



Advanced Embedded System

First Time in Chhattisgarh

Advanced C Programming	Linux Advanced Programming
<p>Fundamentals of C</p> <ul style="list-style-type: none"> • Data types, Variables and Constants • Use of Memory • Arithmetic & Logical Operators and Expressions • Conditional Statements and Loops <p>C Under Linux</p> <ul style="list-style-type: none"> • Text Editors in LINUX • GNU C Compiler • GNU C Debugger <p>Functions</p> <ul style="list-style-type: none"> • Introduction and Importance • Passing Value/Reference • Recursive Functions • Getting values back from Functions • Affects on Stack • Library and User defined function • Passing Variable number of parameters <p>Arrays</p> <ul style="list-style-type: none"> • Definition, initialization and usage • Multi Dimensional Arrays • Strings • Pointer based arrays • Passing arrays to functions • String handling – with function/without function <p>Storage Class and Scope of Variables</p> <ul style="list-style-type: none"> • Scope and Life • Automatic, Static, External, Register • Memory(CPU / RAM) <p>Pointers</p> <ul style="list-style-type: none"> • Introduction to pointers • Defining pointers • Importance of & and * operators • Pointer Assignment • Pointer Arithmetic • Generic and Null Pointer • Function Pointers • Pointers to Arrays and Strings • Array of Pointers <p>Structures & Unions</p> <ul style="list-style-type: none"> • Introduction to structures • Declaration, initialization • Access to members of structure • Array of Structures 	<p>Linux System Programming</p> <ul style="list-style-type: none"> • Universal I/O calls in detail • Process Handling • Process Creation • Process Termination • Thread Handling • Thread Creation • Process Vs Thread • Signals in Linux • Signal Handling <p>Inter Process Communication (IPC) Mechanism</p> <ul style="list-style-type: none"> • Pipes • FIFOs • Message Queues • Shared Memory • Semaphores • Server Client Communication Between Processes <p>Socket Programming</p> <ul style="list-style-type: none"> • Introduction to TCP/IP Protocols • System calls for Socket Creation • Establishing a connection • Sending and Receiving messages • Self looping as well as inter system communication <p>Linux Kernel</p> <ul style="list-style-type: none"> • Understanding User space and Kernel space • Kernel calls Vs Library Calls • Linux Kernel Architecture • Linux Modules • Configuring Kernel • Makefile and Make Utility • Compiling Kernel • Installing new Kernel • Introduction to Linux Device Drivers <p>Introduction to Linux Device Drivers</p> <ul style="list-style-type: none"> • Understanding a PC Mother board Architecture • Need of driver software • Architecture of drivers • Interfacing of Devices on board • Writing a simple device driver • Implementing Universal I/O calls

<ul style="list-style-type: none"> • Passing structures to functions • Memory Allocation • Structure Comparison • Structure bit operation • Unions • Structure Vs Union • Pointers to Structure and Union <p>Dynamic Memory Allocation</p> <ul style="list-style-type: none"> • Malloc(), Calloc(), Realloc(), Free() • Pointers to Dynamic memory <p>Enumerated data types</p> <ul style="list-style-type: none"> • Enum, Indexing • Enum Vs #define <p>Bit wise Operations</p> <ul style="list-style-type: none"> • Bit wise AND (&), OR (), XOR (^) Logic • Compliment (~) operator • Left-Shift (<<), Right Shift (>>) • Masking, Setting, Clearing and Testing of Bit / Bits <p>File Handling Concepts</p> <ul style="list-style-type: none"> • Introduction to File Data • Inode, FILE structure • Pointer to File • File Accessing Concepts <p>Command line Arguments</p> <ul style="list-style-type: none"> • Argc, argv • Variable Inputs to the main <p>Compiler in Practical</p> <ul style="list-style-type: none"> • Preprocessor Directives • Compiler, Assembler, Linker <p>Data Structures</p> <ul style="list-style-type: none"> • Introduction to Data Structure • Types – Linear & non-linear • Concept of Linked List • Single, and Double Linked Lists • Stacks & Queues • Binary Trees • Sorting and Searching Techniques <p>Simple Programming Practice</p> <p>Complex Logic Development</p>	<ul style="list-style-type: none"> • Application and Driver Communication • Understanding the flow of Communication • Interacting with Hardware • Handling CPU interrupts • Understanding the Interrupt Handling Mechanism • Developing a UART Driver • Developing Driver for PCI Devices and USB Devices
--	--

1. Major and Minor Project Guidance for Engineering Students
2. Vocational Training Certification for enrolled students.