

# EMBEDDED LINUX RTOS DEV.

CODE: (EMBD-6M)

## 1. EMBEDDED SYSTEM DEVELOPMENT

- a. HARVARD /SUPER HARVARD ARCH/ VON NEUMANN
- b. TYPES OF MICRO-CONTROLLERS
- c. RISC VS CISC
- d. MEMORY HIERARCHY
- e. 8 BIT MICROCONTROLLERS

## 2. 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

## 3. KEIL (DEVELOPMENT TOOL)

- a. DISCUSSION ON DIFFERENT MEMORY MODULES
- b. POINTERS & EXTENSIONS IN C51, ABSOLUTE VARIABLE DECLARATIONS
- c. STORAGE CLASSES & ITS IMPORTANCE IN EMBEDDED SYSTEMS
- d. FUNCTIONS AND EXTENSIONS IN C51, DISCUSSION ON
- e. RE-ENTRANCY, INTERRUPTS & IMPORTANCE OF GLOBAL VARIABLES
- f. USAGE OF BIT WISE OPERATORS IN EMBEDDED SYSTEMS

## 4. EMBEDDED C

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

 LINKED LIST

## 5. INTRODUCTION TO RTOS

- a. RTOS BASICS
- b. RTOS VS. GENERAL PURPOSE OPERATING SYSTEM
- c. TYPES OF KERNELS
- d. TYPE OF RTOS
  - i. HARD RTOS
  - ii. SOFT RTOS

## 6. TORNADO IDE-2.0

- a. ARCHITECTURE
- b. WIND SHELL
- c. TARGET SERVER
- d. VX-SIM
- e. WIND-DEBUG

## 7. VXWORKS5.4 ARCHITECTURE

- 
- 8. TASK MANAGEMENT IN VXWORKS**
  - 9. CPU SCHEDULING**
    - a. ROUND ROBIN
    - b. RR WITH KERNEL TIME SLICE
    - c. PRIORITY BASED PRE-EMPTIVE
  - 10. PRIORITY INVERSION**
  - 11. DELETION SAFETY**
  - 12. WATCHDOG TIMERS**
  - 13. MULTI TASKING**
  - 14. INTER TASK COMMUNICATION ON VXWORKS**
    - a. SEMAPHORE
    - b. SHARED MEMORY
    - c. MESSAGE QUEUES
  - 15. SOCKET PROGRAMMING**
  - 16. TARGET DEVICE BRINGING-UP ACTIVITY**
    - a. PREPARING TARGET DEVICE
    - b. BOOTING UP TARGET DEVICE
    - c. ATTACHING TARGET DEVICE
  - 17. PORTING APPLICATION ON TARGET DEVICE**
  - 18. REMOTE DEBUGGING**
  - 19. KERNEL CUSTOMIZATION AND PORTING**
  - 20. INTRODUCTION TO BSP DEVELOPMENT**
  - 21. INTRODUCTION TO NANO KERNEL**

## **22. INTRODUCTION TO THE DEVICE DRIVER**

- a. EVOLUTION OF LINUX
- b. ROLE OF DEVICE DRIVER
- c. INTRODUCTION TO DIFFERENT KERNEL MODULE – PEEK INSIDE THE KERNEL
- d. TYPE OF DEVICES AND KERNEL MODULE
- e. KERNEL MODULES DEVELOPMENT V/S APPLICATION DEVELOPMENT.
- f. USER SPACE DRIVER V/S KERNEL SPACE DRIVER.
- g. COMPILING AND LOADING MODULES
- h. BACKWARD COMPATIBILITY AND VERSION DEPENDENCY.
- i. COMPILING AND LOADING FOR MULTI-PROCESSOR SYSTEM

## **23. CHARACTER DRIVERS**

- a. CHARACTER DRIVER BASICS
- b. MAJOR AND MINOR NUMBER ALLOCATION
- c. FILE OPERATION FOR CHARACTER DEVICES
- d. RACE CONDITION AND MODULE USAGE COUNT
- e. ENHANCED CHARACTER DRIVER OPERATIONS
- f. MEMORY MANAGEMENT
- g. IO MEMORY AND IO PORTS
- h. USING IO PORTS AND IO MEMORY
- i. SAMPLE CHARACTER DRIVER FOR PARALLEL PORT.

## **24. INTERRUPT HANDLING**

- a. INSTALLING THE HANDLER
- b. IMPLEMENTING THE HANDLER
- c. TASKLETS AND BOTTOM HALVES
- d. INTERRUPT SHARING.

## **25. SERIAL PORT DRIVERS**

- a. ARCHITECTURE
- b. UART DRIVER DEVELOPMENT
- c. TTY DRIVERS DEVELOPMENT

## **26. INPUT DRIVERS**

- a. INPUT EVENT DRIVER
- b. INPUT DEVICE DRIVER
- c. DEBUGGING AND LOGS MANAGEMENT

## **27. DEVICE DRIVERS IN USER SPACE**

- a. PROCESS SCHEDULING AND RESPONSE TIME
- b. ACCESSING IO REGIONS
- c. ACCESSING MEMORY REGIONS
- d. USER MODE PARALLEL PORT AND SERIAL PORT DRIVERS
- e. ADVANTAGES AND DISADVANTAGES OF USER SPACE DRIVERS

## **28. BLOCK DRIVERS**

- a. INTRODUCTION TO BLOCK DRIVERS
- b. STORAGE TECHNOLOGIES
- c. REGISTERING AND LOADING BLOCK DRIVERS
- d. HANDLING REQUEST
- e. LINUX BLOCK IO LAYER
- f. BLOCK DRIVER AND DATA STRUCTURES
- g. DEVICE EXAMPLE – SIMPLE STORAGE CONTROLLER
- h. HOW MOUNTING AND UN-MOUNTING WORKS
- i. THE IOCTL() METHOD

## **29. MEMORY TECHNOLOGY DEVICES (MTD)**

- a. FLASH MEMORY
- b. LINUX MTD SYSTEM
- c. MAP DRIVERS
- d. NOR CHIP DRIVER
- e. NAND CHIP DRIVER
- f. USER MODULES
- g. MTD UTILITIES
- h. CONFIGURATION OF MTD DEVICES

S-CUBE™