannt. Ein Ressourcen-Profil, das verwendet wird, um einzelne Ressourcen zu schützen, heißt "diskretes" Profil, und ein Profil, das mehrfache Ressourcen durch wild-cards schützt, stellt ein "generisches" Profil dar. Ein RACF-Benutzer-Profil ist in Segmente eingeteilt. Jedes Segment enthält Daten für die OS/390-Nutzer-Administration wie z.B. die Unix-System-Service- oder die TSO-Nutzer-Information.

Profiles, die RACF-geschützte Ressourcen beschreiben, verfügen auch über eine Zugriffsliste, die aussagt, welche Nutzer-IDs, und welche Gruppen das Recht haben, auf die Ressourcen zuzugreifen. Die Zugriffsliste enthält auch die Information, auf welcher Ebene der Zugriff erlaubt ist.

Die Nutzer-Administration basiert auf einem Sicherheits-Grundsatz. Dieser legt fest, welche Ressourcen geschützt werden sollen, wer für diese Ressourcen verantwortlich ist, und wie die Organisation, die sich um die Sicherheit kümmert, aussehen soll.

Die Nutzer-Profiles eines gegebenen OS/390-Systems enthalten neben den normalen Nutzern Manager und Administratoren. Im RACF existiert eine bestimmte Anzahl von Attributen, die den Nutzern Sonderrechte bezüglich des Zugriffs auf Ressourcen einerseits und auf die RACF-Datenbank andererseits einräumt.

Die vier wichtigsten Attribute sind:

- **Special:** Es bedeutet, dass der Benutzer ein RACF-Administrator ist und das Recht hat, alle RACF-Kommandos zu verwenden und jede Profile-Art in der RACF-Datenbank zu definieren. Das Special-Attribut erlaubt aber nicht, auf alle Ressourcen im System ähnlich dem Unix-Root-Nutzer zuzugreifen.
- **Operations:** Es bedeutet, daß der Benutzer auf alle Dateien zugreifen kann sowie auch auf Ressourcen, die in zusätzlichen Ressource-Klassen im System definiert sind. Mit diesem Attribut können auch Dateien irgendeinem anderen Nutzer im System zugeordnet werden.
- **Auditor:** Es ist verantwortlich für das Prüfen der RACF-Datenbank und des Systems. Das Auditor-Attribut gibt einem Benutzer das Recht, sich alle Profiles in der RACF- Datenbank anzusehen und die Auditor-Attribute sowohl für das System als auch für individuelle Profiles zu ändern.
- **Revoke:** Das Revoke-Attribut bietet die Möglichkeit, allen RACF-definierten Nutzern den Zugang zum Rechnersystem zu verweigern.

RACF-Gruppen können für einen unterschiedlichen Zweck verwendet werden. Es wird zwischen drei wesentlichen Gruppentypen unterschieden:

- **Ressource Protection Groups:** Sie sind notwendig, wenn es darum geht, Dateien zu schützen. Es gibt zwei Arten von Dateien: Nutzer-Dateien und Gruppen-Dateien.
- **Administrative Groups:** Sie können für Informationszwecke verwendet werden. Solche Gruppen werden benutzt, um eine Struktur aufzubauen, die eine Firmen-Organisation mit Betriebsteil, Abteilungen usw. emuliert. Die Benutzer im Betriebsteil bzw. in der Abteilung werden an eine entsprechende Gruppe angeschlossen.
- **Functional Groups:** Sie representieren Gruppen, die z.B. Stellen oder Verantwortlichkeiten in einer Firma darstellen.

In einem MVS-Sysplex mit mehreren Systemen, die die RACF-Datenbank gemeinsam benutzen, können Probleme in den Bereichen Systemleistung, Verwaltung und Verfügbarkeit auftreten. Gemeinsam benutzte RACF-Sysplex-Daten adressieren diese Probleme mit:

- **Sysplex Command Propagation:** Wenn ein Kommando auf einem System eingegeben wird, verteilt RACF das Kommando über den gesamten Sysplex.
- **Coupling Facility:** Gemeinsam benutzte RACF-Sysplex-Daten werden von der Coupling Facility benutzt, um die Leistung zu verbessern. Wenn sich das System im Data-Sharing-Modus befindet, liefert die Coupling Facility einen großen zentralen Puffer für RACF-Datenbank-Records. Dieser Puffer kann mehr Daten aus der Datenbank jedes Systems abspeichern als der eigene lokale Puffer des zugehörigen Systems und erlaubt, die gepufferten Informationen gemeinsam zu benutzen.

Die **RACF Remote Sharing Facility (RRSF)** ist in der Lage, RACF-Datenbanken, die überall in einem Unternehmen verteilt sind, zu verwalten und zu warten. RRSF stellt sicher, daß die Datenintegrität bei System- oder Netzausfall erhalten bleibt. Das RRSF-Netzwerk besteht aus einer Menge von Knoten. Jeder Knoten setzt sich aus einer oder mehreren MVS-Systemabbildungen zusammen und benutzt eine spezifische RACF-Datenbank (Abbildung 10).

Die RRSF-Umgebung gestattet es, die Remote-Systeme zu verwalten. Mit Hilfe von RRSF können unterschiedliche Funktionen gebildet werden wie z.B. Password-Synchronisation, Command-Direction und Automatic-Password-Direction.

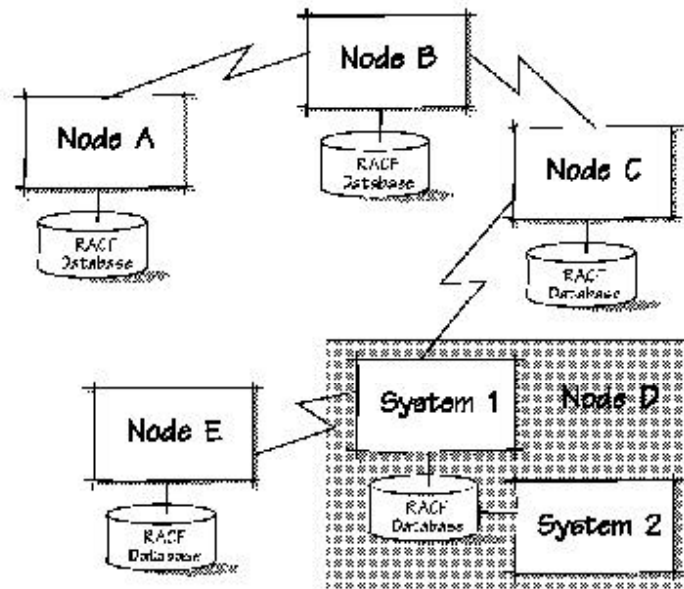


Abbildung 10: Verteilte RACF-Datenbanken