

«به نام خدا»

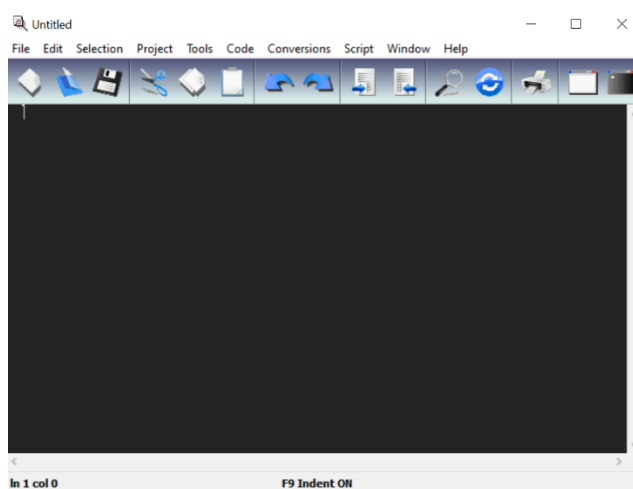
استاد: دکتر میثم عبداللهی

نام: فاطمه زهرا بخشنده

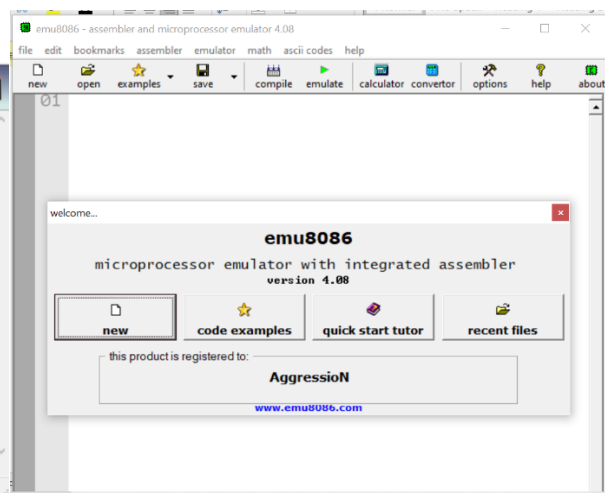
شماره دانشجویی: 98522157

گزارش تمرین دوم:

اسکرین شات های نصب ابزار ها:



masm32



emu8086

توضیحات:

فایل های مربوط به هر دو سوال تمرین، در این فایل زیپ قرار داده شده است. هر سوال، با دو ابزار emu8086 و masm پیاده سازی شده است. همچنین هر سوال، با هریک از ابزار ذکر شده، به دو روش Full Segment-Definition و Simplified-Format پیاده سازی شده است. یعنی برای هر سوال، جمعاً 4 نوع پیاده سازی وجود دارد:

:MASM32

با استفاده از این ابزار، برای هر سوال کد اسمبلی را خط به خط اجرا کرده، و در آخر کامپایل کردم و فایل exe ساختم. با استفاده از دیباگ کد، میتوان در هر لحظه دیتای موجود در هرخانه memory و مقدار هر register و segment را مشاهده کرد.

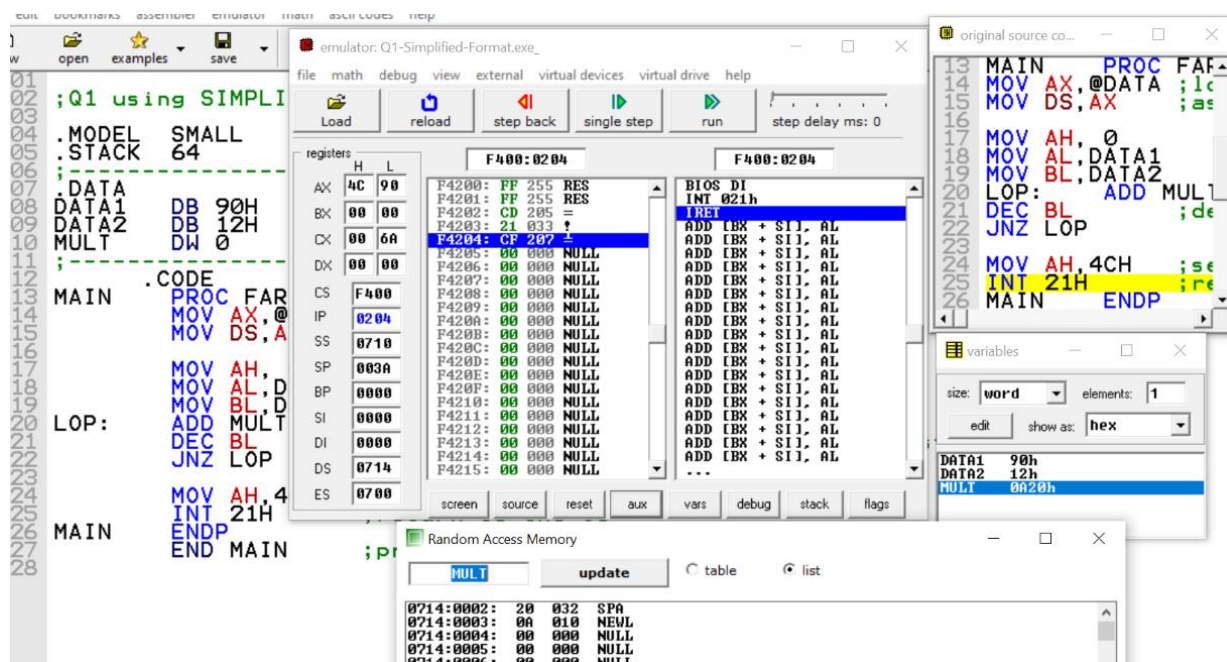
:EMU8086

با استفاده از این ابزار، با syntax ای کمی متفاوت تر از emu8086 کد اسمبلی هر سوال را با استفاده از linker، با کامند های مربوطه، ابتدا به obj تبدیل کرده و سپس از obj ساخته شده فایل exe ساختم.

توضیحات الگوریتم هر سوال:

سوال اول:

در این سوال، برای ضرب دو عدد 8 بیتی به کمک جمع، یک حلقه ایجاد میکنیم که عدد اول را به تعداد عدد دوم با خود جمع کند. توضیحات جزئی هر خط از کد به صورت کامنت کنار آن آمده است. در آخر میتوانیم مقادیر زیر را مشاهده کنیم:



دو عدد DATA1 و DATA2 از حافظه خوانده شده و نتیجه ضرب در حافظه ذخیره شده است (با لیبل MULT). و چون MULT عددی 16 بیتی است دو خانه حافظه را اشغال کرده است. قسمت کم ارزش در خانه 0002 و قسمت پر ارزش آن در خانه 0003 حافظه ذخیره شده است.

سوال دوم:

در این سوال، ده دیتای 8 بیتی مختلف در حافظه قرار می‌دهیم. سپس یک حلقه اصلی در برنامه ایجاد می‌کنیم که در آن SUM و MIN و MAX این ده عدد محاسبه می‌شود. سپس AVG را نیز از تقسیم SUM بر 10 محاسبه می‌کنیم. توضیحات جزئی هر خط از کد به صورت کامنت کنار آن آمده است. در آخر می‌توانیم مقادیر زیر را مشاهده کنیم:

The screenshot shows an x86-64 emulator with the following components:

- Assembly Code:**

```

;Q2 using SIM
.MODEL SMALL
.STACK 64
DATA
MYDATA DB 10, 20, 30, 40, 50, 60, 70, 80, 90, 100
SUM DW 0
AVG DW 0
MAX DB 0
MIN DB 0
.CODE
MAIN PROC
MOV SI, MYDATA
MOV DI, SUM
MOV CX, 10
LOP:
MOV AL, [SI]
ADD DI, AL
MOV AH, 4CH
INT 21H
MAIN ENDP

```
- Registers:**

Register	Value
AX	4C 34
CX	00 00
SI	000A
DI	0000
CS	F400
DS	0714
ES	0700
- Memory:**

Address	Value
0714:0000	10 016
0714:0001	02 002
0714:0002	34 052
0714:0003	5F 095
0714:0004	00 010
0714:0005	00 000
- Variables:**

Variable	Value
MYDATA	00h
SUM	0210h
AVG	34h
MAX	5Fh
MIN	00h

نتیجه SUM و AVG و MAX و MIN این ده عدد را می‌توان به ترتیب در خانه های 000A و 000C و 000D و 000E حافظه مشاهده کرد. چون SUM عددی 16 بیتی است دو خانه از حافظه را اشغال کرده است.