Operating Systems: Course Syllabus

Chapter 2: System Structures

- General OS task: resource mgmt
  - Abstraction from details
  - Sharing, fairness, ...
  - Protection
  - Interfaces: shells, API, system calls, ...
- Architecture of OSs
  - Layered system design
  - Microkernels
  - Modularized approach
- Virtual Machines
- System Boot process

Chapter 3: Processes

Chapter 4: Threads

Chapter 5: CPU Scheduling

Chapter 6: Process Synchronization

- What is the problem? Give examples!
- Mutual exclusion
  - Critical sections
  - Different solutions, on different levels of abstraction
  - „Correctness“ criteria (mut.ex., progress/non-starvation, fairness, bounded waiting)
  - Busy wait vs. Sleep
- Classical synchronization problems
  - Bounded Buffers
  - Readers/writers
  - Dining Philosophers
- Atomic transactions (cf. databases)
  - Logging, UNDO/REDO recovery
  - Concurrency Control, Serializability, Locking Time-Stamping

Chapter 7: Deadlocks

- Define the problem, give necessary conditions!
- Graph-based models (single vs. multiple resources per type)
  - Resource Allocation Graph
  - Wait-For Graph
- Methods for handling deadlocks
  - Prevention
    - What can we do to ensure that no deadlock can ever occur?
  - Detection
    - How/when can we detect that the system/some processes is/are in a deadlock state?
  - Recovery from deadlock
    - If we detected a deadlock: how to resolve the situation?

Chapter 8: Main Memory

Chapter 9: Virtual Memory

Chapter 10: File System Interface

Chapter 11: File System Implementation

Chapter 12: Mass Storage Structure

- Physical characteristics & performance determinants of disk technology
- Connecting disks and processors
- Disk scheduling
  - Different algorithms with their pros & cons
- Overcoming disk failures by the use of RAID-technology
  - General idea
  - Different levels of protection
  - Characterization of those levels
- Storage hierarchy, tertiary storage

Chapter 13: I/O Systems

- Transforming application-level commands into device operations
  - Analyze the steps necessary & point out major OS tasks
- Many details on OS implementation of I/O interface
  - Ports, polling, interrupts, DMA, device drivers, kernel I/O structure
  - Show how interrupt handling is done in the OS!
  - Why several levels (priorities), masking, etc.?
- Blocking vs. non-blocking vs. asynchronous I/O
  - Give an example, show principal steps of interaction between user process and OS components

Chapter 14: Protection

- Goals, principles, mechanisms of protection
- Model used for protection (objects, domains, access-rights)
- Access matrix
  - Organization
  - Addition of rights
  - Revocation of rights
- Capabilities vs. access control lists
- Roles
- Language-based protection (cf. Java)

Chapter 15: Security

- How is security different form protection?
- Threats, attacks, breaches
  - Programs
  - System & network
- Security measures
  - Physical, human, OS, network, ...
  - „weakest link in a chain“!
- Cryptography as THE security tool
  - Symmetric vs. asymmetric cryptography
  - How can these techniques be used in different contexts?
- Authentication
- Firewalls

Chapter 16: Distributed System Structures

Chatper 17: Distributed File Systems

Chapter 18: Distributed Coordination

- What is the problem?
- Essential question: how to **order** distributed events?
  - Global ordering constraint & algorithm
- „Applications“
  - Mutual exclusion
  - Atomicity (2PC)
  - Concurrency Control (distributed locking/timestamp-ordering)
  - Deadlocks (wound-wait vs. wait-die)
  - Election algorithms, reaching agreement

The End