FINAL EXAM GUIDE

The exam is open book and open note, and focuses on material covered in the lectures, labs, assignments, and additional readings. The exam questions will require you to demonstrate a good understanding of the key concepts and the ability to analyze a particular situation and apply your knowledge.

Material Covered: The second half the class concentrates on the following three modules:

1. x86-64 Runtime Stack,
2. Cache Memories,
3. Debugging, Design and Code Optimization,
4. Linking,
5. Heap Allocators

Hence, the final exam will cover all materials contained in Lectures 18-27. Note that, however, some of the questions may require some knowledge about the first half of the class. Specifically, the topics covered in the final exam are listed in detail below:

x86-64 Runtime Stack

- Lecture 18: x86-64 Procedures
  revisiting %rip, the stack, passing control, call instruction, push and pop instructions, passing data, local storage, register restrictions, caller-owned vs callee-owned registers
- Lecture 19: Data and Stack Frames
  implementing one-dimensional, multi-dimensional and multi-level arrays, structures and alignment, floating point instructions
- Lecture 20: Security Vulnerabilities
  memory layout, buffer overflow, *buffer overflow attacks and defences*

Cache Memories

- Lecture 21: The Memory Hierarchy
  storage technologies and trends, principle of locality (temporal locality, spatial locality)
- Lecture 22: Cache Memories
  caching in the memory hierarchy, hits and misses
- Lecture 23: More Cache Memories
  cache memory organization, the memory mountain

Optimization

- Lecture 24: Code Optimization
  rearranging loops to improve spatial locality, using blocking to improve temporal locality, what is optimization, constant folding, common sub-expression elimination, dead code, strength reduction, code motion, tail recursion, loop unrolling, limitations of gcc code optimization

Linking

- Lecture 25: Linking
  static linking, why we need linkers, what do linker do, ELF object file format, symbol resolution, relocation, static libraries, shared libraries

Heap Allocators

- Lecture 26: Managing the Heap
  what is a heap allocator?, heap allocator requirements and goals, fragmentation, implementing heap allocators, bump allocator, implicit free allocator, explicit free allocator, coalescing, in-place realloc