

# EMBEDDED LINUX APPLICATION DEV.

CODE: ELDAPP - 4M

## 1. EMBEDDED C PROGRAMMING

- a. INTRODUCTION TO EMBEDDED C PROGRAMMING
- b. COMPILATION PROCESS, DESIGN APPROACH
- c. DATA TYPES, VARIABLES, LOOPS, NESTED LOOPS
- d. FUNCTIONS, ARRAYS
- e. STRUCTURE, UNIONS, TYPEDEF, ENUMS
- f. POINTERS
- g. STRING OPERATION
- h. WORKING ON BITS MANIPULATIONS
- i. ADVANCED USE OF THE PREPROCESSOR
- j. DATA STRUCTURES
  - \*LINKED LIST

## 2. OPEN SOURCE DEVELOPMENT

- a. OS BASICS
- b. INTRODUCTION TO LINUX OS
- c. LINUX ARCH VS WIN NT ARCH
- d. COMMANDS IN LINUX
- e. LINUX SHELL SCRIPTING
- f. GNU TOOL CHAIN
  - \*GCC COMPILER
  - \*GDB DEBUGGER
  - \*MAKE UTILITY
- g. BASIC LINUX PROGRAMMING
- h. LINUX PROCESS MANAGEMENT
  - \*TASK STRUCTURE AND PROCESS TABLE

\*CREATION, TERMINATION AND DAEMON PROCESS

\*LINUX SCHEDULER

- FIFO
- ROUND ROBIN
- PREEMPTIVE [PRIORITY]

### **3. LINUX SYSTEM PROGRAMMING**

#### **a. SYSTEM CALLS (POSIX)**

#### **b. SIGNAL**

- \*SIGNALS ON 32 BIT OS
- \*SYSTEM CALL IN SIGNALS
- \*WORKING ON SIGNAL HANDLER
- \*SIGNAL MASKING

#### **c. IPC MECHANISMS**

\*PIPES

- PIPES CREATION & CONTROL OPERATION
- PIPES SYSTEM CALL INTERFACES

\*FIFOS

- PIPES VS FIFO
- FIFO SYSTEM CALL INTERFACES

\*SEMAPHORES

- SEMAPHORE SYSTEM CALL INTERFACE
- SEMAPHORE CONTROL OPERATIONS

\*SHARED MEMORY

- SHARED MEMORY SYSTEM CALL INTERFACES
- SHARED MEMORY CONTROL OPERATIONS

\*MSG. QUEUE

- SYSTEM MSG. QUEUE CALL INTERFACES
- MSG. QUEUE CONTROL OPERATIONS

#### **d. SOCKET PROGRAMMING**

\*NETWORK LAYERS

- \*TCP/IP
- \*TYPES OF SOCKETS
- \*SYSTEM CALL INTERFACES
- \*LOCAL HOST -- UNIX BSD
- \*REMOTE HOST – INET
- \*MULTI CLIENTS

## 4. LINUX INTERNALS

### a. LINUX BOOT UP SEQUENCE

### b. LINUX MEMORY MANAGEMENT

\*AN ABSTRACT MODEL OF VIRTUAL MEMORY

\*LINUX MEMORY MANAGEMENT

- PAGE TABLES
- PAGE ALLOCATION AND DE-ALLOCATION
- MEMORY MAPPING
- DEMAND PAGING
- PAGE CACHE
- SWAPPING OUT AND DISCARDING PAGES
- BUFFER CACHE
- SWAPPING OUT SHARED MEMORY PAGES
- SWAP CACHE
- SEGMENTATION

### c. LINUX DEVICE DRIVERS DEVELOPMENT

\*BASIC I/O MECHANISMS

\*LINUX PORT PROGRAMMING

- PARALLEL PORT PROGRAMMING
- SERIAL PORT PROGRAMMING

\*KERNEL INTERRUPT HANDLING MECHANISMS

\*DEVICE DRIVERS

- MEMORY
- INTERFACING MODULES/DRIVERS WITH KERNEL
- INTERRUPT HANDLING (BOTTOM HALVES, KERNEL TIMERS)
- SPIN LOCKS

- CHAR. DD
  - INTRODUCTION OF BLOCK DEVICE DRIVER
    - \* STRUCTURES AND INITIALIZATION
  - INTRODUCTION OF NETWORK DEVICE DRIVER
    - \*STRUCTURES AND INITIALIZATION
- \*CUSTOMIZING LINUX, BOOT IMAGE

## **5. INTRODUCTION TO REAL TIME OPERATING SYSTEMS**

### **6. INTEL 8051**

- a. ARCHITECTURE
- b. INSTRUCTION SET, PIN DIAGRAM
- c. MEMORY (RAM, ROM), STACK
- d. SFR'S, FLAGS, ADDRESSING MODES
- e. TIMERS, CLOCKS
- f. INTERRUPTS, ISR
- g. INTERFACING WITH KEYBOARD, STEPPER MOTOR, ADC/DAC,LCD
- h. 7-SEGMENT DISPLAY, RTC, RELAYS, SERIAL COMMUNICATION