FINAL CLASS!

My next job?

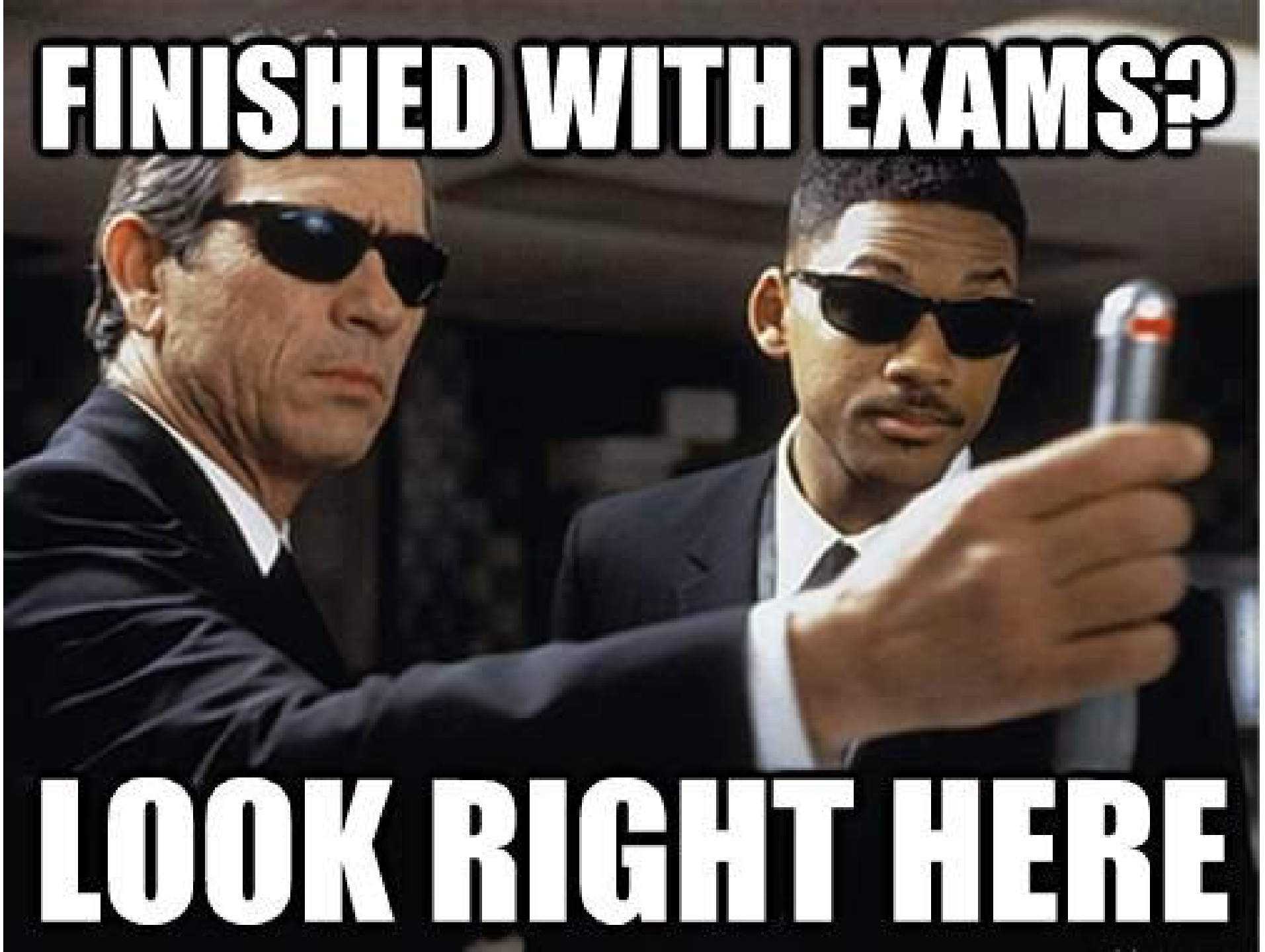

1s
2s p
3s p d
4s p d f
5s p d f
6s p d f
7s p d
Energy
Period
Table
level

Announcements

- Review Session Helpful? Solutions out soon.
- Request Topics review.
- A single handwritten sheet of notes allowed for finals. (Confirmation on this soon.)
- No electronic devices.

_

Today?


FINISHED WITH EXAMS?
LOOK RIGHT HERE

CS 354 ~ What to take away?

Ganesh Kumar . May 6, 2016

General

Everything is represented as a sequence of bits (0s and 1s)!

- Your executable
- Your images
- Your browser application
- Your pdfs
- ...
- Everything!

What they actually represent...

- Depends on **context**
- How we choose to **interpret** them.

General

More specifically,

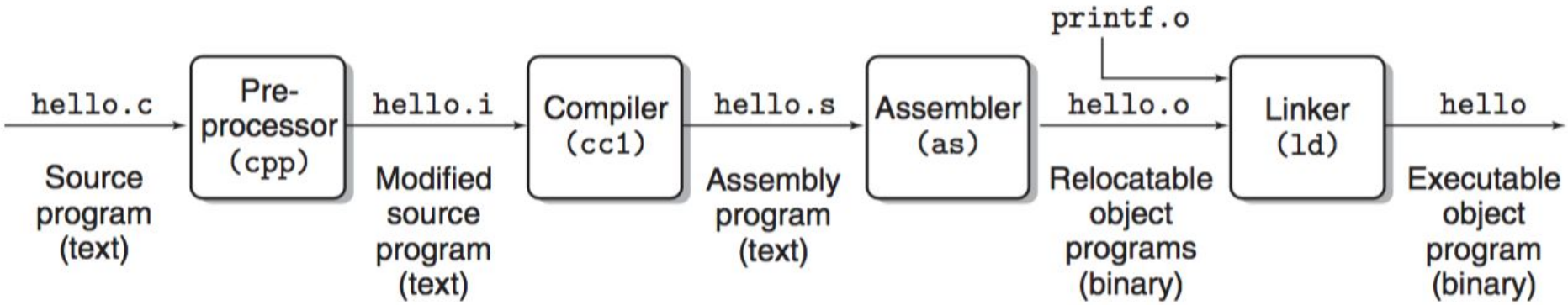
- I have a piece of data that is **0100 0001**. What is it?

- What is 01001010101101101011101010101010100?

  Are we going to split it into 8 bit groups? Or split it into 32 bit groups?

Special Mention: Quantum Computing

Compilation System



C - Pointers

Operators

&varX - ADDRESS OF variable varX.

***varY** - VALUE AT address varY. *(Indirection operator)*

How do you define a char pointer?

```
char ch = 'y';
char * ptr = &ch;
```

This right here is not an indirection operator. It is just pointer declaration syntax;

Data Representation

Endianness

Integer - 0x12 34 56 78 - 4 bytes.

Little Endian

Addr	Data
0x100	78
0x101	56
0x102	34
0x103	12

Big Endian

Addr	Data
0x100	12
0x101	34
0x102	56
0x103	78

Data Representation

4 bit datatype - 0000 to 1111

Unsigned Representation

No specific bits to denote sign. So value goes from DEC 0 to 15.

Signed Representation

MSB allocated for sign.

So value ranges from

BIN	0111	1000
DEC	7	-8

Assembly

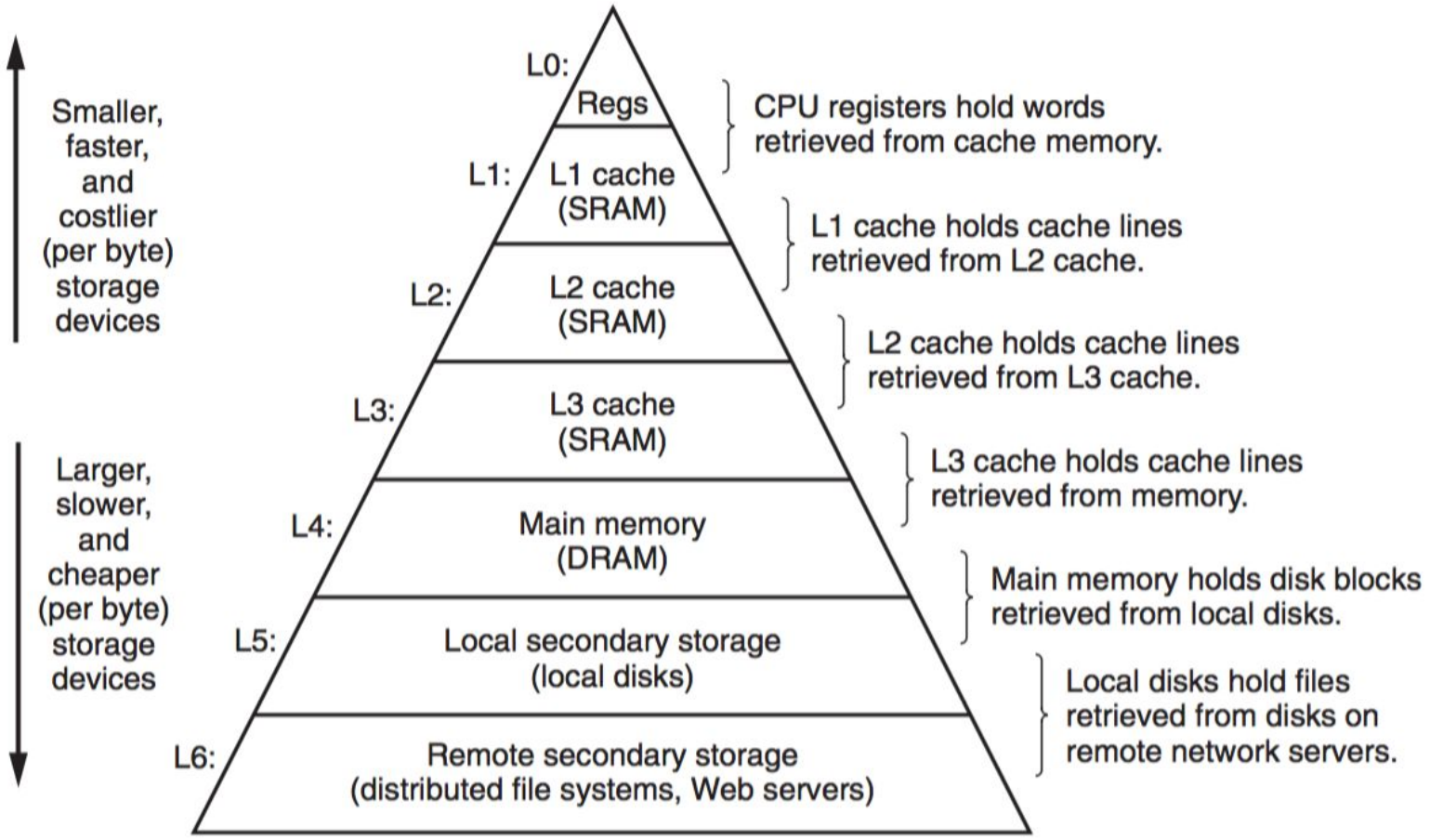
- How to read the x86 Instruction Sheet.
- Registers
- Control Flags and Conditional Jumps.
    - cmp and test instructions followed by jumps.
- Function Stack Frames!
    - All the space associated with a function goes away after it returns i.

      e. Popping it off the stack.
-

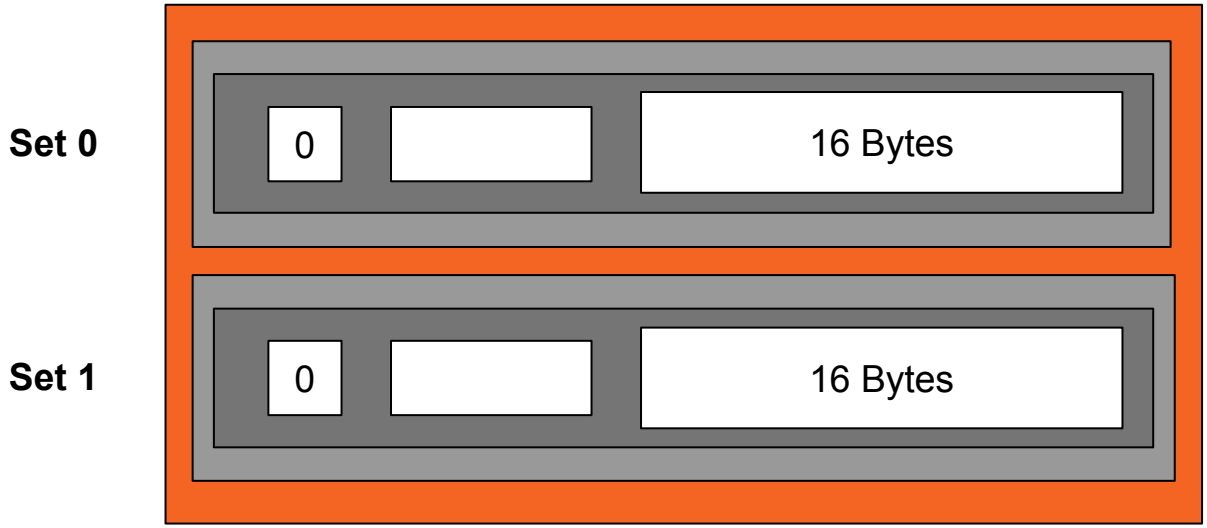
Memory

- Not all storage technology are created equal.
- Some are fast and expensive and some are relatively slow and inexpensive.
- Use a fast memory to serve as a staging area for data from a slower one - CACHING!

Memory Hierarchy



Cache



Design a cache for me!

(S, E, B, m)

Locality

Temporal Locality

Accessing the same memory location over and over again- Good Thing!

Spatial Locality

Accessing the memory in sequence (stride-1 reference pattern) is a good thing as well.

Tip: Algorithm design does not consider the physical limitations of memory.

Virtual Memory

Why Virtual Memory?

- Memory Protection
- Easy Memory Management
- Use more space that what is physically available in the physical DRAM memory.

Virtual Memory

Important Task

Virtual Address ⟶ Physical Address

How?

- Memory Management Unit (In-charge of doing this!)
- Page Tables (Software entity)
- Translation Lookaside Buffer (Separate cache)

Dynamic Memory Allocation

Huge space of unallocated memory - The Heap!

Why an allocator?

- Like any shared resource, access to this resource has to be controlled.
- See Tragedy of the Commons. (Shared resource - Road. Controller - Cops and Traffic Rules).
- So, we need a Dynamic Memory Allocator.
- ... with a lot of rules and control mechanisms - Headers, Alignment Restrictions, Maintenance.

Exceptions

Abrupt change in a processor's control flow.

Different Kinds

- Interrupts (not under our control ~ Asynchronous)
  - Press Ctrl + C!
- System Calls (we cause the abrupt change)
  - Write to display..
- Faults (we cause this)
  - Accessing a page that is not in the main memory.
- Abort (we cause this)

Context Switching

Switch from executing one process to another... while storing the status (or context) of the switched out process.

Why?

CPU should not wait for a slow task that a process needs to be performed. (Relative times - 1 sec for a CPU is 10 months for a Hard Disk)

Process has two modes - Kernel and User

Needs to be in kernel mode before performing a context switch. Why?

Linking

Hope this is still fresh in memory!

Goodbye
&
Good Luck!