1 Einführung

1.1 Hardware

-

1.2 Software

-

1.3 Adressraum

- Aufbau
  - Text
  - Static
  - Heap
  - Stack(s)(Mehrere Threads)
  - wrx Rechte

2 Memory

2.1 Raum-Zeit-Kontinuum

- Kontrollfluss (Thread of Control)
- Reference String
- von Neumann Architektur
- Reference Locality

2.2 Memory

- Internal Protection
- External Protection
- Ideal Address Space Usage

2.3 Pages

- MMU(!)
- Virtual/Physical Address(!)
- Single-Step Mapping(!)
- Multi-stage Mapping(!)
- Translation Lookaside Buffer(!)
- Page Descriptor(!)
- Page Table Descriptor(!)

2.4 Page Faults

- Efficiency(50): $t_{effective} = (1-p) * t_{Memory} + p * t_{PageFault}$
- Page Fault: $t_{PageFault} = t_{Interrupt} + t_{Search} + t_{Write} + t_{Read} + t_{Instr}$
- Replacement Algorithms(!)
  - Reference Strings
  - Optimal Replacement by Belady
  - FIFO
  - Belady's Anomaly
  - Least Recently Used(LRU)
  - LRU with Second Chance
  - Clock Algorithm
  - 2 Hand Clock Algorithm
  - Vergleich zwischen Algorithmen
  - Other (LFU, Random,...)
- Average Access Time and Costs (66)(!)
- Multi-level caching(!)

2.5 Working Set Theory

- Working Set of a thread(70)
- Thrashing(!)
- Gobal/Local Replacement(!)

3 Threads

- Context Switch
- Multiplexing
- Bursts(!)
  - CPU
  - IO
  - alternation

3.1 State Model

States (!)

- Ready
- Running
- Blocked

Transitions (!)

- Add
- Assign
- Block
- Ready
- Resign
- Terminate

Dispatcher

3.2 Scheduling

- Short/Long-term Scheduling
- Criteria
  - CPU utilization
  - Throughput
  - Tunraround Time
  - Waiting Time
  - Response Time
  - Real time
- Non-/Preemptive(!)
- Scheduling
  - FCFS
  - Priority-Based(Priority Inheritance)(!)
  - Round-Robin
  - Multi-level
  - Feedback

4 Synchronization

- Amdahl(!)
- Moore(!)
- Konkurrenz(!)
- Kooperation(!)
- Mutual Exclusion(Mutex)
- IPC
- Semaphore
- Producer/Consumer(!)

5 IPC

- Pipes (named/unnamed)(!)
- Client/Server Architecture(!)

6 Filesystems

- RAID

7 Übungen

- Internal/External Fragmentation
- Buddy/ First Fit
- Barrier
- linux scheduler
- Fork
- Demand scheduling