UNIVERSITÄT LEIPZIG

Mainframe Internet Integration

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

SS2013

WebSphere Application Server Teil 1

z/OS as a Unix System

Unix

Welche Server-Betriebssysteme findet man in der Praxis ?

- **Windows, verschiedene Varianten**
- **Unix, verschiedene Varianten**
- **i-Series, OS/400**
- **zSeries Betriebssysteme – z/OS, zVM, zVSE, TPF**

Welche wesentlichen Unix Varianten existieren ?

- **HP/UX**
- **SunSolaris**
- **IBM AIX**
- **Siemens Sinix**
- **MacOS (BSD)**
- **Linux, einschließlich zLinux**
- **z/OS Unix System Services**

} **Unix Systeme sind weitestgehend, aber nicht 100 % kompatibel.**

Was ist ein Unix System ?

Ein Betriebssystem ist ein Unix System, wenn es den Posix (1003.1 and 1003.2) Standard und die X/Open XPG4 und XPG4.2 Standards erfüllt. Linux wurde nie Posix und XPG4.2 zertifiziert, dürfte aber zu 99,9 % Posix und XPG4.2 kompatibel sein. Ob Linux als Unix System bezeichnet werden kann, ist umstritten.

z/OS vs. Unix

Führende Unix Großrechner sind

- **Integrity Superdome von HP mit Itanium Prozessoren und dem HP-UX Betriebsssystem**
- **M9000 bzw. SPARC M5-32 Server von Oracle mit Sparc Prozessoren und dem Solaris Betriebssystem. Ein verwandtes Produkt wird von der Firma Fujitsu unter dem Namen „SPARC Enterprise Server“ vertrieben.**
- **System p von IBM mit PowerPC Prozessoren und dem AIX Betriebssystem**

Neben den proprietären Unix Dialekten ist auf diesen Rechnern auch Linux verfügbar.

Die I/O Leistung eines Rechners wird gemessen in der Anzahl von I/O Operationen pro Sekunde unter realistischen Betriebsbedingungen. Konkrete Untersuchungen sind nie veröffentlicht worden, aber es wird allgemein angenommen, dass die z/OS I/O Leistung vielleicht um einen Faktor 3 - 10 höher als die I/O Leistung von großen Unix Rechnern ist.

Es hat mehrfache Unix Implementierungen für Mainframes gegeben. Am erfolgreichsten war Amdahl UTS (Universal Time Sharing System), was seinerzeit auf etwa 300 Mainframe Rechnern installiert war. Die AT&T Portierung von Unix System V lief auf etwa 30 Mainframes. Andere Beispiele von geringerer Bedeutung sind Hitachi HI-OSF/1-M und IBM AIX/ESA.

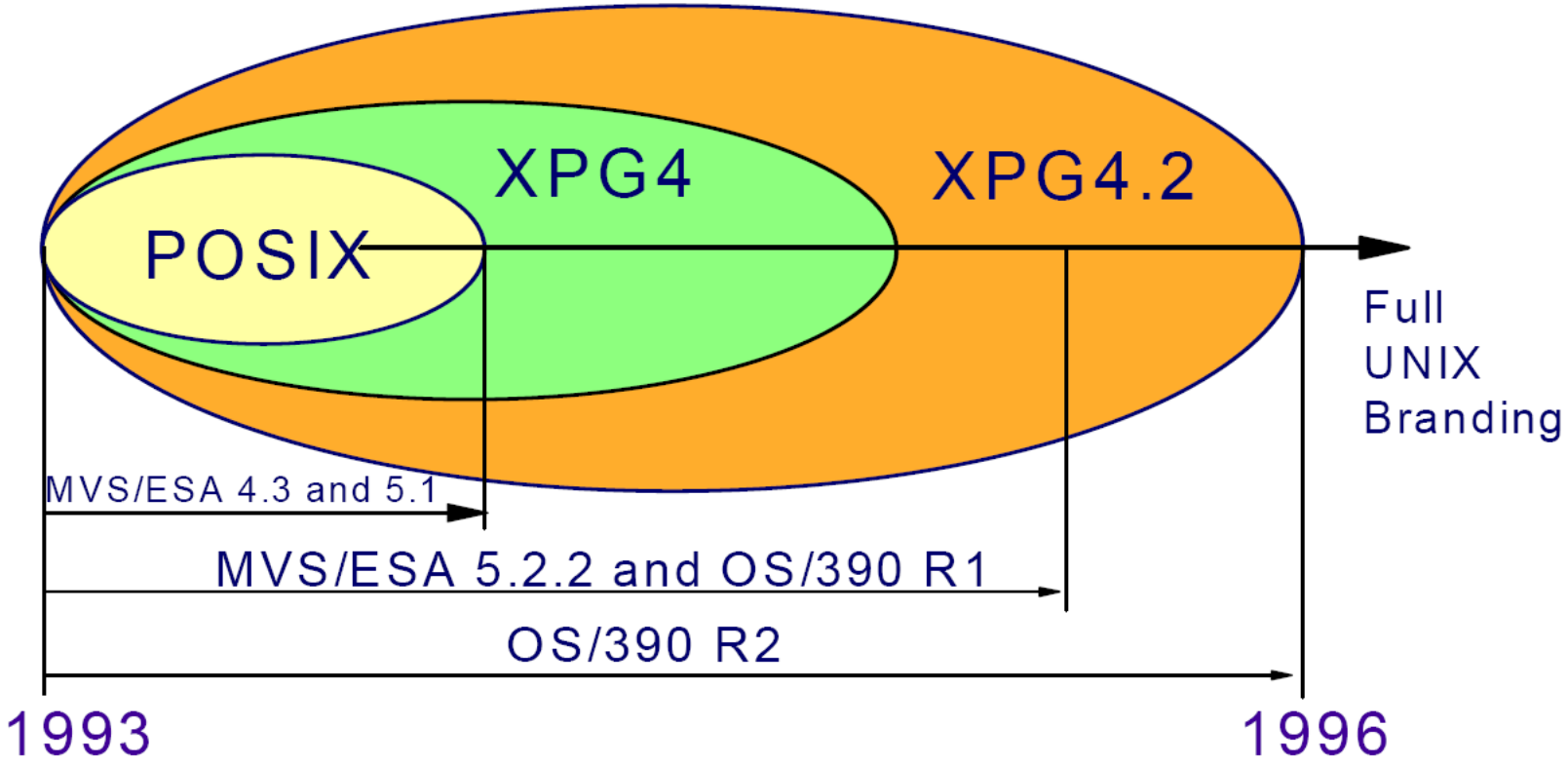
Was definiert ein Unix System ?

Unix wurde ursprünglich von Ken Thompson und Dennis Ritchie an den Bell Telephone Laboratories für Rechner der Digital Equipment Corporation (DEC) entwickelt und 1971 erstmalig im praktischen Betrieb eingesetzt. Schon bald entstanden (fast) kompatible Implementierungen für andere Rechner.

Aus Bemühungen um einen einheitlichen Unix Standard entstand die Portable Operating System Interface (POSIX). Dies ist ein gemeinsam von der IEEE und der Open Group für Unix entwickeltes standardisiertes Application Programming Interface, welche die Schnittstelle zwischen Applikation und dem Betriebssystem darstellt.

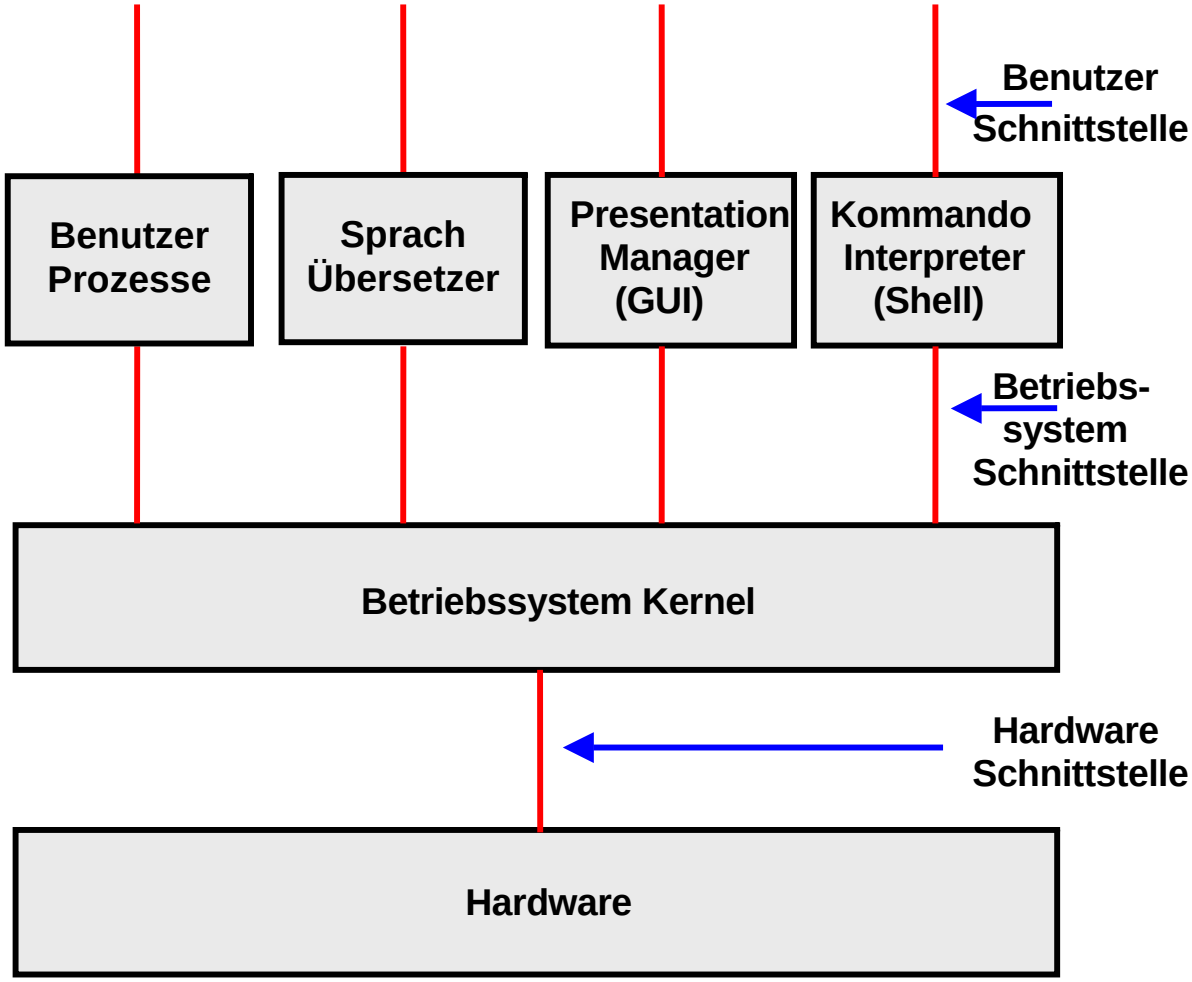
X/Open ist eine Erweiterung des POSIX Standards, welche ursprünglich von einem Konsortium mehrerer Europäischer Hersteller von Unix Systemen erarbeitet wurde. Die Erweiterung wurde in mehreren X/Open Portability Guides (XPG) veröffentlicht. Die letzten Versionen haben die Bezeichnungen XPG4 und XPG4.2.

HP-UX (HP), Solaris (SUN/Oracle), AIX (IBM) und z/OS Unix System Services (IBM) sind heute die wichtigsten Unix Betriebssysteme. Sie sind POSIX und X/Open zertifiziert. Dies garantiert, dass der Quellcode beliebiger Anwendungen mit minimalem Aufwand von einem Unix System auf ein anderes Unix System portiert werden kann, obwohl die Implementierungen sehr unterschiedlich sind.



Mainframe „Unix System Services“ wurde 1993 erstmalig unter dem Namen Open MVS (OMVS) verfügbar. Es gilt als vollwertiges Unix Betriebssystem.

Wie ist es möglich, dass z/OS gleichzeitig ein Unix Betriebssystem sein kann ?

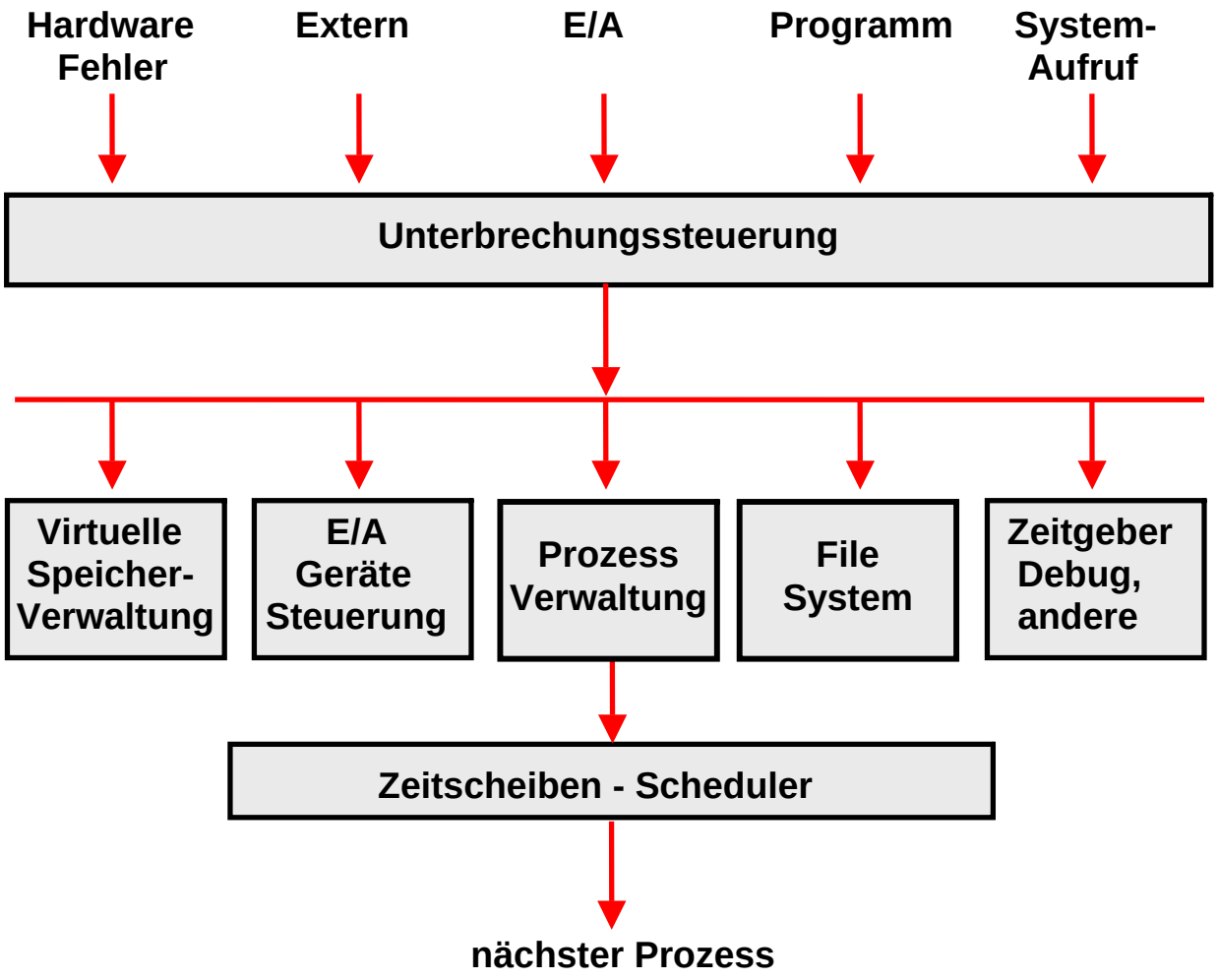


Der z/OS Kernel (Supervisor, Überwacher) wird auch als „Basic Control Program“ oder BCP bezeichnet.

Schichtenmodell der Rechnerarchitektur

Betriebssystem Kernels haben grundsätzlich immer eine sehr ähnliche Struktur. Sie bestehen aus den in der folgenden Abbildung dargestellten Komponenten.

Diese Folien sind teilweise eine Wiederholung von Verarbeitungsgrundlagen Teil 3 aus dem Vorlesungsscript „Einführung in z/OS“. Siehe http://jedi.informatik.uni-leipzig.de/de/Vorles/Einfuehrung/Grundlag/vg03.pdf#page=08



Struktur des Supervisors

Der Aufruf des Überwachers (Supervisor, Kernel) erfolgt grundsätzlich über Unterbrechungen.

Je nach Art der Unterbrechung werden unterschiedliche Komponenten des Überwachers aufgerufen.

Benutzerprozesse nehmen Dienste des Überwachers über eine architekturierte Schnittstelle, den Systemaufruf (System Call, Supervisor Call, SVC) in Anspruch. Diese Begriffe haben alle die gleiche Bedeutung.

Der Scheduler (Zeitscheiben-steuerung) sucht den nächsten auszuführenden Prozess aus.

Supervisor Calls

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

Supervisor Calls im weiteren Sinne sind Library Routinen, die eine Dienstleistung des z/OS Kernels in Anspruch nehmen. Andere Bezeichnung: System Calls.

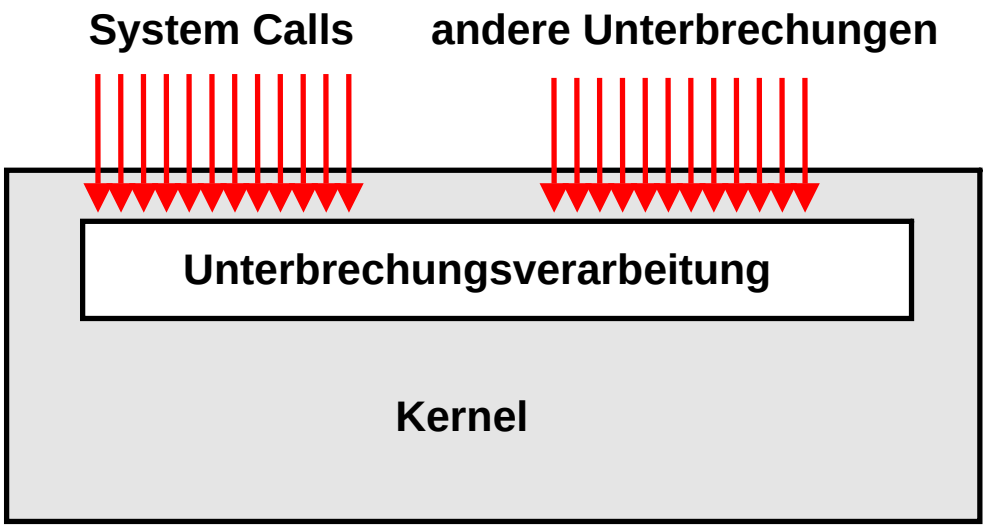
Supervisor Calls im engeren Sinne sind Maschinenbefehle, die mittels eines Übergangs vom User Mode (Problem Status) zum Kernel Mode (Supervisor Status) einen Interrupt Handler des Kernels aufrufen. Äquivalente Maschinenbefehle in der X86 (Pentium) Architektur sind int 0x80 Call Gate, und SYSENTER. Für die IA-64 Architektur wird die EPC (Enter Privileged Mode) instruction benutzt.

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**

Unix System Services sind eine Erweiterung des z/OS Kernels (BCP) um 1100 Unix System Calls, die als z/OS SVCs implementiert sind.



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

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

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

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

Der Kernel eines Betriebssystems wird fast ausschließlich über Unterbrechungen oder über System Calls aufgerufen.

Unix System Services (USS)

Was ist ein Unix Betriebssystem ?

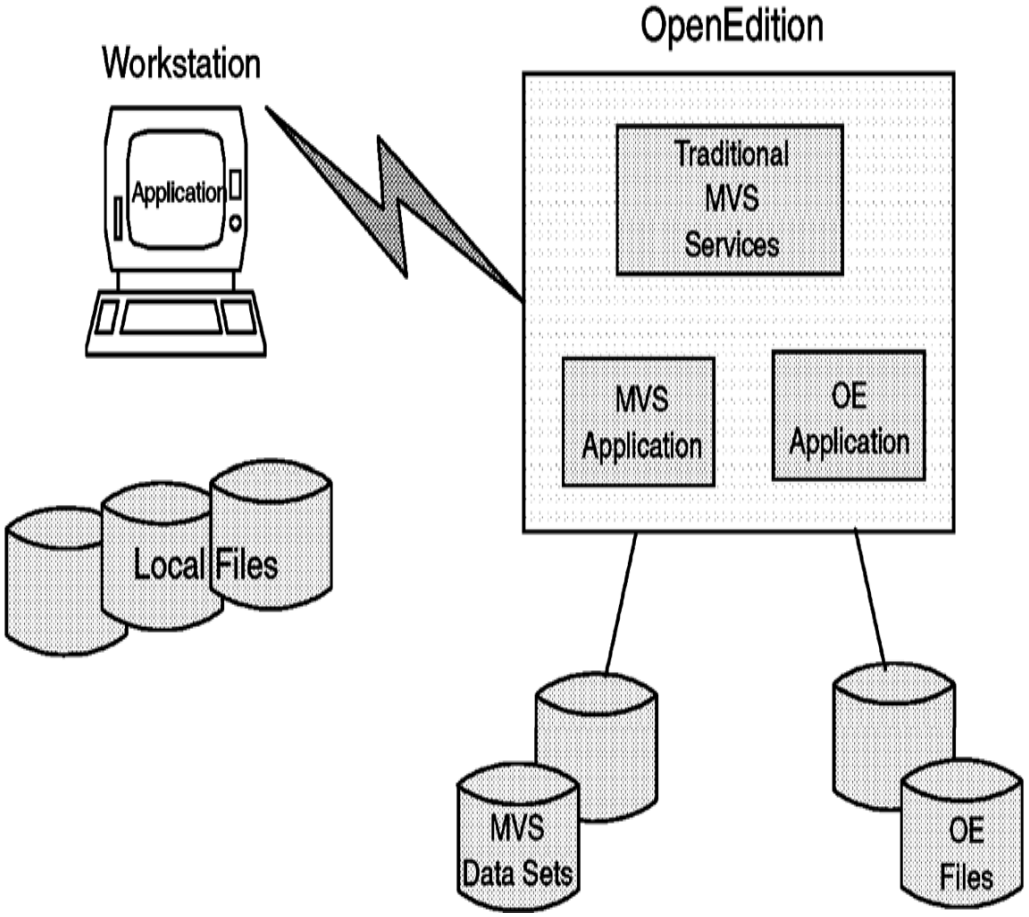
Ein Unix Betriebssystem erfüllt den „POSIX“ Standard. Das Portable Operating System Interface ist ein gemeinsam von der IEEE und der Open Group für Unix entwickeltes standardisiertes Application Programming Interface, das die Schnittstelle zwischen Applikation und dem Betriebssystem darstellt. Der Standard 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.



Unix System Services
OpenEdition MVS

z/OS verhält sich entweder wie ein traditionelles MVS (z/OS) System oder wie ein Unix System. Letzteres wird durch eine Zusatz Einrichtung, den Unix System Services ermöglicht. Frühere Bezeichnung : OpenEdition.

Literatur

ABCs of z/OS System Programming Volume 9, November 2005, IBM Form No. SG24-6989-01

z/OS UNIX System Services User ’s Guide, IBM Form No. SA22-7801-04

OS/390 Unix System Services. IBM Form No. SC28-1891-8

Die z/OS UNIX Komponenten

Ein typisches UNIX-Betriebssystem besteht aus einem Kernel, der eine direkte Schnittstelle mit der Hardware darstellt. Aufgebaut auf dem Kernel ist die Shell und eine Utilities Schicht, die eine Kommando-Interface definiert. Dazu gibt es Anwendungsprogramme, die auf der Schell oder auf Utilities aufgebaut sind.

Beim z/OS Systemstart (IPL) werden UNIX System Services Kernel-Dienste automatisch gestartet. Der Kernel bietet z/OS USS als Service für Anforderungen von Unix Programmen und von der USS Shell an. Der Kernel verwaltet ein Unix hierarchischs Filesystem (HFS), die Kommunikationseinrichtungen und die UNIX System Services (USS) Prozesse. Das hierarchische Filesystem wird als Teil des Kernels betrachtet, und als Komponente des DFSMS (Data Facility Storage Management Subsystem) installiert. Siehe http://jedi.informatik.uni-leipzig.de/de/Vorles/Einfuehrung/Bs/zos03.pdf#page=016.

Auch nicht-Unix Programme können auf das hierarchische File System zugreifen. So kann z.B. ein CICS Programm im Zusammenhang mit CICS Web Support auf HFS zugreifen.

Um die APIs unterstützen, muss das z/OS-System einige System-Dienste in seinem Kernel enthalten, z. B. das hierarchische Filesystem-und Kommunikationsdienstleistungen.

Daemons sind Programme, die typischerweise gestartet werden, wenn das Betriebssystem initialisiert wird. Sie bleiben aktiv, um Standard Services auszuführen. z/OS UNIX (USS) enthält Daemons, die auf Anforderung Anwendungen oder eine User Shell Sitzung starten.

Unix System Services Komponenten

z/OS UNIX shell and utilities

Eine interaktive Interface zu z/OS UNIX Services, welche Commands von interaktiven Benutzern und von Programmen interpretiert.

Die Shell und Utilities Komponente kann mit den TSO-Funktion unter z/OS verglichen werden.

Hierarchical File System

Neben den vorhandenen z/OS Dateisystemen (zB VSAM, PDS) wurde ein zusätzliches Unix kompatibles hierarchischen Filesystem eingerichtet. Jedes Programm kann auf das hierarchische Filesystem zugreifen.

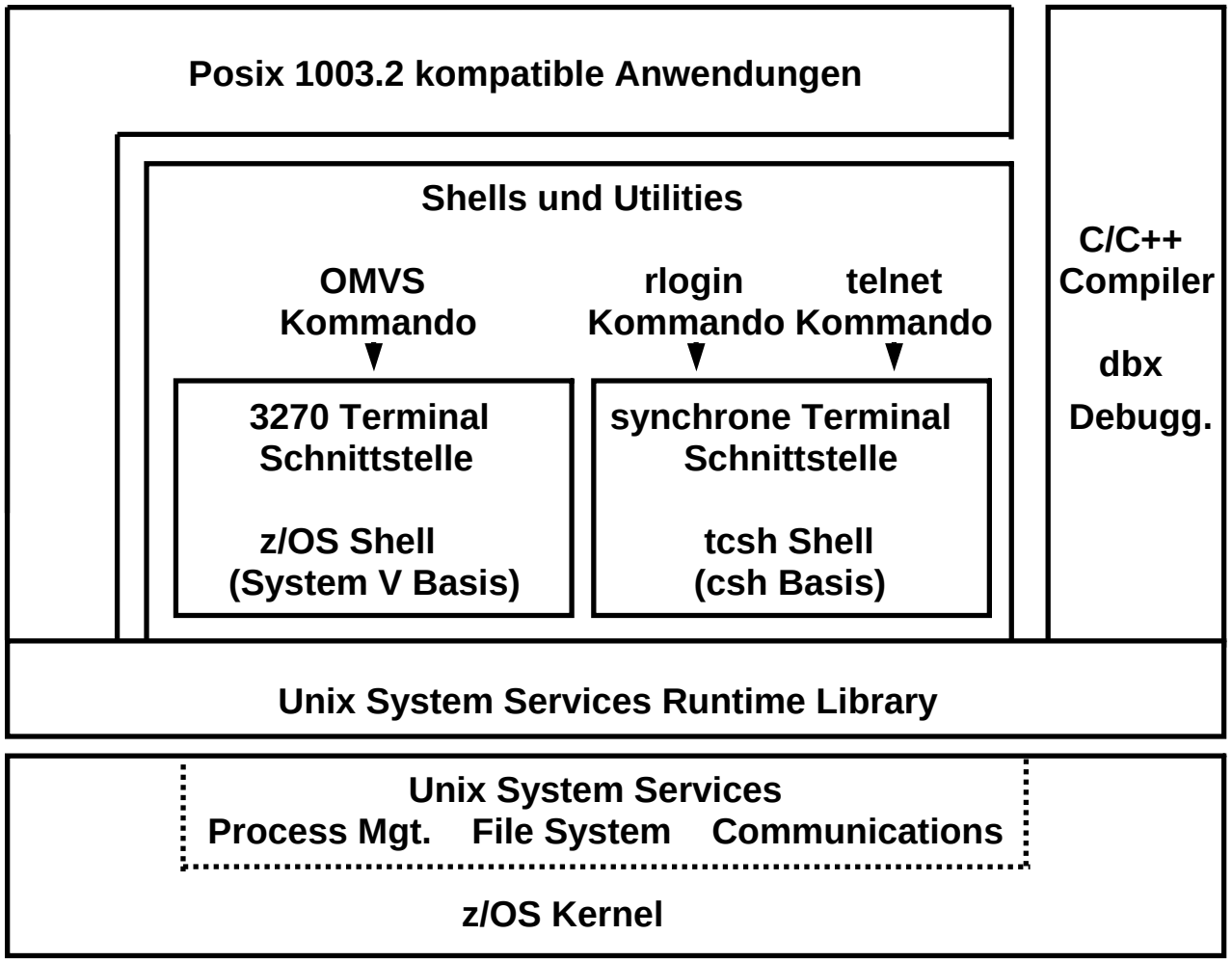
z/OS UNIX debugger (dbx)

Die z/OS UNIX-Debugger ist ein Werkzeug, das Anwendungsprogrammierer verwenden, um interaktiv ein C-Programm zu debuggen. Der dbx-Debugger ist nicht Teil des POSIX-Standards. Es basiert auf dem Unixdbx Debugger, der auch in vielen UNIX-Umgebungen vorhanden ist.

C/C++ Compiler and C run-time Libraries

Die C/C++ Compiler und C-Laufzeitbibliotheken werden benötigt, um ausführbare Dateien zu erstellen, die der Kernel verstehen und verwalten kann. Die C/C++ Compiler und Language Environment (LE) Feature-Bibliothek wurde verändert und erweitert, um Unterstützung für die POSIX-und XPG4 C Funktionsaufrufe zu enthalten. Das LE Produkt stellt eine gemeinsame Laufzeitumgebung und sprachspezifische run-time Services für kompilierte Programme zur Verfügung.

Um einen Shell-Befehl oder Dienstprogramm, oder ein vom Benutzer bereitgestelltes Anwendungsprogramm in C oder C++ auszuführen, benötigen Sie die C/C++ Runtime Library, die mit LE zur Verfügung gestellt wird.

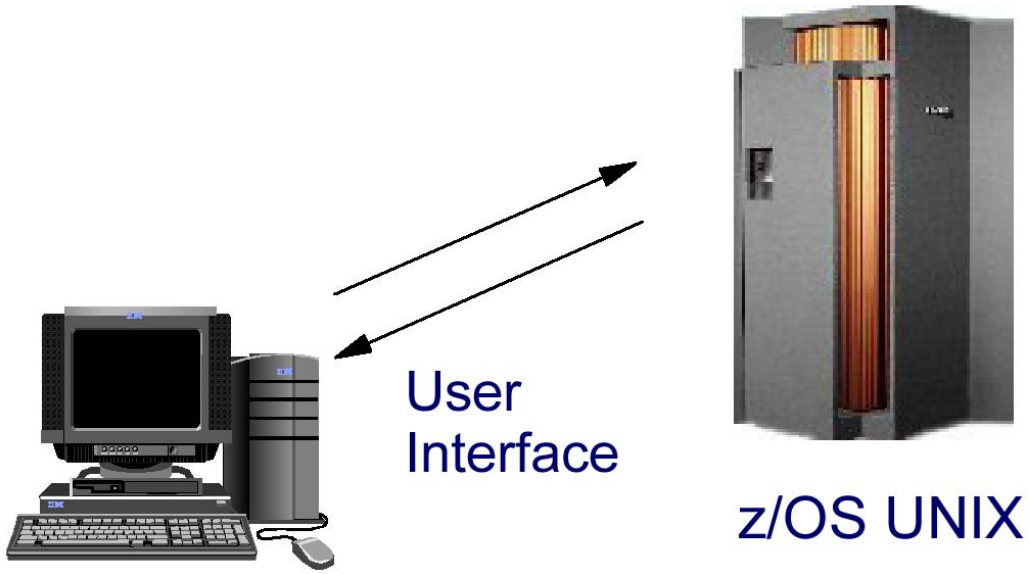


Die Unix System Services Kernel Funktion läuft in einem eigenen virtuellen Adressenraum, der als Teil des IPL (Boot) Vorgangs hochgefahren wird. Er wird wie jeder Unix Kernel über eine API aufgerufen, die aus C/C++ Function Calls besteht.

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.

Häufig werden auf dem gleichen z/OS Rechner Unix System Services und zLinux parallel betrieben.

Unix System Services (USS)

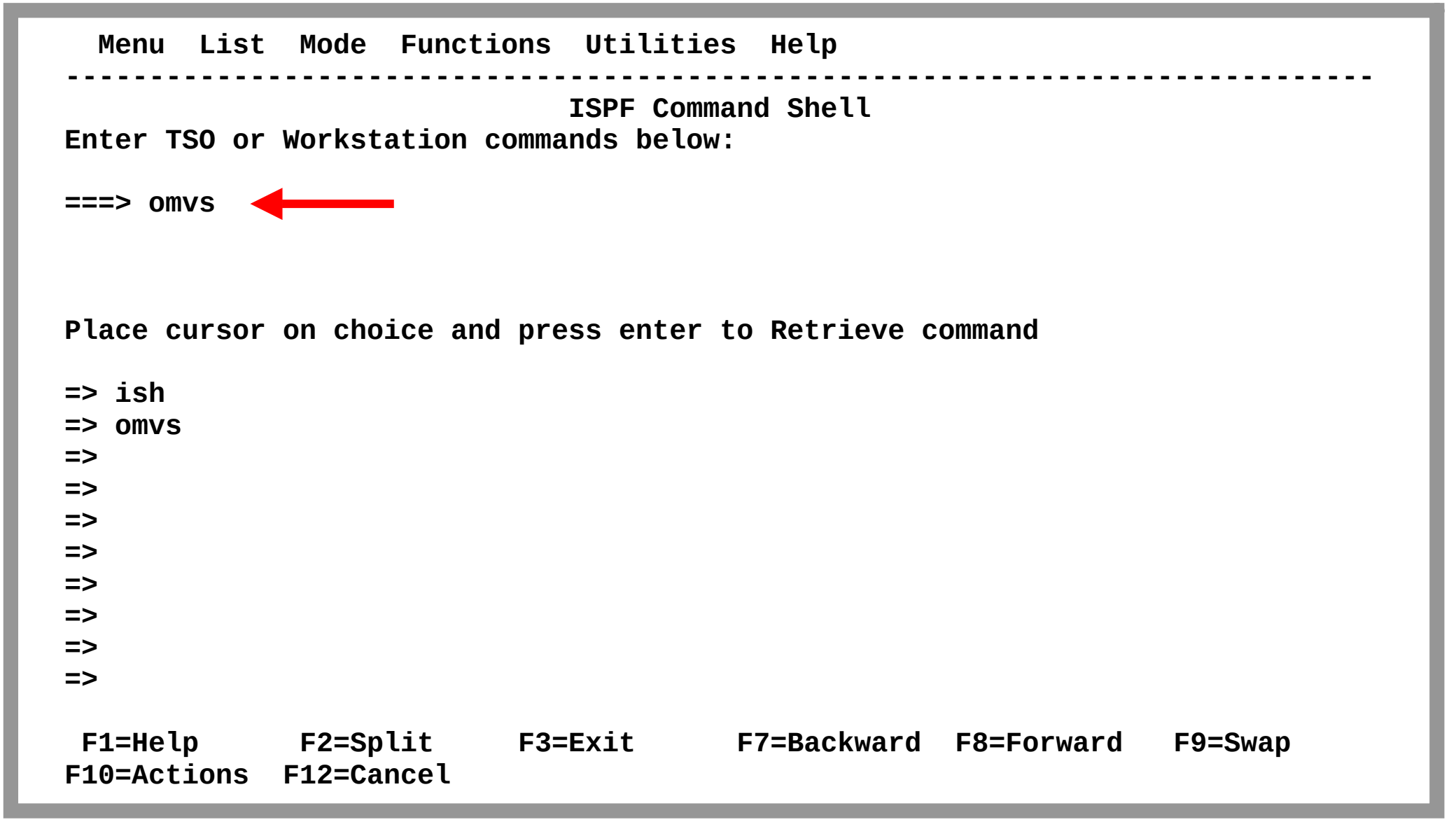


TSO und z/OS UNIX

Es existieren zwei alternative Möglichkeiten für den Zugriff auf Unix System Services (zwei unterschiedliche Shells)

Eine Möglichkeit ist die Benutzung der z/OS Shell mittels eines 3270 Terminals und TSO. Die z/OS Shell gleicht der Unix System V Shell mit einigen zusätzlichen Eigenschaften der Korn Shell. Sie benutzt den ISPF Editor. Ein Benutzer startet eine TSO Session und gibt das OMVS Kommando ein. Dies ist die bevorzugte Methode für TSO Experten.

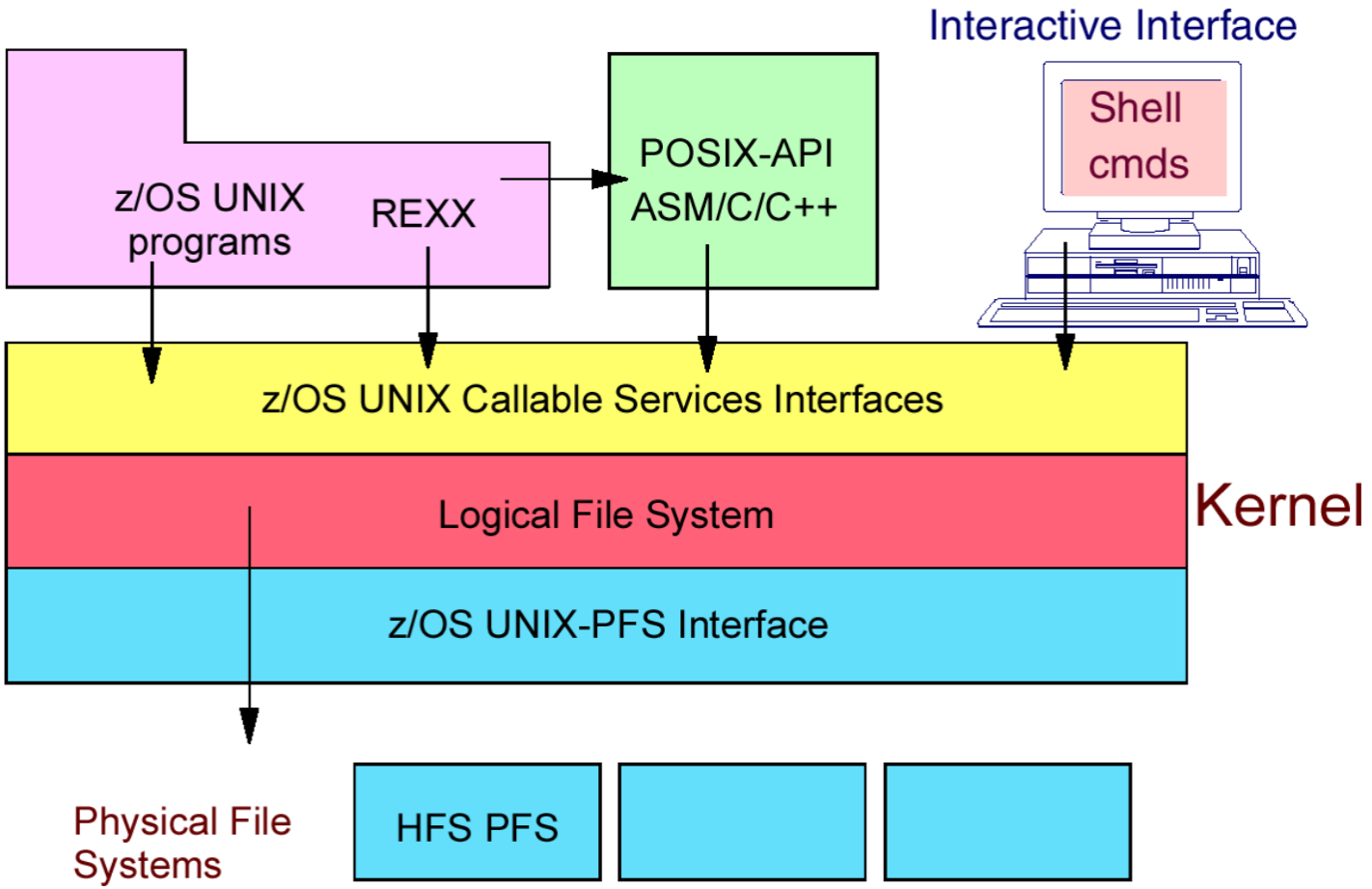
Die andere Möglichkeit ist die tcsh Shell. Diese ist kompatibel mit der csh Shell, der Berkley Unix C Shell. Sie wird über rlogin oder telnet (z.B. putty) aufgerufen und verwendet den vi Editor. Dies ist die bevorzugte Methode für Unix Experten.



Die Unix System Services Shell wird vom ISPF Command Shell Screen durch die Eingabe des TSO Kommandos “OMVS” gestartet.

```
(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



Mit beiden Shells kann ein Standard Unix hierarchisches Filesystem benutzt werden. z/OS verfügt über ein logisches hierarchisches File System, welches auf eins von mehreren verfügbaren physischen File Systemen abgebildet wird. So wie Linux über mehrere physische File Systems verfügt (ext2, ext3, ext4, ReiserFS), existieren auch für z/OS USS mehrere physische File Systeme.

REXX ist eine populäre z/OS und USS Script Sprache.

Benutzung von z/OS UNIX

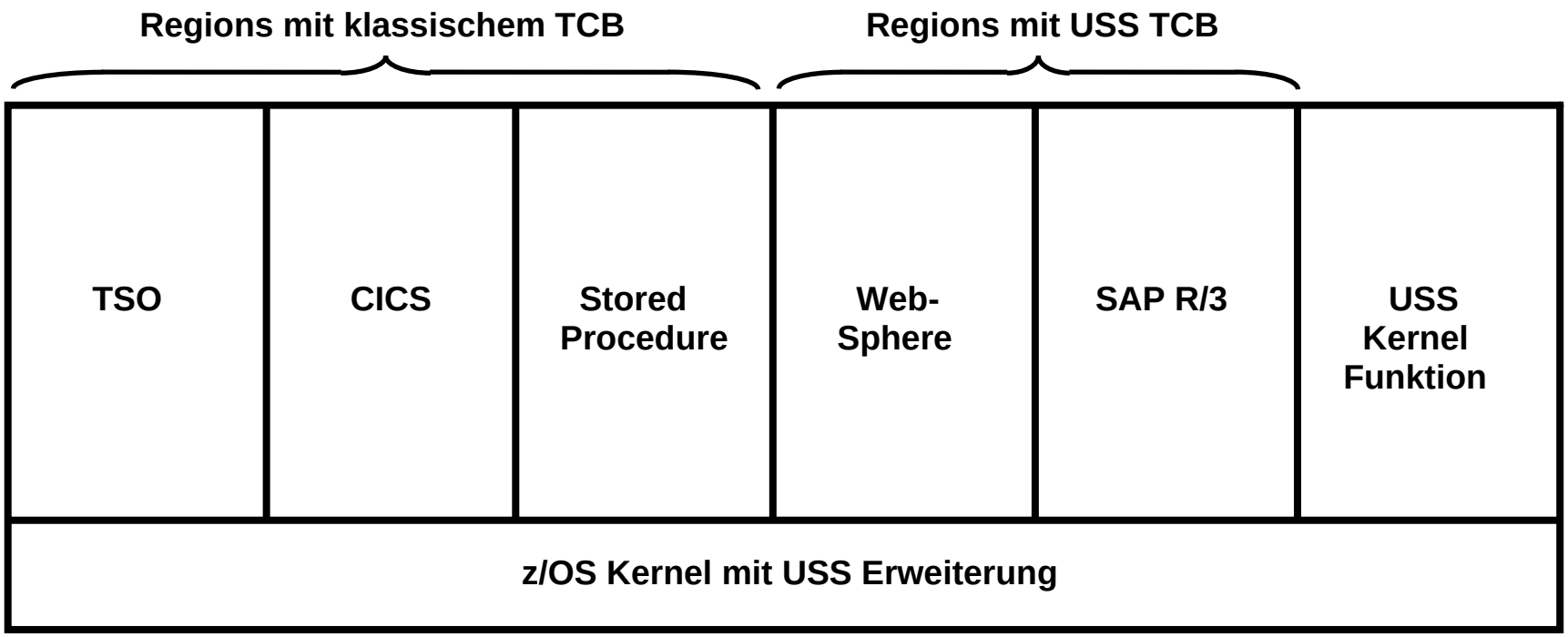
Ein Benutzer kann mit z/OS UNIX über die folgenden Schnittstellen interagieren:

- **Das Application Programming Interface (API) besteht aus Aufrufen (Calls), die von C/C++ Programmen verwendet werden können, um auf z/OS UNIX zuzugreifen. Diese C Aufrufe sind in dem POSIX 1003.1 Standard definiert.**

  Die aufrufbaren Dienstleistungen können direkt vom Assembler-Programmen verwendet werden, um auf z/OS UNIX zuzugreifen, z. B. um auf Dateien in dem hierarchischen Filesystem zuzugreifen. Diese Möglichkeit erlaubt es anderen Hochsprachen und Assembler, z/OS UNIX zu verwenden. Die API-Schnittstelle bietet die Möglichkeit, XPG4.2 Programme auf z/OS auszuführen. Ein Programm, das dem XPG4.2 Standard entspricht, kann auf einem System entwickelt werden, und dann auf ein anderes System portiert werden, dort kompiliert und gelinked werden, und dann auf diesem ausgeführt werden. Ein solches Programm wird als portable bezeichnet.

- **Die interaktive Schnittstelle wird als z/OS UNIX-Shell bezeichnet. Die Shell ist ein Kommando-Interpreter, der Befehle akzeptiert, die in dem POSIX 1003.2 Standard definiert sind. Shell-Befehle können in eine Reihenfolge gebracht werden, in einer Textdatei als ein Shell-Skript gespeichert werden, und dann ausgeführt werden. Das Shell-Skript arbeitet ähnlich wie z/OS CLISTs oder REXX EXECs.**

  TSO REXX enthält Erweiterungen, um den Zugang zu den z/OS UNIX abrufbaren Dienstleistungen zu ermöglichen. Eine REXX EXEC mit UNIX System Services kann von TSO/E ausgeführt werden, als z/OS Batch Job oder in der Shell.



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.

Unix System Services vs. zLinux

Für die heutigen Mainframes sind heute zwei Unix Implementierung von Bedeutung:

- **zLinux**
- **z/OS Unix System Services (USS)**

Sie sind vor allem von Interesse, wenn existierende Unix/Linux Anwendungen von einer distributed Platform auf den Mainframe portiert werden sollen.

Warum zwei Alternativen, und was ist der Unterschied ?

zLinux ist ein regulärer Port von Suse oder Red Hat Linux auf System z Hardware. Es läuft entweder in einer eigenen LPAR, oder als virtuelle Maschine unter z/VM. Beide Alternativen sind auf Mainframe Servern relativ weit verbreitet und werden fast immer nicht an Stelle von z/OS, sondern parallel zu z/OS benutzt.

Auf der Betriebssystem Ebene ist der zLinux Kernel performanter als der z/OS Kernel. Es fehlen aber alle z/OS spezifischen Funktionen, beispielsweise Unterstützung für Sysplex, Coupling Facility, Work Load Manager, RACF, FICON, Protection Keys, Krypto und vieles mehr. Je nachdem ob diese Funktionen gebraucht werden, laufen Unix Programme entweder unter zLinux oder USS. Häufig wird in einer Mainframe Installation beides genutzt.

Im Gegensatz zu zLinux ist Unix System Services (USS) kein eigenes Betriebssystem, sondern ein integrierter Bestandteil von z/OS. Es ist damit in der Lage, alle z/OS Funktionen zu nutzen. Es ermöglicht eine relativ problemlose Portierung von existierenden Unix Anwendungen auf einen Mainframe Server, ohne dabei auf existierende z/OS Eigenschaften verzichten zu müssen.

Zwei der wichtigsten Software Pakete, die unter z/OS Unix System Services laufen (oder auch unter zLinux), sind:

- **SAP/R3**
- **WebSphere**

Microsoft Windows Services for UNIX

Microsoft Windows Services for UNIX (SFU) ist ein Software-Paket von Microsoft, welches ähnlich wie z/OS Unix System Services arbeitet. Es handelt sich hierbei um ein Unix-Subsystem (und weitere Komponenten einer Unix-Umgebung) nach dem POSIX-Standard, welches unter den Windows Betriebssystemen verfügbar ist. Dieses Subsystem wird als Interix, SFU oder SUA bezeichnet. Siehe
http://de.wikipedia.org/wiki/Microsoft_Windows_Services_for_UNIX

Microsoft Windows Services for UNIX setzt als Implementierung eines User-Mode-Subsystems unmittelbar auf dem Windows-Kernel auf. Die aktuelle Version trägt die Nummer 3.5. Als Veröffentlichungsdatum wird der 21. September 2006 angegeben.

Microsoft Windows Services for UNIX kann auf den folgenden Windows-Varianten installiert werden:

- **Windows XP Professional,**
- **Windows Server 2003,**
- **Windows Server 2008,**
- **Windows 7 Enterprise und Ultimate.**

Microsoft Windows Services for UNIX hat als Bestandteil von Windows eine vergleichbare Funktionalität wie Unix System Services als Bestandteil von z/OS. Im Gegensatz zu z/OS Unix System Services wird es aber weniger häufig eingesetzt.