The following topics are present in the 2004-2007 CSE Systems Quals.

1 Operating Systems

- Three-state process model
  - Transitions
  - Events to trigger transitions
- Threads
  - User-level
  - Kernel-level
  - Which is better for high concurrency?
- Processes
  - Use of fork()
  - Messages
- Synchronization and Deadlock
  - Conditions for deadlock
    * Safe, Unsafe, Deadlocked States
    * Resource Allocation Graphs
  - Semaphores
    * `sem_wait`
    * `sem_signal`
  - Busy wait v. Blocking wait
  - Dining Philosophers Problem
- Race avoidance
  - Four conditions for race avoidance?
- Job Scheduling
  - Round Robin
    * Quantum
    * Run Time
    * CPU Efficiency formula
  - Multilevel Feedback Queue
    * Is starvation possible?*
    * Effect on time-sharing system
    * CPU vs. I/O bound processes
  - First-come First-served (FIFO)
  - Shortest-Job-First (SJF)
  - Priority
  - Burst Time
  - Waiting Time
• Virtual Memory
  – Advantages
  – Page Tables
    * Table fields (sometimes top+second levels)
    * Offset
    * Be able to compute sizes
  – Replacement Policies
    * Least Recently Used (LRU)
    * FIFO
    * Least Frequently Used (LFU)
    * Working Set Page Replacement
    * Belady’s optimal algorithm
      · Forward distance: distance between the current page references in the reference stream to the
        next place where each candidate is once again referenced.
      · Evict the page with greatest forward distance.
  – Performance
    * Average Address Translation Time
    * Faults given code
    * Code optimization
    * Hit/Miss Rates
    * Mean Overhead (read time)

2 Computer Organization

• Amdahl’s Law

• Pipelining
  – Classic RISC five-stage processor:
    * Fetch
    * Register, Read & Decode
    * Execute/Arithmetic Logic Unit (ALU)
    * Memory Access
    * Write Back
  – Data Hazards
    * Memory RAW
    * Memory WAR
    * Memory WAW
    * Register RAW
    * Register WAR
    * Register WAW
    * Control
  – Stalls
  – Dependences
    * Recognize dependencies in assembly.
    * Optimize with loop unrolling
- Forwarding
- Performance vs. Sequential Processor

- Parallelism (Instruction-level parallelism?)
  - Scoreboard
  - Tomasulo Algorithm
  - Tomasulo Algorithm with Speculation (i.e. with Re-order Buffer)
  - What limitations are improved with the above?

- Out-of-order (dataflow-order)
- Temporal and Spatial Locality
- Speculation
- Victim Cache
- Trace Scheduling
- Software Pipelining
- Instruction Prefetching
- Reorder Buffer (ROB)

- Processors
  - Types
    - Speculative? (Tomaulo’s algorithm?)
      - Costs
      - Benefits
      - Frequencies
      - Affects on performance metrics
    - Branch delay slot and instruction reordering?
  - Performance
    - dynamic Instruction Count (IC)
    - Cycles Per Instruction (CPI)
    - Clock Cycle Time (CCT)
    - Million Instructions Per Second (MIPS)
      \[
      MIPS = \frac{\text{InstructionCount}}{\text{ExecutionTime} \times 10^6} = \frac{\text{ClockRate}}{\text{CPI} \times 10^6}
      \]
      
      What are the pros and cons of this metric?

- Disk Storage
  - Measurements of performance
    - Reliability (Mean Time To Failure [MTTF])
    - Latency
    - Throughput-1
    - Throughput-s
  - Arrays of Independent Disks (AID)
    - Striping
Mirroring
- Fault tolerance
- RAID (and the types)
  - RAID 1
  - RAID 3
  - RAID 5

- Memory Heirarchy
  - Cache Options
    - What size data to store
      - Cache Index
      - Block Offset
      - Cache Tag
      - Data
    - What to do with hit
      - Write-through
      - Write-back
    - What to do with miss
      - Write-allocate
      - No-write-allocate
    - Expected speedup
  - Write buffers
    - Merging Write Buffer
    - Non-merging Buffer
  - Blocking (hit under miss)
  - Non-blocking (miss under miss)
  - Fully Associative
  - Direct Mapped
  - 4-way Associative
  - Set associative
    - Tag
    - Index
    - Block offset
  - Hit rate
  - Miss rate
  - Miss penalty
  - Hit Ratio Given an \( n \)-level memory system, the hit ration \( H_i \) associated with the memory \( M_i \) at level \( i \) is the probability that the information requested by the CPU is stored in level \( M_i \). This can be used to compute the average memory access time in \( n \)-level memory given hit-times \( T_i \).
  - Average memory access time
  - Split cache (I-cache, D-cache)
  - Unified cache
  - Virtual vs. Physical cache
  - Cache Misses*
* Compulsory
* Conflict
* Capacity
* Optimization Techniques to reduce cache miss rates
* Optimization Techniques to reduce cache hit time
* Software/Hardware Optimizations

– Page faults
– Translation Lookaside Buffer (TLB)
  * Hits and Misses
  * Index, Tag, Data
  * Virtual Page Number, Page Offset
  * Address Space Number (ASN) to prevent flushes on a context switch. (Why?)
– Cache affect on Cycles Per Instruction (CPI)
  * Affect on CPI
  * How many cycles are due to cache misses?

• Instruction Set Architecture (ISA)
  – Stack ISA
  – Load-Store ISA