

CMPE 230 Systems Programming

Project (due March 23th)

(This project can be implemented in groups of at most two students. You can use C/C++ or Java)

In this project, you will implement an interpreter for an assembly language of a hypothetical 8086-like CPU called HYP86. You can make the following assumptions about HYP86:

1. It has 64KB of memory.
2. Each instruction has a fixed length of 6 bytes.
3. It supports immediate, register, register indirect, memory addressing and stack addressing modes.
4. It has 64K memory. Instructions start at address 0. Stack starts at high address (FFFF) and grows towards low address. SP points to free word location on top of stack.
5. It has 16-bit registers **AX, BX, CX, DX, DI, SP, SI, BP**.
6. It has 8-bit registers **AH, AL, BH, BL, CH, CL, DH, DL**.
7. It has the following flags: **ZF** zero flag, **CF** carry flag, **AF** auxillary flag, **SF** sign flag , **OF** overflow flag.
8. Assume all registers with the exception of **SP** and flags are initialized to zero at the beginning. **SP** initially contains **FFFE**.
9. The following instructions are available: **MOV, ADD, SUB, INC, DEC, MUL, DIV, XOR, OR, AND, NOT, RCL, RCR, SHL, SHR, PUSH, POP, NOP, CMP, JZ, JNZ, JE, JNE, JA, JAE, JB, JBE, JNAE, JNB, JNBE, JNC, JC, PUSH, POP, INT 20h** (exit to dos), **INT 21h** (read/write character).
10. Interpreter should not allow reading or writing to instruction area starting from **0**, up to and including **INT 20** (which is required to be the last instruction at the end of instruction area).
11. Labels can be used in the assembly language.
12. Directives **dw** and **db** can be used to define words and bytes. A **variable-name** can be put in front of **dw** and **db**. When using variable names **offset variable-name** accesses the address, just the **variable-name** accesses the value.
13. If there is overflow, you should exit the interpreter with a message saying overflow and give the line number.

Grading

Your project will be graded according to the following criteria:

Documentation (written document describing how you implemented your project)	12%
Comments in your code	8%
Implementation and tests	80%

Late Submission

If the project is submitted late, the following penalties will be applied:

- 0 < hours late <= 24 : 25%
- 24 < hours late <= 48 : 50%
- hours late > 48 : 100%

Timestamping

Project file should include your names in it. Please timestamp your project file using

<https://opentimestamps.org/> before you submit it. Keep the project file and its corresponding timestamp .ots file.