

Real Time Group

Real Time & Embedded Linux Solutions

Course: C for Embedded Systems

Duration: 90 Hours

Hands-On-Training: 85%

רח' רוז'נסקי 14 ראשון לציון טל. 077-7067057 / 050-3309318 פקס 077-5067058

www.rt-hr.co.il

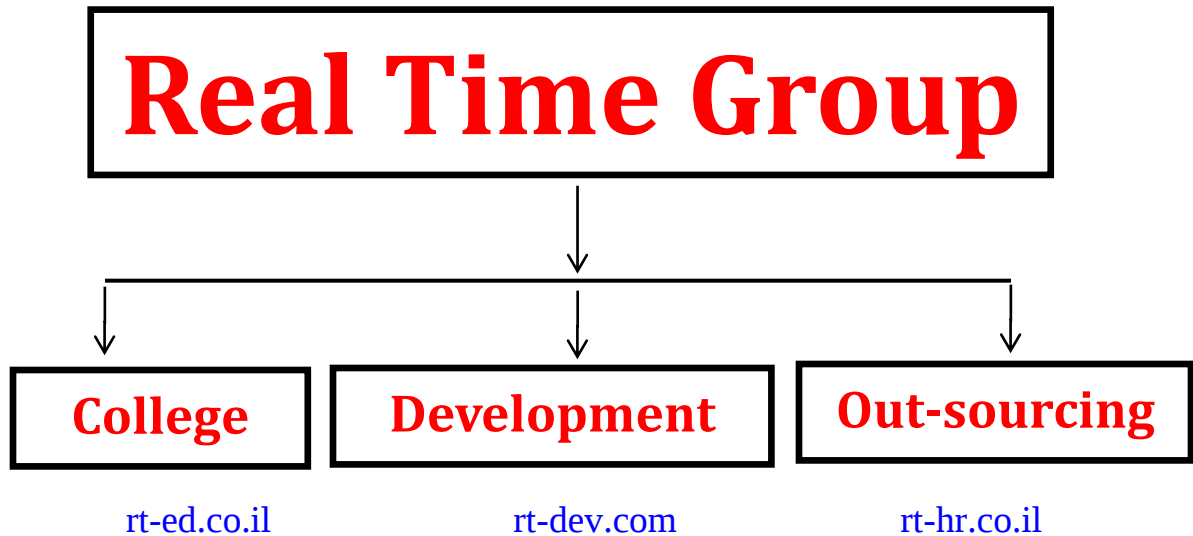
www.rt-ed.co.il

www.rt-dev.com

Real Time Group is a multi-disciplinary dynamic and innovative Real-Time O.S. and Embedded Software Solutions Center, established in 2007.

Providing Bare-Metal and Embedded Linux solutions, professional services and consulting, end-to-end flexible system infrastructure, outsourcing, integration and training services for Hardware, Software and RT-OS \ Embedded Systems.

The company is divided into the following three Divisions:



Training Division:

Professional Training Services for Hardware, Software, RT-OS and Embedded systems industries.

We provide the knowledge and experience needed to enable professional engineers to Develop, Integrate and QA Hardware, Software and Networking Projects.

In order to insure experience, all courses are practical – hands-on-training. The latest Development, QA and Automation equipment which are adopted by the industry are used.

All students are supplied with Development-Boards for home-work and course projects.

רח' חז'נסקי 14 ראשון לציון טל. 077-7067057 / 050-3309318 פקס 077-5067058

www.rt-hr.co.il

www.rt-ed.co.il

www.rt-dev.com

Course Overview:

This course will teach you to program the C language from the ground up.

You will learn everything from the very fundamentals of programming right through to the complexities of pointers, File IO , Structure , Linked list and more...

C is one of the most important of all programming languages.

It is used to program desktop applications, compilers, tools utilities and hardware devices.

Because The C language mostly used for low level programming , embedded linux and kernel programming , This course will emphasize on C language from an embedded point of view, How to increase performance, efficiency and flexibility when using C.

in this course we will have an introduction to C and move on to the most Complex elements and tools in the C language .

Who should attend:

- Anyone that wish to learn C programming language.
- Hardware/Electrical/software engineers who need to use C for Embedded systems.

Prerequisite:

- English language – must
- Knowledge in operation systems - advantage
- Background in software implement - advantage

C for Embedded Systems :

1. Introduction to Embedded Systems:

- a. How Embedded System Works
- b. Where is the Program Located in Memory?
- c. The steps to run a Program
- d. How the Program is Represented in RAM – Code, Data and Stack Segments
- e. Your first C Program - "Hello World"
- f. The main () function.
- g. Basic I/O Commands.

2. Compiling and Linking Process:

- a. What is a "Compiler", why is it essential?
- b. The Pre-Processor
- c. The Cross-Compiler
- d. The Linking Processes
- e. COFF & ELF file structures

3. Data Types

- a. Declaring Variables
- b. Variable Names (by Code Conventions)
- c. Variable Types.
- d. Storage considerations (how Variables are stored in Memory).
- e. Initializing Variables
- f. The sizeof () Usage
- g. Signed VS Unsigned variables
- h. Typedef Usage & Declaration
- i. Casting
- j. I/O commands – printf () & scanf ()
- k. Printf () Format specifier

4. Constants & Enumeration

- a. Using the #define macro
- b. The const keyword
- c. Differences

רח' רוזנסקי 14 ראשון לציון טל. 077-7067057 / 050-3309318 פקס 077-5067058

5. Expressions:

- a. What is an Expression in C?
- b. Operators
- c. Assignment Operators
- d. Mathematical Operators
- e. Integer Division and Modulus
- f. Increments and Decrement
- g. Prefix and Suffix
- h. Boolean Expressions.
- i. Logical Operators.

6. Statements:

- a. The if Statement
- b. The else Keyword
- c. Nested if Statement
- d. Else if Statements
- e. Ternary operators
- f. The switch Statement

7. Using Loops:

- a. Loop Types.
- b. While Loops
- c. The break Statement
- d. The continue Statement
- e. The do.... while Statement
- f. The for Loop
- g. "for" VS "while"
- h. Nested for Loops

8. Functions and Headers:

- a. What is a Function in C?
- b. Declaring and Defining a Function
- c. Modular usage of functions
- d. Variable Scope
- e. Local Variable

רח' רוזנסקי 14 ראשון לציון טל. 077-7067057 / 050-3309318 פקס 077-5067058

- f. Global Variable
- g. Recursion

9. Arrays:

- a. What is an Array ?
- b. Declaring Arrays in C.
- c. Accessing Array Elements.
- d. Array Calculations in Memory size
- e. Using Initiators to initialize Arrays
- f. Multidimensions Arrays

10. Pointers:

- a. Direct ver Indirection access.
- b. What are pointers ?
- c. Using pointers
- d. Pointer arithmetic's
- e. By-ref parameters
- f. Invalid pointers
- g. Pointers vs. arrays:
- h. Function Pointers – Declaring and Usage

11. Strings:

- a. How "string" Define in C?
- b. String location Memory
- c. Strings Initialization
- d. Char [] VS char*
- e. Useful <string.h> routines
- f. Working String on bytes resolutions

12. Dynamic Memory Allocation

- a. The Free Store (Heap)
- b. malloc ()
- c. realloc ()
- d. free ()
- e. Pointers calculations
- f. Memory Leaks.

רח' רוזנסקי 14 ראשון לציון טל. 077-7067057 / 050-3309318 פקס 077-5067058

- g. Malloc/free Warning
- h. Dynamic Arrays on the Free Store
- i. Stray or Dangling Pointers

13.Arguments to main() from the command line:

- a. argc & argv – What are they?
- b. Usage of args to main ()

14. Structure in C:

- a. What is a Structure in C?
- b. the struct keyword
- c. Accessing Structure Members Directly
- d. Accessing Structure Members Indirectly
- e. Dynamic Memory Allocation of Structures
- f. Allocating structures and arrays of structures on Heap.

15.Bitwise Operations:

- a. Bitwise Operations – Usage
- b. bitwise Operators
- c. bitwise shift Operator
- d. Changing Specific Bits
- e. Read Modify Write

16.Big & Little Endian:

- a. Big & Little Endian – Usage
- b. Endianness in Networking

17.Bit fields & Unions:

- a. What are bit-fields in C?
- b. bit-field Declaration
- c. Accessing bit-field Members.
- d. Bit fields – Example in Embedded System
- e. Bit fields – Note of Caution
- f. What is a Unions in c?
- g. Union Definition
- h. Unions Usage in embedded system.

רח' חז'נסקי 14 ראשון לציון טל. 077-7067057 / 050-3309318 פקס 077-5067058

18.I/O commands:

- a. Background to I/O
- b. FILE structure
- c. FILE API
- d. Standard Stream in Unix
- e. Getchar () VS getc ()
- f. Putchar () VS putc ()
- g. Scanf () – Related Function
- h. printf () – Related Function

19.The extern Keyword:

- a. The extern Keyword - Usage
- b. The extern Keyword - Declaration
- c. extern – compiling and linking
- d. extern – Behind the scene

20.Static Variable:

- a. Static Variable – What is it?
- b. Static Variable – Declaration and Usage.
- c. register Variable
- d. Volatile Variable

21. Linked List:

- a. Singly Linked List
- b. add () & remove () examples
- c. Link list VS Array
- d. Double Linked List
- e. Using Stacks & Queue with Linked list
- f. Binary Trees in c
- g. Advance Tree Concept

22.Hash Tables:

- h. Hash Tables – Usage
- i. Hash Tables – implementation
- j. Hash Function Example

23.Sorting Methods: Insertion Sort

- e. Selection Sort
- f. Shell Sort
- g. Merge Sort
- h. Binary Tree Sort

רח' רוז'נסקי 14 ראשון לציון טל. 077-7067057 / 050-3309318 פקס 077-5067058

- i. Heap Sort
- j. Quick Sort