

FINAL EXAM GUIDE

The exam is a **closed book and notes exam**, and focuses on material covered in the lectures, labs, and assignments. But you are allowed to have a **single-page A4 sized copy sheet** during the exam (you can use both sides of these sheets). Moreover, **you can additionally bring the Stanford CS107 x86-64 Reference Sheet** with you (no extra notes allowed on this printout). 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. Introduction to x86-64 Assembly
2. x86-64 Runtime Stack,
3. Cache Memories,
4. Debugging, Design and Code Optimization,
5. Linking

Hence, the final exam will cover all materials contained in Lectures 14-24. 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:

Introduction to x86-64

- Lecture 14: Introduction to x86-64, Data Movement
gcc and assembly, looking at an executable, registers, the mov instruction, operand forms, data and register sizes, mov and data sizes
- Lecture 15: Arithmetic and Logic Operations
the lea instruction, logical and arithmetic operations, reverse engineering assembly code
- Lecture 16: x86-64 Control Flow
executing instructions, the program counter register (%rip), unconditional and conditional jump instructions, control mechanics (condition codes, cmp and test instructions), implementation of if statements in assembly
- Lecture 17: More Control Flow
implementation of while and for loops in assembly, other instructions that depend on condition codes (the set and cmov instructions)

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

Optimization

- Lecture 23: 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 24: Linking
static linking, why we need linkers, what do linker do, ELF object file format, symbol resolution, relocation, static libraries, shared libraries