Assembly Language Programming

Modeling Motion Week 5 Computer Lab

This week we will investigate how programs written in high-level languages like Python are translated into machine language that can be executed on real microprocessors. We will try to translate simple Python programs into the assembly language for a simulated microprocessor, the MM1 (Modeling Motion I).

1) Write an MM1 assembly language program to compute 8! (eight factorial) leaving the result in memory at address 63. First write a Python program that computes factorial and then “compile” it into assembly language.

2) Write an MM1 assembly language program to compute the y position of a body under uniform acceleration. Start the body out at y=30 with a velocity of zero and a constant downward acceleration of 1 unit per step. Simulate bouncing by setting y to zero whenever the body drops below y<0 and making its velocity positive.

3) Write an MM1 assembly language program which calls a function which adds two arguments and returns the result in register *r8*. The function call should preserve the contents of the first four registers *r0-r3*.