UNIVERSITÄT LEIPZIG

Enterprise Computing
Einführung in das Betriebssystem z/OS

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

WS 2012/13

z/OS Betriebssystem Teil 4

z/OS Subsysteme

z/OS Subsysteme

Einige der wichtigsten z/OS Subsysteme sind:

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

Einige dieser Subsysteme werden wir uns näher anschauen.

Prozessverwaltung

Unix System Services

Datenbanken

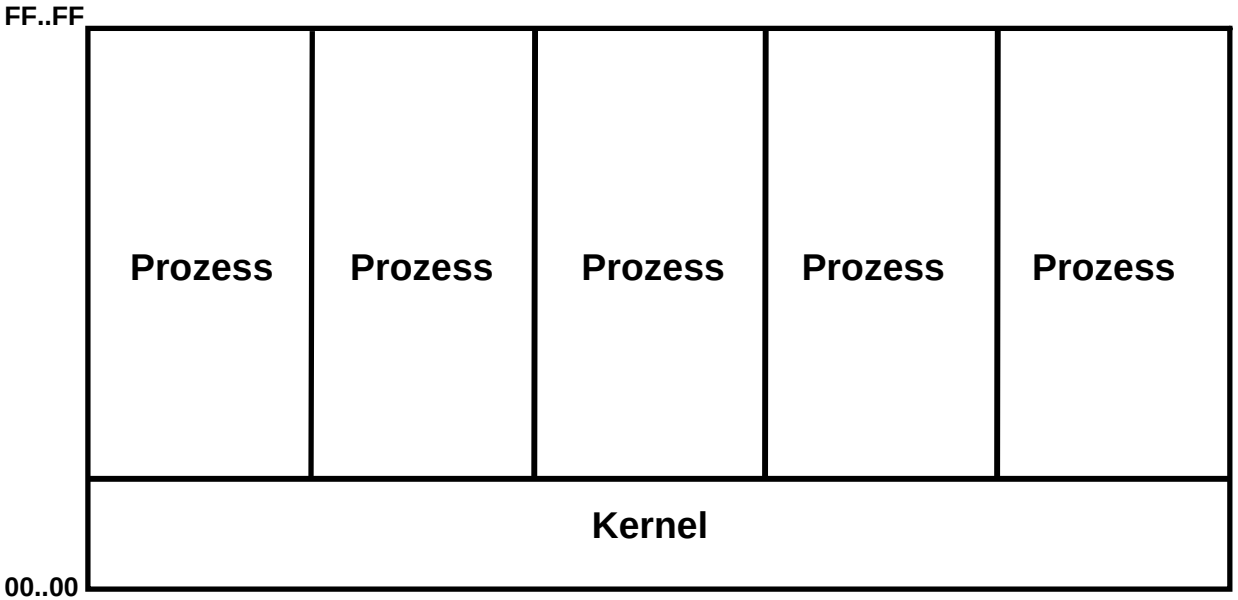
Communication Server

z/OS Prozessverwaltung

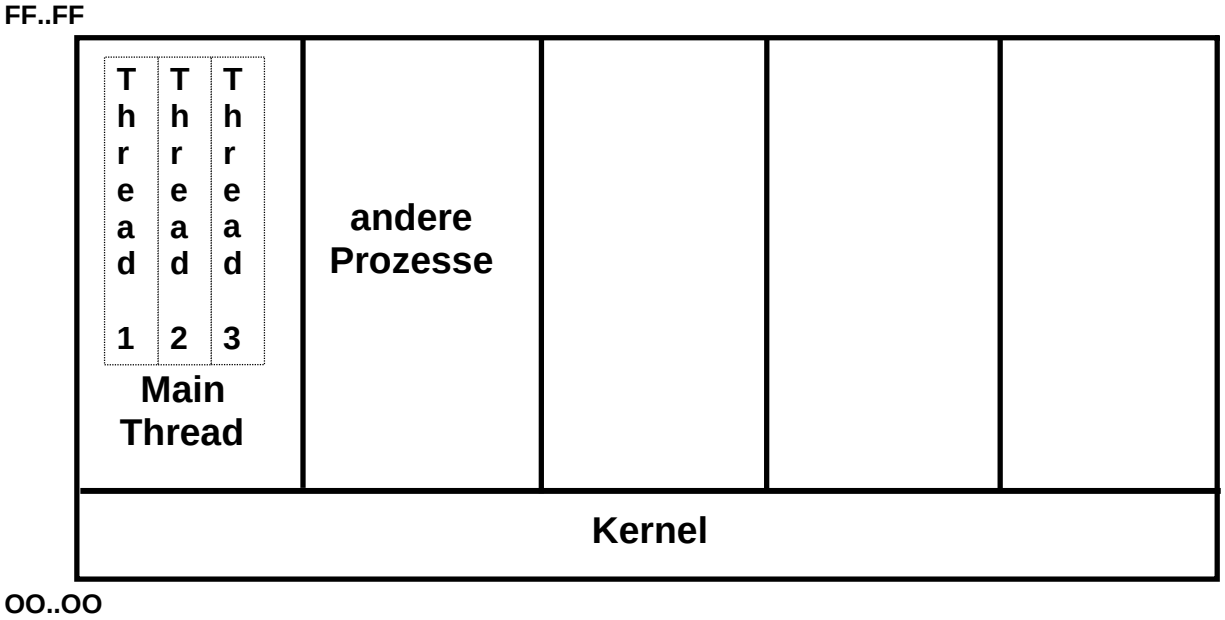
Ein z/OS Prozess besteht aus mehreren Arbeitseinheiten, die als „Tasks“ bezeichnet werden. Eine „Main Task“ repräsentiert einen Prozess und typischerweise einen Address Space. Eine Main Task kann Subtasks generieren; eine Subtask entspricht in etwa einem Thread in Unix oder Windows. Subtasks laufen im gleichen Address Space wie die Main Task.

Wenn ein Prozess gestartet wird, erstellt z/OS hierfür einen Main Task Control Block (TCB). Ein TCB entspricht in etwa einem Unix Process Control Block (PCB).Das Programm kann weitere Subtasks erstellen mit Hilfe des ATTACH System Aufrufes. Hierbei wird jeweils ein eigener Subtask TCB erzeugt. Da der ATTACH Overhead relativ groß ist, implementieren zeitkritische Subsysteme wie z.B. CICS ihr eigenes Subtasking. Bei der Nutzung der Unix System Services (siehe weiter unten) wird empfohlen, die POSIX Funktion `pthread_create` an Stelle der z/OS ATTACH Makro zu benutzen.

Im Kernelmodus (Supervisor State) existieren zusätzliche Mechanismen, die als Service Request Blocks (SRB) bezeichnet werden. SRBs werden benutzt, um Systemroutinen auszuführen. Windows kennt im Kernelmodus einen ähnlichen Mechanismus, der als „Fibers“ (light weight threads) bezeichnet wird. Fibres werden durch die Anwendung gescheduled. Die CICS Portierung auf Windows benutzt Fibers.



Prozess Ansatz



Thread Ansatz

z/OS Prozessverwaltung

Bei der Ausführung von Programmen unterscheiden wir zwischen Prozessen und Threads.

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

Prozessverwaltung

Unix System Services

Datenbanken

Communication Server

Supervisor Calls

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

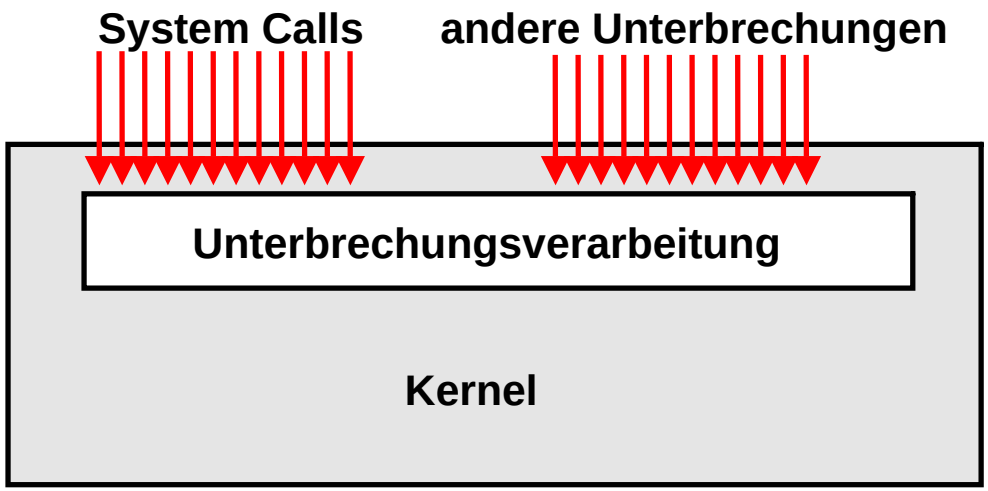
Supervisor Calls im weiteren Sinne sind Library Routinen, die eine Dienstleistung des Kernels in Anspruch nehmen.

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

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

Beispiele sind:

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



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

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

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

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

Der Kernel eines Betriebssystems wird fast ausschließlich über Unterbrechungen aufgerufen.

Unix System Services (USS)

Was ist ein Unix Betriebssystem ?

is a UNIX environment that runs in an *z/OS* environment.

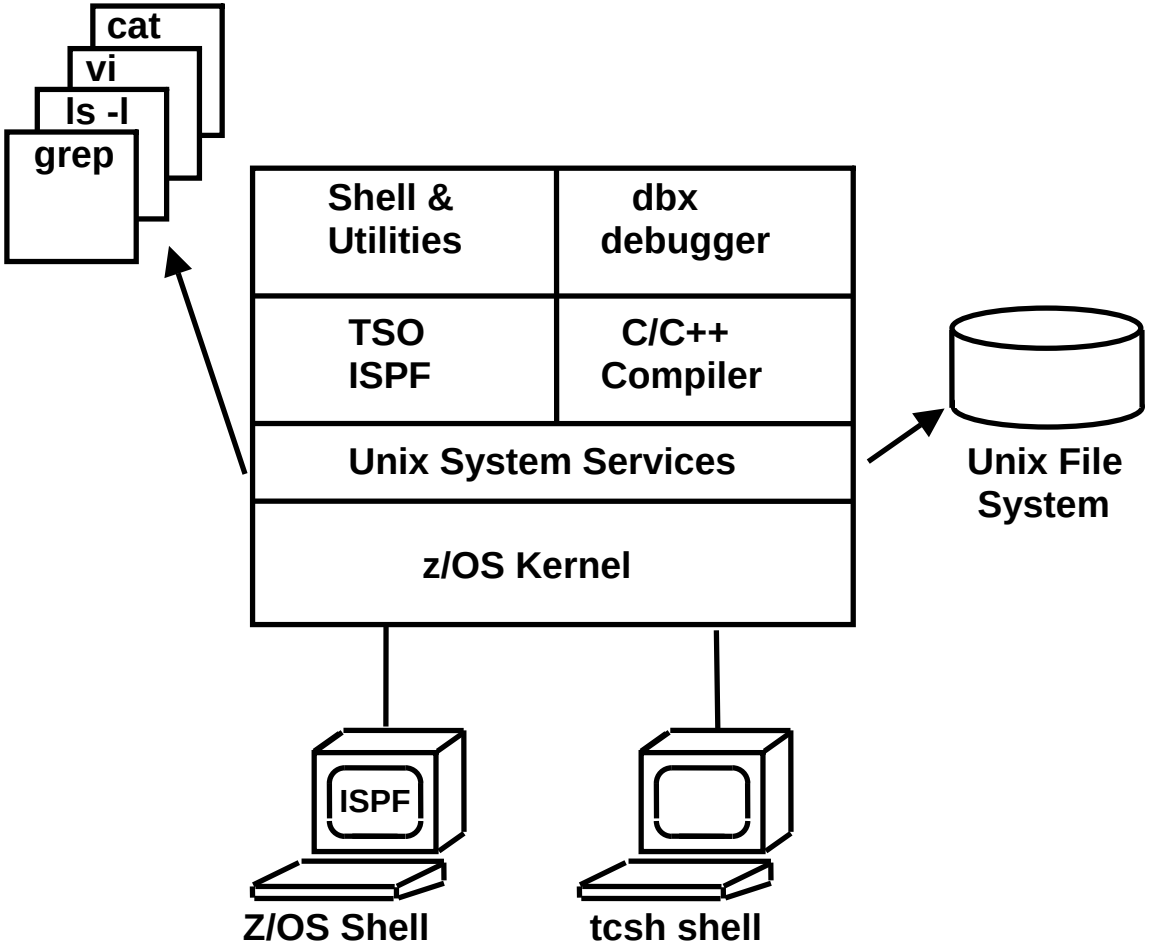
Ein Unix Betriebssystem erfüllt den „POSiX“ Standard. Das Portable Operating System Interface ist ein gemeinsam von der IEEE und der Open Group für Unix entwickeltes standardisiertes Application Programming Interface, das die Schnittstelle zwischen Applikation und dem Betriebssystem darstellt. Der Standard beinhaltet außerdem Spezifikationen für eine Command Line Interface, Services wie basic I/O (File, Terminal und Netzwerk), sowie eine threading Library API.

Die wichtigsten Implementierungen sind AIX, HP-UX, Solaris und Mac OS X

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

Damit wird aus z/OS ein Unix Betriebssystem. Dies ermöglicht eine einfache Portierung von Unix Anwendungen nach z/OS. Ein typisches Beispiel ist das SAP R/3 System, welches ursprünglich für das Unix Betriebs system entwickelt wurde, aber auch unter z/OS Unix System Services lauffähig ist.

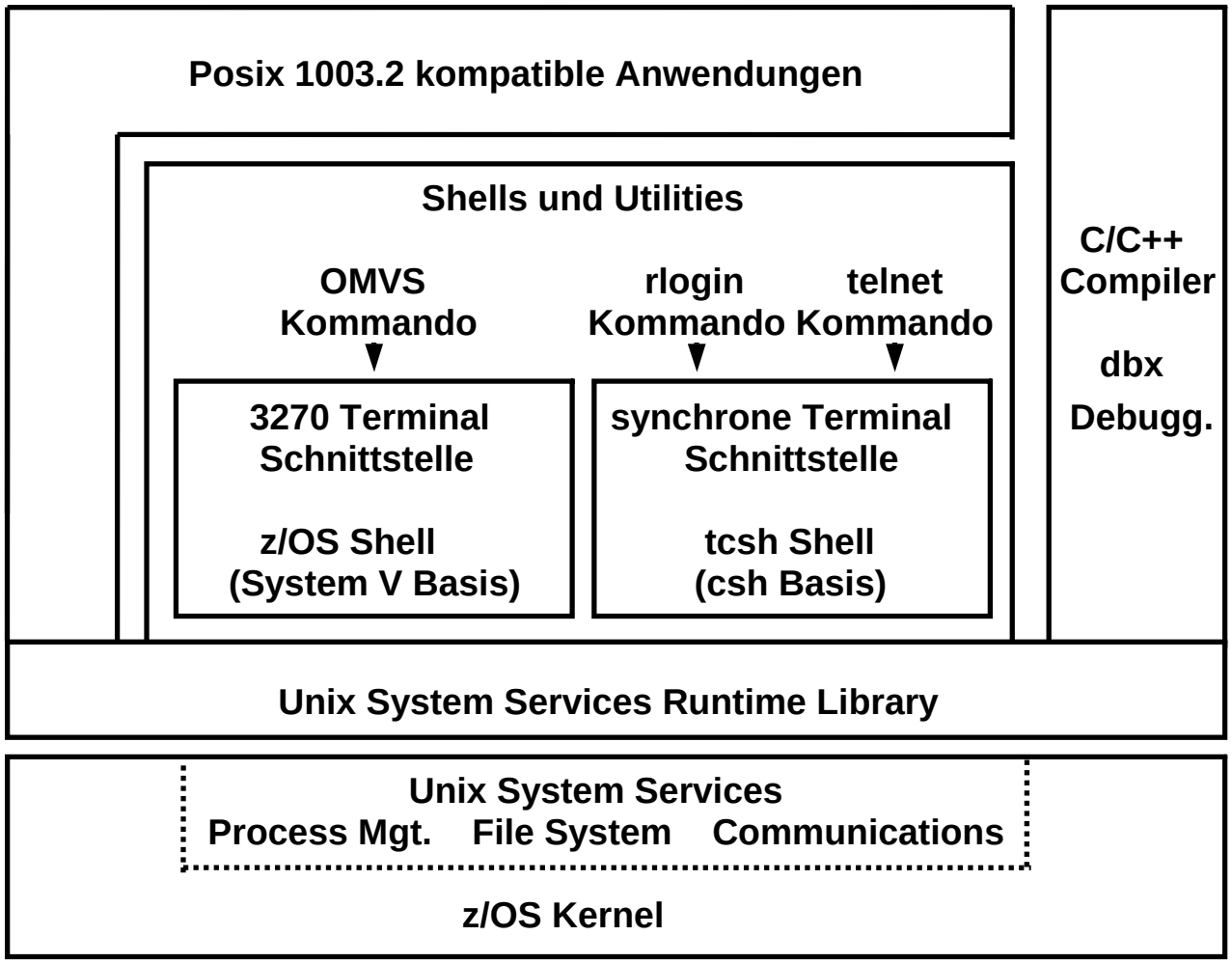
z/OS Unix System Services wurde früher als Open Edition MVS bezeichnet.



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

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

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



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

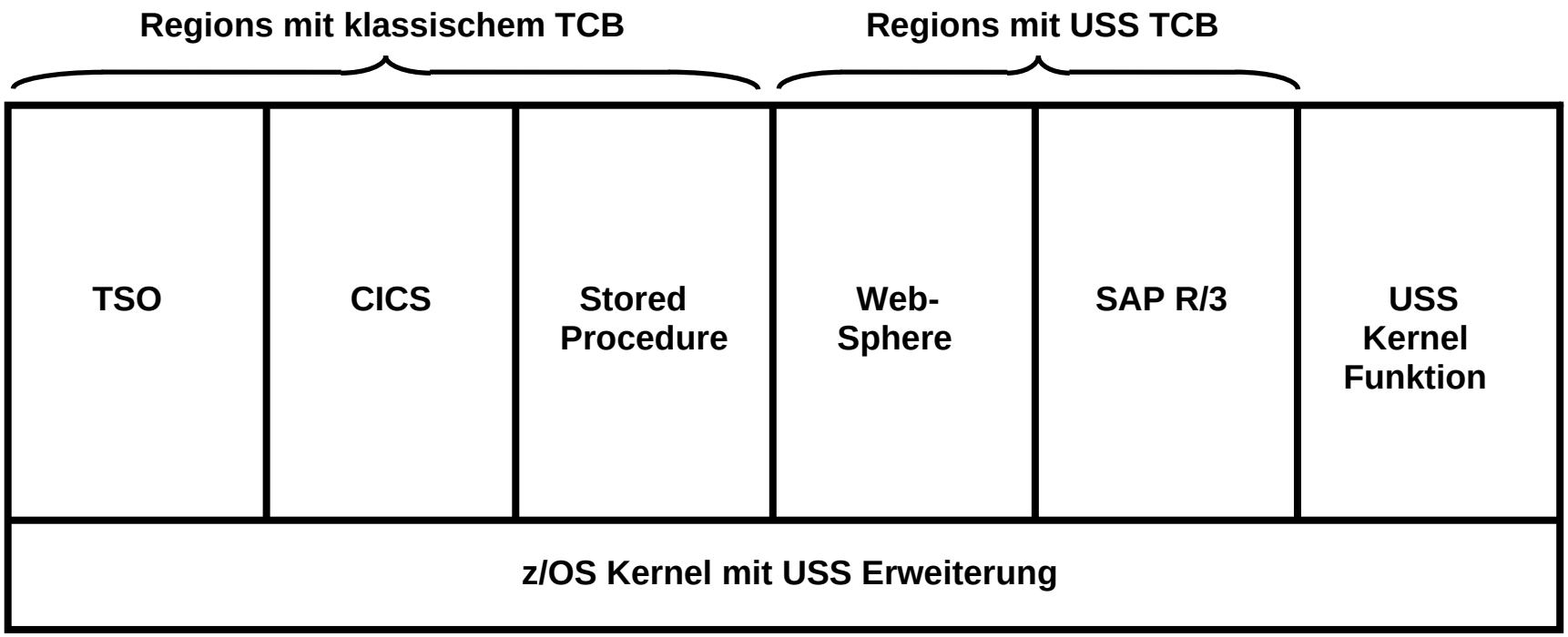
Das Byte-orientierte hierarchische File System arbeitet wie jedes Unix File System. Es wird in z/OS Data Sets abgebildet. Alle Files sind dem Unix System Services Kernel zugeordnet, und alle Ein-/Ausgabe Operationen bewirken Calls in den Kernel.

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

Unix System Services (USS)

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

All Rights Reserved.

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

IBM is a registered trademark of the IBM Corp.

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

In dem gezeigten z/OS Shell Beispiel hat der Benutzer /u/spruth den Unix Befehl `ls –als` eingegeben



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

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

Prozessverwaltung

Unix System Services

Datenbanken

Communication Server

DB2 relationale Datenbank

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

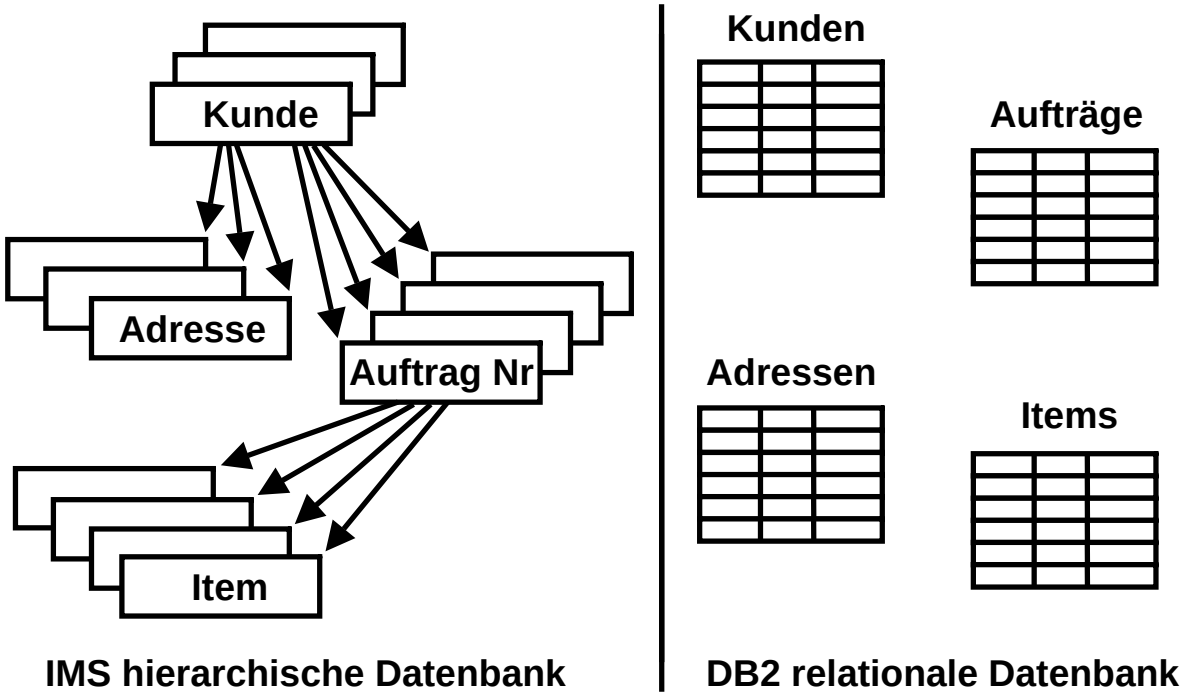
Identische Implementierung für alle UNIX-, Linux-, OS/2-, und Windows-Betriebssysteme.

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 DB2 neben IMS die am häufigsten eingesetzte Datenbank. Andere populäre z/OS-Datenbanksysteme sind IDMS der Fa. Computer Associates sowie Adabas der Fa. Software AG. Oracle ist unter z/OS nicht mehr verfügbar.

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.

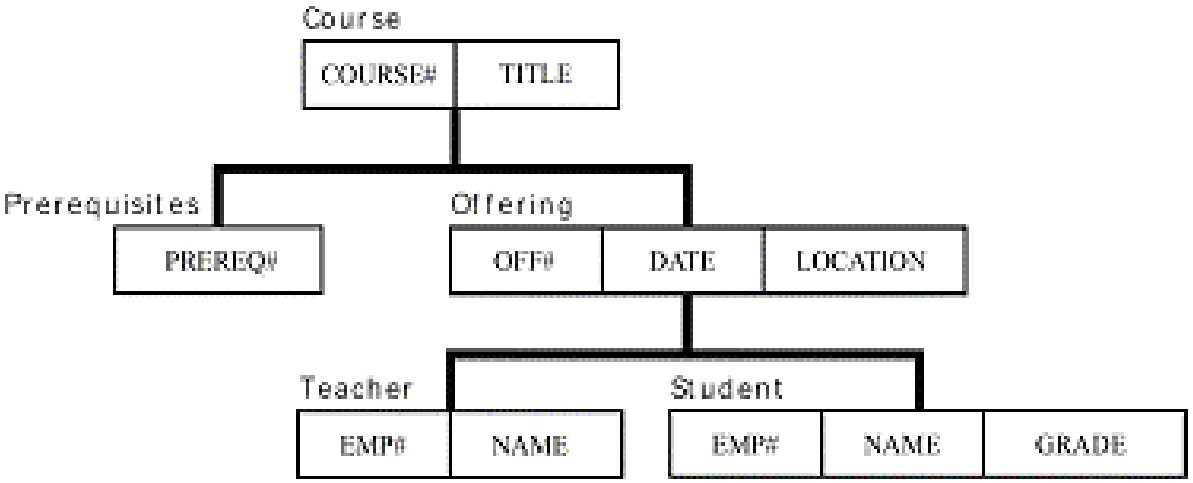
Der Zugriff kann mit Hilfe von SQL-Statements erfolgen, die in einem Anwendungsprogramm eingebettet sind. Alternativ existieren eine Reihe spezifischer SQL-Klienten-Anwendungen.

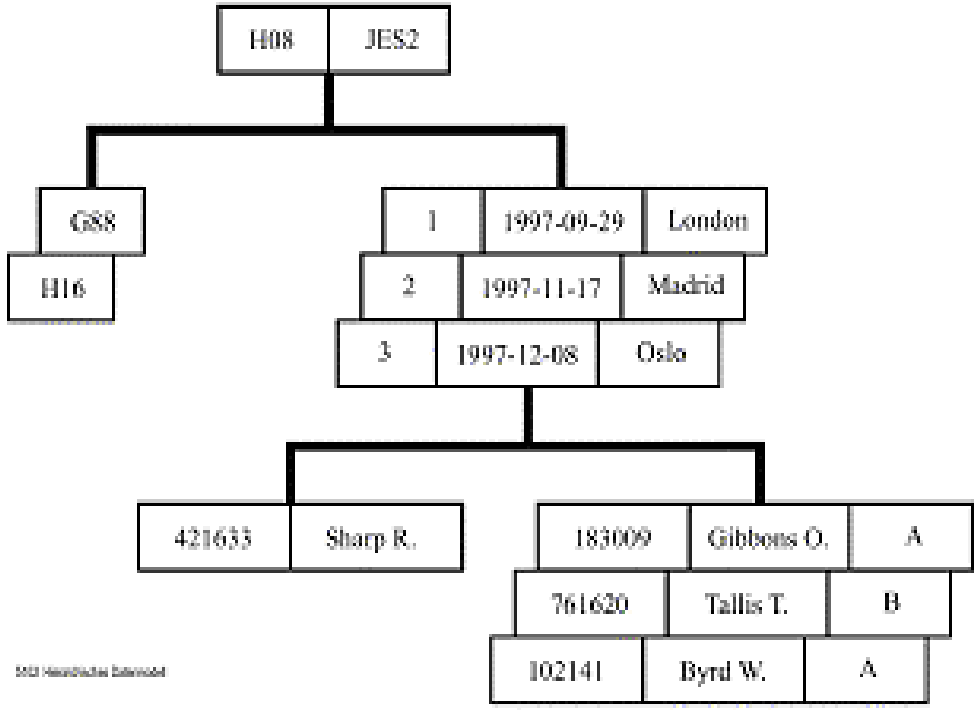


IMS Datenbanksystem

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

Neben DB2 ist IMS ein häufig unter z/OS eingesetztes Datenbanksystem.





IMS Datenbank-system

IMS benutzt Bäume (Trees) an Stelle der Tabellen in DB2.

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

IMS Datenbanksystem

IMS besteht aus zwei Komponenten, dem Database Manager (IMS DB) und dem Transaction Manager (IMS DC).

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.

Eines der ersten großen DBMS war IMS mit der Sprache DL/I (Data Language One). Die damit verwalteten Datenbanken waren hierarchisch strukturiert. Parallel dazu definierte CODASYL ein Modell für netzwerkartig strukturierte Datenbanken.

Geschäftsvorgänge als Geschäftsprozesse modellieren und dokumentieren ist heute Qualitätsstandard. Mit den Geschäftsobjekten (Stammdaten) aus der zentralen IMS Datenbank werden auch für komplexe Organisationsstrukturen die Veränderungen einfach und schnell dokumentiert.

The latest capabilities enable SOA exploitation

Die logische Struktur der Datenbank ist in der Database Description (DBD) und in den Program Communication Blocks (PCB) definiert. Dabei entspricht ein DBD grob einer Tabelle und ein PCB einer Sicht in relationalen Datenbanken, genauer einem Create-Table- bzw. einem Create-View-Befehl.

Data Language/I (DL/I) function codes used to query and update the databases within application programs.

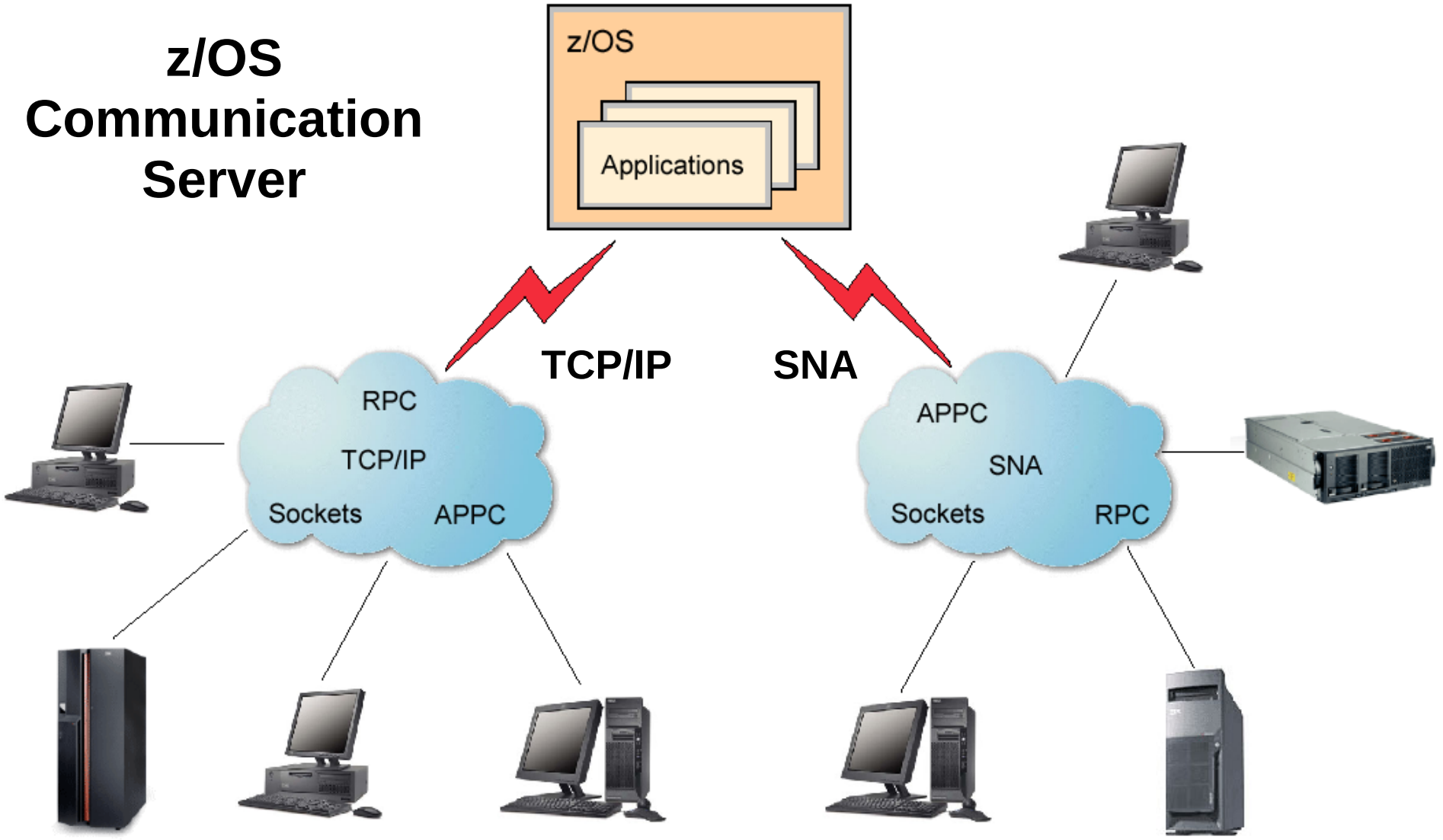
Prozessverwaltung

Unix System Services

Datenbanken

Communication Server

z/OS Communication Server



Der z/OS Communication Server ist ein eigenständiges Subsystem in einem eigenen virtuellen Adressenraum. Er implementiert die Software (Netzwerk Architektur Stacks) für eine ganze Reihe von Netzwerk Architekturen, besonders für TCP/IP und SNA.

SNA

Die Systems Network Architecture (SNA) wurde von IBM entwickelt und im Jahre 1974 vorgestellt. TCP/IP kam erst wesentlich später.

SNA ist vom Funktionsumfang her immer noch sehr modern und in etwa mit TCP/IP Version 6 vergleichbar. Bis in die 90er Jahre war SNA der de facto Standard in den allermeisten Mainframe Installationen. Danach erfolgte ein gradueller Wechsel zu TCP/IP, getrieben durch die Verbreitung des Internet. Heute sind physische SNA Netze fast überall durch TCP/IP Netze ersetzt worden.

Unter der Decke benutzen viele Systemkomonenten nach wie vor SNA. So kommuniziert Ihr 3270 Emulator über eine SNA Verbindung mit TSO oder CICS. Für den Benutzer ist dies unsichtbar, weil SNA Pakete in TCP/IP Pakete verpackt und über einen TCP/IP Tunnel versandt werden. Für die Kommunikation zwischen einem 3270 Emulator und einem Mainframe verwendet man hierzu ein aufgebohrtes Telnet Protoll, welches als TN3270 bezeichnet wird. Port 23 ist der Standard Telnet Port, und aus diesem Grunde ist Port 23 der Standard Port für Zugriffe auf einen Mainframe Rechner.

Der z/OS Communication Server hat Zusatz Einrichtungen, mit denen getunnelte TCP/IP Pakete entpackt, und der Inhalt an den SNA Stack weitergegeben werden kann.