TI III: Operating & Communication Systems

Winter 2006/07

Prof. Dr.-Ing. Jochen H. Schiller

Computer Systems & Telematics
Freie Universität Berlin, Germany
schiller@inf.fu-berlin.de

TI III: Operating & Communication Systems

Introduction and Motivation

Tasks, Services,

Virtual Resources,

Historical Perspective,

Examples, Tools

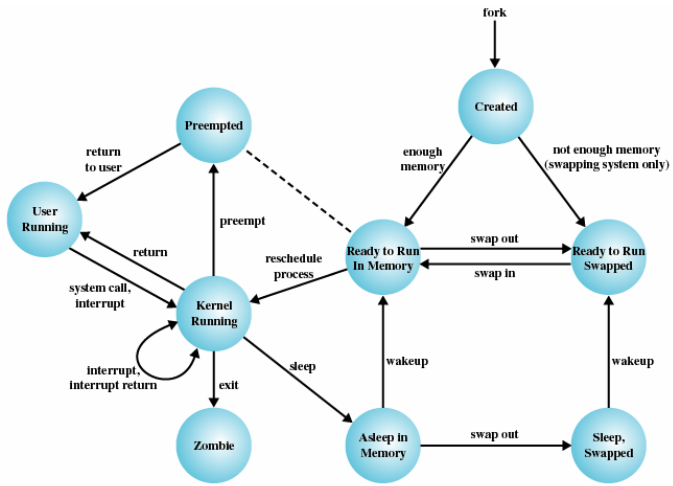


Figure 3.17 UNIX Process State Transition Diagram

Structure and Content of the CST-Lectures

Freie Universität Berlin

Praktikum Mobilkommunikation (8, 6)
Medienzugriff, Mobile IP, Mobiles Web

Embedded Sensor Web (7+)
Projekte rund um Sensornetze

Seminar Technische Informatik (6+)
Forschung in Mobilkommunikation, eingebettete Systeme, Internet

Mobilkommunikation (8, 6)
Drahtlose Übertragung, Medienzugriff, GSM, 3G, WLAN, Mobile IP, Ad-hoc-Netze, WAP

Embedded Internet (7)
Protokolle, Dienste, Internet, TCP/IP, Betriebssysteme für eingebettete Systeme

Mikroprozessorpraktikum (7, 5)
Programmierung eingebetteter Systeme, mobile Endgeräte, Mikrocontroller, Steuerungssysteme

Telematik (5)
Protokolle, Dienste, Standards, LAN, Internet, TCP/IP, WWW, Sicherheit, ISDN/IN/ATM, Dienstgüte, Multimedia, IPv6, MPLS

Praktikum Technische Informatik (TI IV) (4)
Eingebettete Systeme, Schnittstellen, Treiber, Betriebssystem – programmieren, vernetzen, interagieren

Betriebs- und Kommunikationssysteme (TI III) (3)
Ein-/Ausgabe, DMA/PIO, Unterbrechungen, Puffer, Prozesse/Threads, UNIX/Windows, Netze, Medienzugriff, Protokolle, TCP/IP, Internet

Rechnerarchitektur (TI II) (2)
Harvard/v. Neumann, Mikroarchitektur, RISC/CISC, VLIW, Pipelining, Cache, Speicherhierarchie, Assembler, Multiprozessorsysteme

Grundlagen der Technischen Informatik (TI I) (1)
Schaltnetze, Schaltwerke, Logikminimierung, Gatter, Speicher, Halbleiter, Transistoren, CMOS, AD/DA-Umsetzer

④ Semester
Bachelor
Master

Content (1)

Course organization

- Lecture
  - Friday, 10:00-12:00h, HS, Takustr. 9
- Office hours
  - Prof. Schiller: Tuesday, 14:00-15:00h, room 156, Takustr. 9
  - Tutors: during tutorials
- News and updates
  - http://www.inf.fu-berlin.de/inst/ag-tech/
- Tutorials
  - Groups of approx. 30 students
  - Time/location depends on group
  - Registration via WWW

Assignments

- During the lecture
  - Distribution of new assignments with fixed deadline
- Handing in
  - Right on time!
    Each tutor has his/her own box, 1st floor, Takustr. 9
  - Solutions handed in too late will be ignored!
- Discussion
  - During the tutorials
- Practical assignments
  - Pool computers available
  - More during lecture/tutorials

Criteria for successful participation

Freie Universität Berlin

- Active participation in the tutorials is essential!
  - Minimum n-2 times present
- Hand-in your assignments on time
  - Teamwork is required - 2 students in a team
- At least 50% of the max. number of points from the assignments required
- Successful submission of at least n-2 assignments
  - Successful = at least 25% of the max. number of points
- Each student with a correct answer must be able to present the assignment during the tutorials
  - At least 1x presentation during the tutorials
- At least 50% of the max. number of points required in the exam
  - Only the exam counts for grading!

Literature

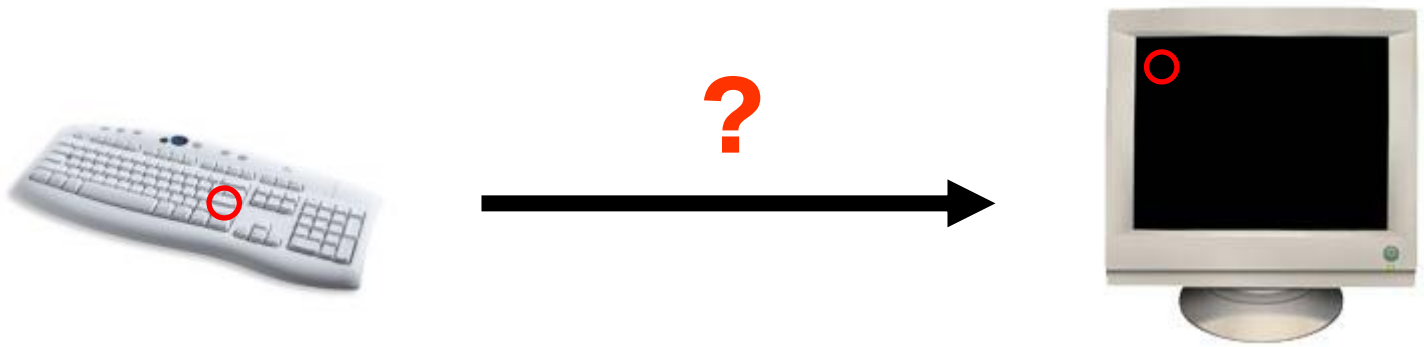
- Copies of the slides
  - Will be available at the secretary‘s office, room 155, 5€ - not on the WWW (© reasons…)

- The course follows (roughly) the book:
  - William Stallings. Betriebssysteme – Prinzipien und Umsetzung, 4. Auflage. Pearson Studium. November 2002. ISBN 3-8273-7030-2.

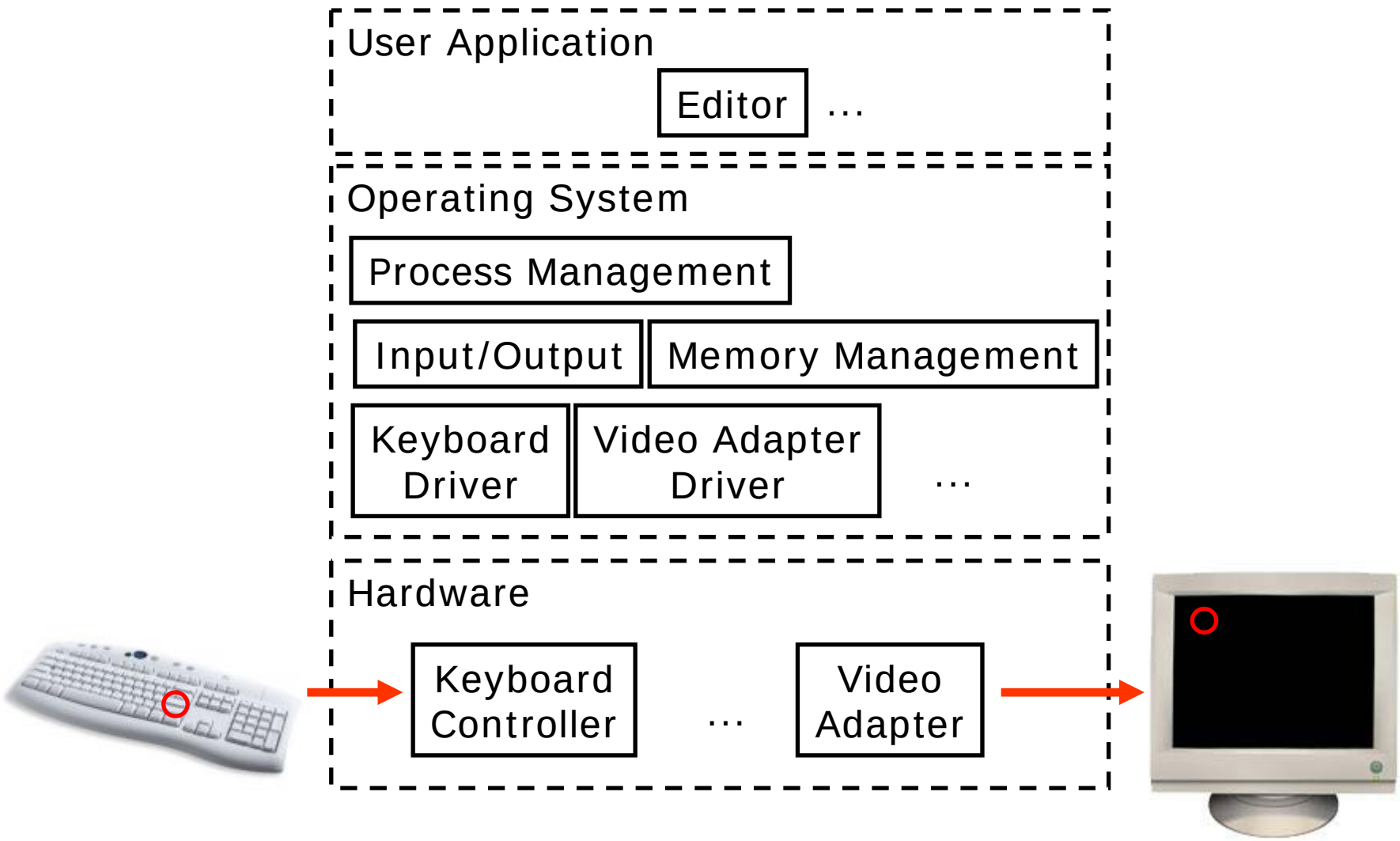
- Additional literature:
  - Erich Ehses, Lutz Köhler, Petra Riemer, Horst Stenzel und Frank Victor. Betriebssysteme, 1. Auflage. Pearson Studium. August 2005. ISBN: 3-8273-7156-2.
  - Gary Nutt. Operating Systems – A Modern Perspective, 2nd Edition. Addison Wesley. 2002. ISBN: 0-201-74196-2
  - Andrew S. Tannenbaum. Modern Operating Systems, 2nd Edition. Prentice Hall. February 2001. ISBN: 0-130-31358-0

Operating System Example

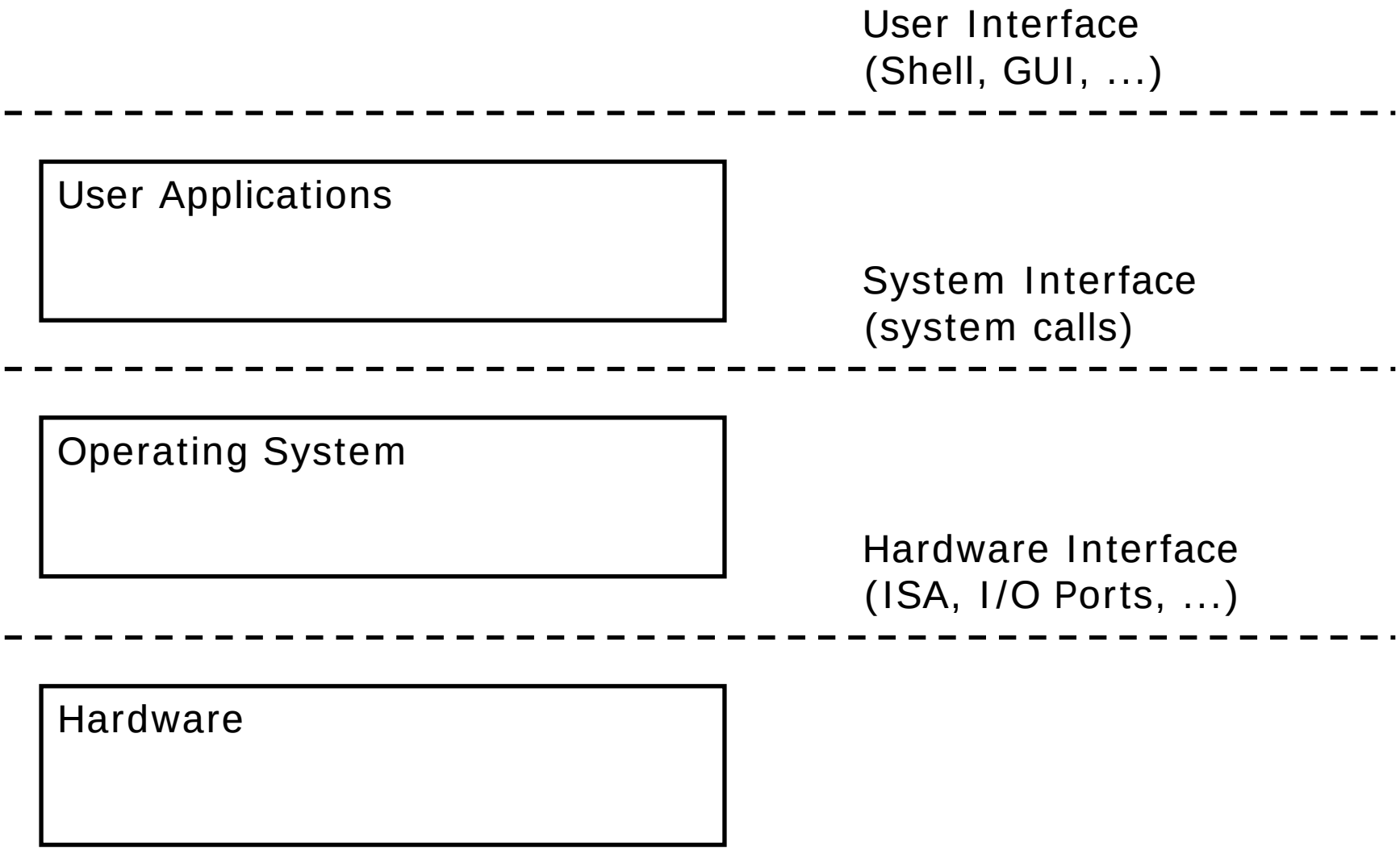
- What happens if one presses a key on the computer?



Operating System Example

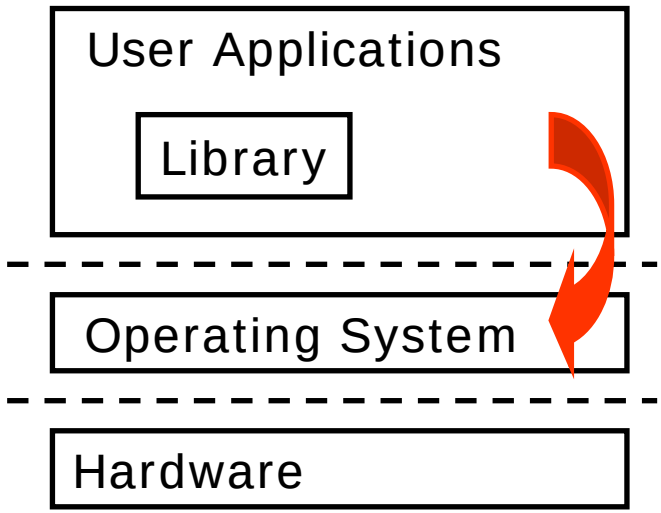


Layers of Abstraction



System Interface und System Calls

- System interface is the only way for user applications to interact with the operating system.
- System interface consists of system calls (supervisor calls), e.g. POSIX.



- High-level programming languages hide systems calls in their library routines.

Tasks of Operating Systems

- Goals:
  - Ease of use
  - Efficiency
  - Possibility to evolve

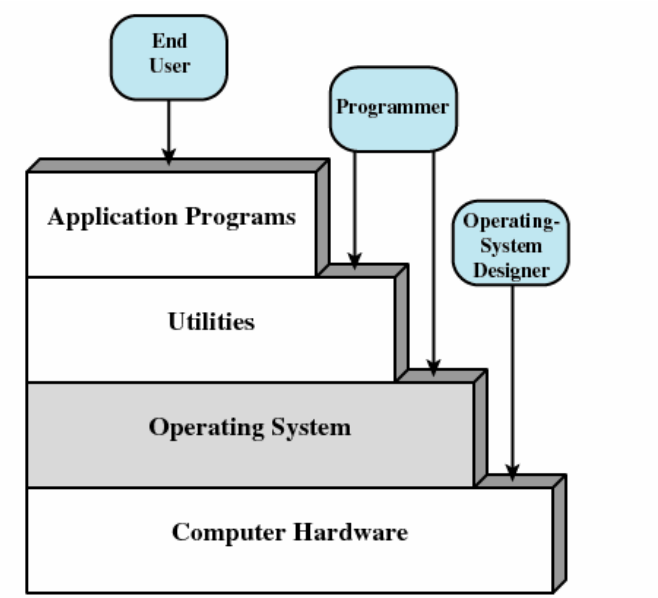


Figure 2.1 Layers and Views of a Computer System

- Typical services:
  - Program execution (and program development)
  - Access to I/O-devices
  - Controlled access to files
  - System access / access control
  - Error detection and error handling
  - Logging

Managing Resources

- Computer consists of resources.
  - Transfer, storage and processing of Data
  - Control of these functions
- Operating system manages these resources.
  - Program, like user applications, but with direct access to hardware
  - Allows other programs to access hardware (CPU, ...)

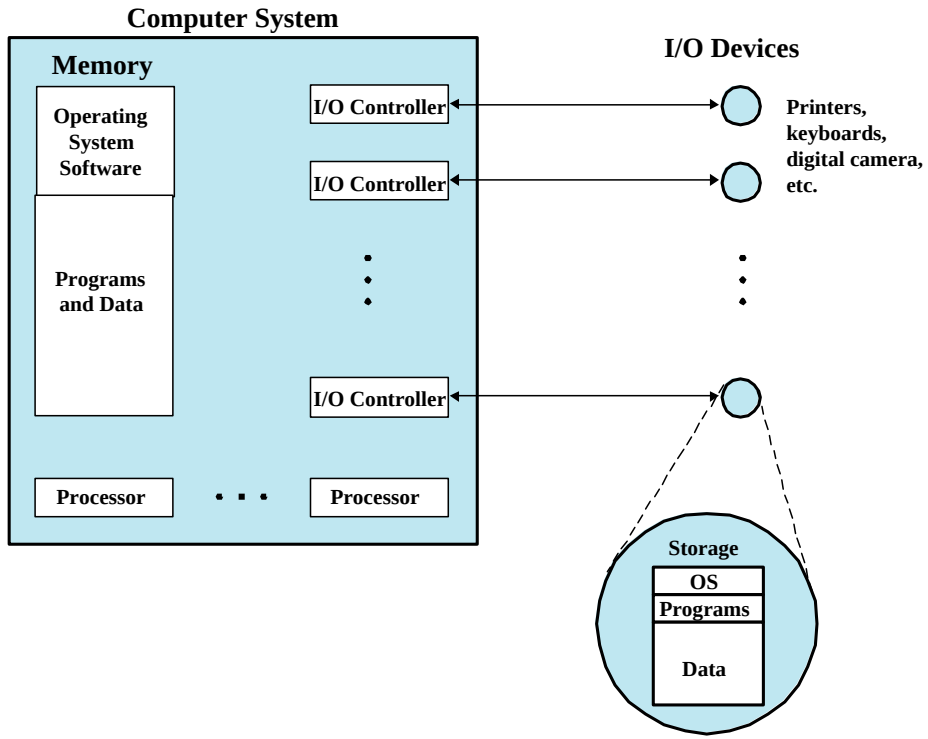


Figure 2.2 The Operating System as Resource Manager

Virtual Resources

- Virtual resources instead of real resources
  - Processes — instead of (only one) processor
  - Virtual Memory — instead of (limited) RAM
  - File — instead of persistent memory (hard disk, ...)
- Advantages:
  - easy to use — through high-level, procedural interface
  - secure — against hardware and software errors / manipulation
  - efficient — as compared to available real resources

Virtual Resources - Processes

- Multitasking (multiprogramming):
  - Number of processes is not limited by the number of processors
  - Terminal and client/server operations
  - Efficient utilization of hardware in light of different usage patterns of user applications

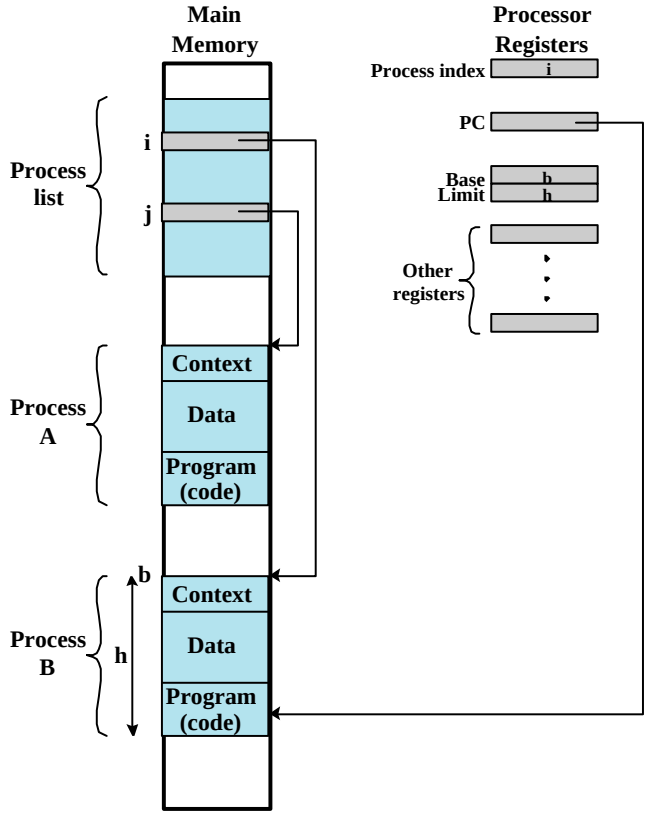


Figure 2.8 Typical Process Implementation

- Virtual Memory:
  - Address space of a process larger than available physical memory
  - May be stored in secondary storage

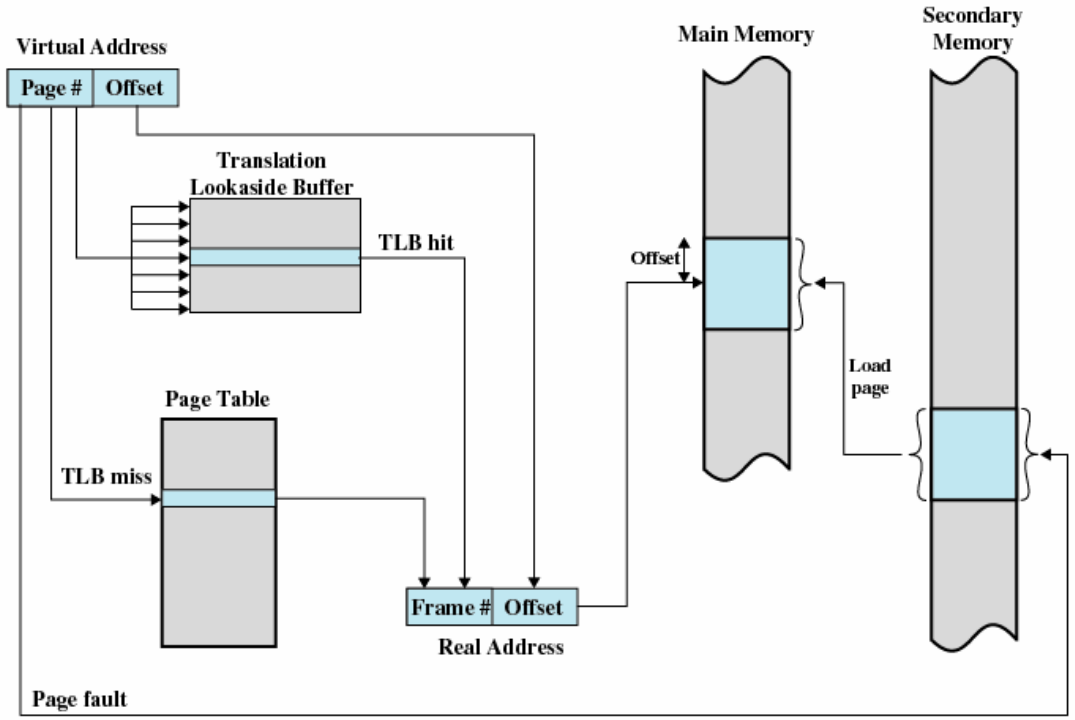


Figure 8.7 Use of a Translation Lookaside Buffer

Virtual Resources - Files

- Files
  - Named, persistent objects
  - Stored on secondary storage (tape, hard disk, floppy disk, USB disk drive)

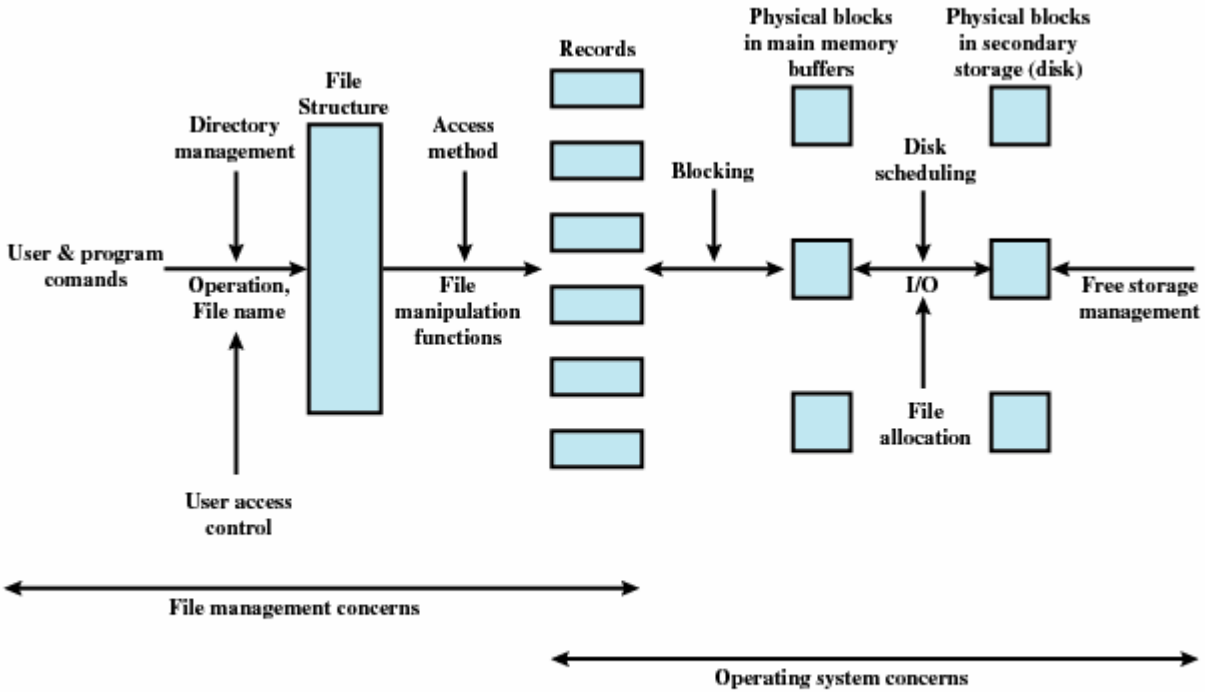


Figure 12.2 Elements of File Management

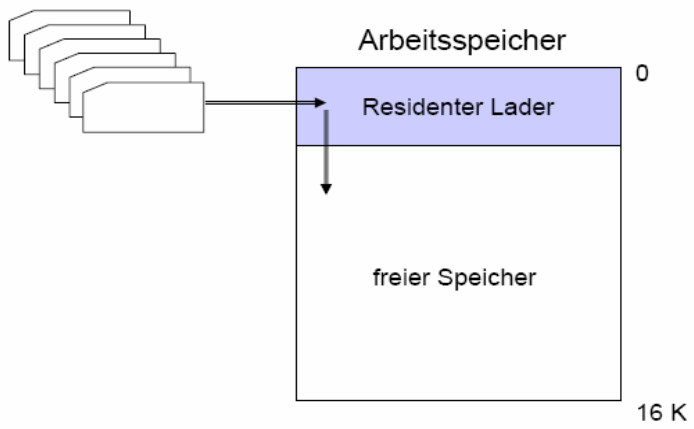
A Word About Synchronization

- Concurrency handling is outside the scope of this lecture.
  - Algorithmen und Programmierung IV (SS07)

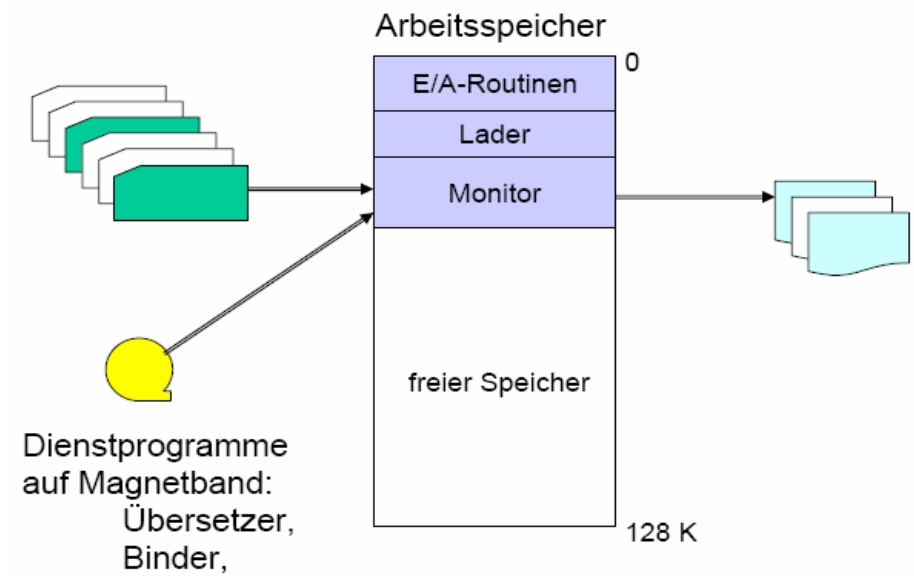
Some pointers

- Disabling interrupts:
  - Implemented in hardware, enabled/disabled via special instructions
  - Allows other interrupt handlers to run to completion
- Spinlocks:
  - Short-term synchronization mechanism in software
  - Low overhead, avoid re-scheduling
- Semaphores:
  - Long-term synchronization mechanism in software
  - Better suited for longer delays or events

History of Operating Systems



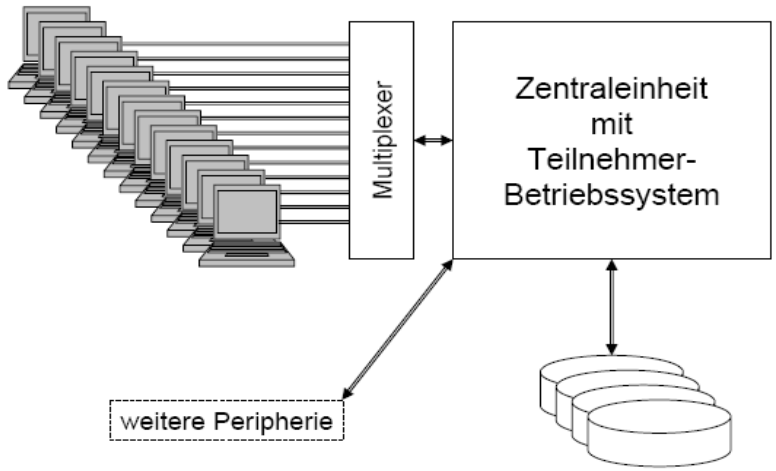
- Loader (1950, e.g. IBM 704)
  - Loads programs into memory



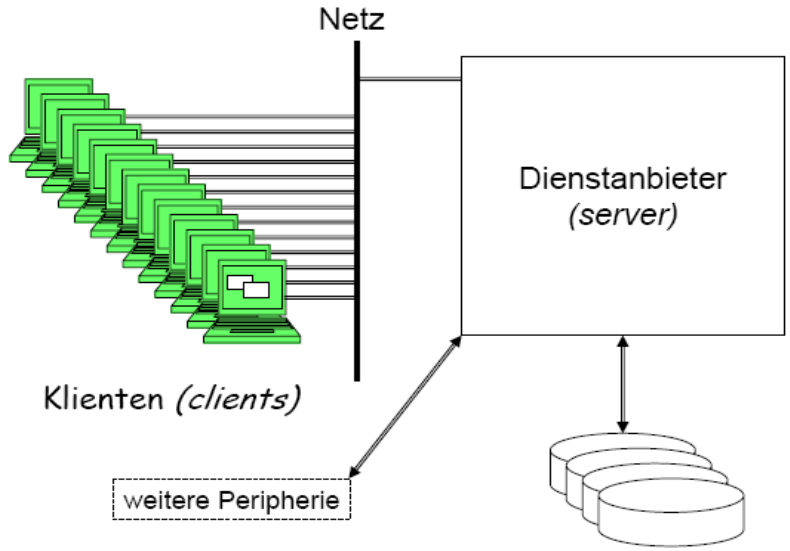
- Batch System (1960, e.g. IBM 7090, Zuse Z 23, Telefunken TR4)
  - Processing of jobs stored on punch cards
  - Manual job control by human operator

History of Operating Systems

- Multi-User / Time Sharing Systems (1970, e.g. IBM OS/360, TSS, T.H.E., Multics, UNIX)
  - Many terminals connected to one computer
  - Interactive control for users
  - Multitasking

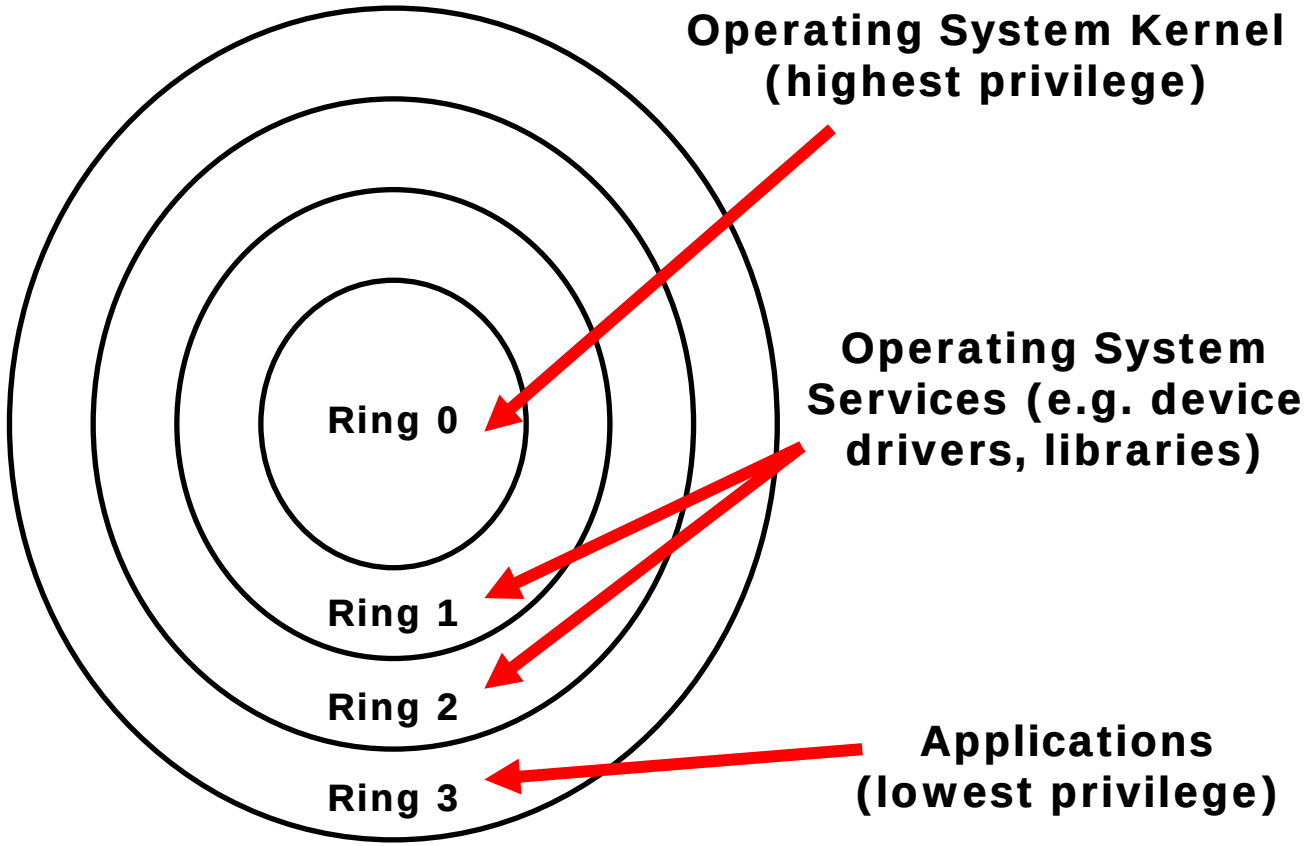


- Personal Computing und Client/Server (1980/90, e.g. Apple Lisa, MS Windows, Linux, Solaris, HP-UX, ...)
  - Intelligent workstations
  - GUI / Window mode



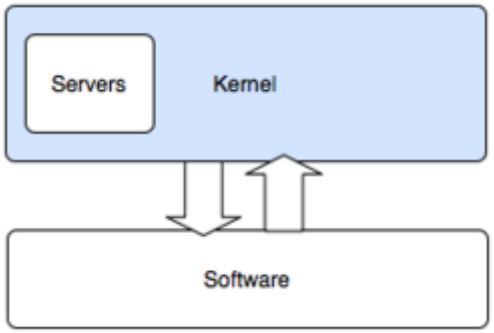
Security: Protection Rings

- Hardware-enforced hierarchical privilege levels:
  - Inner rings have access to outer ring's resources,
  - Outer rings may access inner rings through predefined gateways.



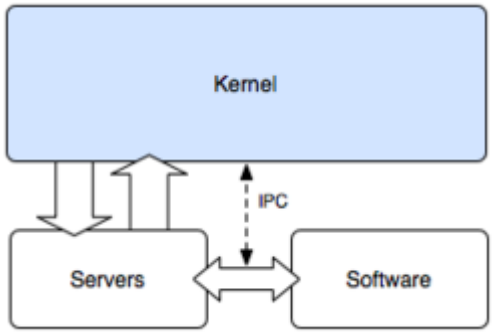
Conceptual Perspective

- Monolithic Kernel

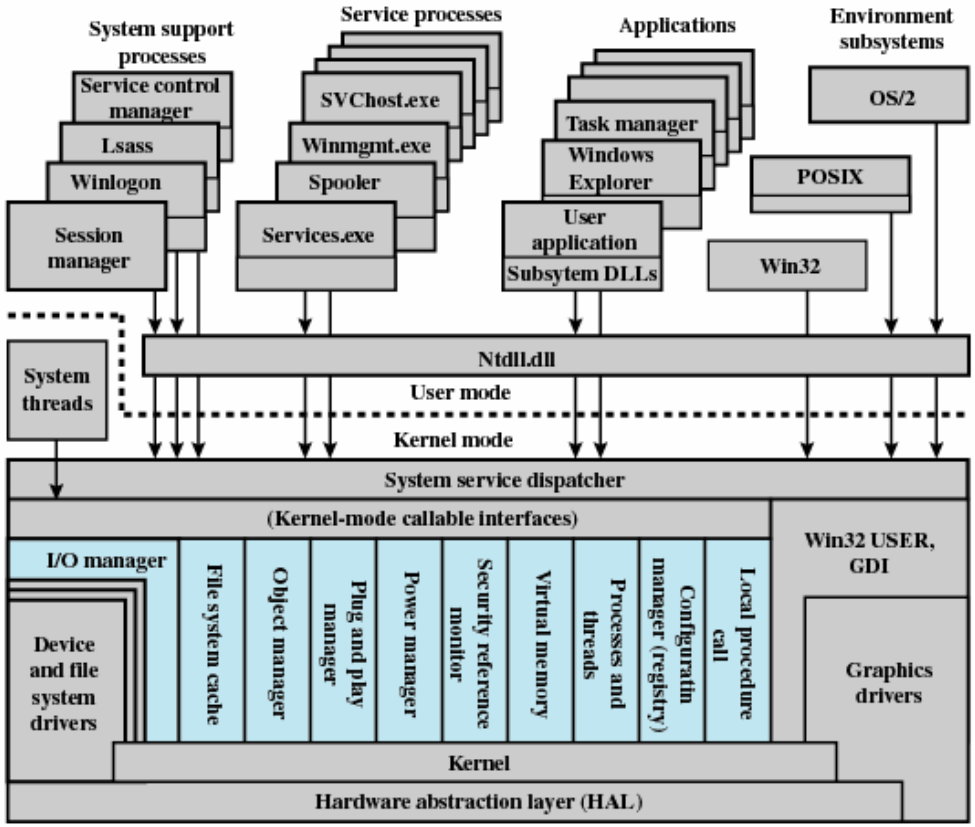


- Pro:
  - Less context switches
  - No expensive communication
- Contra:
  - Complications when exchanging functionality
  - Cost of development

- Microkernel



- Pro:
  - Strict interfaces, components
  - Less Complexity
- Contra:
  - Speed
  - Synchronization



Lsass = local security authentication server
POSIX = portable operating system interface
GDI = graphics device interface
DLL = dynamic link libraries

Colored area indicates Executive

Figure 2.13 Windows 2000 Architecture [SOLO00]

UNIX (traditional)

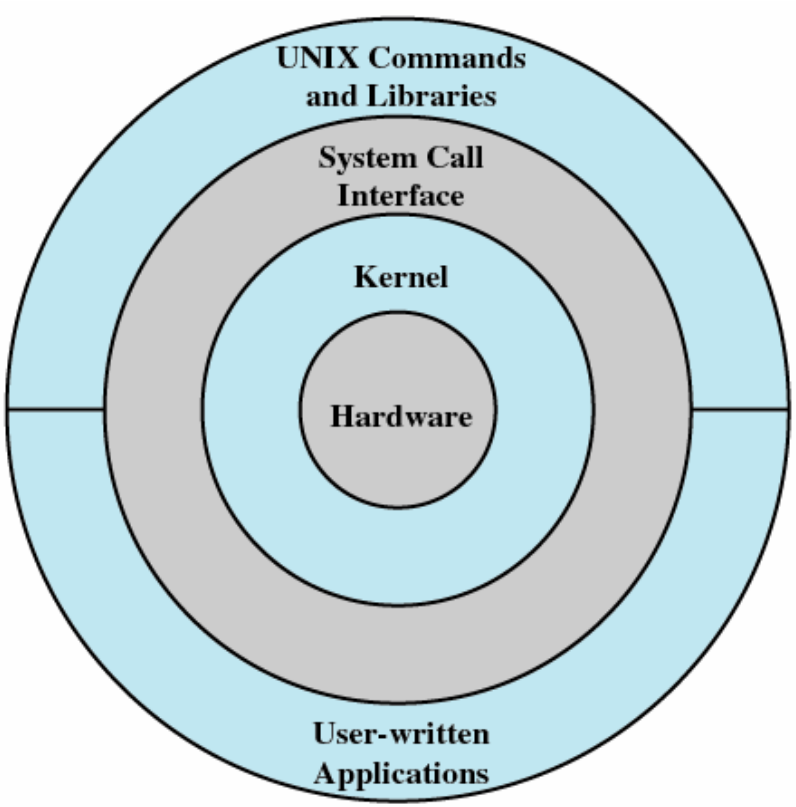


Figure 2.14 General UNIX Architecture

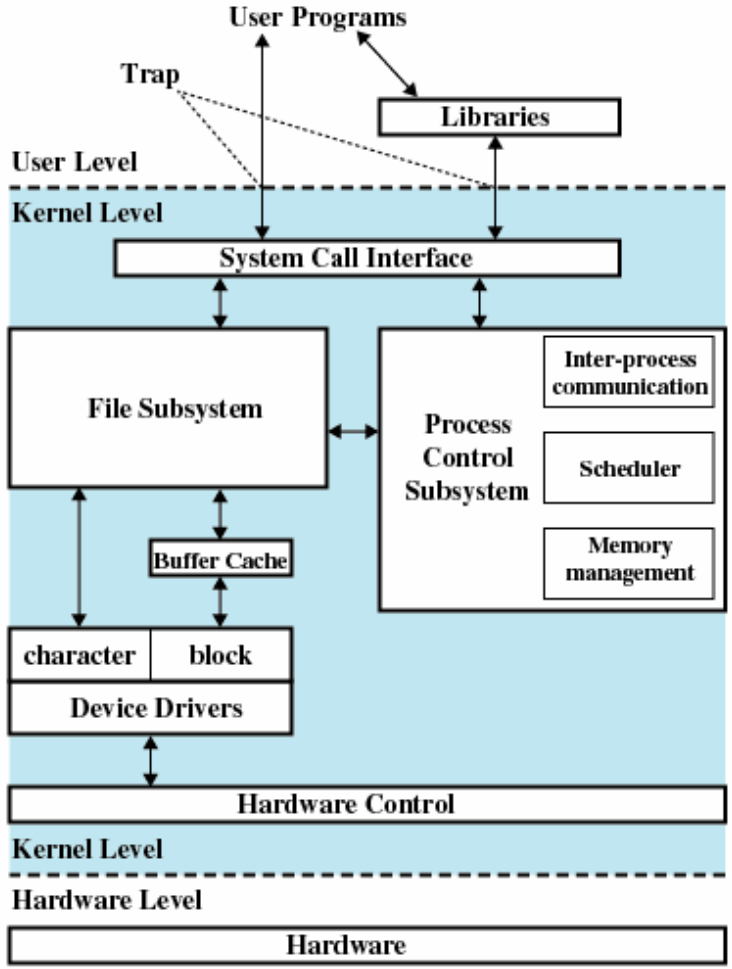


Figure 2.15 Traditional UNIX Kernel [BACH86]

UNIX (modern)

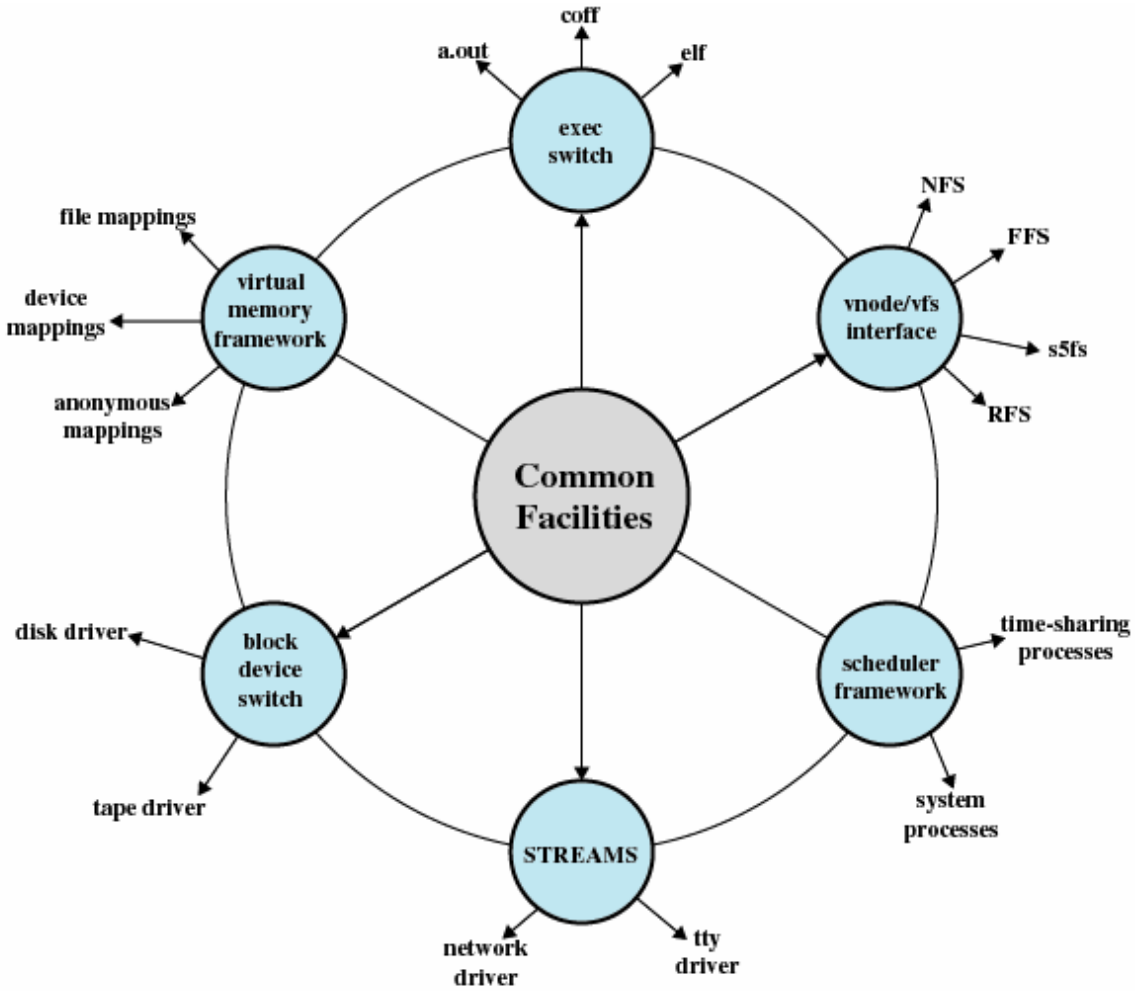
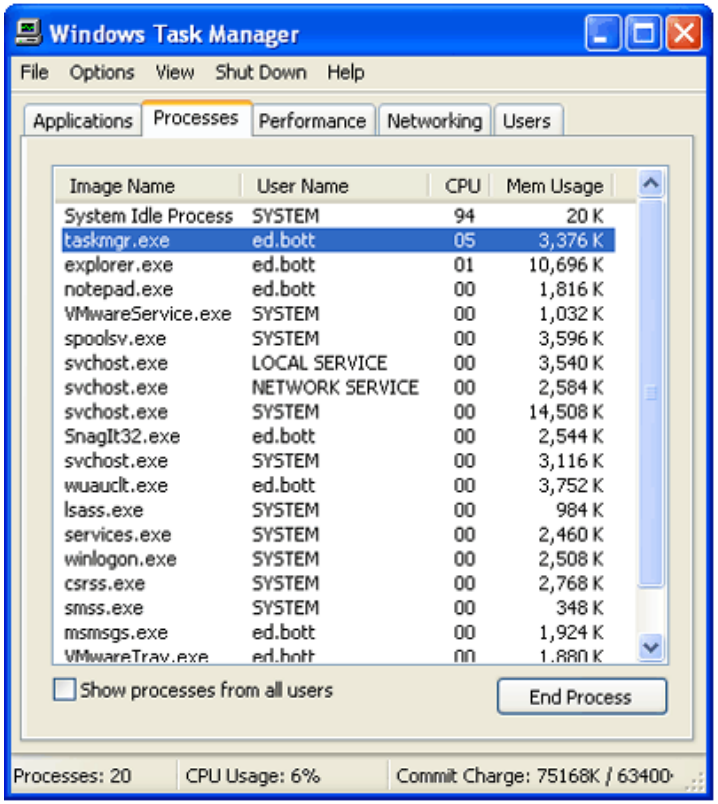


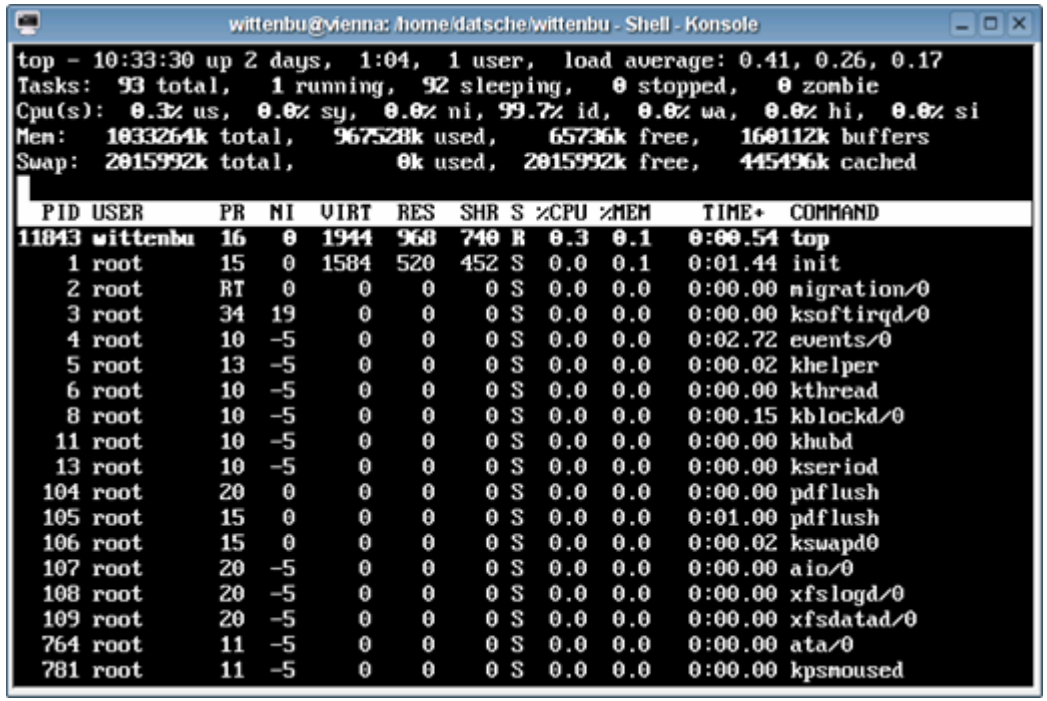
Figure 2.16 Modern UNIX Kernel [VAHA96]

Common Tools

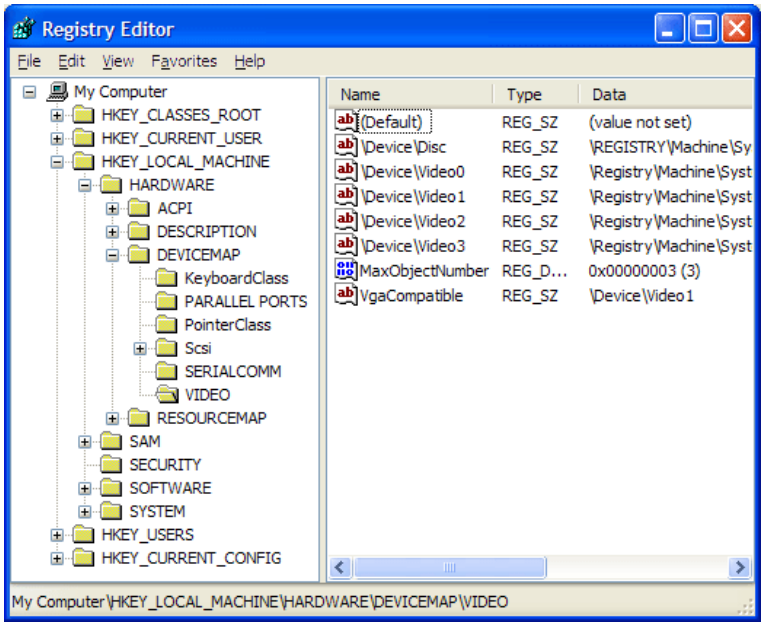
- Graphical User Interface (GUI) / Text Mode (Shell)
- Process monitor:
  - Task Manager
  - top
- Kernel parameters:
  - Registry
  - /proc, /sys
- Tool chain:
  - Programming languages (e.g. C), Compiler, Linker, (Boot-)Loader

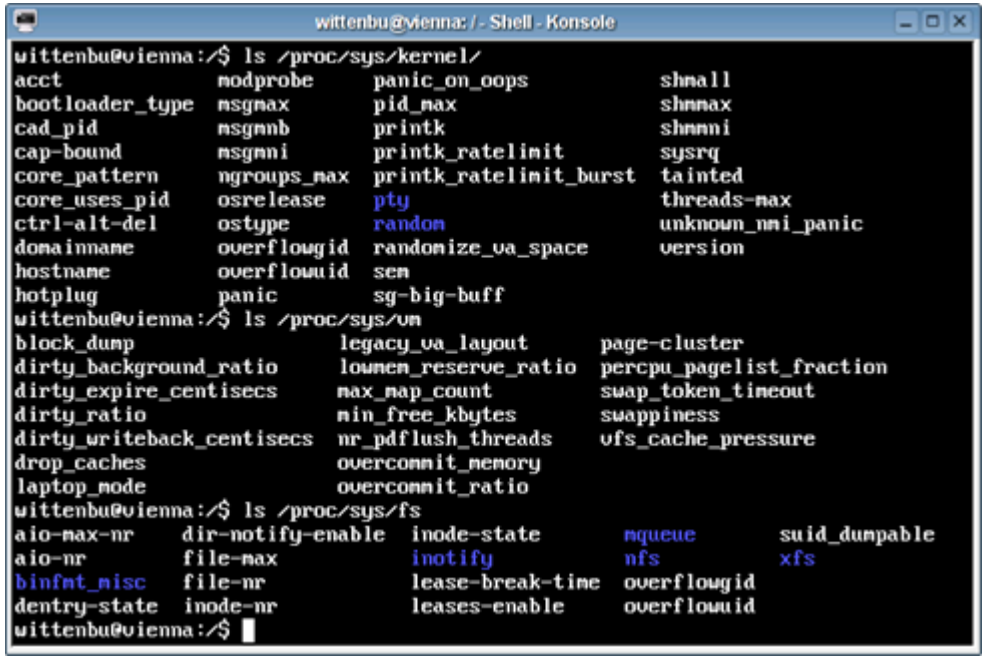
Process Monitor



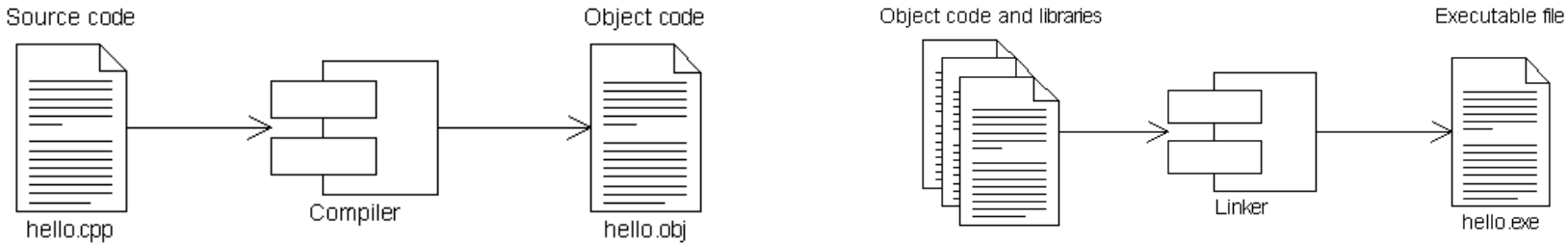


Kernel Parameters





Tool Chain




```
wittenbu@vienna: /home/datsche/wittenbu/hello - Shell - Konsole
wittenbu@vienna:~/hello$ ls hello.c
hello.c
wittenbu@vienna:~/hello$ file hello.c
hello.c: ASCII C program text
wittenbu@vienna:~/hello$ cat hello.c
#include <stdio.h>

main() {
        printf ("Hello World!\n");
}
wittenbu@vienna:~/hello$ cc -c hello.c
wittenbu@vienna:~/hello$ ls
hello.c  hello.o
wittenbu@vienna:~/hello$ file hello.o
hello.o: ELF 32-bit LSB relocatable, Intel 80386, version 1 (SYSV), not stripped
wittenbu@vienna:~/hello$ cc hello.o -o hello
wittenbu@vienna:~/hello$ ls
hello  hello.c  hello.o
wittenbu@vienna:~/hello$ file hello
hello: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2
.2.0, dynamically linked (uses shared libs), not stripped
wittenbu@vienna:~/hello$ ./hello
Hello World!
wittenbu@vienna:~/hello$
```