CS 271 Computer Architecture and Assembly Language

Self-Check for Lecture #3

Solutions

1. What's wrong with the following data segment?

```
.data
  x_value    DWORD    5
  7Eleven   BYTE     "My job",0
  X_VALUE   DWORD    500
  Age       ;user’s age; DWORD ?
```

- x_value and X_VALUE are the same variable, since MASM is not case-sensitive.
- 7Eleven is an invalid variable name (can’t start with a digit).
- Age ;user’s age; DWORD ? The first semi-colon makes all of the rest of the line into a comment.

2. The following data segment starts at memory address 1400. What is the address of each variable?

- `.data`
- myName BYTE "Elmer Fudd",0 ; Address = 1400, size = 11
- yourName BYTE 30 DUP(0) ; Address = 1400+11 = 1411, size = 30
- myAge DWORD 45 ; Address = 1411+30 = 1441, size = 4
- yourAge DWORD ? ; Address = 1441+4 = 1445, size = 4
- myScore DWORD ? ; Address = 1445+4 = 1449, size = 4
- yourScore DWORD ? ; Address = 1449+4 = 1453, size = 4

3. Why is it a good idea to implement a program’s output first?

As soon as the output is displayed, you can check to see if it fulfills layout specifications. The greatest advantage, however, is that the rest of the program development will be much easier to debug, since results will be displayed as the program’s processes are implemented.

4. What’s the result of the following code fragment? I.E., what registers are changed?

```
mov    eax,100
cdq
mov    ebx,13
div    ebx
```

```
Registers changed:
edx contains 9 (integer remainder of 100 / 13)
edx contains 13 (assigned, unchanged by division)
eax contains 7 (integer quotient of 100 / 13)
```

Given the following constant definition and data segment:

```
MY_CREDITS = 12
.data
  x      DWORD    12
  y      DWORD    13
  z      WORD     25
```

5. What’s wrong with the following code segment statements?
Given the following data segment:

```plaintext
.data
intro_1       BYTE    "Welcome,"
username      BYTE    "Fred."
intro_2       BYTE    "What's up?"
count         DWORD   0
```

6. What is displayed by the following code segment statements?

```plaintext
mov    edx, OFFSET intro_1
CALL   WriteString
CALL   CrLf
mov    edx, OFFSET username
CALL   WriteString
CALL   CrLf
mov    edx, OFFSET intro_2
CALL   WriteString
CALL   CrLf
```

```
Welcome, Fred. What's up?
Fred. What's up?
What's up?
```

Each call to WriteString displays memory until a zero is encountered.