Formula Readme

Paul Jones
Computer Architecture (01:198:211)
School of Arts and Sciences
Rutgers University

April 4, 2013

1 Challenges

Figuring out how to make a C file "talk to" an Assembly file was the biggest challenge I ran into. The linking process is made clear when you visualize what it is that is actually happening when you compile a C file on a particular machine, but from a programmer's perspective this does not trivially follow.

Conceiving the level on which software finally meets hardware is still hugely difficult to me. It is not *a priori* obvious to me that software actually can meet hardware. I am cognizant of the fact that it does, and all the time, otherwise running a program would be impossible, including this operating system and the programs it runs for me. Yet my confusion remains.

The difficulty I have with the point which software becomes hardware is analogous to the mind-body distinction. I am aware that I have a physical brain, yet my mind seems distinct from it. Consciousness seem distinct from the brain the way that software seems distinct from hardware.

2 Big-O Analysis

2.1 Space

My implementation only ever requires a 32-bit integer at any given time. Furthermore, my assembly is written for a 32-bit processor, so any value that cannot be stored in a 32-bit binary integer will flag overflow and return nothing.

This means that the maximum value one can input is 12, as 13 will cause overflow on the `mull` Assembly command.

2.2 Time

It is amazing that the most "expensive" thing that my program does is print to the terminal. The comparisons are done on an incredibly low level, and the process that takes the most time is printing each integer.