Lab 4: Setup a LAN that can communicate among the virtual devices inside the packet tracer
Lab 5: Extend the LAN that is set up in Lab 4 for so that it works in the real world. Setup aphysical LAN network using UTP cables, RJ45, Crimpers, switch/hub and machines connectedusing this setup should communicate with each other.
Lab 6: Explore possibility of setting up fiber optic connection physically. Identify variouscomponents required for setting up a fiber optic connection.

Unit 3: Data link layer
Lab 7: Using packet tracer/wireshark identify the data link layer frame structure
Lab 8: Perform some lab work that demonstrates MAC, ARP etc.

Unit 4: Network Layer
Lab 9: Create a network and multiple subnetworks in the packet tracer and make them able tocommunicate with each other.
Lab 10: Lab that demonstrate routing in the packet tracer
Lab 11: Configure routing with various protocols like RIP, BGP, EGP etc

Unit 5: Transport Layer
Lab 12: Write a C/C++/Java program to demonstrate socket programming
Lab 13: Write a program to demonstrate client/server communication protocol

Unit 6: Application
Lab 14: Configure an SMTP/IMAP/POP to send/receive email, DHCP server to allocate IPaddresses, HTTP server to serve html documents, ftp to access files, ssh to access remote server.

References:
Software: CISCO Packet tracer, Boson NetSim
OS: Linux/Windows having specialised software installed for the specific purpose.
Application Softwares: DHCP Server, FTP Server: filezilla server, openftp, opensmtpd, HTTP-Apache, nginx, SSH-OpenSSH, termius,sshd,putty

Operating System Lab (UCSE572)

L-T-P: 0-0-3

Credits: 3

List of Programs:
1. Simple Unix-C (at least two) programs using system calls to read and write strings onstandard I/O devices and files.
2. Implementation of starting a new process, replacing a process image, duplicating aprocess image, waiting for a process, zombie process.
3. Implementation of the Dining Philosopher problem using shared memory and semaphore.

4. Implementation of a bounded-buffer problem using shared memory and semaphore.
5. Implementation of FCFS process scheduling techniques.
6. Implementation Shortest Job First (both preemptive and non-preemptive version)process scheduling techniques.
7. Implementation Round Robin process scheduling techniques.
8. Implementation for simulating page replacement algorithms like FIFO, Optimal andLRU.
9. Implementation of threads using POSIX or using thread class in Java.
10. Implementation of (at least one) deadlock avoidance techniques.

Text Books:

1. Stevens, “UNIX programming”, Pearson Education, Pearson Education, 2004.

Hardware Lab (UCSE573)

L-T-P: 0-1-3

Credits: 5

HDL: Verilog/VHDL

List of experiments:

1. Realization of basic digital circuits: Half adder, Full Adder, Ripple Carry Adder, Adder/Subtractor, Multiplexer/Demultiplexer.
2. Complex Arithmetic Units:Carry Lookahead Adder, Unsigned Multiplication, Signed Multiplication, Systolic Array Multiplication, Division
3. Realization of Logic Units: 16 bits greater than, 16 bits less than, 16 bit equals to
4. Development of a 16-bit ALU

Books:

The Verilog® Hardware Description Language 5th Edition by Donald E. Thomas , Philip R. Moorby