1-512-256-0197
www.mindshare.com
training@mindshare.com

x86 Assembly Language Programming

Let MindShare Bring “x86 Assembly Programming” to Life for You

This course covers the basics of programming in the x86 assembly language. Assembly language is virtually the lowest level programming that can be done on x86 processors which makes it the most flexible. All code written in this class can run on the latest processors from Intel (e.g. Nehalem, Core 2, Atom) and AMD (Opteron, Phenom). Topics include logic, arithmetic, control transfer, and data transfer instructions, data addressing modes, assembling and linking programs, modular programming, linking to higher level languages and inline assembly. This course also includes discussions of both 32-bit and 64-bit application instructions. All topics are reinforced during lab exercises. Emphasis is on understanding the fundamentals of x86 application programming.
Note: Protected Mode is not a focus in this class; see Protected Mode Class description for details on protected mode.

You Will Learn:

- How to write assembly language programs for x86-based processors
- How to modularize and link these programs to other languages like C

Course Length: 4 Days

Who Should Attend?

This course is a must for processor validation engineers who will be writing low-level assembly code to validate features and functionality of the processor. This course is also beneficial for software and hardware engineers, technicians, and others needing a fundamental working knowledge of x86 assembly language programming.

Course Contents:

- **Introduction**
  - We introduce several basic concepts. Topics include memory space, I/O space, memory and I/O access, hexadecimal numbers, CPU registers, flag word, instruction format, data transfer, and the loop instructions.
- **Debugging**
  - We explain how to use a debugger to help debug our lab exercises. Topics include go, single step, breakpoints, displaying memory, displaying registers, displaying I/O ports, disassembly, assembly, and saving memory to a disk.
- **Control Transfer**
  - We discuss "short" and "near" control transfers. Topics include relative jumps, unconditional jumps, conditional jumps, procedures, and the stack.
- **Program Development**
  - We discuss program development. Topics include assembling, linking, equates, and list files.
- **A Few DOS Functions**
  - We discuss a few simple DOS I/O functions for use in the lab exercises. Topics include character input, character output, string output, and keyboard status checking.
- **Defining and Accessing Data**
  - We discuss data declarations. Topics include declarations for bytes, words, dwords, fwords, qwords, tbyte, declarations for arrays, accessing memory, accessing arrays, and addressing modes using base registers, index registers, and scale factors.
- **Arithmetic, Logic, and Shifts**

1-800-633-1440
www.mindshare.com
training@mindshare.com

  - We present the arithmetic, logic, and shift/rotate instructions. Topics include addition, subtraction, multiplication, and division instructions, shift and rotate instructions, logic instructions, and flag updating and usage.
- **Segmentation**
  - We begin a discussion of segmentation. Topics include segment registers, base addresses, offsets, address calculation, defining segments, and segment register initialization. Note: Protected Mode is not discussed in this class, see Protected Mode Class for details about protected mode.
- **Segmentation Revisited**
  - We continue the discussion of topics related to segmentation. Topics include "far" control transfers (jumps, calls, and returns), segment override prefixes, and the need for the "ASSUME" statement.
- **Programming Techniques**
  - We discuss two useful advanced programming concepts. Topics include indirect transfers (near and far), jump tables, and parameter passing on the stack for high-level language interfacing.
- **Modular Programming**
  - We show how to combine modules. Topics include modules, external symbols, "EXTRN" and "PUBLIC" declaratives, referencing external code and data, combining segments, and linking.
- **Using the 32-bit and 64-bit Extensions**
  - We discuss the use of the 32-bit and 64-bit extensions. Topics include a review of 32-bit registers, 64-bit registers, eflags, address and operand override prefixes, REX prefix, data definitions, available instructions, new addressing modes, and several new application instructions.
- **String Instructions**
  - We show how to use the "string" (block transfer) instructions. Topics include "string" instructions, default pointers, direction flag, and repeat prefixes.
- **Structures**
  - We discuss assembly language structures. Topics include structure definitions, structure initialization, structure field access, and using structures for data structure templates.

Recommended Prerequisites:

Some previous programming experience with a computer language (high- or low-level).

Course Material:

MindShare will supply a hardcopy and electronic version of the presentation slides including the lab descriptions.