2825 Lesson 1

Jason\_Wilder@bcit.ca

Jason will post the actual notes we create together in class, and ALSO more-formal notes (longer, typed), and also videos from last year (optional to watch).

How to do well in this course:

* Don’t fall behind
* Read ahead of time
* Ask questions, especially in class
* Focus on arithmetic and diagrams

Typical lesson:

Q&A

Lesson

Quiz

Official grades are at my.bcit.ca

How to get 100% in this course:

Focus on arithmetic and diagrams, namely THESE diagrams (references the textbook, 6th ed.):

Chapter 1: page 5\*, 18, 50\*

Chapter 2: 57, 66\*, 75\*, 82

Chapter 3: 161(b)\*, 164, 165, 167, 173, 176, 193\*, 195

Chapter 4: 245, 292, 307\*\*

Chapter 5: 366

Chapter 7: 527

The final exam is not cumulative.

Work with a partner and come back at 648 and tell us about John von Neumann.

Your computer is the result of many compromises: simpler, cheaper, backwards compatible

Bit: binary digit: 0 or 1

Byte: 8-bit word

Word size: 64 bits probably; the size depends on the size of a register

Register: fast cpu memory; by definition it stores a single word (an instruction, an address, an operand)

32 bits versus 64 bits: 2^64 = 2^32 \* 2^32

Cpu: FDE cycle: fetches, decodes, and executes instructions

RAM: main memory

Bus: a common electrical pathway along which data moves between devices

2^5 = 32

2^10 = 1024 = 1k

2^15 = 2^5 \* 2^10 = 32k

2^20 = 1M

2^30 = 1G

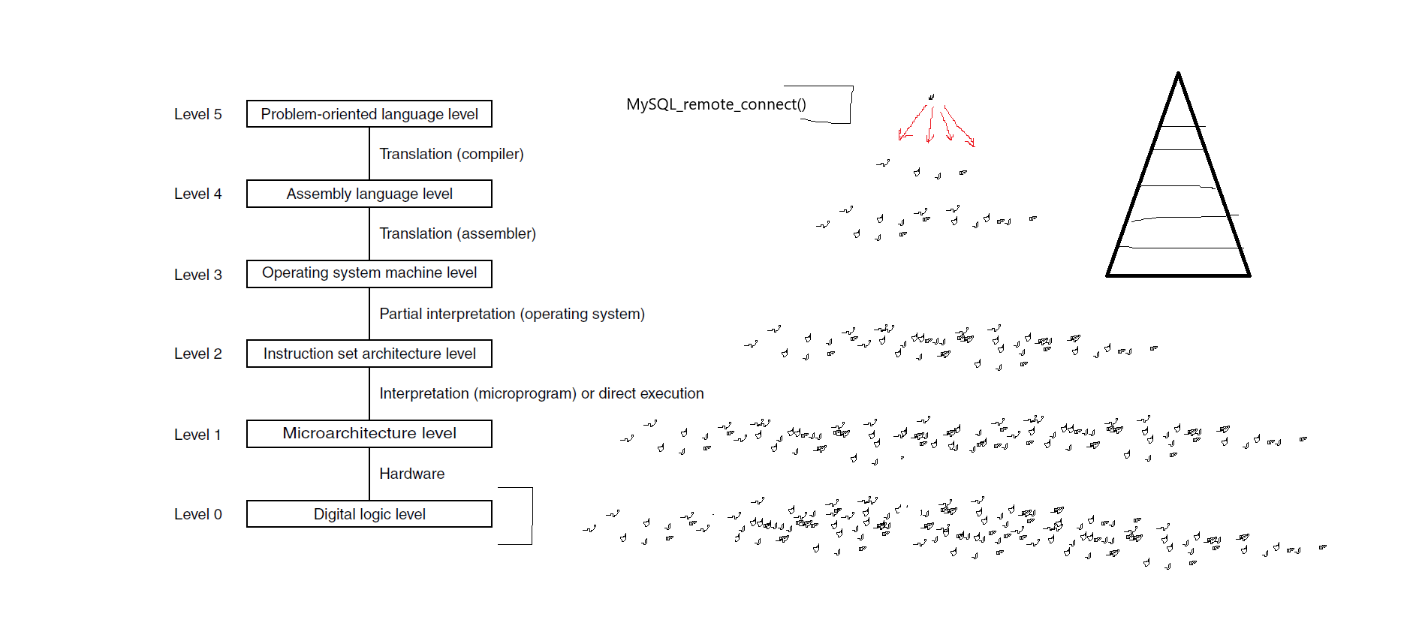
2^40 = 1T

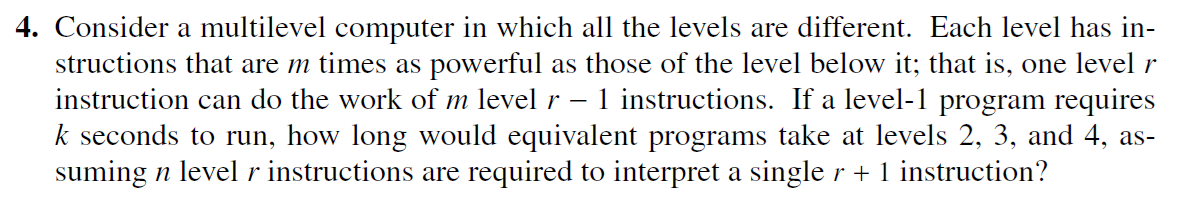
2^37 = 2^7 G = 128G

2^49 = 512T

Imagine a Hello World C program: 170 bytes level-5 program

The .exe file was 16kB level-0 program





ANSWER:

Level 1 takes k seconds

Level 2 TAKES LONGER because level 2 must first be converted into level 1 and THEN is run

A diagram of a system

Description automatically generated

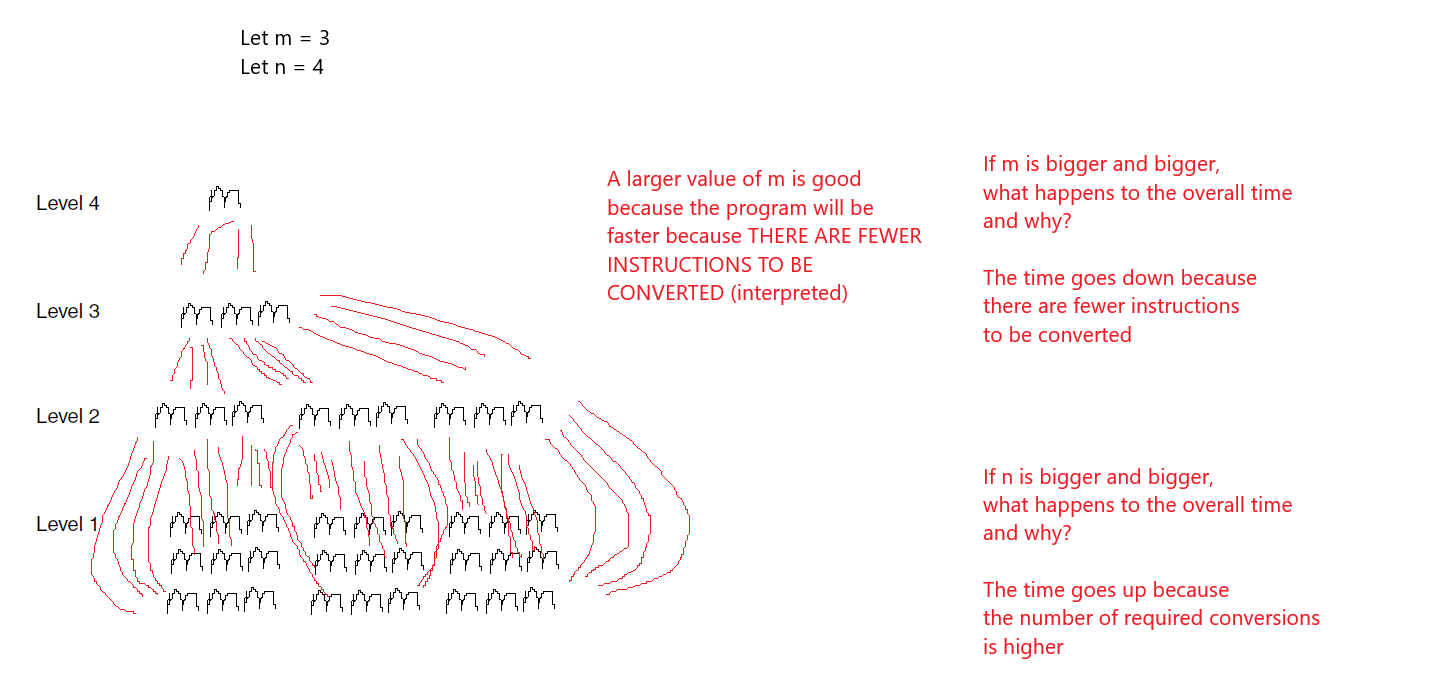
Clearly define the variables:

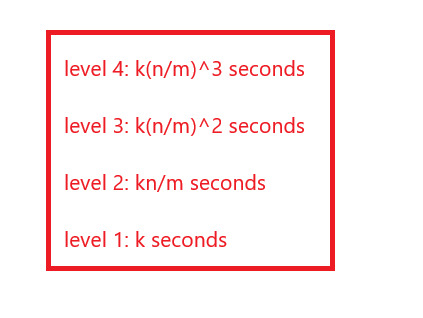
r the level in question

k time in seconds for the program to run at level one

m the number of LLL instructions required to the same work as one HLL instruction

n the number of INTERPRETER instructions required to transform each HLL instruction (into the equivalent set of LLL instructions)





TRANSLATION aka COMPILATION

Converting an entire HLL (high-level language) program into the equivalent lower-level-language program, all at once. Then you no longer need the original HLL program.

INTERPRETATION

Converting a HLL (high-level language) program into the equivalent lower-level-language program, one instruction at a time (convert one instruction, run it, repeat with next instruction)

