BITS Pilani
DIGITAL
Excellence made yours

Computing Systems

Presented by : Prof. Shan Sundar Balasubramaniam
Prof. Biju Raveendran

Unit :
3 Credits

Course Description

This introductory course provides a cross-sectional understanding of Computing Systems from a bottom-up perspective - starting with the hardware components of a stand-alone computer, the software (i.e. Operating Systems and virtual environments) that manage such hardware and enable execution of other software (i.e. programs), leading towards the supporting landscape of systems for a high-level programmer (source code repositories, profilers and debuggers, co-processors for offloading niche computation, mobile/embedded computers) and concluding with large-scale computing systems (distributed systems and Cloud Computing).

BITS Pilani DIGITAL
Excellence made yours

Week 1

Getting Started with Programming

- Introduction to Computer Systems
- Basic Components of a Computer System
- Unicore systems with Stored Program Concept
- Harvard and Modified Harvard Architecture
- Multi-core Systems
- Multi-Processor Systems
- Various Intel Architectures [i3 Vs i5 Vs i7 Vs i9]
- Module Summary

Discussion Prompt | Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

BITS Pilani DIGITAL
Excellence made yours

Week 2

Computer System Architecture - Memory and I/O

- Module Introduction
- Introduction to Locality of Reference
- Spatial Locality
- Temporal locality
- Importance of Locality of Reference in Programming
- Introduction to Primary Memory - RAM and ROM
- SRAM
- DRAM
- Introduction to Secondary Memory & I/O Devices
- Block Devices & Character Devices
- Secondary Memory - Hard Disk
- Secondary Memory - SSD
- SSD as Primary Memory Vs Secondary Memory
- Other I/O Devices
- Module Summary

Discussion Prompt | Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

BITS Pilani DIGITAL
Excellence made yours

Week 3

Introduction to an Operating System

- Module Introduction
- Working with Linux OS
- Commandline Interface
- Taking different datatypes from commandline
- UNIX commands: ls, pwd
- UNIX commands: cd, mkdir, passwd
- UNIX commands: man, info
- UNIX commands: date, cal
- Text Processing Commands: cat, cp
- File Processing Commands: mv, rm, chmod
- Text Processing Commands: IO Redirection
- Text Processing Commands: Filters - wc, sort, head, tail
- Text/File Search Commands: grep, find
- Text Processing Commands: Piping
- File Processing Commands: Hard link and Symbolic link
- Module Summary

Discussion Prompt | Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 4

Shell Scripting

- Module Introduction
- Shell Scripting: Variables and Constants
- Shell Scripting: Conditional statements
- Shell Scripting: Looping constructs
- Shell Scripting: Handling Files
- Shell Scripting: Programming
- Module Summary

Discussion Prompt | Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Graded Quiz

- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 5

Program Execution in a System

- Module Introduction
- Program Vs Process
- Managing Process in a UNIX OS: Creation
- Managing Process in a UNIX OS: Termination
- Managing Process in a UNIX OS: Wait
- Managing Process in a UNIX OS: Signal
- Threads Vs Process
- Mapping between User level and Kernel level threads
- Thread operations: Joinable / Detached, Priority etc.
- Thread Data Sharing
- Thread Pooling
- Zeroed out threads
- Thread Concurrency and Synchronization
- Module Summary

Discussion Prompt | Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Graded Quiz

- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 6

Virtual Environment and Isolation

- Module Introduction
- Abstraction - Definition and Example
- Abstractions in the context of Computer Systems - Examples
- The difference between abstraction and virtualization
- Process and Resources required for a Process
- Process Virtualization - Use cases and Examples
- Hardware Resources and Resource Management - OS as Resource Manager
- Virtualized Hardware and Virtualized Resource Management
- System Virtualization - Use cases and Examples
- High Level Language Virtual Machines as a special case of Process Virtualization
- Python Virtual Environment
- Performance Overhead of Virtual Machines
- Performance Overhead of System VMs
- Minimalistic approach to Isolation and Reducing the Performance Overhead
- Containers
- Module Summary

Discussion Prompt | Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 7

Source Code Repositories and Versioning Systems

- Module Introduction
- Distributed Development Teams and Source Code Usage Scenarios
- Program as a collection of files / modules and Building Executables
- Nature of Requirements - Bags of Requirements and Requirements Change
- Need for Versioning and typical use cases / practices
- Distributed Versioning
- Why code repositories and versioning tools are related and built as one piece?
- Typical Code Repository and Versioning Tool. Example: GitHub and Git (Structure and features)
- Operations on Repositories - Git Commands/Actions
- Module Summary

Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 8

Profiling and Debugging

- Module Introduction
- Compiler vs Interpreter
- Execution of a program and running time (aka run-time)
- Static vs. Dynamic
- Just-in-time and bytecode
- What is Profiling?
- Static vs. Dynamic Profiling - Scenarios, Applications
- Using a profiler (typical profiling attributes and options) - 1
- Using a profiler (typical profiling attributes and options) - 2
- What is Debugging?
- Debugging - Exceptions vs. Errors. Bug Finding vs. Bug Fixing
- Debugging a program (beyond printf) 1 - Stepping, Breakpoints, and Halting
- Debugging a program 2 - Inspecting memory (reading / setting mem. Locations)
- Using a debugger (debugger features / options)
- Module Summary

Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 9

Introduction to GPUs

- Module Introduction
- Why are GPUs useful?
- Basic GPU architecture
- Program Execution in CPU-GPU configuration
- CPU-GPU communication and data transfer
- Threads in GPUs - Introduction
- Shared Memory Multi-Threading
- Multi-Threading in GPUs (Single Instruction Multiple Threads)
- Warps / Wavefront (Nvidia/AMD)
- Thread Blocks and Grids, Indexing
- Understanding GPU (parallel) execution Model in GPU : 2-level hierarchy of threads
- Estimating Performance of Multi-threading in GPUs

Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

BITS Pilani DIGITAL
Excellence made yours

Week 10

Program Execution and Performance Tuning in GPUs

- Module Introduction
- Different levels and types of memory in a GPU memory hierarchy
- Data movement and transfer costs between different levels of memory
- Locality-aware Programs
- Multi-threading and Tiling in GPUs
- A Tiled version of Matrix Multiplication
- Muliple possible tilings and performance impact
- Performance of Basic Neural Network Operations on GPUs - Element-wise operations, Reductions, Dot-Products
- Impact of the performance os NN operations - Examples
- Performance Analysis of NN code - Example 1
- Performance Tuning of NN code - Example 1
- Performance Analysis of NN code - Example 2
- Performance Tuning of NN code - Example 2
- Typical Heuristics for Performance Tuning on GPUs with example code snippets
- Module Summary

Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 11

Introduction to Distributed Systems

- Module Introduction
- Dividing and Distributing Complex Computations (into multiple computers) as opposed to multile cores
- Distributing Data into multiple computers (or into memories / storage units of multiple computers)
- A (Distributed) Program as a collection of processes
- Multi-process vs. Multi-threaded program : Communication or Data Exchange instead of shared memory
- Client-Server Systems
- Remote Procedure Calls (RPC) as a speical case of Client Server
- gRPC Case Study
- The Request-Response Model
- The role of a server - scalability
- Typical Performance Attributes of a Server: Throughput and Response Time
- Server and Service
- Typical Perfrormance Attributes of a Service: Reliability and Availability
- Homogeneous vs. Heterogeneous nodes in a distributed system
- Distributed Cluster
- Compute Clusters and Scale-out Clusters - Economies of Scale
- Homogeneous Clusters and the Data Distribution Problem
- Partitioning vs. Replication as Strategies/Solutions
- Partioninng vs Replication: Scenarios, Advantages, and Overheads
- Partitioning and Replication: Case Study: Backend of a Search Engine

Reading | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Experiential Learning

Objective

Description

BITS Pilani DIGITAL
Excellence made yours

Week 12

Introduction to Cloud Computing

- Module Introduction
- The technical need for Cloud Computing
- The business need for Cloud Computing
- Virtualization and Virtual Machines
- Resource Provisioning and Multi-Tenancy
- Resource Management - Scalability, Server Consolidation,
- Elasticity, Measures of Elasticity
- Virtualization Overhead
- Isolation vs. General Purpose Abstraction
- Alternative to Virtual Machines - Containers: Isolation and Fine-Grained Abstraction
- VMs vs. Containers: Performance, Use Case Scenarios
- Module Summary

Discussion Prompt | Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

BITS Pilani DIGITAL
Excellence made yours

Week 13

Introduction to Mobile/Embedded Computing

- Module Introduction
- Device Attributes and Constraints: Communication & Bandwidth, Small Form Factor, Space & Chipset constraint, Battery/Power Constraints, Memory Size Limitation, and Computing Constraints
- Tyoical Components and Features of Mobile phone as a computing device
- Typical Mobile Phone hardware (Chipset Architecture)
- Mobile OS (Process, Programs, Memory Layout, I/O and Communication - Android as an example
- A typical web-and-mobile application: mobile pobone as a client, communication with the web-server; web front-end on mobile
- A typical mobile-and-cloud application with mobile phone as a client, communication with the cloud, mobile front-end.

Practice Quiz

Live Session

- Date: <To be filled>
- <To be filled>

Experiential Learning

Objective

Description