Operating Systems WS 2014/2015

Prof. Dr. Neeraj Suri

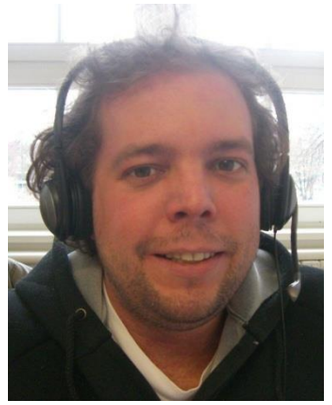
DEEDS (Dependable Systems & SW Group)

www.deeds.informatik.tu-darmstadt.de

Stefan Winter

Thorsten Piper

Habib Saissi



Daniel Germanus

Oliver Schwahn

DEEDS Group

Contact Info

- Course homepage:

http://www.deeds.informatik.tu-darmstadt.de/teaching/courses/ws20142015/operating-systems/

- Contact Info:

Person	Email Address	Office Hours
Prof. Dr. Neeraj Suri	suri@cs.tu-darmstadt.de	by appointment
Daniel Germanus	germanus@cs.tu-darmstadt.de	by appointment
Stefan Winter	sw@cs.tu-darmstadt.de	by appointment
Thorsten Piper	piper@cs.tu-darmstadt.de	by appointment
Habib Saissi	saissi@deeds.informatik.tu-darmstadt.de	by appointment
Oliver Schwahn	os@cs.tu-darmstadt.de	by appointment

Teaching Event	Reason	Email Address
OS Course	Generic Questions	teaching@deeds.informatik.tu-darmstadt.de
	Lab Questions	os-lab@deeds.informatik.tu-darmstadt.de
Seminars	Generic Seminar Questions	seminars@deeds.informatik.tu-darmstadt.de

You've received an email from us and need more clarification? Push "reply all"!

Course Contents

- Classical OS topics
  - Processes, Threads, Memory Management, Scheduling, …
  - You already know that from GdI3/ICS3
    (though may need a little recap? ☺)

- Lecture
  - Compact review of classical topics
  - After that more advanced (& interesting) stuff!

- Exercises
  - In-depth review of classical topics
  - Critical discussions

- Labs
  - Hands-on experience

Course Structure

- **Credit points:**
  - SWS: 5 (2+3)
  - CPs: 8

- **Schedule:**
  - Lecture (Wed, 11.40-13.20, S2|02 C205)
  - Exercise (Thu, 11.40-13.20, S2|02 C110)
    - First exercise: Oct 23rd 2014 → no exercise tomorrow!

- **Exams**
  - Mid-Term exam
  - Final exam

- **Course attendee limit 80 students** (via lottery)

Relevant Literature + Lecture Foils

- ***Modern Operating Systems***; A. Tanenbaum, Prentice Hall
- ***Operating System Concepts***, Silberschatz et al., John Wiley and Sons

- Both books are available (in limited numbers!) from the ULB library
  - http://www.ulb.tu-darmstadt.de/ulb
  - Bldg. S1|20, computer science section is on 4th floor

- More reading recommendations (books, papers) will be posted online

- Slides will be made available via TUCaN
  - We will try to upload the foils shortly before the lecture
  - Login with your TU ID and assign to the Operating Systems course
  - Let us instantly know if you have trouble accessing the material!

Labs

- Lab assignments are optional, but you gain bonus points!
- Topics:
  1. Process Management & Linux IPC
     - Fork & wait
     - Pipes, message queues, shared memory
  2. Linux Kernel Modules
     - Introduction to kernel modules
     - /proc FS and /proc/clock device driver
     - FIFO Device Driver
  3. Linux Kernel Modules
     - Producer & Consumer
     - Semaphores, Mutual exclusion
     - Work queues

- Attestation takes place on Nov 21st, Dec 12th and Jan 16th
- Also see the course website and lab instructions for further details!

Bonus points

- Exams ONLY from lecture foils AND material covered in class!!!
- Bonus points (to improve final grade): From (a) Mid-term exam, (b) Labs and (c) Exercises - apply **if** you pass the final exam successfully! ☺

- Total 100 BP:
  - Mid-term exam - 30 BP
  - Labs - 45 BP
  - Exercises - 25 BP

- Increase:
  - 00 - 59 BP -> +0.0 points
  - 60 - 100 BP -> +0.3 points

Optional, yet, we strongly recommend it!

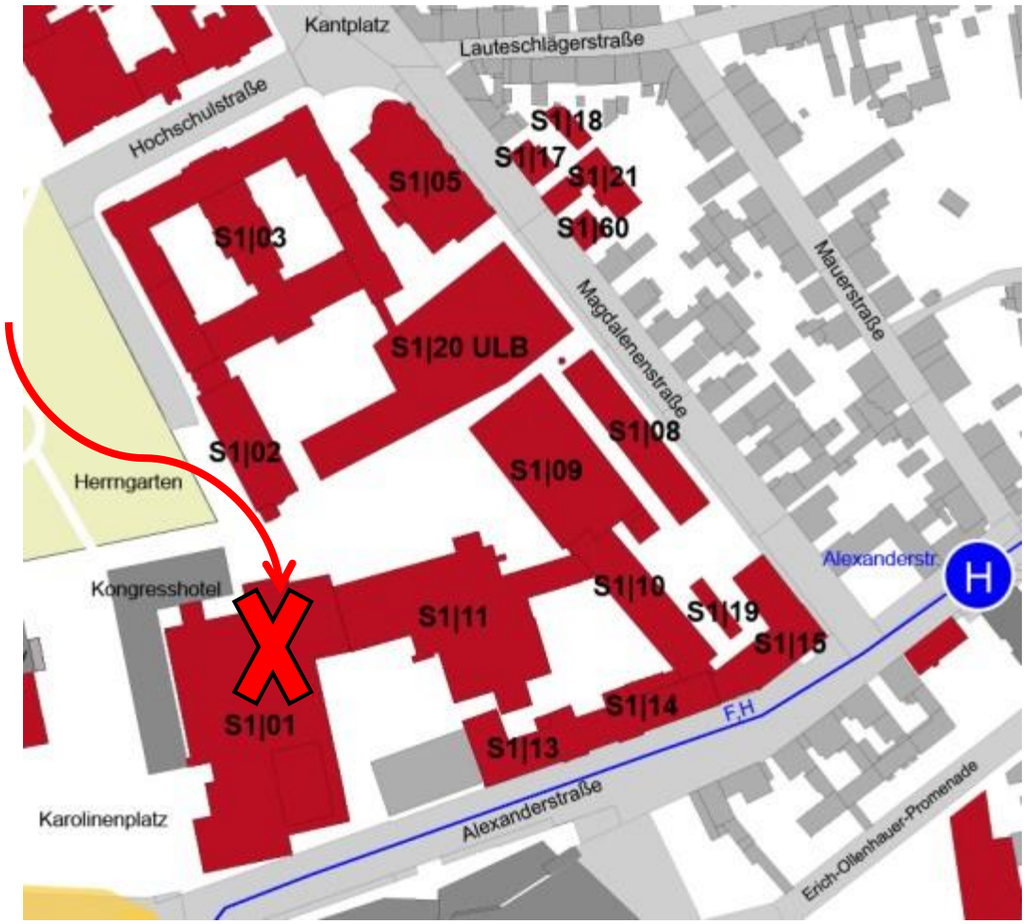
Attend both, lecture & exercises consequently in order to prepare for the exam! (and improve the final grade w/BP)

Prepare for the exercises & discuss!

Exams

- Mid-Term exam
  - Wed Dec 10th, in this lecture hall (and time slot)
  - Be there 10 minutes early!
  - Earn bonus points for the final exam
  - No registration needed

- Final exam
  - Wed Feb 25th, 13:00 – 15:00, S101/A1
  - Be there 10 minutes early!
  - Bonus points are only valid for this exam and not for subsequent ones!
  - Register in time on TUCaN!



Related seminars and projects

- **Related seminars:**
  - **Building and Breaking Complex Software Systems:**

http://www.deeds.informatik.tu-darmstadt.de/teaching/courses/ws20142015/building-and-breaking-os/

  - **Implementing Secure & Reliable Software:**

http://www.deeds.informatik.tu-darmstadt.de/teaching/courses/ws20142015/implementing-secure-reliable-software/

  - **Security and the Cloud - the Issues and Metrics:**

http://www.deeds.informatik.tu-darmstadt.de/teaching/courses/ws20142015/security-and-the-cloud-the-issues-and-metrics/

NEW **Smart Grid Informatics & Trustworthiness:**

http://www.deeds.informatik.tu-darmstadt.de/teaching/courses/ws20142015/smart-grid-informatics-and-trustworthiness/

- **Related HiWi / MSc/ BSc/ Diploma projects examples:**
  - *"Which Operating System is the best?"*
  - *"Workload Impacts on System Robustness"*
  - *"Malicious behavior in P2P systems: Detection & Mitigation "*
  - *"Security Level-based scheduling for IaaS Clouds"*
  - *"Benchmarking Virtual Machines"*
  - More MS/BS: http://www.deeds.informatik.tu-darmstadt.de/deeds/teaching/bsms-theses/
  - More HiWi: http://www.deeds.informatik.tu-darmstadt.de/jobs/

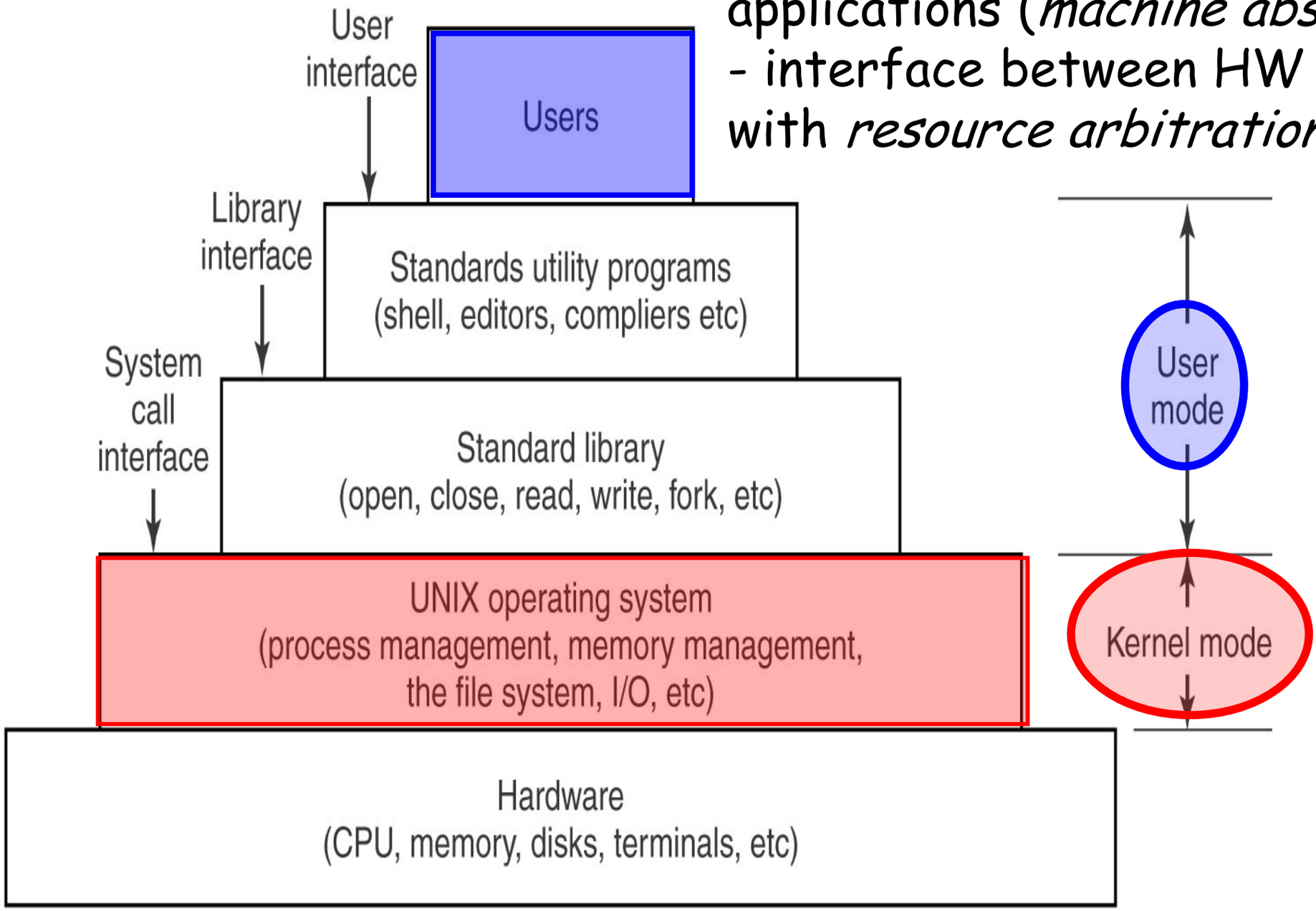
OS History

- Initially there was no OS (just a single program)

- Now:
  - distributed, networked, virtual machines, multi-user, multi-core, smart cards …
  - running everywhere (on mobile phones, tablets, iPods, mainframes, Google, etc.)
  - commercial-off-the-shelf (COTS) products

OS Definition

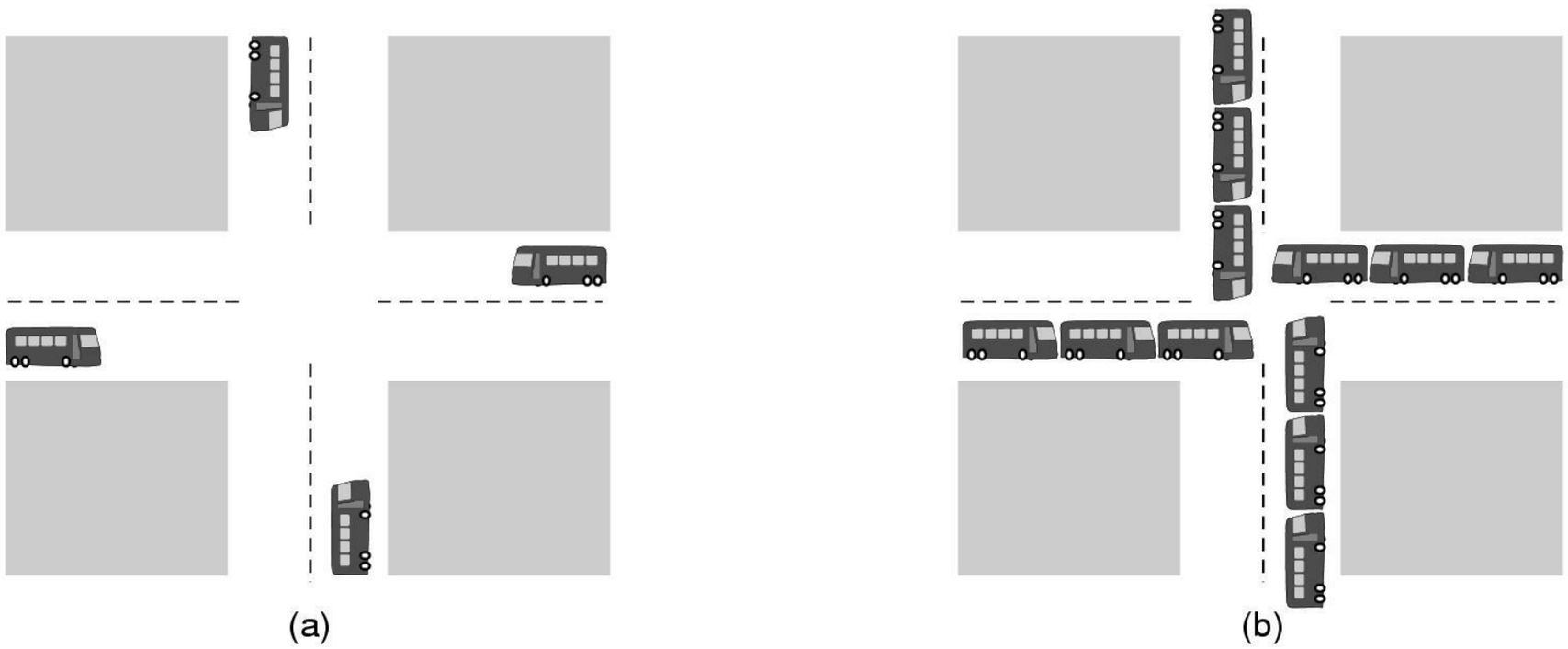
OS: program (or set of programs) providing
- execution environment for user applications (*machine abstraction*)
- interface between HW resources and user with *resource arbitration/management*



Operating System Functionality

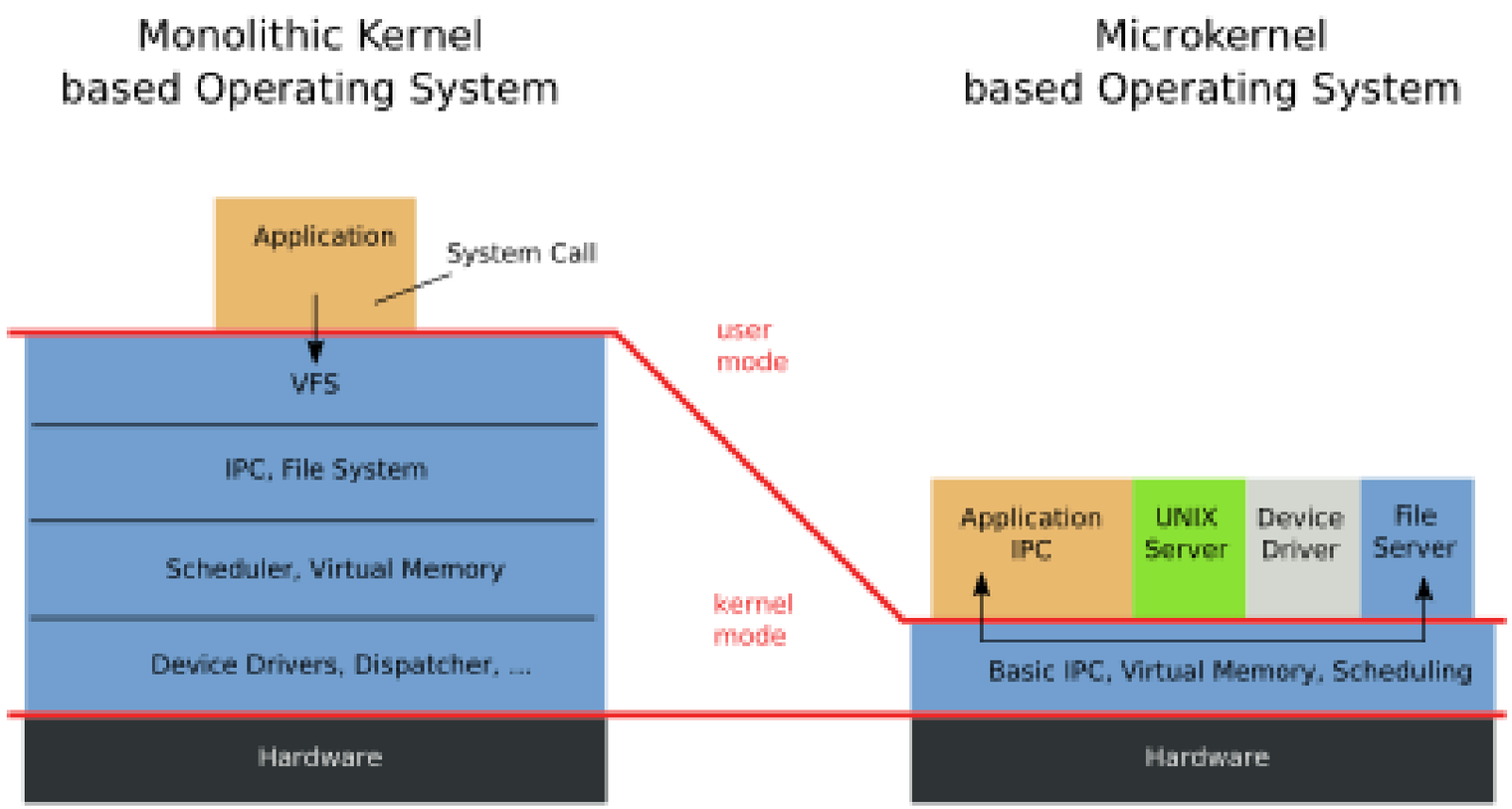
- OS is a **resource allocator**
  - Manages all resources (HW, applications, etc)
  - Decides between conflicting requests for efficient and fair resource use
    - Each program gets its time with the computing resource
    - Each program gets its space with the computing resource

- OS is a **control program**
  - Controls execution of programs to prevent errors and improper use of the resources (HW and programs) – ordering, sequencing, …



OS Structures?

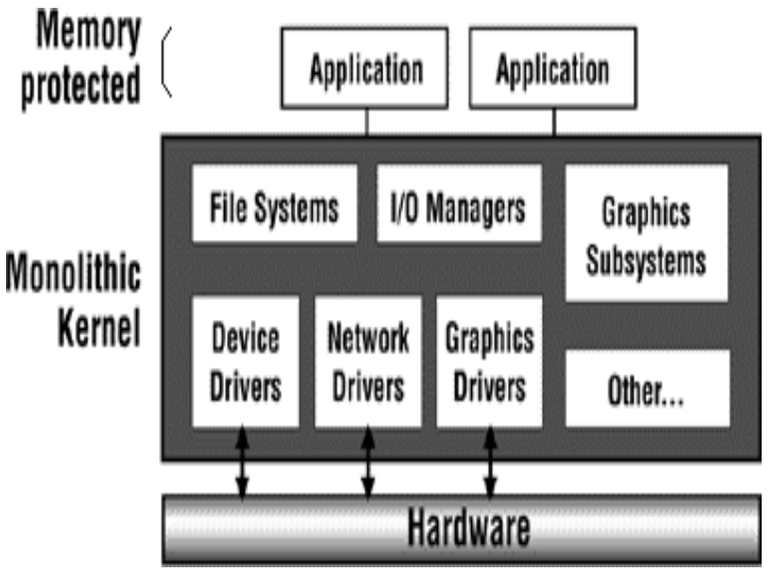
- **Monolithic**: all user/kernel functions inside a single kernel (**pro**: all OS procedures can call on each other, visible to all. **con**: all procedures need to be compiled and linked to each other – static!)



- **Client-Server/Microkernel** based: non-basic services float as servers with a basic kernel for primitive functions: IPC based!
- **Virtual machines?**

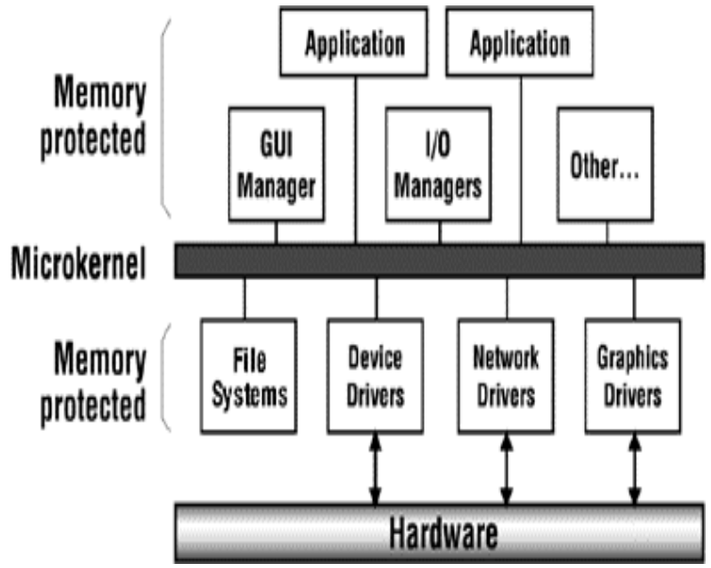
Kernel Structures & Examples

Monolithic kernel structure



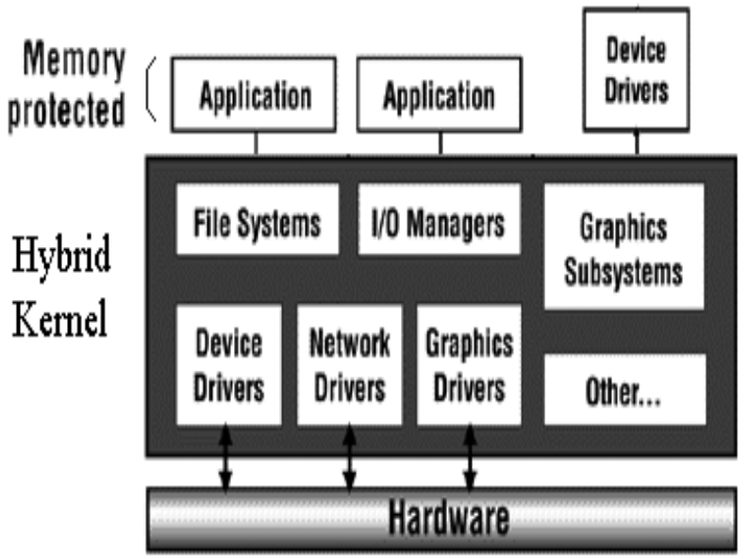
- Windows 9x Kernels
- Solaris
- Unix BSD

Microkernel structure



- Symbian
- L4
- Minix
- GNU Mach/Hurd

Hybridkernel structure



- Windows NT
- Linux
- MacOS X

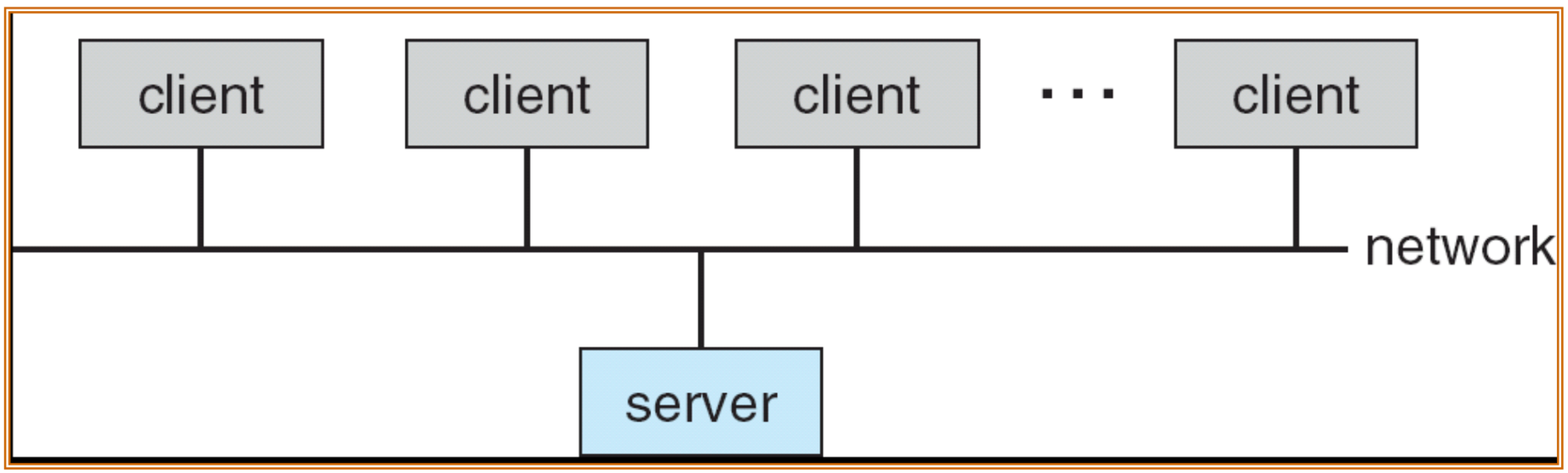
- OS with monolithic kernel and hybridkernel structures are widely-used.

Client-Server Computing Environments

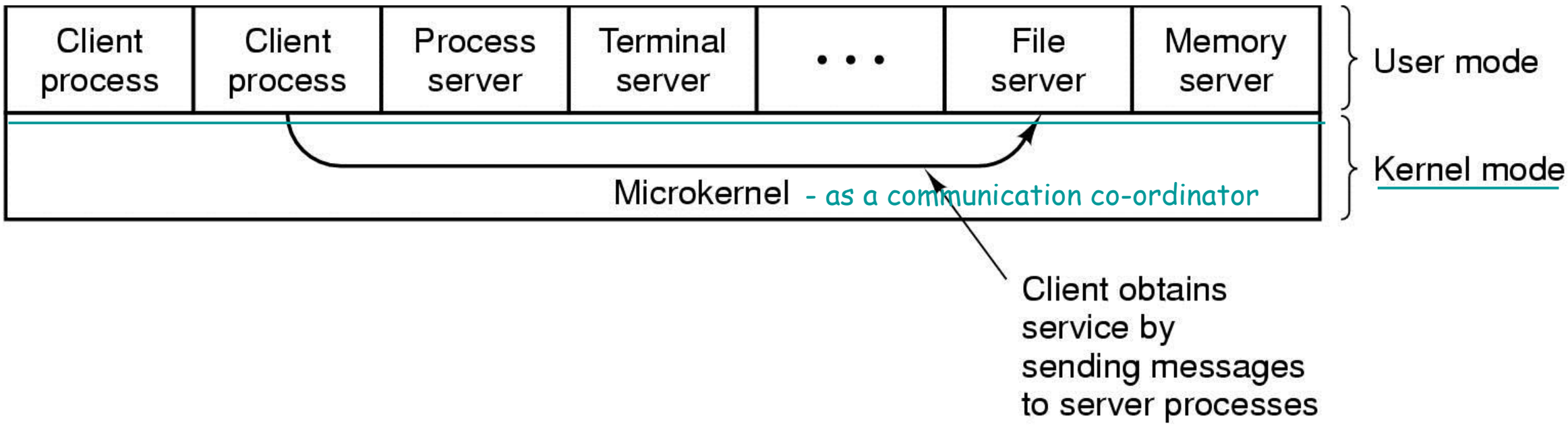
Split by "functionality" into Providers (Servers) & Consumers (Clients):

Servers respond to requests generated by **clients**

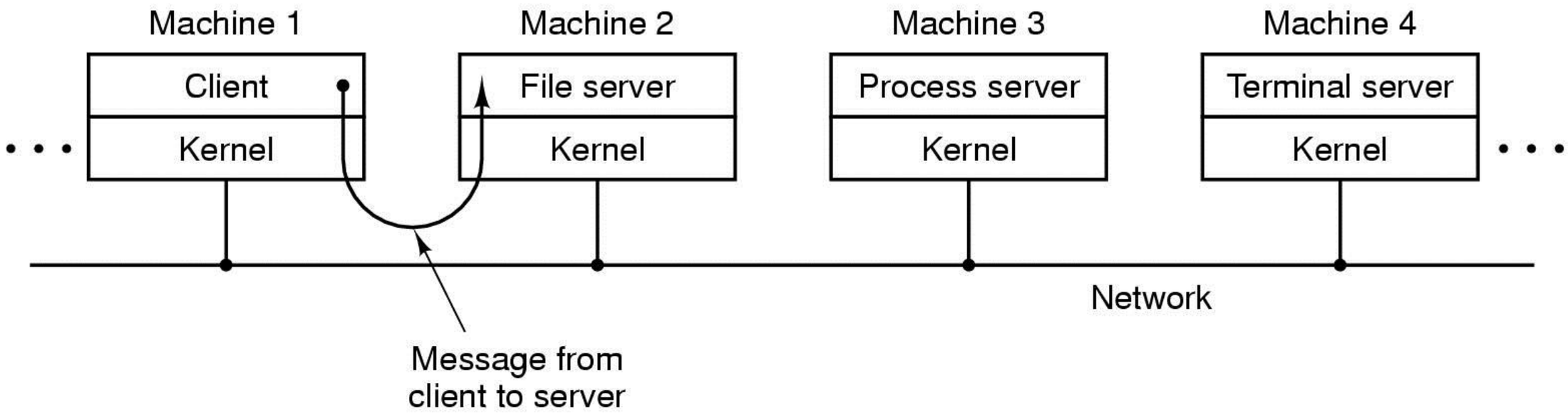
- **Compute-server** provides an interface to client to request services (i.e. database)
- **File-server** provides interface for clients to store and retrieve files
- …



Client-Server Model 1: The Flat Model

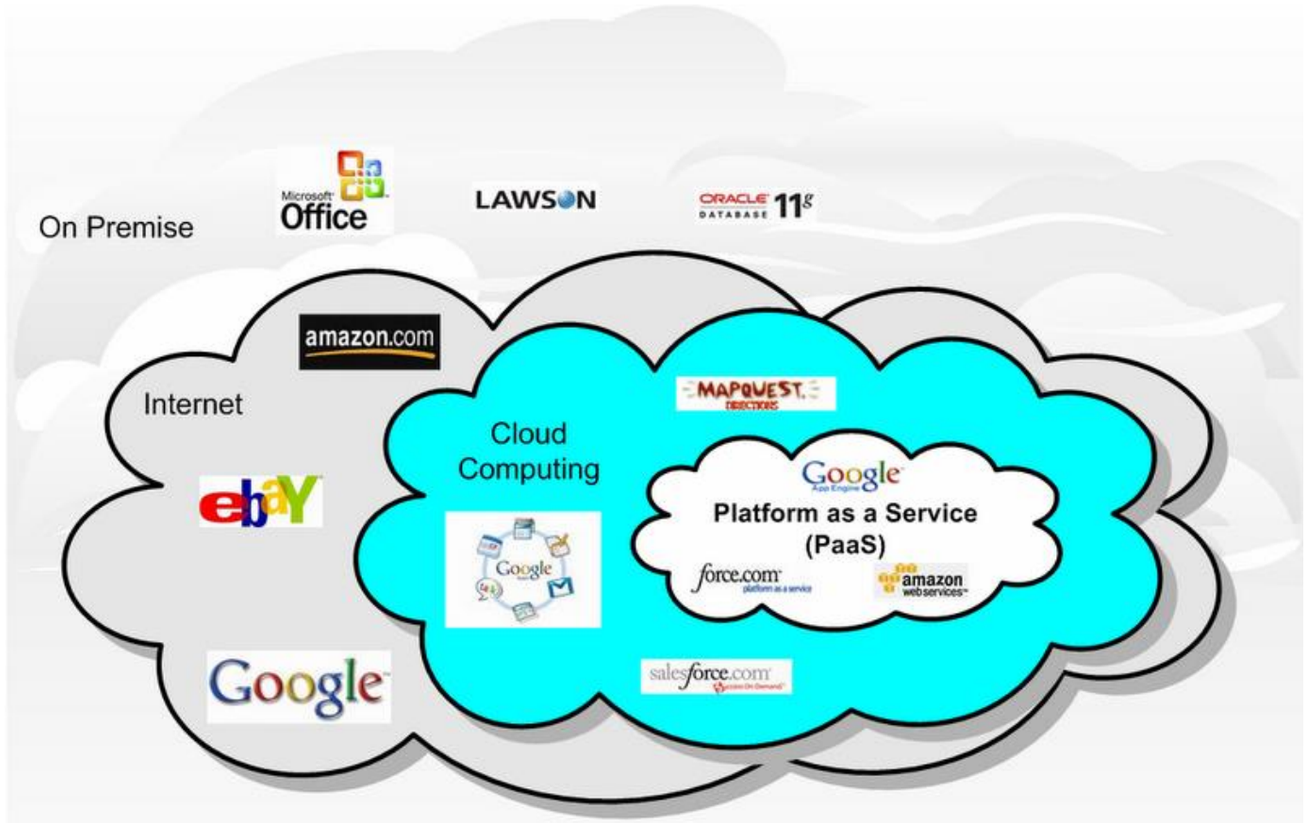


Client-Server Model 2: Distributed Model



Distributed/Networked OS's?

- Flexible: resources/functionality can be added
- Sharing of data, messages, devices
- Cheaper? price/performance
- Faster?
- Secure, networked, load balancing?



Course Outline

- Processes and Threads
- Memory Management and I/O

- Deadlocks/Distributed Concurrency Management
- Resource Allocation/Scheduling

- File Systems
- OS Architectures and Implementation Issues
- Multicore Utilization
- Mobile, Embedded, and Real-Time Operating Systems
- Distributed/Networked OS: Models, RPC, IPC…

Process Management

- Process is a program in execution. Program is a *passive entity (code)*, process is an *active entity (executable)*.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data
- Process termination requires reclaim of any reusable resources
  - Process executes instructions sequentially, one at a time, until completion
  - Single-threaded process has one **program counter** specifying location of next instruction to execute
  - Multi-threaded process has one program counter per thread
- Typically, (modern) systems have many processes, some user, some operating system running concurrently on one or more CPUs
  - Concurrency by multiplexing the CPUs among the processes / threads

Process Management Activities

OS responsible for process management:

- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

Memory Management

- Memory contains
  - All data before and after processing
  - All instructions and their order to execute
- Memory management determines what is in memory in order to
  - Optimize CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

I/O Subsystem

- OS hides operations of hardware devices from the user; application to HW interfaces
- I/O subsystem is responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred)
  - Caching (storing parts of data in faster storage for performance)
  - Spooling (the overlapping of output of one job with input of other jobs)
  - Drivers for specific hardware devices

Storage Management – File Systems/Mass Storage

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - file
- File-System management
  - Files usually organized into directories
  - Access control to determine who can access what & when (RW!)
- OS activities include
  - Creating and deleting files and directories
  - Primitives to manipulate files and dirs
  - Mapping files onto alternate storage
  - Backup files onto stable (non-volatile) storage media
- Speed of OS operations hinges on disk subsystem and its algorithms
  - Free-space management
  - Storage allocation
  - Disk scheduling

Course Attendee Limit

- **Course attendee limit 80 students**
  - Enforced by department via lottery
- **Line up in 1 of the queues**
  - Ordered by Last Name: A-F / G-L / M-R / S-Z
  - Written OS1 exam before? (if so when)
  - Signature
- **Results will be published by the department via TUCaN**
  - Students who did not attend todays lecture will be unsubscribed
  - Students who are not in the 80 student pool will be unsubscribed