Introduction to 8086 Assembly

Lecture 1

Behrooz Nasihatkon

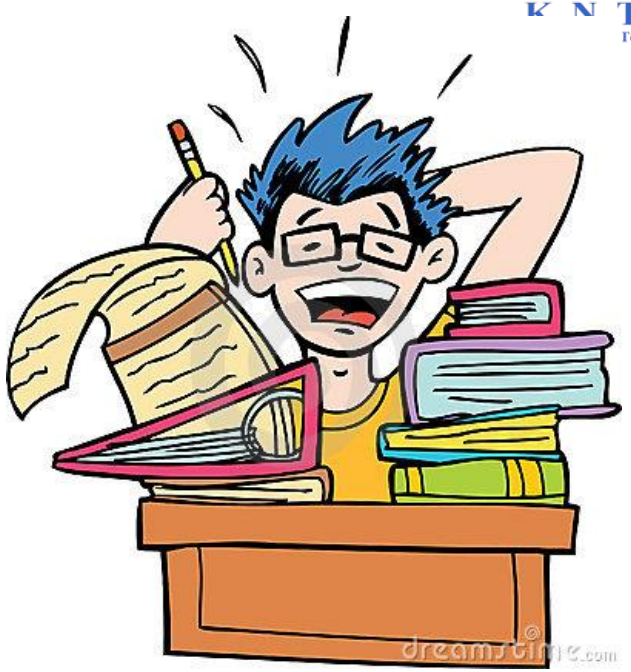
Introduction to 8086 Assembly Language

K. N. Toosi
University of Technology

- 3 credits
- Saturday, Wednesday
  - Group 1: 10:30-12:30 AM
  - Group 2: 13:30-15:30 AM
- Instructor: Behrooz Nasihatkon
- Email: nasihatkon@kntu.ac.ir
- Room: EC building, level 3

Grading

- Homework Assignments
- Project(s)
- Midterm Exam
- Final Exam



What is considered cheating?

K. N. Toosi
University of Technology

Roll call

K. N. Toosi
University of Technology

Special needs

K. N. Toosi
University of Technology

Auditing the course

K. N. Toosi
University of Technology

Recording the lectures

K. N. Toosi
University of Technology

Eating in class

K. N. Toosi
University of Technology

How to get help?

K. N. Toosi
University of Technology

Asking questions!

K. N. Toosi
University of Technology

How to give feedback?

K. N. Toosi
University of Technology

Anonymous form:
https://goo.gl/zPxBAS

Join the Telegram Channel

K. N. Toosi
University of Technology

https://t.me/asmkntuf98

Course Website

- https://wp.kntu.ac.ir/nasihatkon/teaching/asm/f2019/index.html

Previous offerings:

- https://wp.kntu.ac.ir/nasihatkon/teaching/asm/s2019/index.html
- https://wp.kntu.ac.ir/nasihatkon/teaching/asm/f2018/index.html
- https://wp.kntu.ac.ir/nasihatkon/teaching/asm/s2018/index.html
- https://wp.kntu.ac.ir/nasihatkon/teaching/asm/f2017/index.html

Resources

K. N. Toosi
University of Technology

- Carter, Paul A. ***PC Assembly Language***, 2007
  - http://pacman128.github.io/pcasm/
- **NASM tutorial**
  - http://cs.lmu.edu/~ray/notes/nasmtutorial/
- **TutorialsPoint**
  - https://www.tutorialspoint.com/assembly_programming
- ***GOOGLE!***

Further study:

- Hyde, Randall. ***The art of assembly language***. No Starch Press, 2010.
  - **Linux:** http://www.plantation-productions.com/Webster/www.artofasm.com/Linux
  - **Windows:** http://www.plantation-productions.com/Webster/www.artofasm.com/Windows/
- Blum, Richard. ***Professional assembly language***. John Wiley & Sons, 2007.

What is Assembly language?

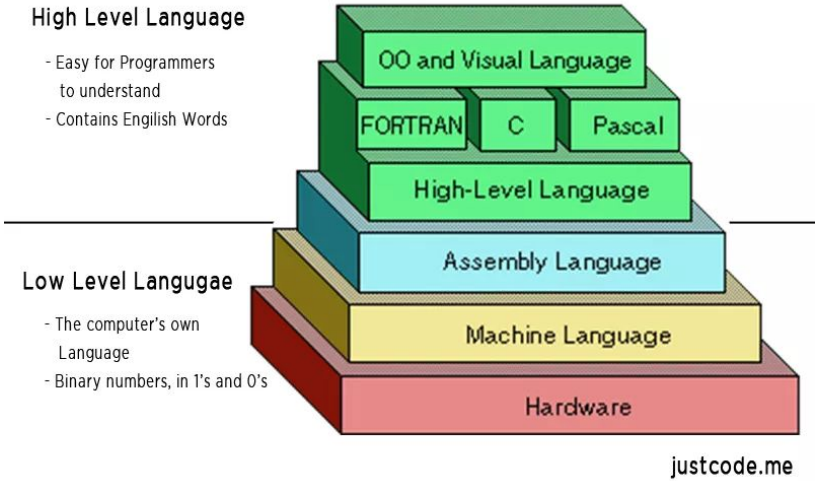
What is Assembly language?

K. N. Toosi
University of Technology

https://me.me/i/ssembly-gu-ge-cs-student-llove-programming-5644fb641baa4609aec4adc8ff5742cf

What is Assembly language?

K. N. Toosi
University of Technology



http://justcode.me/assembly/introduction-assembly-language-examples/

How many assembly languages are there?

K. N. Toosi
University of Technology

https://knowyourhandheld.weebly.com/blog/what-are-the-necessary-features-in-latest-smartphones

Why assembly?

Why assembly?

- Going low-level!
- **Getting insight**
  - **How programming languages are implemented (code, variables, arrays, functions, etc.)!**
  - **How compilers work**
- Writing efficient programs (?)
- System programming
- Writing device drivers
- Interfacing with high-level languages like C
- Reverse engineering
- New CPU features

x86 & x86-64 Assembly

K. N. Toosi
University of Technology

AT&T vs Intel Syntax

K. N. Toosi
University of Technology

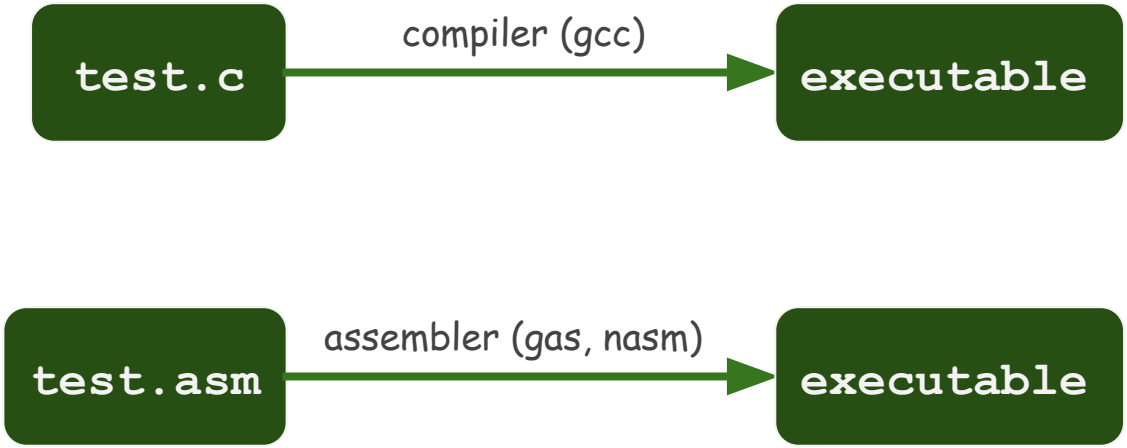
https://en.wikipedia.org/wiki/X86_assembly_language#Syntax

```
movq    %fs:40, %rax
movq    %rax, -8(%rbp)
xorl    %eax, %eax
leaq    -16(%rbp), %rax
movq    %rax, %rsi
movl    $.LC0, %edi
movl    $0, %eax
call    __isoc99_scanf
movl    -16(%rbp), %eax
addl    %eax, %eax
leal    3(%rax), %edx
movl    -16(%rbp), %eax
imull   %edx, %eax
movl    %eax, -12(%rbp)
```

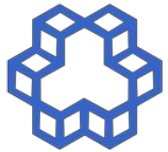
```
sub     rsp, 16
mov     rax, QWORD PTR fs:40
mov     QWORD PTR [rbp-8], rax
xor     eax, eax
lea     rax, [rbp-16]
mov     rsi, rax
mov     edi, OFFSET FLAT:.LC0
mov     eax, 0
call    __isoc99_scanf
mov     eax, DWORD PTR [rbp-16]
add     eax, eax
lea     edx, [rax+3]
mov     eax, DWORD PTR [rbp-16]
imul    eax, edx
```

What is an **Assembler**?

K. N. Toosi
University of Technology



Major Assemblers

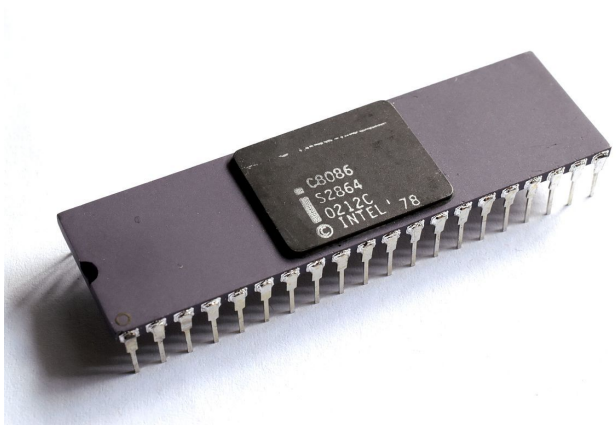


- Microsoft Assembler (MASM)
- GNU Assembler (GAS)
- Flat Assembler (FASM)
- Turbo Assembler (TASM)
- **Netwide Assembler (NASM)**

Backward compatibility

K. N. Toosi
University of Technology

- Look at
  - https://en.wikipedia.org/wiki/X86



Our platform

- **Hardware**: 80x86 processor (**32**, 64 bit)
- **OS**: Linux
- **Assembler**: Netwide Assembler (NASM)
  - + GNU Assembler (GAS)
- **C Compiler**: GNU C Compiler (GCC)
- **Linker**: GNU Linker (LD)

How does an assembly code look like?

Write a C program named **test.c**.

Compile it to x86 assembly language, the **AT&T syntax**

>>> gcc -S -o att.s test.c

Now compile to the **Intel syntax**:

>>> gcc -S -masm=intel -o intel.s test.c

Compare the two assembly syntaxes (output files att.s and intel.s)