X86 Assembly Language – Programming Exercise

Assignment 03 (Ch07)

Practice in Chapter 7

- Multiplication and division instructions

Greatest Common Divisor

- The greatest common divisor (GCD) of two integers is the largest integer that will evenly divide both integers. The GCD algorithm involves integer division in a loop, described by the following pseudocode:

```
int GCD(int x, int y)
{
    x = abs(x)        // absolute value
    y = abs(y)
    do {
        int n = x % y
        x = y
        y = n
    } while (y > 0)
    return x
}
```

Greatest Common Divisor (Cont'd)

- Implement this function in assembly language. Write a non-recursive procedure **CalcGcd** to calculate the GCD of two integers received from eax and ebx, and return EAX as GCD calculated for display. This is an example in action:

x	%	y	=	n
10		24		10
24		10		4
10		4		2
4		2		0
2		0		

Greatest Common Divisor (Cont’d)

- The program will be run like this:

```
Enter a 32 bit number: 10
Enter a 32 bit number: 24
Greatest common divisor is: 2

Enter a 32 bit number: -100
Enter a 32 bit number: 48
Greatest common divisor is: 4
```

Submission Format

- Turn in your report **in group**
- Observe the changes of registers and memory while executing. **Screenshot when needed.**
- Pack (archive) the following files
  - xxxx.asm
  - Report (file format: .doc/.docx/.pdf)
    - Follow the given programming exercise report format
    - Please include screenshots, the source code with comments, and your feedback about the assignment in one file.
- Upload to Tronclass