Operating Systems & Computer Networks
1. Introduction & Motivation

Dr. Larissa Groth
Computer Systems & Telematics (CST)

Roadmap

1. **Introduction and Motivation**
2. Interrupts and System Calls
3. Processes
4. Scheduling
5. Memory
6. I/O and File System
7. Booting, Services, and Security

Lernziele

Sie nennen:

- die Aufgaben und Ziele eines Betriebssystems
- welche realen Ressourcen durch welche virtuellen Ressourcen abgebildet werden

Sie nennen und beschreiben:

- die Schnittstelle zwischen Betriebssystem und Anwendungsschicht

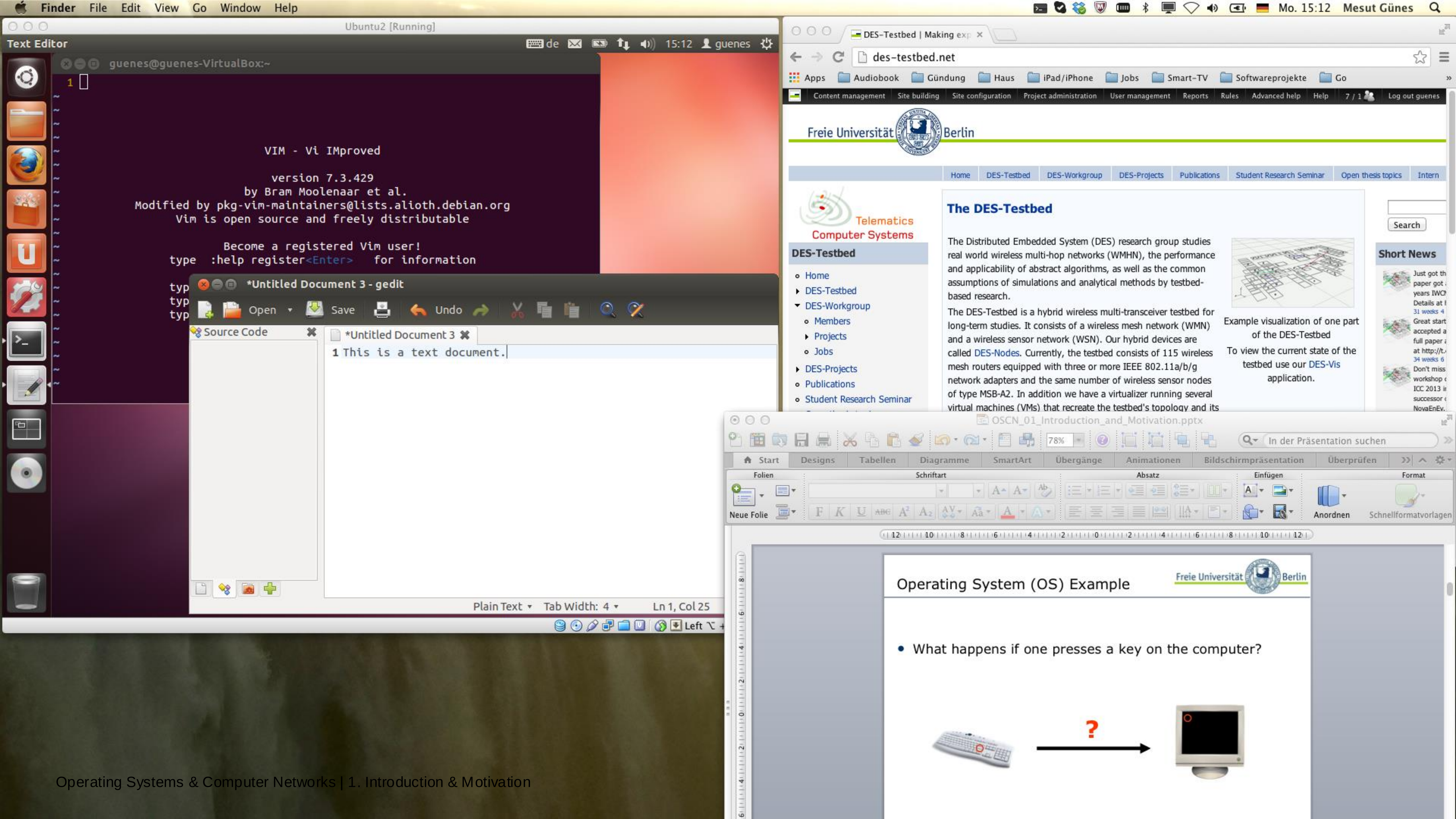
Sie beschreiben:

- was unter Protection Rings zu verstehen ist und wie sie mit der Microkernel-vs-Monolithischer-Kernel-Diskussion zusammenhängen
- was unter einem Microkernel BS zu verstehen ist
- was unter einem Monolithischen Kernel BS zu verstehen ist

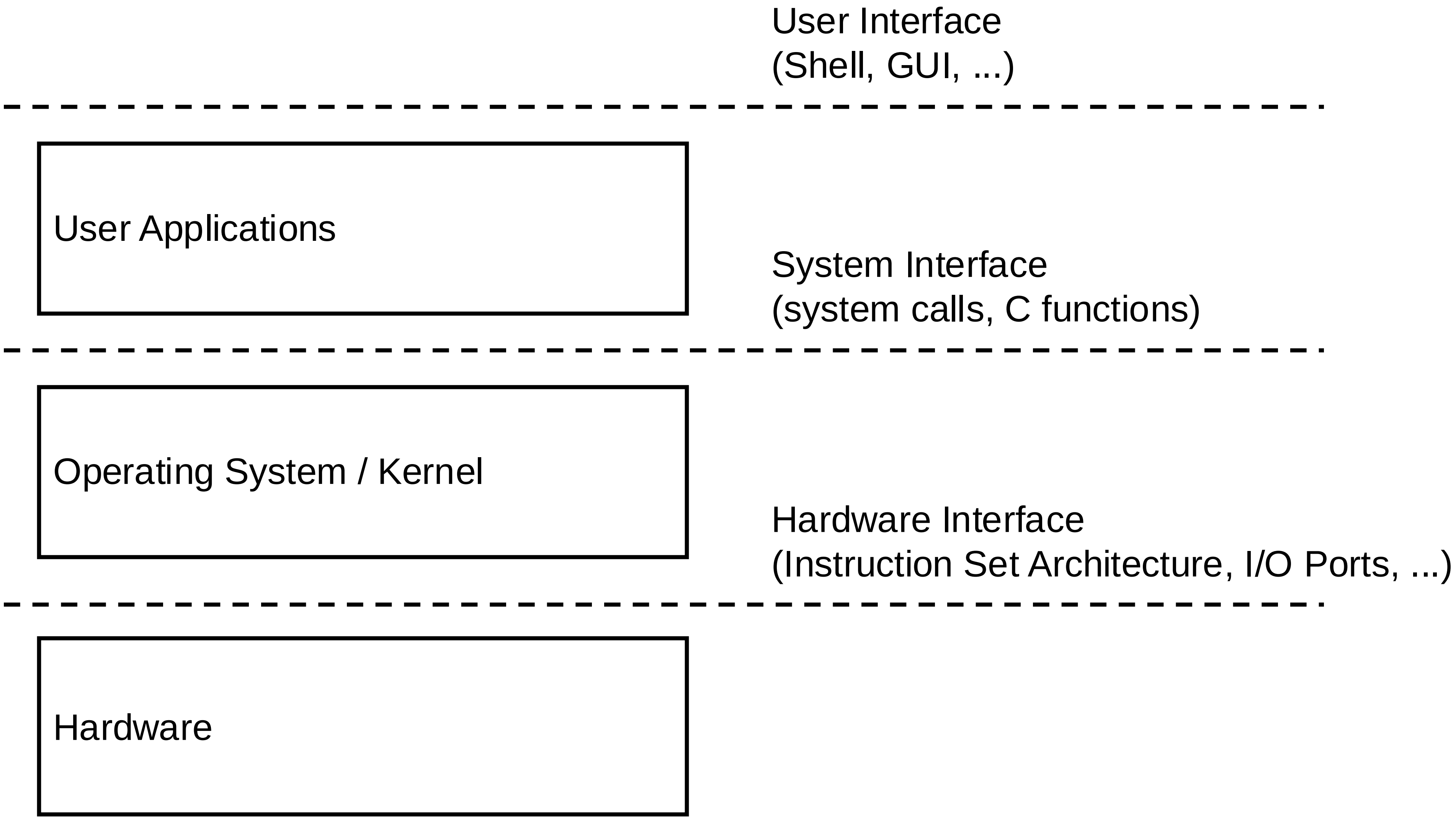
Sie wägen die Vor- und Nachteile ab:

- zwischen Microkernel und Monolithischem Kernel

Motivation


Finder File Edit View Go Window Help
Mo. 15:12 Mesut Günes
Ubuntu2 [Running]
Text Editor
15:12 guenes
guenes@guenes-VirtualBox:~
VIM - Vi IMproved
version 7.3.429
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable
Become a registered Vim user!
type :help register<Enter> for information
*Untitled Document 3 - gedit
Open
Save
Undo
Source Code
*Untitled Document 3
This is a text document.
Plain Text
Tab Width: 4
Ln 1, Col 25
DES-Testbed | Making exp
des-testbed.net
Apps Audiobook Gündung Haus iPad/iPhone Jobs Smart-TV Softwareprojekte Go
Content management Site building Site configuration Project administration User management Reports Rules Advanced help Help Log out guenes
Freie Universität Berlin
Home DES-Testbed DES-Workgroup DES-Projects Publications Student Research Seminar Open thesis topics Intern
Telematics
Computer Systems
DES-Testbed
Home
DES-Testbed
DES-Workgroup
Members
Projects
Jobs
DES-Projects
Publications
Student Research Seminar
The DES-Testbed
The Distributed Embedded System (DES) research group studies real world wireless multi-hop networks (WMHN), the performance and applicability of abstract algorithms, as well as the common assumptions of simulations and analytical methods by testbed-based research.
The DES-Testbed is a hybrid wireless multi-transceiver testbed for long-term studies. It consists of a wireless mesh network (WMN) and a wireless sensor network (WSN). Our hybrid devices are called DES-Nodes. Currently, the testbed consists of 115 wireless mesh routers equipped with three or more IEEE 802.11a/b/g network adapters and the same number of wireless sensor nodes of type MSB-A2. In addition we have a virtualizer running several virtual machines (VMs) that recreate the testbed's topology and its
Example visualization of one part of the DES-Testbed
To view the current state of the testbed use our DES-Vis application.
Search
Short News
OSCN_01_Introduction_and_Motivation.pptx
In der Präsentation suchen
Start Designs Tabellen Diagramme SmartArt Übergänge Animationen Bildschirmpräsentation Überprüfen
Folien Schriftart Absatz Einfügen Format
Neue Folie
Anordnen
Schnellformatvorlagen
Operating System (OS) Example
Freie Universität Berlin
What happens if one presses a key on the computer?
?

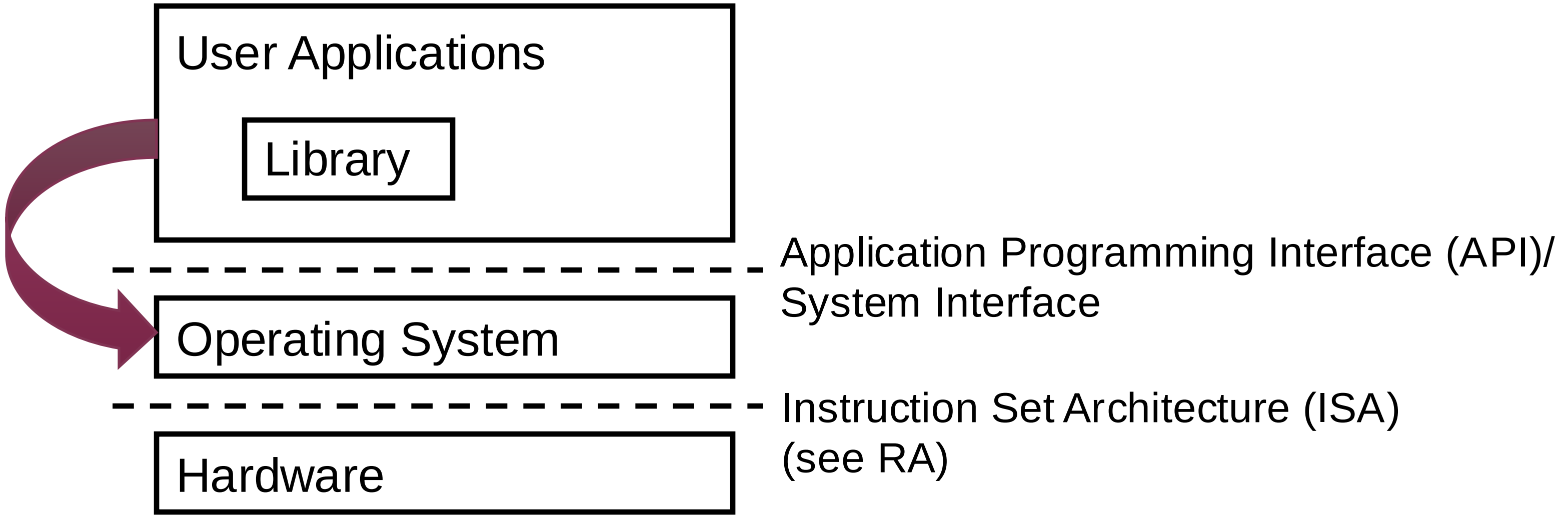
Layers of Abstraction



System Interface and 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) → POSIX.



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

POSIX

Portable Operating System Interface (POSIX)

- E.g. https://standards.ieee.org/standard/1003_1,2013Edition.html

POSIX defines

- Application programming interface (API)
- Command line shells
- Utilities

UNIX like Operating Systems

POSIX oriented operating systems

- Unix
- Linux
- Windows
- Mac OS X
- …

Tasks of an Operating System

Typical services of a **general** purpose OS includes:

- Program execution
- Access to I/O-devices
  - Hardware abstraction
- Controlled access to files
  - Non-volatile memory
- Access control
  - Security / user management
- Error detection and error handling
  - Both hardware and software
- Logging

Special purpose operating systems focus on different services, e.g., real-time or communication requirements.

See: Stallings, W. (2017). Operating Systems. (9th ed.). Pearson International, chapter 2.1, p. 70

Goals of an Operating System

Convenience
→ ease of use for users and programmers

Efficiency
→ when managing limited resources

Ability to evolve
→ New hardware standards
→ Changing user requirements

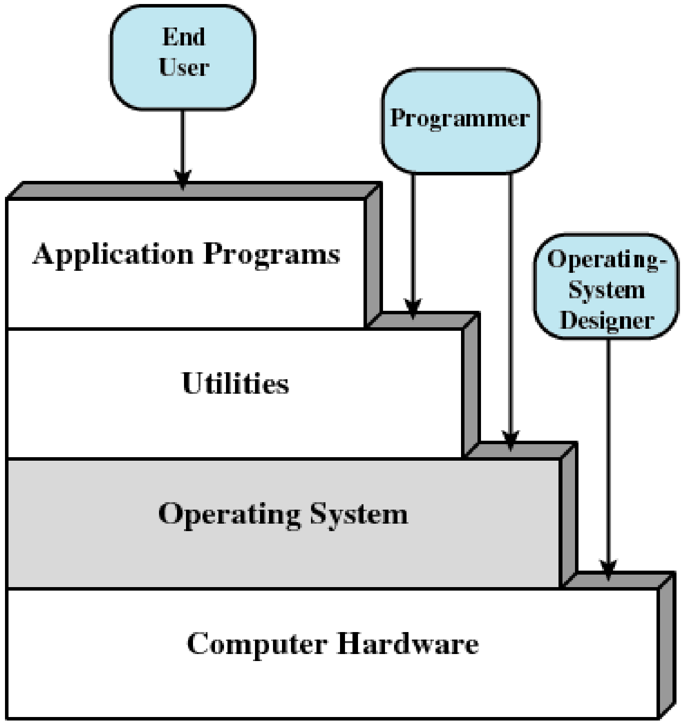


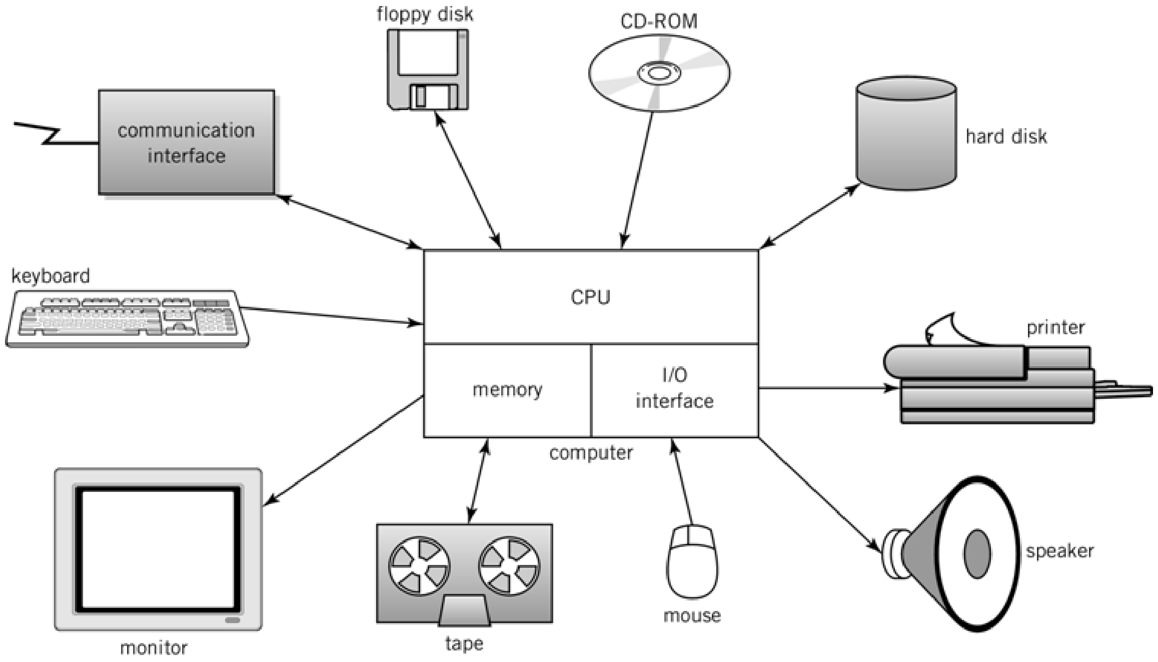
Figure 2.1 Layers and Views of a Computer System

See: Stallings, W. (2017). Operating Systems. (9th ed.). Pearson International, chapter 2.1, p. 69

Managing Resources

Hardware provides the basic computing resources such as

- Processor(s)
- Memory
- Persistent storage
- Network connection



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 1, Figure 01-06

OS virtualizes resources to permit controlled sharing and isolation

- virtual instances of a resource are created

OS provides virtual resources for user applications

Virtual Resources

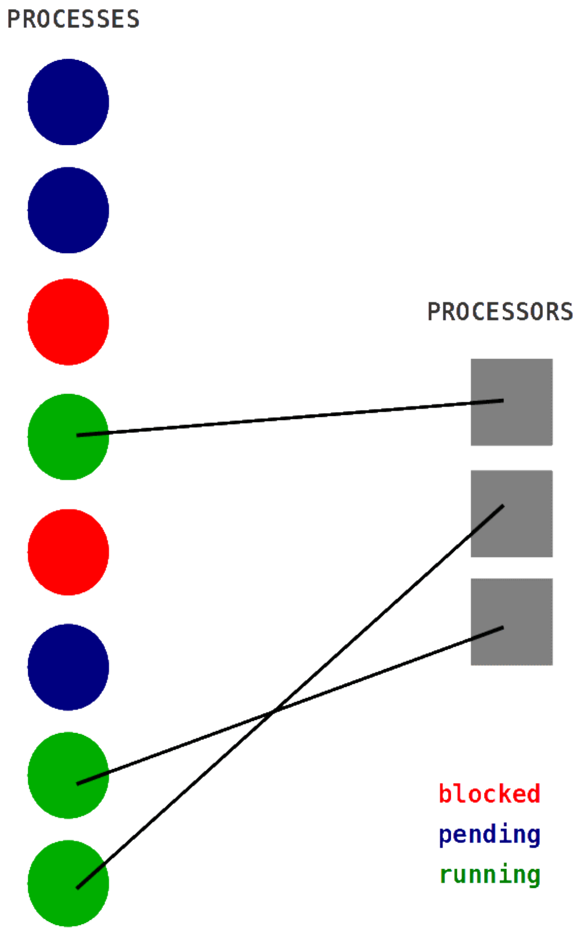
Virtual resources and corresponding real resources:

• Processes	processor(s)
• Virtual Memory	main memory
• Files	persistent memory
• Ports	network adapter

Advantages:

- Easy to use through procedural interface (system calls)
- Secure against hardware and software errors or manipulation

Processes



Number of processes is not limited by the number of processors: Multitasking

Processor is used efficiently:

- Time is not wasted by processes that are waiting on I/O devices

Reduced latency (=response time)

Different **process states**, e.g.,

- running – executing
- pending – ready to execute
- blocked – not ready to execute

Virtual Memory

Managed by the Memory Management Unit (MMU)

Transportability:

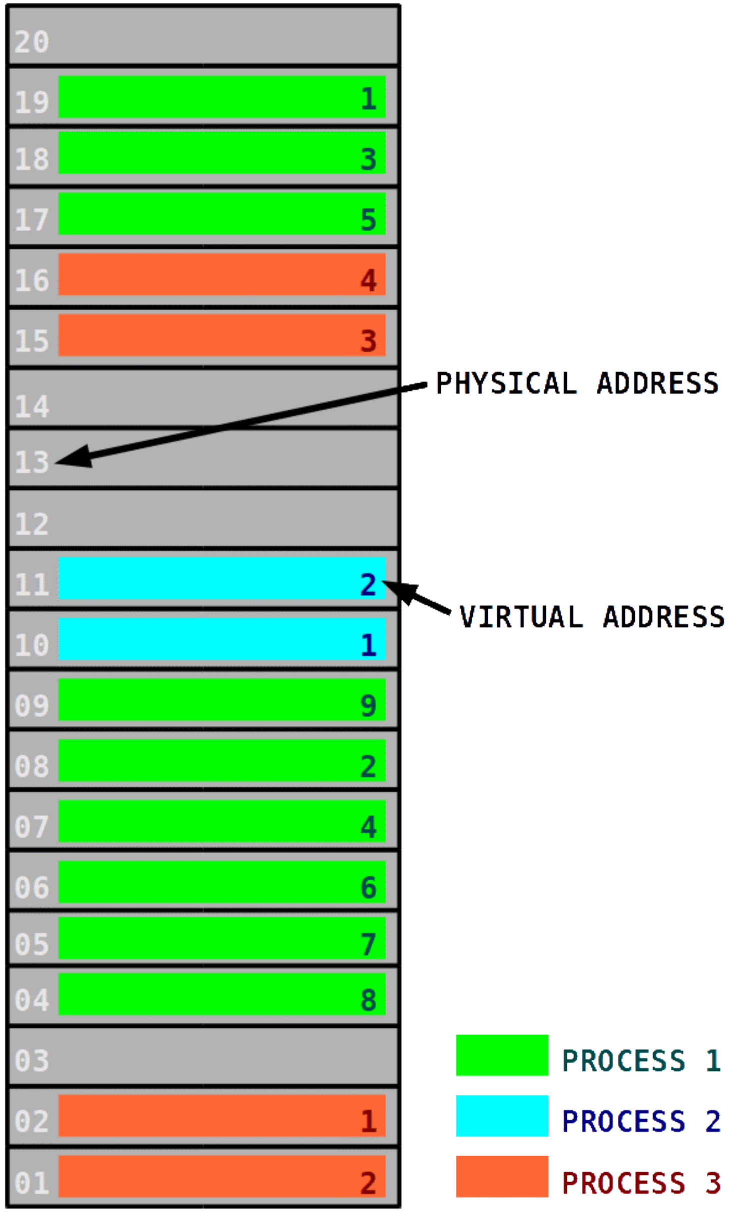
- position independent code – program does not depend on
  memory architecture

Security:

- memory access is restricted to memory units “owned” by a process

Efficiency:

- external fragmentation is avoided



Files

Managed by a file system

Persistent objects for long-term data storage

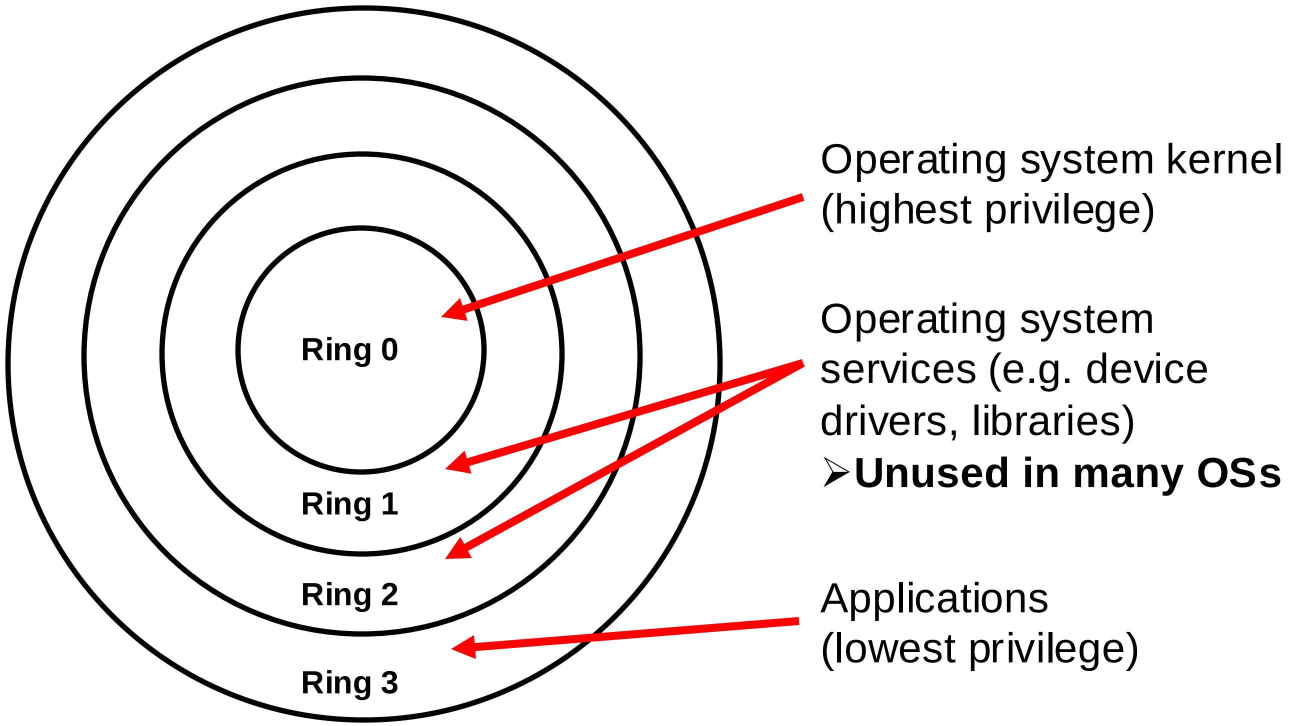
Stored in secondary memory (e.g., tape, hard disk, USB flash drive)

Similar to virtual memory - file name instead of virtual address

Protection Rings

Hardware provides hierarchical privilege levels

- Inner rings have access to outer rings’ resources
- Outer rings may access inner rings through predefined gateways



Operating System Kernel / Ring 0

Kernel implements basic layer of abstraction

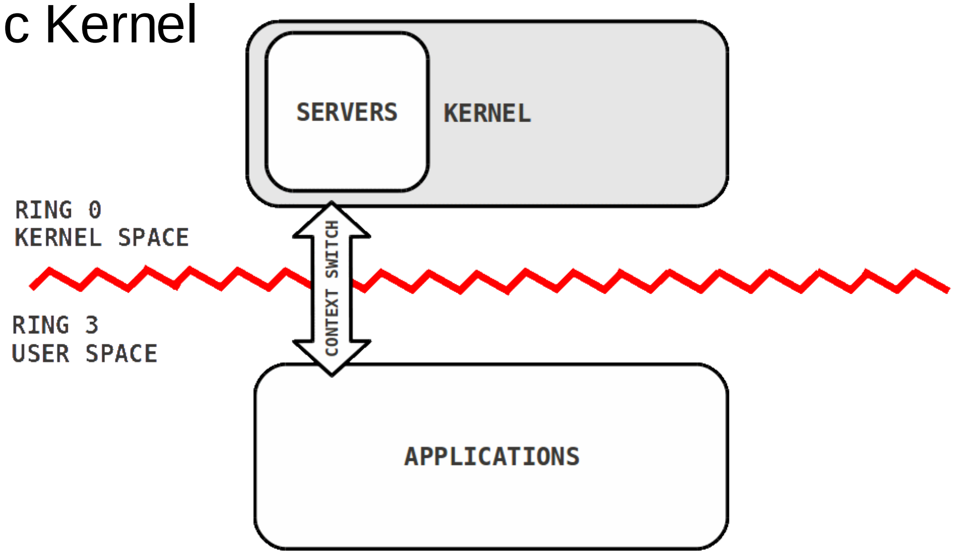
Runs with full access to hardware (Ring 0)

Context Switch: switching from one process to another

- A certain amount of time is required for doing the administration, e.g., saving and loading registers.

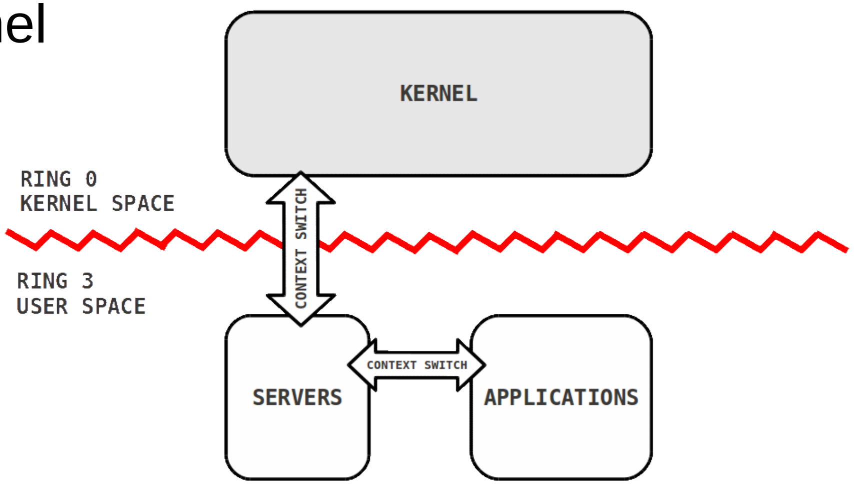
Monolithic versus Microkernel

Monolithic Kernel



- Implemented as single process
- Sharing same kernel address space

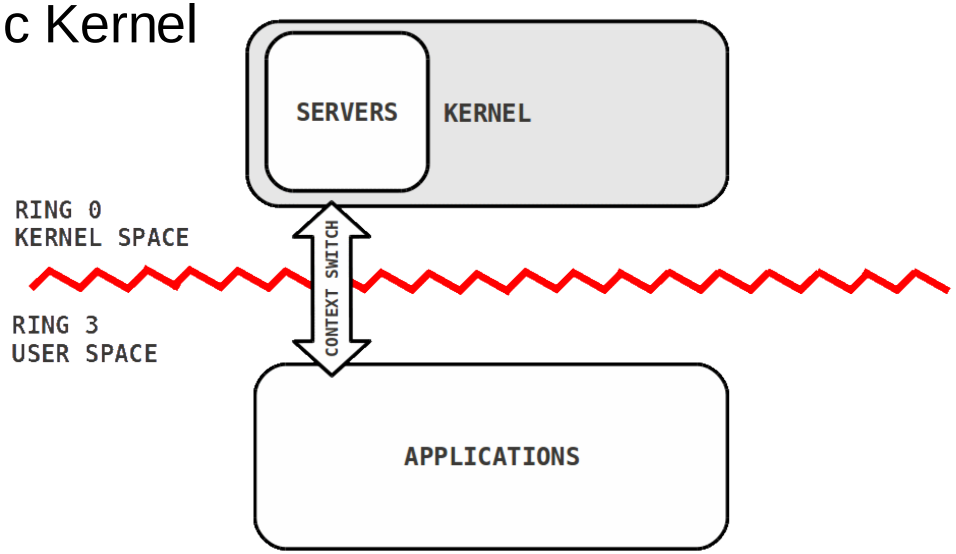
Microkernel



- Very few dedicated functions in kernel:
  - address space management
  - interprocess communication (IPC)
  - basic scheduling
- All other OS services are processes in user space

Monolithic versus Microkernel

Monolithic Kernel



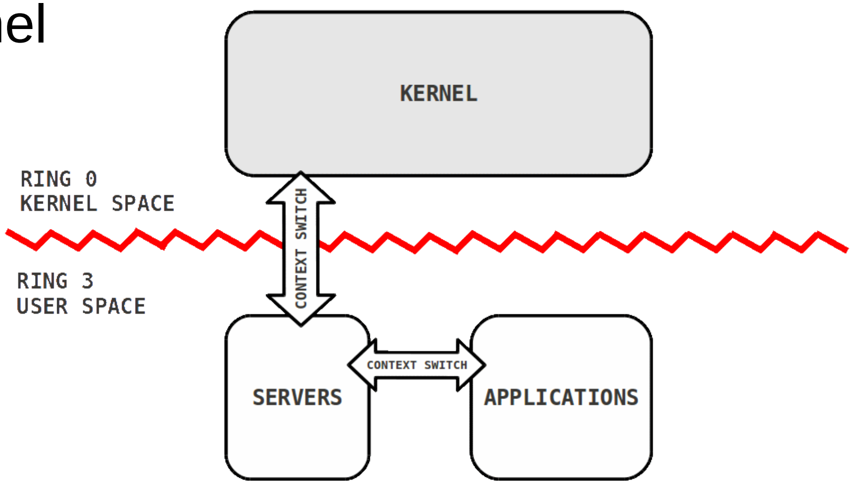
Pro:

- less context switches
- no expensive communication

Contra:

- complications when exchanging functionality

Microkernel



Pro:

- strict interfaces
- less complexity, clear structure

Contra:

- speed
- synchronization

Roadmap

1. **Introduction and Motivation**
2. Interrupts and System Calls
3. Processes
4. Scheduling
5. Memory
6. I/O and File System
7. Booting, Services, and Security