Linux Programming with C and Reverse Engineering Essentials

By Milad Kahsari Alhadi

Last Update: **Friday - 2019 02 August**

- **Procedural Programming with C – 565 Min**
  - Integrated Development Environment
    i. Introduction to Linux's IDEs
      1. Introduction to the Eclipse
      2. Download Eclipse IDE for Linux
      3. Creating a C/C++ Project
      4. C Project Toolchain
  - Compile and Linking of the Project
    i. GCC Project' Profile
      1. Cross GCC
      2. Linux GCC
    ii. Executable
      1. Debug Mode
      2. Release Mode
  - Debugging of the Project
    i. GDB Debugger
    ii. Eclipse Local Debugger
      1. Disassembly Perspective
      2. Registers
      3. Breakpoint
      4. Runtime
      5. Console
      6. Memory
  - Eclipse Settings
    i. GCC Compiler Commands
      1. ggdb
      2. O3

3. Wall
4. fmessage-length

ii. Configuration Settings

iii. Matintenance of the Project

- Introduction to Programming Language

i. Introduction to Algebra

1. Matlab
2. Input Value – X Variable
3. Functions / Equations
4. Output Results – Y Variable
5. Graphing the Output

ii. Mathematical Thinking

1. Problem Analysis with Mathematics
2. Designing an Algorithm
3. Write Pseudo Code for the Problem
4. Rewrite Pseudo Code with Matlab Script
    a. Clearing
    b. Figure
    c. Arrays
    d. Plot
    e. Title
    f. Labeling
5. Rewrite Matlab Scripts with C
    a. Arrays
    b. Strings
    c. Functions
    d. Operators

- Overview of the C Language

i. Introduction to C

1. Ken Thompson's B
2. C Specification
3. Unix's Interfaces
    a. Portable Operating System Interface
    b. Single UNIX Specification
4. C Paradigms
    a. Procedural Programming
    b. Structural Programming
    c. Inline Programming

ii. Why we need C?
    1. Security
    2. Efficiency
    3. Protability
    4. Performance
    5. Compatibility
    6. and Multithreading
iii. C17 ISO Standard

— Problem Solving and Solution Engineering

i. Problem Solving Phases:
    1. Engineering Mindsets
    2. Improvement and Maintenance
        a. Profiling
        b. Debugging
        c. Disassembling
    3. Performance Measurement
ii. First I/O Program
    1. Analysis the Problem
    2. Devising an Algorithms
    3. Implement the Program
iii. Debugging the Project with GDB in Eclipse
    1. C Components
        a. Variables
        b. Headers
        c. Functions
        d. Macros
        e. Comments
    2. Linux Manual Page
    3. GDB Dashboard
        a. Disassembly
        b. Source-code
        c. Registers
        d. Memory Map

— Compile and Preprocessing

i. What is Preprocessing?
    1. Macros
    2. Macro based Processing
        a. Substitutions
        b. Expansions

ii. Gnu Compiler Phases
    1. –E Parameter – Preprocessing
        a. Source Code Expansion
        b. Lexical Analysis
        c. Syntax Analysis
        d. Context Analysis
    2. –S Parameter – Compilation
        a. Assembly List Generating
        b. Assembly List Optimization
    3. –O Parameter – Linking
        a. Assembling
        b. Linking
            i. Final Executable
            ii. ELF File
iii. Headers and Implementation
    1. Headers
    2. Implementation
    3. Hide Implemenation of Functions

– Machine Memory and Addressing

i. Memory Addresses
    1. Address
    2. Labels
    3. Values
ii. Declaration and Defination
    1. Memory Name – Variables
    2. Pointer Variables – Memory Parsing
iii. Variable Data Types and Their Size
    1. Int
    2. Char
    3. Void
    4. Float
    5. Double
    6. Strings
iv. Expressions and Statements

– Advanced C-based Data Types

i. Void Data Type
    1. Address Translation
    2. Address Casting

- ii. Boolean Data Type
  1. Duality of Values
     - a. True
     - b. False
  2. Boolean Equations
     - a. Logical Operations
       - i. And
       - ii. Or
       - iii. Xor
       - iv. Not
       - v. Nand
       - vi. Nor
     - b. Hardware Gates
     - c. Stdbool C Header
- iii. Strings Data Type
  1. C Based Chars Array
     - a. Null-Terminated Strings
     - b. Multidimensional Arrays
     - c. Quotation Marker
  2. Strings-based Functions
     - a. Printf
     - b. Scanf
     - c. Putchar
     - d. Getchar
     - e. Strcmp
     - f. Strcpy
     - g. Memcpy
     - h. Strlen
  3. Challenge 1 and 2

— Procedural Programming with C

- i. C-based Functions
  1. Signture
  2. Return and Input Value
  3. Passing by Value
  4. Passing by Pointer
- ii. Return Value of Functions
  1. Return Value Analysis with Radare2
  2. Return Value Analysis with Hopper
- iii. Recursive Function

        1. Recursive Analysis with GDB
        2. Recursive Analysis with Radare2
        3. Recursive Analysis with Hopper
    iv. Challenge 3 and 4
- Conditional Statements
    i. Nuclear Program Implementation
        1. If-else statement
        2. Switch statement
        3. Runtime Error Handling
        4. Disassembly analysis with Hopper
    ii. Challenge 5
- Loop and Repeation Statements
    i. Repeator Program Implementation
        1. For-loop
        2. While-loop
        3. Do-while-loop
        4. Breaking and Continue
        5. Disassembly analysis with Hopper
    ii. Challenge 6
- Structures, Enumeration and Unions
    i. Declaration, Definiation and Identifier
    ii. Pointers to Functions and Its Disassembly
    iii. Unions and Its Initialization Issue
    iv. Enumeration and Simulating an ICS Environment
- Dynamic Memory Managment
    i. Heap Memory Layout
    ii. Memory Allocation
    iii. Memory Management
    iv. Stack vs. Heap
    v. Disassembly Analysis of Malloc
    vi. Basics of Linked Lists Data Structure
    vii. Macros and Logging with Functions Like Macros
    viii. Pointers to Functions and Its Disassembly Formation