

# UNIX & Linux Kernel Training Syllabus

**Eligibility :** 2 years of exp with Linux and C as a student or as a professional and a BE/MCA or equivalent degree.

## Course content:

### 1. OS introduction

- ✚ OS fundamentals and Linux kernel introduction
- ✚ Process and memory management and file systems introduction.
- ✚ Kernel synchronization and cross-system facilities

### 2. Process Management Topics

- ✚ Concepts of processes, Kernel threads and systems call interfaces.
- ✚ Linux kernel structures for processes
- ✚ Kernel stack internals
- ✚ process queues , kernel schedulers
- ✚ CFS and multi-core scheduler.
- ✚ CPU affinity and scheduling domains.
- ✚ Load balancing.
- ✚ System calls for processes.
- ✚ Signals and IPC
- ✚ signal generation and delivery of
- ✚ signals. Pipes, Shared memory and
- ✚ semaphores. sleep and wakeup.

### 3. Memory Management Topics

- ✚ Segmentation, Paging and Virtual memory
- ✚ High memory and bounce buffers. Logical,
- ✚ Linear and physical addressing
- ✚ Physical Page extension, Physical Memory Layout and virtual pages, Page tables and
- ✚ TLBs. Hardware caching and Multi-processor caching
- ✚ Cache hierarchies, Cache coherence and MESI
- ✚ protocol. Page replacement policies
- ✚ Page frame management, memory zones and Temporary Kernal
- ✚ mappings. Page allocations, slab allocators and object caches
- ✚ kmalloc implementation.
- ✚ Swapping concepts.
- ✚ Page frame reclamation algorithms
- ✚ The out of memory killer
- ✚ Linux page cache
- ✚ address\_space object
- ✚ Buffer cache versus page
- ✚ cache syncing dirty buffers.
- ✚ Swap cache.
- ✚ Page cleaners and system calls for
- ✚ caching, Process address space and VMAs
- ✚ demand paging and page fault
- ✚ handler Copy on write.

## 4. Kernel Synchronization

- Kernel pre-emption and control paths.
- Pre-emptive kernels.
- Concurrency and making the code re-entrant.
- Synchronization primitives
  - per-CPU variables, atomic operations. Memory barriers
  - spin locks
  - seqlocks.
  - RCU
  - Semaphore variants,
  - Synchronization across interrupts.

## 5. System calls

- Covers most common systems calls that interacts with various kernel modules.

## 6. Timing controls.

- Clocks and timer interrupts
- Jiffies and software timers
- Timing wheel idea.
- Udelay and ndelay()
- system calls for timings.

## 7. Virtual File Systems

- VFS primitives, File systems abstractions

- ✚ dentry caches, super-block, fs-struc and files\_struct , inode and directory operations
- ✚ File system mounting and un-mounting.
- ✚ Local and Network file system concepts.
- ✚ Linux Block layer - BIOS- Overview of EXT4.

## 8. Playing with kernel

- ✚ obtaining kernel source
- ✚ installation of the kernel
- ✚ kernel source tree and code walk thru.
- ✚ Configuring and Building the kernel
- ✚ GNU C extentions for kernel coding
- ✚ inline assembly.

## 9. Lab session:

- ✚ Building a kernel extension /module in a chosen area.

## Contact Us:

**NetDiox R&D Team**

**Netdiox Computing System Pvt. Ltd.**

**Address:** #1514, First Floor, 19th Main Road,  
HSR Layout Sector-1, Bangalore 102,  
Karnataka, India

**Contact No:** 08289884406, 09946711160

E-mail: [info.netdiox@gmail.com](mailto:info.netdiox@gmail.com)