Systems Comprehensive Exams Topics

Patrick G. Bridges

November 23, 2004

1 Introduction

I've divided each of the subjects into three areas: undergraduate-level topics, graduate-level topics, and other topics for basic familiarity. We will assume you have in-depth, comprehensive familiarity with the undergraduate level topics of architecture and operating systems since they are fundamental to much of computer science. We also expect a solid grasp of the issues on the graduate-level topics, as well as basic familiarity with the issues that arise in some specialized sub-fields.

As far as the exam format goes, we are still discussing exactly how many questions and their particular content and format. However, you may expect a moderate number of questions out of which you may choose a slightly fewer number to answer - 7 choose 5, or 6 choose 4, for example. In addition, we may ask you to relate relevant material from other areas that you will also be studying for the exams to portions of this exam.

2 Computer Architecture

2.1 Undergraduate-level Topics

- Amdahl's Law
- Integer data representation, though some familiarity with the basic floating point representation issues is also expected
- Basic computer organization - CPU, Memory Bus, Memory, I/O Bus, I/O devices, etc.
- Basic processor architectures - RISC, CISC, Register-Register, Memory-Memory, Stack machines
- Basic pipelining - forwarding, stalls, hazards, scheduling
- Memory hierarchy basics - uniprocessor caching basics (addressing the cache, direct-mapped vs. associative, split vs. unified, write-back vs. write through, block size issues), virtual memory hardware (TLBs, hardware page tables, software-loaded TLBs)

- I/O Basics - DMA vs. PIO, Polling vs. interrupt-driven

2.2 Graduate-level topics

- Super-scalar systems - in-order and out-of-order execution
- Super-pipelining
- Branch prediction, speculative execution
- Symetric multi-threading/Hyper-threading
- Multi-processor (SMP) caching - coherency, bus snooping, invalidation

2.3 Other topics for basic familiarity

- VLIW
- NUMA multiprocessors - consistency/coherency, etc. (closely related to consistency/coherency in distributed systems such as DSMs)
- Distributed-memory/message-passing machines (e.g. clusters)
- Vector architectures

3 Operating Systems

3.1 Undergraduate-level material

- Basic system structure (user-level, kernel, servers, device drivers, interrupt processing)
- Scheduling
- Memory-management (e.g. virtual memory management, page replacement, etc.)
- Syncrhonization (test-and-set, spin locks, semaphores, monitors, deadlock *basics*, etc.)
- File-system basics (format of files, directories on a file system)

3.2 Graduate-level material

- Basic O.S. architecture tradeoffs (macro/micro/extensible)
- System design heuristics (e.g., the end-to-end principle)
- Advanced scheduling and threading issues (kernel vs. user threads, priority inversion, livelock)

- Distributed systems
  - RPC
  - Naming/resource discovery (e.g. DNS or grapevine)
  - Consistency/coherency issues
  - Distributed shared memory
  - Distributed file system basics (e.g. Andrew FS, NFS, Sprite FS - how they deal with naming, file lookup, and caching, for example

3.3 Other Material for Basic Familiarity

- Journalling and logging in modern file systems
- Hard real-time (e.g. process control) and soft real-time (e.g. multimedia) systems
- Mobile systems
- Fault tolerance
- optimistic syncrhronization and dynamic code generation (see Massalin's paper on threads and I/O in Synthesis)

3.4 Networking

- Basic layers in a networking stack and what their roles are (the most important are the standard internet four layers five - physical (sometimes data-link is added above this layer), network, transport, application (which OSI subdivides into app, presentation, and session)
- Packet-switched networks vs. circuit switched networks
- Basic characteristics of physical networks like Ethernet
- Routing *basics*
- TCP/IP basics - addressing, flow control, congestion control, error detection