Toolchain Basics

Address Spaces:
* Physical Address Space
  - PC System Architecture
* Virtual Address Space
  - Association of Application code/data to virtual address space

Object File Format
* ELF
  - Understanding relevant sections of ELF from debugging stand point.

GNU Tools
* Compiler, Linker, objcopy, objcopy, strip, nm

Libraries
* Static Library
* Shared Library, DLL

Debugging Tools
* addr2line, readelf, objdump, GNU Debugger, valgrind

Debugging Techniques
* Analysing memory leaks in Data, heap and stack sections
* Effectively using /proc/<pid>/maps

Introduction to Embedded System development environment

Build Utilities
- GNU Make

Introduction to Linux Kernel

Execution Environments
Types of Kernel
User Mode and Kernel Mode
System Call Mechanism
Writing a Kernel Module (insmod, rmmod, modprobe)
Linked Lists in Kernel
Exploring the Linux Kernel Source tree
Configuring and building Linux

Writing a kernel module

Memory Management

* Virtual Address Management
    * Address Mapping
        - Physical Memory to User Virtual Address
            ~ Hardware Support
                - MMU
                - Paging Hardware
            ~ Kernel Support
                - Virtual Address Region Descriptors
                    (vm_area_struct)
                - Page Table
    * mmap()

        - Program Execution Sequence

Task Management

* Understanding of Process, Thread, Task from kernel perspective
* Task's State Space
* schedule()
* Scheduling Algorithm for Realtime tasks
    ~ FIFO
    ~ Round Robin
* Scheduling Algorithm for non-real time tasks
    ~ CFS
* Signals

Virtual File System:
inode, dentry, file, path, mounting

File System:
Ext2 File System

Inter Process Communication

Pipes
Message Queues
Semaphores
Shared Memory

---

* Introduction to IPC Mechanisms

* POSIX Message Queues
    - Introduction
    - Creating a message queue
    - Configuring a message queue
    - Sending messages to a message queue
    - Receiving messages to a message queue
    - Closing a message queue
    - Deleting a message queue

* POSIX Shared Memory
    - Introduction
    - Creating a shared memory object
    - Map the shared memory to the address space
      of the calling process
    - Writing and Reading into the shared memory
    - Removing a shared object

* POSIX Semaphores
    - Introduction
    - Creating a semaphore
    - Acquiring a semaphore
    - Releasing a semaphore
    – Removing a semaphore.

Multithreaded Programming

* Need for Asynchronous programming

* Address Space
    - Physical Address Space
    - Virtual Address Space
    - /proc/<pid>/maps

* Context and Content from the Kernel Perspective

* Process as seen by User and Kernel
    - fork(), wait(), exec(), vfork()

* Thread as seen by User and Kernel

* Difference between Process, Thread and Task

* Introduction to POSIX compliant pthread library.

* Creattion of thread using pthread function
    - pthread_create()
    - pthread_join()
    - pthread_detach()
    - pthread_self()
    - pthread_exit()

* Synchronization Mechanisms

    * Mutual Exclusion
        - pthread_mutex_init()
        - pthread_mutex_lock()
        - pthread_mutex_unlock()
        - pthread_mutex_trylock()
        - pthread_mutex_destroy()

    * Condition Variable
        - pthread_cond_init()
        - pthread_cond_signal()
        - pthread_cond_wait()
        - pthread_cond_timedwait()
        - pthread_cond_destroy()

* Real Time Scheduling
    - pthread_attr_init()
    - sched_get_priority_max()
    - sched_get_priority_min()
    - pthread_attr_getschedpolicy()
    - pthread_attr_setschedpolicy()
    - pthread_attr_getschedparam()
    - pthread_attr_setschedparam()
    - pthread_attr_getinheritsched()
    - pthread_attr_setinheritsched()
    - pthread_attr_destroy()

    - pthread_getschedparam()
    - pthread_setschedparam()

* Priority Inversion
* Priority Inheritance
* Priority Ceiling

Networking Fundamentals

Introduction to basic networking infrastructure
- Hub
- LAN
- Switch
- WAN
- Router

Addressing

- MAC address
- IP Address
- Hierarchical addressing
- Classfull IP address classification
- Classless IP Address
- Subnetting
- Private IP Address Vs Public IP Address
- Network Address Translation (NAT)
- Port Address Translation (PAT)
- DHCP, DNS

TCP/IP Stack
- Transportation protocols
- TCP vs. UDP

Packet Journey between two systems.

Socket Programming
- Introduction:
- Why socket programming ?
- Sockets and its related functions
- socket address structure
- Comparision of file operations and socket programming

- Client/Server paradigm:
- TCP Connection establishment and Connnection termination
- TCP State Space
- TCP based socket program
  - Introduction to basic socket calls
    - socket()
    - bind()
    - listen()
    - connect()
    - accept()
    - read()
    - write()
    - select()
    - poll()
    - setsockopt()

Implementation of TCP/UDP client/server example
Implementation of Chat Application
Implementation of RAW socket over IP layer
Implementation of RAW socket over Data Link Layer
Implementation of network stack

Use of tools like tcpdump, wireshark, ss
