Source Code Translation

CPU: Qualcomm Snapdragon 845

CPU : Intel Core i5 6300U

von Neumann architecture

v = \pi r^2 h

CPU 1 (maybe Snapdragon 845) Machine Instructions

load r into register 1
multiply r1 by r1, store result in r1 \( r^2 \) in R1
load h into r2
multiply r1 by r2, store result in r2 \( r^2 h \) in R2
load pi into r3
multiply r2 by r3, store result in r3
store r3 in v

C compiler

\[ v = \pi r^2 h \]
Interpreters vs Compilers

Translation from high-level source code to CPU instructions happens here (likely not explicitly as in by producing a list of instructions) but implicitly by choosing to run different parts of the interpreter, which is itself a list of machine instructions.

Like the interpreter above, except the input is machine instructions for the virtual machine inside the .class files instead of high-level source code.
C compiler translates to native CPU instructions.

Gathers instructions from multiple object files into a single executable.