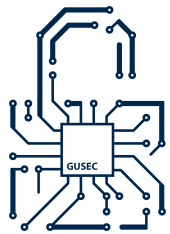
Introduction to x86 Assembly

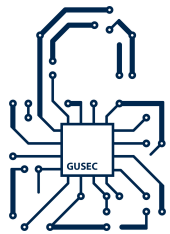
Joe Rose

GUSEC

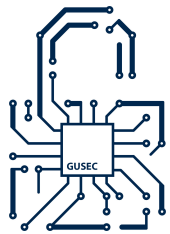
December 15, 2022



Outline of the Workshop

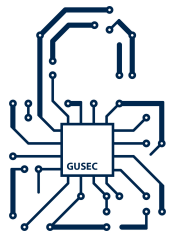


Outline of the Workshop

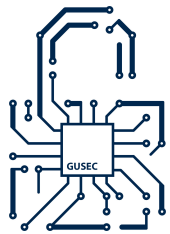


Who am I?

- 3rd Year Comp. Sci. Student
- Secretary of GUSEC
- Big nerd

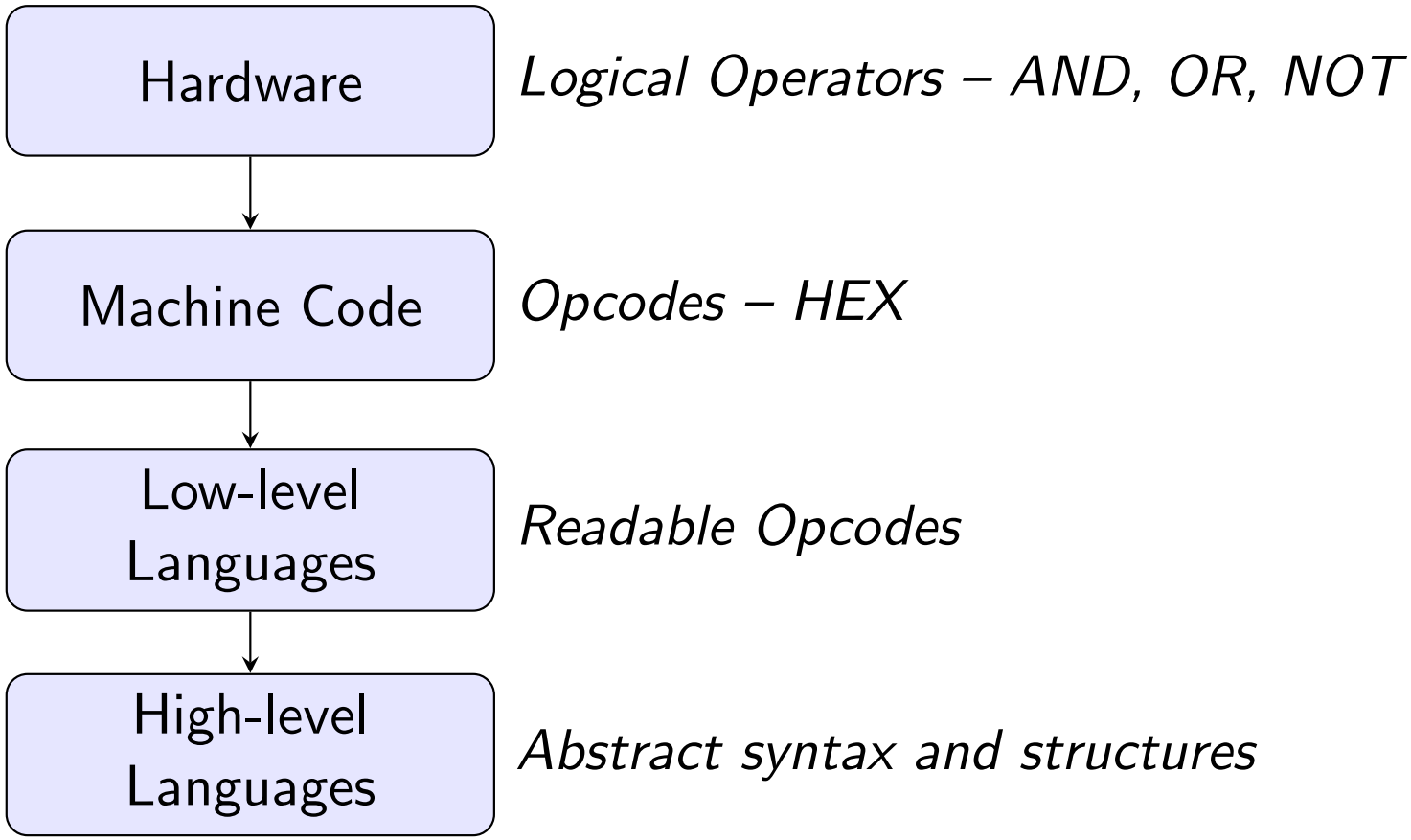


Outline of the Workshop



Abstraction

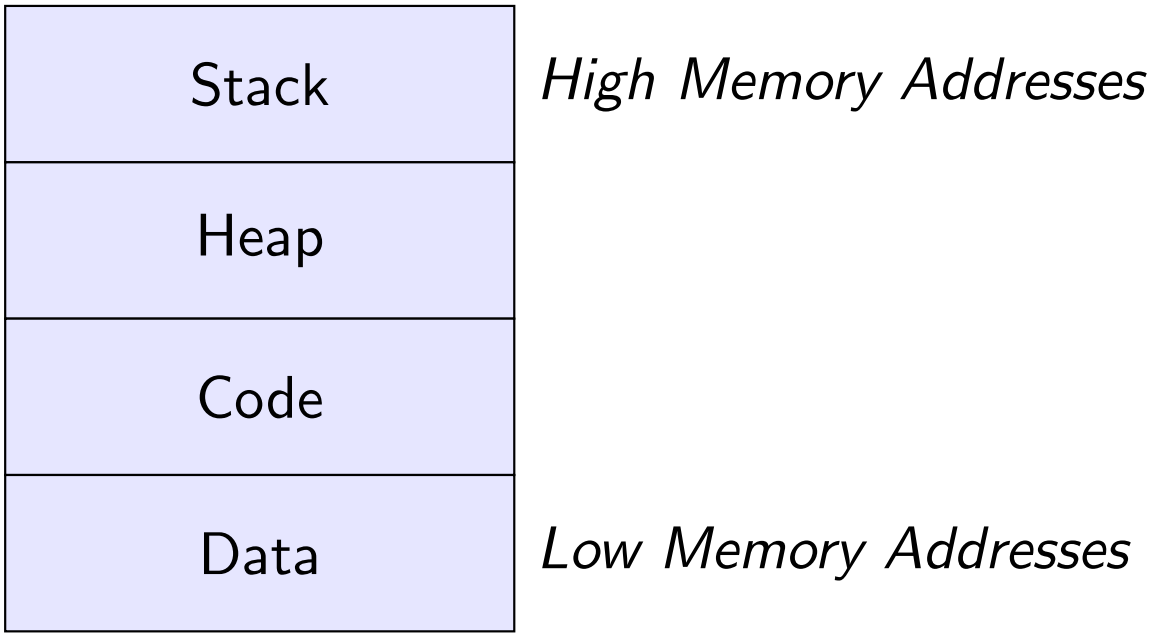
Modern computer systems can be represented as layers of *abstraction*. As software engineers, we generally work in a 'high level' highly abstracted space, which is then converted by our compiler into low level *machine code*.



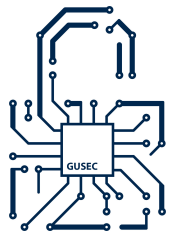
Memory

Memory, as it has come to be defined in some current literature, is somewhat of an abstract concept.

Every process running at a given time is carved out a little piece of virtual memory. It follows roughly this construction



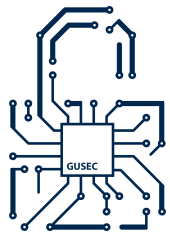
Outline of the Workshop



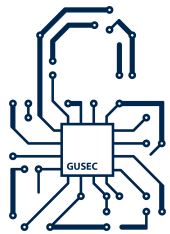
The Registers

In low-level languages we deal with the regsiters. Registers are extremely small (32-bit) pieces of memory within your CPU which are quicker than RAM or cache to access. These are what store the pieces of data currently being worked with. The x86 specifies some general purpose registers and some dedicated registers.

Register	**Purpose**
EAX	Accumulator
ECX	Counter in loops
ESI	Source in string & memory operations
EDI	Destination in string & memory operations
EBP	Stack base pointer
ESP	Stack Pointer



Outline of the Workshop



Sources

These are the books I got the content for this workshop from. However, there is a huge amount of free literature on x86 and these are by no means necessary for you to learn the ropes.

- *Practical Malware Analysis* - Chapter 4. A Crash Course in x86 Disassembly
- *Practical Reverse Engineering* - Chapter 1. x86 and x64
- *Practical Binary Analysis* - Appendix A. A Crash Course in x86 Disassembly