University of Mosul / College of Engineering / Department of Computer Engineering

Level 3 Laboratories

Description of the Computer Networks Laboratory

1. General Information:	
Laboratory Name and Lab. Number:	Computer Networks Laboratory (Lab. 312)
Associated Course Name:	Computer Networks
Department:	Computer Engineering
Weekly Lab Hours:	3 hours
Number of Weeks in the Semester:	15 weeks
Academic Level:	Third Year
Lab Supervisor:	Prof. Dr. Salah Abdulghani

2. General Description of the Laboratory:

The Computer Networks Laboratory aims to reinforce the theoretical concepts covered in the Computer Networks course through practical applications and simulations using specialized tools and software for network design and analysis. The lab provides an interactive environment that enables students to gain skills in setting up and configuring both wired and wireless networks, as well as implementing routing protocols and securing networks.

3. Laboratory Objectives:

- Train students to design and set up basic computer networks.
- Familiarize students with network components including hardware, equipment, and software.
- Apply various network protocols such as IP, TCP, UDP, ICMP, DNS, and DHCP.
- Configure routers and switches using simulation environments.
- Analyze network traffic using tools like Wireshark.
- Enhance troubleshooting skills for network issues.

4. Learning Outcomes:

By the end of the laboratory, the student should be able to:

- Design a local area network (LAN) using a network simulator.
- Configure IP addresses and set up routers and switches.
- Analyze data packets and their contents using monitoring tools.
- Document experiments and prepare professional lab reports.
- Collaborate effectively within a team to solve network-related issues.

5. Weekly Experiment Schedule:

Week	Experiment Title	Tools / Software Used	Main Objective
1	Introduction to Computer Networks	Packet Tracer	Understanding network concepts and devices
2	LAN Setup (Local Area Network)	Packet Tracer / Physical Equipment	Creating a local area network and connecting computers
3	IP Address Configuration and Subnetting	Packet Tracer	Assigning IP addresses and applying subnetting
4	Connectivity Testing Using Ping Commands	Command Line + Packet Tracer	Verifying network connectivity
5	Router Configuration (Routing)	Packet Tracer / Physical Router	Configuring a router using routing protocols
6	Routing with RIP (Routing Information Protocol)	Packet Tracer	Setting up and enabling the RIP protocol
7	Routing with OSPF (Open Shortest Path First)	Packet Tracer	Setting up and enabling the OSPF protocol
8	DHCP Service Configuration	Packet Tracer	Configuring DHCP for automatic IP address distribution
9	DNS Service Configuration	Packet Tracer	Configuring a DNS server and mapping names to addresses

10	Data Traffic Analysis Using Wireshark	Wireshark	Monitoring and understanding the structure of network packets
11	Wireless Network Setup	Packet Tracer / Access Point	Designing a Wi-Fi network
12	Network Security – Basic Firewall Configuration	Packet Tracer	Controlling access and securing the network
13	Troubleshooting	Network Diagnostic Tools	Diagnosing and troubleshooting network connectivity issues
14	Comprehensive Review and Practical Project Discussion	All Tools	Integrating acquired skills into a final project

6. Tools and Equipment Used:

- Routers and switches (if available).
- Ethernet cables.
- Laboratory computers.
- Cisco Packet Tracer software.
- Wireshark software.

7. Safety Guidelines:

- Ensure that the power is turned off when connecting devices.
- Do not touch electrical outlets or network components without the supervisor's permission.
- Maintain silence and organize cables to prevent accidents.
- Use the simulation software for routing experiments before working on real devices.

8. Evaluation Method:

Evaluation Item	Percentage
Attendance and participation	10%
Weekly lab reports	30%
Quizzes	20%

Final practical exam	30%
Practical project	10%

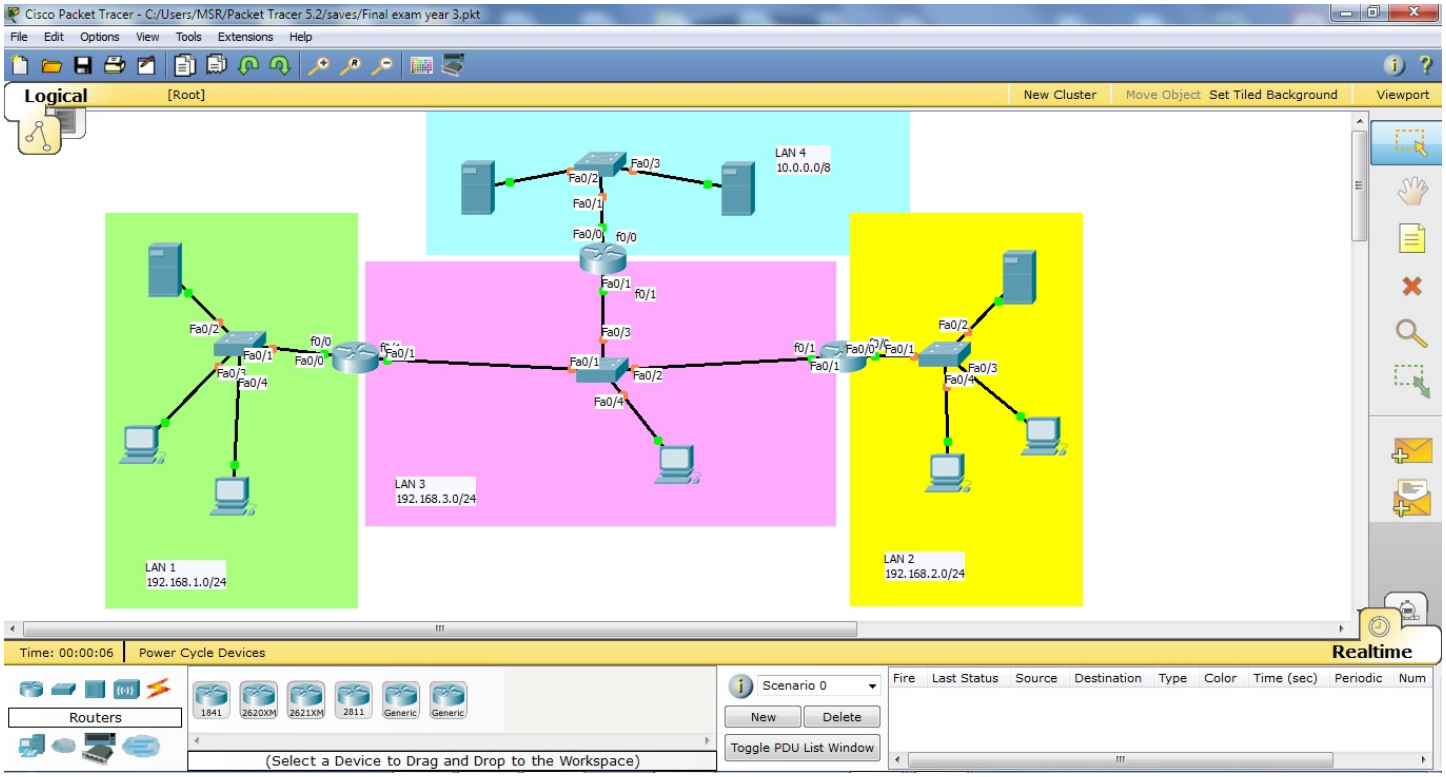
9. References and Sources:

- Cisco Networking Academy Materials
- *Computer Networking: A Top-Down Approach* by Kurose & Ross
- Official Packet Tracer User Guides
- Educational websites such as networklessons.com

10. Attachments:

- Experiment Report Template
- Weekly Work Plan
- Packet Tracer Usage Instructions
- Router and Switch Configuration Guide


USTOUCH




```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\MSR>ipconfig

Windows IP Configuration


Wireless LAN adapter Wireless Network Connection 2:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Wireless LAN adapter Wireless Network Connection:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::ec66:c94c:ba1c:3218%13
   IPv4 Address. . . . . . . . . . . : 192.168.0.104
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 192.168.0.1

Ethernet adapter Bluetooth Network Connection:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{2C1FDE0E-80C2-4FEC-967F-A7F23BD54D01}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{4883F51D-7C13-44B8-8780-5779C86E4A73}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

Tunnel adapter isatap.{085E7CA8-EEE2-41C9-9BA4-1E746DBABA4F}:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . :

C:\Users\MSR>ping www.google.com

Pinging www.google.com [142.250.75.100] with 32 bytes of data:
Reply from 142.250.75.100: bytes=32 time=110ms TTL=111
Reply from 142.250.75.100: bytes=32 time=108ms TTL=111
Reply from 142.250.75.100: bytes=32 time=103ms TTL=111
Reply from 142.250.75.100: bytes=32 time=102ms TTL=111

Ping statistics for 142.250.75.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 102ms, Maximum = 110ms, Average = 105ms

C:\Users\MSR>
```

Description of the Computer Networks 2 Laboratory

1. General Information:	
Laboratory Name:	Computer Networks 2 Laboratory (Lab. 312)
Associated Course Name:	Computer Networks 2
Department:	Computer Engineering
Weekly Lab Hours:	3 hours
Number of Weeks in the Semester:	15 weeks
Academic Level:	Third Year
Lab Supervisor:	Prof. Dr. Salah Abdulghani

2. General Description of the Laboratory:

The Computer Networks 2 Laboratory aims to reinforce the theoretical concepts covered in the Computer Networks course lectures through practical applications and simulations using specialized tools and software for network design and analysis. The lab provides an interactive environment that enables students to gain skills in configuring and setting up wired and wireless networks, implementing routing protocols, securing networks, configuring servers, and identifying and troubleshooting network issues.

3. Laboratory Objectives:

- Expanding students' skills in designing advanced networks, including Local Area Networks (LAN) and Wide Area Networks (WAN), using the OSI and TCP/IP layer models.
- Implementing advanced configurations for routers and switches using the Command Line Interface (CLI) in realistic simulation environments.
- Applying dynamic routing concepts using protocols such as RIP and OSPF.
- Configuring advanced network services such as NAT, VLANs, and ACLs.
- Conducting advanced traffic analysis and monitoring using professional tools like Cisco Packet Tracer and Wireshark.
- Troubleshooting and diagnosing network faults using systematic methodologies such as troubleshooting models and loopback testing.

- Understanding the fundamentals of network security and applying protection mechanisms such as packet filtering and identity verification.
- Preparing students for professional certification concepts such as CCNA through practical scenarios that simulate real-world environments.

4. Learning Outcomes:

By the end of the lab, the student should be able to:

- Design and implement LAN and WAN networks using dynamic routing protocols.
- Perform advanced configuration of routers and switches (VLAN, ACL, NAT).
- Analyze network traffic and diagnose faults using tools such as Wireshark.
- Apply basic network security concepts in a simulated environment.
- Prepare professional reports documenting configurations and troubleshooting solutions.
- Collaborate in teams on network projects that simulate real-world scenarios.

5. Weekly Experiment Schedule:

Week	Experiment Title	Tools / Software Used	Main Objective
1	Subnetting	Packet Tracer	Calculate and divide network addresses
2	VLSM (Variable Length Subnet Masking)	Packet Tracer	Efficient distribution of IP addresses
3	IPv6 Addressing	Packet Tracer / Physical equipment	Configure and use modern IPv6 addressing
4	ICMP (Internet Control Message Protocol)	Command Line + Packet Tracer	Test and troubleshoot network issues
5	ARP (Address Resolution Protocol)	Command Line + Packet Tracer	Map IP to physical (MAC) address
6	VLAN (Virtual LAN)	Packet Tracer / Physical switch	Separate networks within switches
7	Inter-VLAN Routing	Packet Tracer	Connect VLANs using routing

8	Router-on-a-Stick	Packet Tracer / Physical switch + router	Connect VLANs via a single interface
9	NAT (Network Address Translation)	Packet Tracer	Share internet through address translation
10	ACLs (Access Control Lists)	Packet Tracer / Physical router	Restrict access using filtering rules
11	Traffic Analysis with Wireshark	Wireshark	Monitor and understand packet structure
12	STP (Spanning Tree Protocol)	Packet Tracer	Configure redundancy and prevent network loops
13	Troubleshooting	Network testing tools	Diagnose and resolve connectivity issues
14	Final Project & Review	All tools	Integrate skills in a final practical project

6.

7. Tools and Equipment Used:

- Routers and switches (if available)
- Ethernet cables
- Lab computers
- Cisco Packet Tracer software
- Wireshark software

8. Safety Guidelines:

- Ensure the power is turned off when connecting devices.
- Do not touch electrical outlets or network components without supervisor permission.
- Maintain silence and organize cables to prevent accidents.
- Use simulation software for routing experiments before working on real devices.

9. Evaluation Method:	
Evaluation Item	**Percentage**
Attendance and participation	10%
Weekly lab reports	30%
Quizzes	20%
Final practical exam	30%
Practical project	10%

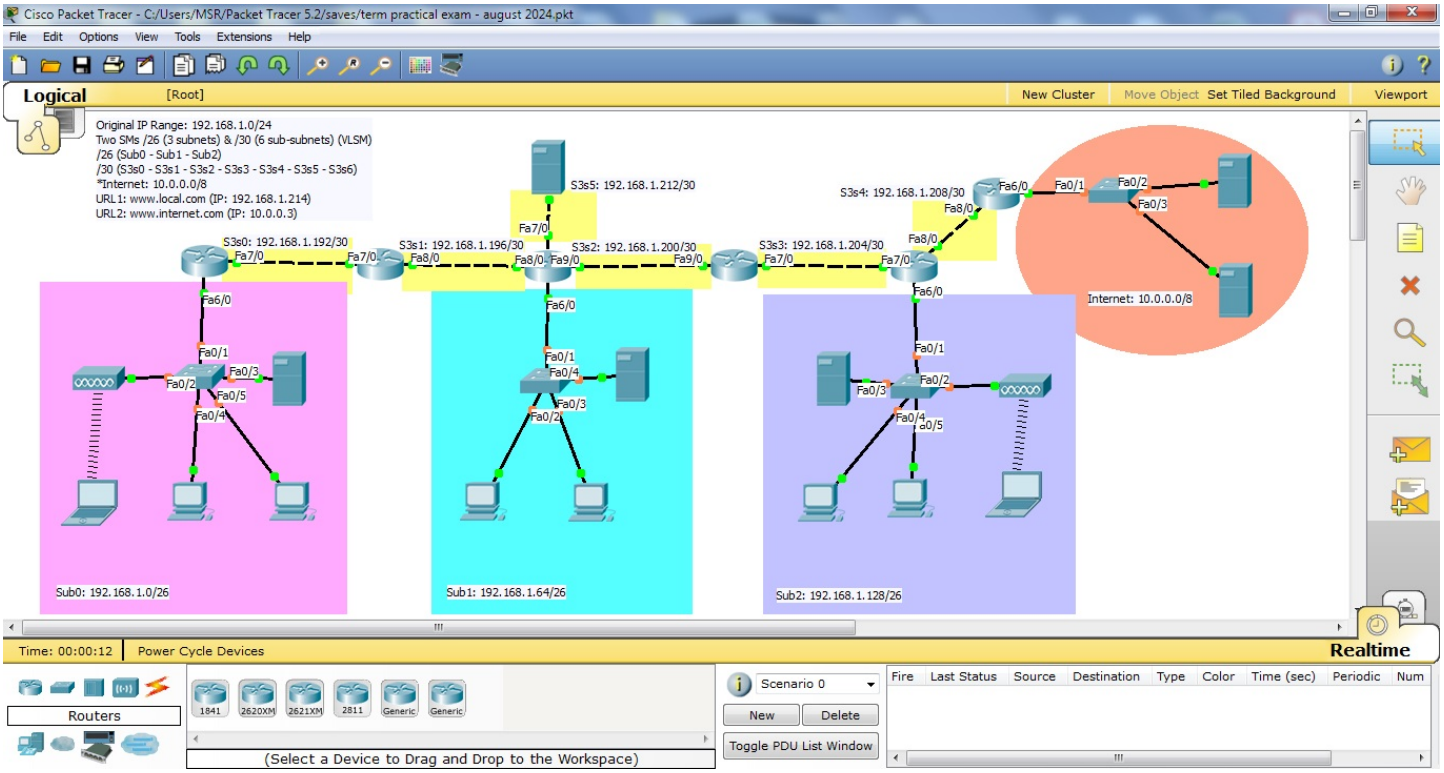
10.References and Sources:

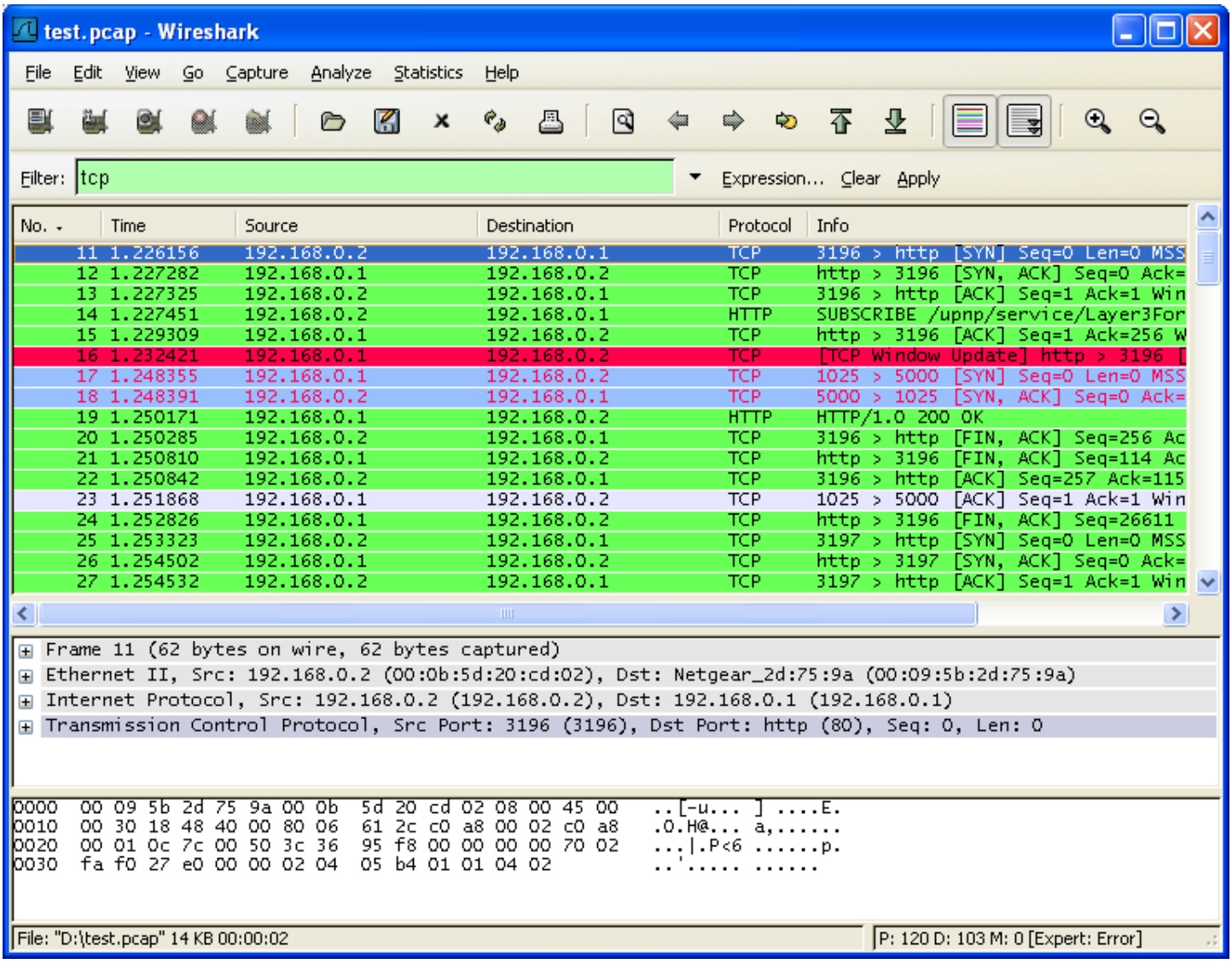
- Cisco Networking Academy Materials
- *Computer Networking: A Top-Down Approach* by Kurose & Ross
- Official Packet Tracer User Guides
- Educational websites such as networklessons.com

11. Attachments:

- Experiment Report Template
- Weekly Work Plan
- Packet Tracer Usage Instructions
- Router and Switch Configuration Guide


USTOUCH


Cisco Packet Tracer - C:/Users/MSR/Packet Tracer 5.2/saves/term practical exam - august 2024.pkt
File Edit Options View Tools Extensions Help
Logical [Root]
New Cluster Move Object Set Tiled Background Viewport
Original IP Range: 192.168.1.0/24
Two SMs /26 (3 subnets) & /30 (6 sub-subnets) (VLSM)
/26 (Sub0 - Sub1 - Sub2)
/30 (S3s0 - S3s1 - S3s2 - S3s3 - S3s4 - S3s5 - S3s6)
*Internet: 10.0.0.0/8
URL1: www.local.com (IP: 192.168.1.214)
URL2: www.internet.com (IP: 10.0.0.3)
S3s5: 192.168.1.212/30
S3s0: 192.168.1.192/30
S3s1: 192.168.1.196/30
S3s2: 192.168.1.200/30
S3s3: 192.168.1.204/30
S3s4: 192.168.1.208/30
Internet: 10.0.0.0/8
Sub0: 192.168.1.0/26
Sub1: 192.168.1.64/26
Sub2: 192.168.1.128/26
Time: 00:00:12 Power Cycle Devices
Realtime
Routers
(Select a Device to Drag and Drop to the Workspace)
Scenario 0
New Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time (sec) Periodic Num


test.pcap - Wireshark
File Edit View Go Capture Analyze Statistics Help
Filter: tcp
Expression... Clear Apply
No. Time Source Destination Protocol Info
11 1.226156 192.168.0.2 192.168.0.1 TCP 3196 > http [SYN] Seq=0 Len=0 MSS
12 1.227282 192.168.0.1 192.168.0.2 TCP http > 3196 [SYN, ACK] Seq=0 Ack=
13 1.227325 192.168.0.2 192.168.0.1 TCP 3196 > http [ACK] Seq=1 Ack=1 Win
14 1.227451 192.168.0.2 192.168.0.1 HTTP SUBSCRIBE /upnp/service/Layer3For
15 1.229309 192.168.0.1 192.168.0.2 TCP http > 3196 [ACK] Seq=1 Ack=256 W
16 1.232421 192.168.0.1 192.168.0.2 TCP [TCP Window Update] http > 3196 [
17 1.248355 192.168.0.1 192.168.0.2 TCP 1025 > 5000 [SYN] Seq=0 Len=0 MSS
18 1.248391 192.168.0.2 192.168.0.1 TCP 5000 > 1025 [SYN, ACK] Seq=0 Ack=
19 1.250171 192.168.0.1 192.168.0.2 HTTP HTTP/1.0 200 OK
20 1.250285 192.168.0.2 192.168.0.1 TCP 3196 > http [FIN, ACK] Seq=256 Ac
21 1.250810 192.168.0.1 192.168.0.2 TCP http > 3196 [FIN, ACK] Seq=114 Ac
22 1.250842 192.168.0.2 192.168.0.1 TCP 3196 > http [ACK] Seq=257 Ack=115
23 1.251868 192.168.0.1 192.168.0.2 TCP 1025 > 5000 [ACK] Seq=1 Ack=1 Win
24 1.252826 192.168.0.1 192.168.0.2 TCP http > 3196 [FIN, ACK] Seq=26611
25 1.253323 192.168.0.2 192.168.0.1 TCP 3197 > http [SYN] Seq=0 Len=0 MSS
26 1.254502 192.168.0.1 192.168.0.2 TCP http > 3197 [SYN, ACK] Seq=0 Ack=
27 1.254532 192.168.0.2 192.168.0.1 TCP 3197 > http [ACK] Seq=1 Ack=1 Win
Frame 11 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: 192.168.0.2 (00:0b:5d:20:cd:02), Dst: Netgear_2d:75:9a (00:09:5b:2d:75:9a)
Internet Protocol, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)
Transmission Control Protocol, Src Port: 3196 (3196), Dst Port: http (80), Seq: 0, Len: 0
0000 00 09 5b 2d 75 9a 00 0b 5d 20 cd 02 08 00 45 00 ..[-u...] ....E.
0010 00 30 18 48 40 00 80 06 61 2c c0 a8 00 02 c0 a8 .0.H@... a,......
0020 00 01 0c 7c 00 50 3c 36 95 f8 00 00 00 00 70 02 ...|.P<6 ......p.
0030 fa f0 27 e0 00 00 02 04 05 b4 01 01 04 02 ..'..... ......
File: "D:\test.pcap" 14 KB 00:00:02
P: 120 D: 103 M: 0 [Expert: Error]

Description of the Computer Interfacing Laboratory

1. General Information:	
Laboratory Name:	Computer Interfacing Lab.
Associated Course Name:	Computer Interfacing
Department:	Computer Engineering
Weekly Lab Hours:	4
Number of Weeks in the Semester:	15
Academic Level:	Third class
Lab Supervisor:	Dr. Ina'am Fathi

2. General Description of the Laboratory:

The Computer Interfacing Lab aims to reinforce the theoretical concepts covered in the Computer Interfacing course lectures through practical applications using the MTS-86 microprocessor training kit and tools and software dedicated to writing, testing, and executing programs. The lab provides an interactive environment that enables students to acquire the skills to write, design, and implement programs practically, and upload them for practical implementation on the MTS-86 microprocessor training kit, using auxiliary software, tools, and laboratory equipment.

3. Laboratory Objectives:

- Learn about the MTS-86 microprocessor training kit and how to program the operation of the supporting circuits and electronic modules connected to the 8086 microprocessor.
- Learn how to write and test programs in a programming environment.
- Learn how to download and drop program executable files onto the 8080 microprocessor.
- Train students to write programs to program and operate the auxiliary and supporting electronic modules connected to the 8086 microprocessor.
- Train students on how to use the necessary laboratory auxiliary devices to perform inspections, display, read, and measure important signals.
- Enhance their ability to troubleshoot and resolve errors practically.

- Enhance their understanding and comprehension of the theoretical material through practical implementation and monitoring.

4. Learning Outcomes:

By the end of the lab, the student should be able to:

● Write and test programs for programming and configuring the electronic modules supporting the 8086 microprocessor.

● Learn how to make changes and modifications to the program sections programmed for the microprocessor and its associated supporting modules.

● Learn how to download and drop executable files onto the microprocessor and its associated supporting modules.

● Programs for programming and configuring the electronic modules supporting the 8086 microprocessor.

● Study, verify, and analyze results and signals.

● Document experiments and prepare professional reports.

● Work within a team to solve programming and practical problems.

5. Weekly Experiment Schedule:

Week	Experiment Title	Tools / Software Used	Main Objective
1	Learn about the MTS-86 microprocessor training kit	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	Identify the parts and components of the MTS-86 kit.
2	The7-Segment Display and 74LS373	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	How to connect, program and operate the 7-Seg. display and 74LS373 adapter with the processor
3	8255Programmable Peripheral interface Mode0	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	How to connect, program, and operate the 8255 support module and identify the working mode 0
4	8255Programmable Peripheral interface	DosBox,HyperTerminal, MASM, Linker , Notpad	How to connect, program, and operate the 8255 support

	Mode1	TextEditor	module and identify the working mode 1
5	8253 Programmable Interval Timer (PIT) Modes of Operation	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor Oscilloscope	How to connect, program, and operate the programmable interval timer (PIT-8253)
6	Sound Generator using 8253PIT	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor, Oscilloscope	How to generate sound waves using 8253PIT
7	The Programmable Keyboard / Display8279	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	How to connect, program, and operate the 8279 programmable keyboard/display
8	Programmable Keyboard/Display8279 Modes of operations	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	Learn about programmable keyboard/display operation modes 8279
9	Digital to Analog Convertor DAC 0808	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor Oscilloscope	How to connect, program, and operate an 0808 DAC
10	Analog to Digital Convertor 0809	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	How to connect, program, and operate an 0809 analog-to-digital converter
11	Digital Sound Recorder	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	How to design, program, and implement a digital voice recorder
12	USART 8251 Universal Synchronous Asynchronous Receiver Transmitter USART (Asynchronous Mode(	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor Oscilloscope	How to connect, program, and operate a USART universal synchronous and asynchronous transceiver module (Asynchronous mode)
13	USART 8251 Universal Synchronous	DosBox,HyperTerminal, MASM, Linker , Notpad	How to connect, program, and operate a USART universal

	Asynchronous Receiver Transmitter USART (Synchronous Mode(	TextEditor Oscilloscope	synchronous and asynchronous transceiver module (Synchronous mode)
14	8259The Programmable Interrupt Controller PIC	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	How to connect, program, and operate the PIC-8259 programmable interrupt controller
15	Modes of operation of 8259The Programmable Interrupt Controller PIC	DosBox,HyperTerminal, MASM, Linker , Notpad TextEditor	Understanding the PIC-8259 Programmable Interrupt Controller Operating Modes

6. Tools and Equipment Used:

- Laboratory computers
- Computer, microprocessor training kit, MTS-86 digital signal display, auxiliary software
- Connection cables
- DOSBox, HyperTerminal, Notepad Text Editor

7. Safety Guidelines:

- Ensure the power is off when connecting devices.
- Do not touch electrical outlets or electrical appliances without the permission of your supervisor.
- Remain calm and organize cables to avoid accidents.

8. Evaluation Method:

Evaluation Item	Percentage
Attendance and Participation	10%
Weekly Experiment Reports	30%
Quizzes	20%
Final Practical Exam	40%

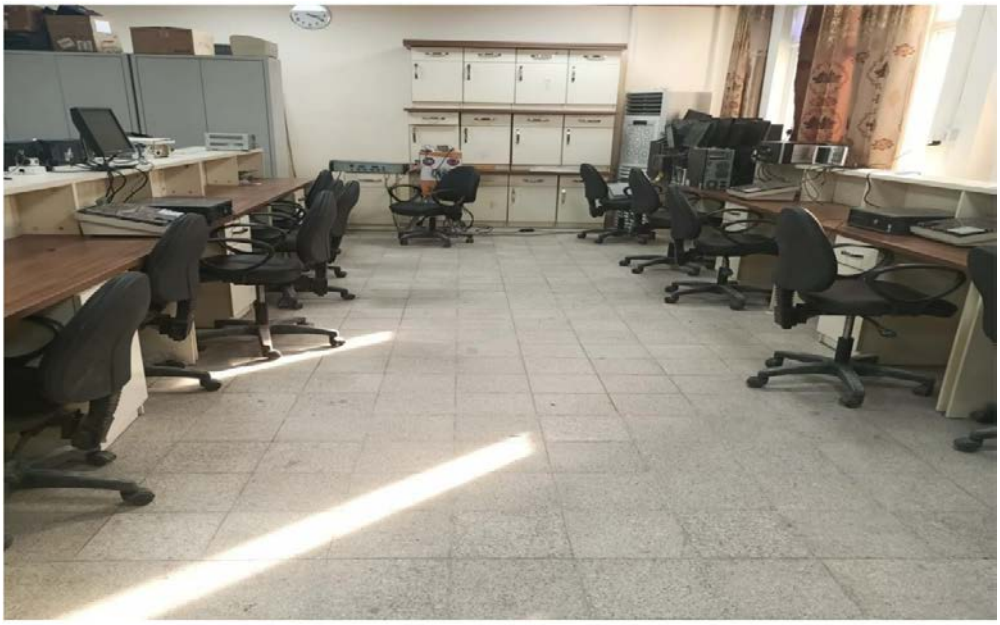
9. References and Sources:

- Intel 80x86 and other chips hardware reference manuals, Intel.
- Data Sheets (8255, 8253,8254,DAC808-ADC809,8251,1650,8237,8259, 8279) by Intel.

10. Attachments:

- Experimental Report Form.
- Weekly Work Plan.
- Guide to Using and Programming the Electronic Circuits and Modules Supporting the Microprocessor.

- **Laboratory Pictures:**



Laboratory.

MTS-86 MTS Kit.

Desktop Computer.

Oscilloscope.

Description of the Embedded Systems Laboratory

1. General Information:	
Laboratory Name:	Embedded Systems Laboratory
Associated Course Name:	Embedded Systems
Department:	Computer Engineering
Weekly Lab Hours:	4 hours
Number of Weeks in the Semester:	15weeks
Academic Level:	Third Level
Lab Supervisor:	Dr. Ina'am Fathi

2. General Description of the Laboratory:

The Embedded Systems Lab aims to reinforce the theoretical concepts covered in the Embedded Systems course lectures through practical applications using the Arduino educational kit and dedicated tools and software for writing, testing, and executing programs. The lab provides an interactive environment that enables students to acquire the skills to write, design, and implement programs practically, and upload them for practical implementation on the Arduino microcontroller kit using auxiliary software, tools, and laboratory equipment.

3. Laboratory Objectives:

- Learn about the Arduino ATmega2560 microcontroller training kit and how to program the operation of the supporting electronic circuits and modules connected to the ATmega2560 microcontroller.
- Learn how to write and test programs in a programming environment.
- Learn how to download and drop program executable files onto the 2560 ATmega microcontroller.
- The microcontroller in the Arduino educational kit.
- Training students to write programs and operate the electronic modules embedded and connected internally to the 2560 ATmega microcontroller board.
- Train students on how to use the necessary laboratory auxiliary devices to perform inspections, display, read, and measure important signals.

- Enhance their ability to troubleshoot and resolve problems practically.
- Enhance their understanding and comprehension of theoretical material through practical implementation and monitoring.

4. Learning Outcomes:

By the end of the lab, the student should be able to:

● Write and test programs for programming embedded modules connected internally to the microcontroller.

● Learn how to make changes and modifications to the programming sections of the microcontroller and its supporting and associated modules.

● Learn how to download and drop executable files onto the microcontroller and its associated supporting electronic modules.

● Understand and work with programs for programming and configuring the microcontroller's supporting electronic modules.

● Study, verify, and analyze results and signals.

● Document experiments and prepare professional reports.

● Work within a team to solve programming and practical problems.

5. Weekly Experiment Schedule:

Week	Experiment Title	Tools / Software Used	Main Objective
1	Introduction To ArduinoATmega2560 microcontroller training kit	Proteus 8 Professional, Studio Atmel7.0	Identify the components of ArduinoATmega2560 microcontroller training kit
2	Microcontroller Chip Install and configure lab software (Programming Arduino)	Proteus 8 Professional, Studio Atmel7.0	Install and identify the required software programs
3	Basic I/O programming	Proteus 8 Professional, Studio Atmel7.0	How to use and programming Arduino
4	Traffic light system using Arduino	Proteus 8 Professional,	How to use Arduino to

		Studio Atmel7.0	control the TLS
5	I/O LEDs and Buzzer with Arduino	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino to control the LEDs and generating Alarm
6	Using Analogue to digital convertor in Arduino	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino with ADC
7	Servo motor with Arduino	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino to control the servo motors
8	Stepper motor with Arduino	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino to control the stepper motors
9	Arduino with 7_ Segments	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino with 7Segment display
10	Up And Down Counter On The 7-Segment	Proteus 8 Professional, Studio Atmel7.0	Using the counters of Arduino
11	Applications of counter\timer	Proteus 8 Professional, Studio Atmel7.0	Implementing timer and counter applications for Arduino
12	Arduino with Keypad	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino with keypad
13	Using DotMatrix with Arduino	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino with DotMatrix
14	Arduino wih LCD	Proteus 8 Professional, Studio Atmel7.0	How to use Arduino with DotMatrix
15	Term exam	Proteus 8 Professional, Studio Atmel7.0	Term exam

6. Tools and Equipment Used:

- Computer and required software programs(Proteus 8 Professional, Studio Atmel7.0).
- ArduinoATmega2560 Board .

- ArduinoATmega2560 Board microcontroller training kit.
- Oscilloscope and wires

7. Safety Guidelines:

- Ensure the power is off when connecting devices.
- Do not touch electrical outlets or electrical appliances without the permission of your supervisor.
- Remain calm and organize cables to avoid accidents.

8. Evaluation Method:

Evaluation Item	Percentage
Attendance and Participation	10%
Weekly Experiment Reports	30%
Quizzes	20%
Final Practical Exam	40%

9. References and Sources:

- The ATmega640/1280/2560/V Microcontroller Data sheet.
- Embedded system Design: Embedded systems Foundations of Cyber-Physical Systems, Peter Marwedel, Spriner Nov. 16, 2010.

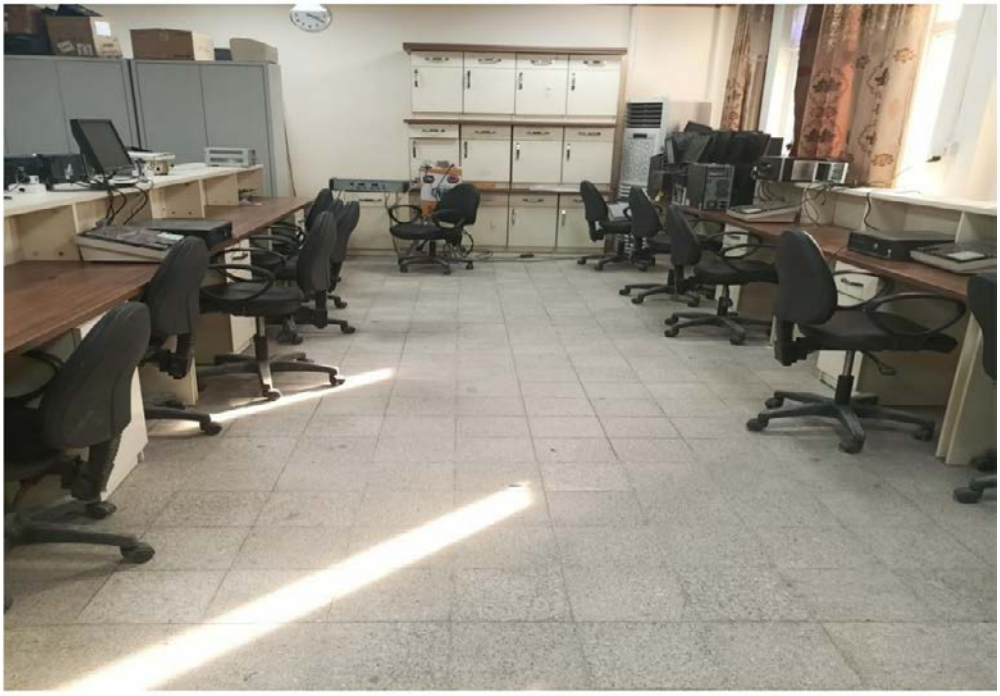
10. Attachments:

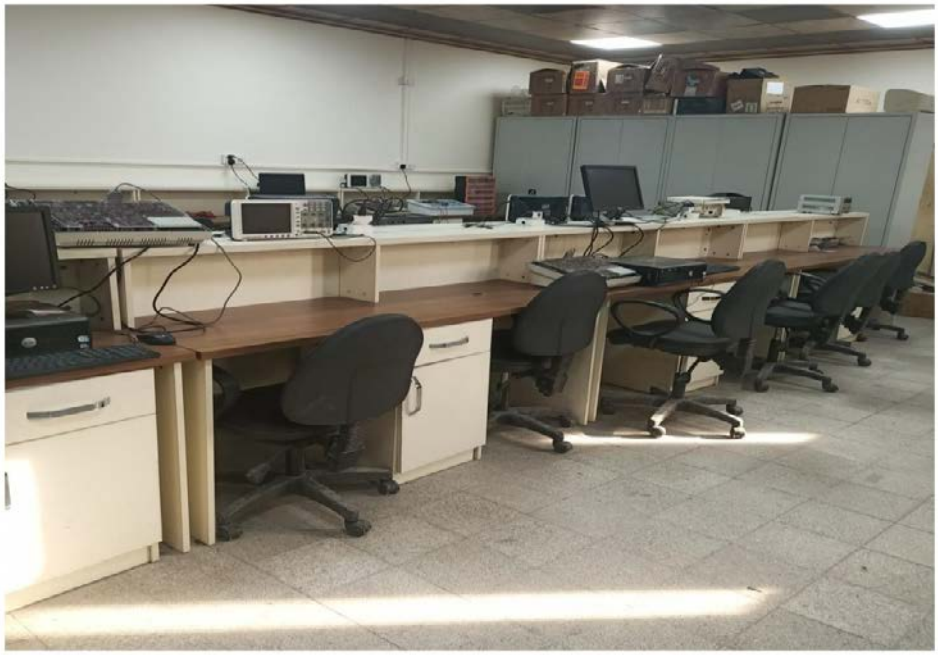
• Experimental Report Form

• Weekly Work Plan

• AVR Family AT-mega2560 Microcontroller Instructions User Guide

- **Laboratory Pictures:**





Laboratory.

Arduion Training Kit.

Desktop Computer.

Oscilloscope.

Description of the Operating System I Laboratory

1. General Information:	
Laboratory Name:	Operating System I
Associated Course Name:	Operating System I
Department:	Computer engineering
Weekly Lab Hours:	3 hours
Number of Weeks in the Semester:	15 weeks
Academic Level:	Third level
Lab Supervisor:	Dr. Sura Ramzi Shareef

2. General Description of the Laboratory:

Available Attendance Forms:

Teach students the concept of Linux programming, how to schedule it on the CPU, and how to execute it using a variety of algorithms. They also learn how to manage the process architecture (processes, threads, CPU scheduling, synchronization, and understanding the concept of deadlocks). They also learn how to solve system freezes and try to prevent or avoid them.

3. Laboratory Objectives:

Providing lectures in the designated classroom, in addition to creating a special electronic classroom for the subject.

Lectures are presented on paper, in addition to an electronic Power Point presentation presented to students.

Giving and explaining lectures in detail to students.

Asking students to submit periodic reports and homework assignments on the basic topics of the subject.

4. Learning Outcomes:

Introduction to operating systems, basic definitions of the hardware components and software used in operating systems, types of systems, their origin and development, and types of modern systems.

Learn about the structure of the operating system, how it works, and its most important basic components.

Basic principles and concepts of process management in operating systems, including process creation, scheduling, synchronization, communication.

Analyze examples of synchronization problems in operating systems, such as producer, consumer, readers, writers, and food philosophers, and propose solutions using appropriate synchronization techniques.

Identify the concept of stagnation and ways to solve the problem of system stagnation and try to prevent it or avoid its occurrence.

5. Weekly Experiment Schedule:

Week	Experiment Title	Tools / Software Used	Main Objective
1	Installing and dealing with the Linux system using the Command Line Interface. Within an Oracle Virtual Box environment.	Oracle Virtual Box	Training students on installing and dealing with the Virtual Box Using c language in unix system
2	Basics of UNIX commands and Implementation of Shell Programming	Virtual Box	Identifying the basic instructions in the Linux system (Bash Command Line).and review students of C Language Programming.
3	Management System Shell Programming	Using Command prompt implementation Unix commands .	UNIX commands and explain Implementation of Shell Programming (Command Line Interface)
4	Management System	UNIX Bash Command Line	▪ Create a file • Copy one file to another

			• Linking a file ▪ Delete a file.
5	Implementation of Process and thread (Life cycle of process): (i) Process creation and Termination; (ii) Thread creation and Termination	Bash Command Line using C Language	How can creation Process , more process and Termination process
6	Process creation as (Parent process include Child process.	Bash Command Line using C Language	Training students relationship between processes , how to deal parent process and the child process
7	Producer-Consumer Problem using Semaphores and Reader Writer Problem	Virtual Box, Bash Command Line using C Language	Synchronization process produce and consumer problem in same time
8	CPU scheduling algorithms	Virtual Box, Bash Command Line using C Language	Training students on process scheduling
9	CPU scheduling algorithms FCFS, (ii) SJF, (iii) Shortest Remaining Time First and (iv) Priority based.	Virtual Box, Bash Command Line using C Language	apply CPU scheduling algorithms
10	Simulate algorithm for deadlock prevention and detection	Virtual Box, Bash Command Line using C Language	Explain Bankers algorithm to safe state
11	Collection all experiment in main programming	Bash Command Line using C Language	Training students to create general program includes all function of lab operating

			system through course
12	Comprehensive review and discussion of the practical project	programs	Integrate acquired skills into a final project.

6. Tools and Equipment Used:

- Laboratory computers
- Linux software

7. Safety Guidelines:

- Make sure the power is turned off when connecting devices.
- Do not touch electrical outlets or network components without supervisor permission.
- Keep calm and organize cables to avoid accidents.

8. Evaluation Method:

Evaluation Item	Percentage
Attendance and participation	10%
Weekly trial reports	30%
Short tests	20%
Final practical exam	30%
Practical project	10%

9. References and Sources:

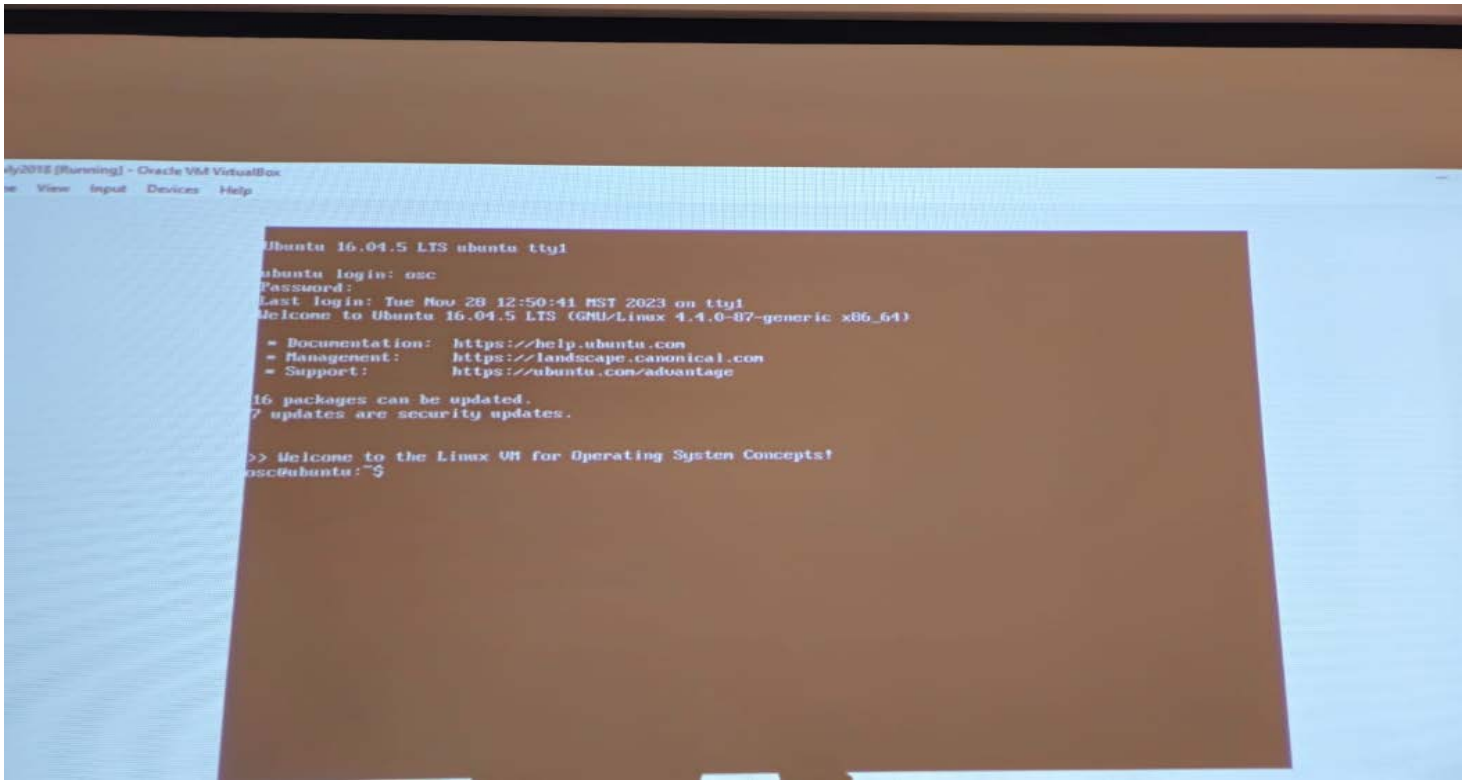
1. Operating Systems Concepts, 10th Edition Silberschatz, Abraham, Galvin, Peter B., a Gagne, Greg JohnWiley&Sons.,Inc. ISBN: 9781119320913.
2. Operating Systems Concepts, 10th Edition Silberschatz, Abraham, Galvin, Peter B., a Gagne, Greg JohnWiley&Sons.,Inc. ISBN: 9781119320913.
3. An Introduction to GCC: For the GNU Compilers GCC and G++, Brian J. Gough, Richard M. Stallman, Network Theory Ltd, ISBN : 978-095416179.

Lectures notes at www.tutorial.com

10. Attachments:

- Experiment report form
- Weekly work plan


ORACLE
VM
VirtualBox
6.0
04:09
VSTOUCH


View Input Devices Help
Ubuntu 16.04.5 LTS ubuntu tty1
ubuntu login: osc
Password:
Last login: Tue Nov 28 12:50:41 MST 2023 on tty1
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-87-generic x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
16 packages can be updated.
7 updates are security updates.
>> Welcome to the Linux VM for Operating System Concepts!
osc@ubuntu:~$

Description of the Operating System 2 Laboratory

1. General Information:	
Laboratory Name:	Operating System 2
Associated Course Name:	Operating System 2
Department:	Computer engineering
Weekly Lab Hours:	3 hours
Number of Weeks in the Semester:	15 weeks
Academic Level:	Third level
Lab Supervisor:	Dr.Sura Ramzi Shareef

2. General Description of the Laboratory:

The Operating Systems Lab aims to provide students with the knowledge and practical skills necessary to understand and apply the principles and fundamentals of operating systems. The lab focuses on providing a practical environment for applying theoretical concepts related to resource management, process control, memory management, handling devices and files, and providing a user interface. It also contributes to encouraging students to think critically and constructively in order to improve and expand their skills in how the system works and interacts with its environment through the application of exercises and experiments covered during the prescribed semester and the interactive programs required by students during the lab.

3. Laboratory Objectives:

- Explore the importance, objectives, and functions of operating systems.
- Introduce the student to the basic concepts and structure of various operating systems, how they work internally, their most important components, and their scheduling methods on the central processing unit.
- Covers the concept of operating system design principles and the various techniques used by operating systems and their implementation for memory management.
- Teach the student how an operating system manages memory: fixed partitions, variable partitions, virtual memory, paging, page replacement algorithms, fragmentation; input/output circuitry; systems practices: Linux operating system.

4. Learning Outcomes:

By the end of the lab, the student should be able to:

● Understand the basic principles and concepts of process management in operating systems, including process creation, scheduling, synchronization, and communications, to effectively manage system resources and facilitate the efficient execution of user programs.

● Gain knowledge of various memory management techniques, such as main memory management.

● Virtual memory, including concepts such as paging, segmentation, and on-demand paging, to optimize memory utilization and support multitasking in operating systems.

● Explore the structure and functions of mass storage systems, including disk organization, file systems, and input/output systems, to ensure efficient and reliable storage and retrieval of data in operating systems.

● Understand the file system interface, implementation, and internals, including file organization, directory structures, and access methods, to effectively manage and manipulate files and directories in operating systems.

5. Weekly Experiment Schedule:

Week	Experiment Title	Tools / Software Used	Main Objective
1	Installing and dealing with the Linux system using the Command Line Interface. Within an Oracle Virtual Box environment.	Virtual Box	Training students on installing and dealing with the Virtual Box Using c language in unix system
2	Basics of UNIX commands and Implementation of Shell Programming	Virtual Box	Identifying the basic instructions in the Linux system (Bash Command Line).and review students of C Language Programming.

3	Management System Shell Programming	Using Command prompt implementation Unix commands .	UNIX commands and explain Implementation of Shell Programming (Command Line Interface)
4	Management System	UNIX Bash Command Line	▪ Create a file • Copy one file to another • Linking a file ▪ Delete a file.
5	Implementation of thread creation and Termination	Bash Command Line using C Language	How can creation thread , more threads and join between two threads
6	Implementation Multithreading	Bash Command Line using C Language	How can creation thread and Join Multithreading
7	Synchronization thread-MUTEX	UNIX Bash Command Line using C Language	Explain How to synchronize thread with mutex
8	Memory Management	UNIX Bash Command Line using C Language	Teaching students concept of memory management
9	Memory Management Paging Model of Logical and Physical Memory	UNIX Bash Command Line using C Language	Logical and physical transformations

10	MMU (Page Table Map-V) Logical and Physical Address Translation	UNIX Bash Command Line using C Language	Logical and Physical Address Translation
11	Sharing Memory Implementation of share a variable	UNIX Bash Command Line using C Language	Teach students how to share memory between multiple processes.
12	Virtual memory Simulate page replacement algorithms: FIFO, LRU and Optimal	UNIX Bash Command Line using C Language	Teach students how Simulate page replacement algorithms
13	File system implementation	UNIX Bash Command Line using C Language	Teach students how open file as read/write ,protection file as group ,user, owner
14	Collection all experiment in main programming (Over all Review)	Bash Command Line using C Language	Training students to create general program includes all function of lab operating system through course
15	Comprehensive review and discussion of the practical project	programs	Integrate acquired skills into a final project.

6. Tools and Equipment Used:

- Laboratory computers.
- Linux software.

7. Safety Guidelines:

Ensure the power is off when connecting devices.

● Do not touch electrical outlets or network components without supervisor permission.

● Remain calm and organize cables to avoid accidents.

8. Evaluation Method:

Evaluation Item	Percentage
Attendance and participation	10%
Weekly trial reports	30%
Short tests	20%
Final practical exam	30%
Practical project	10%

9. References and Sources:

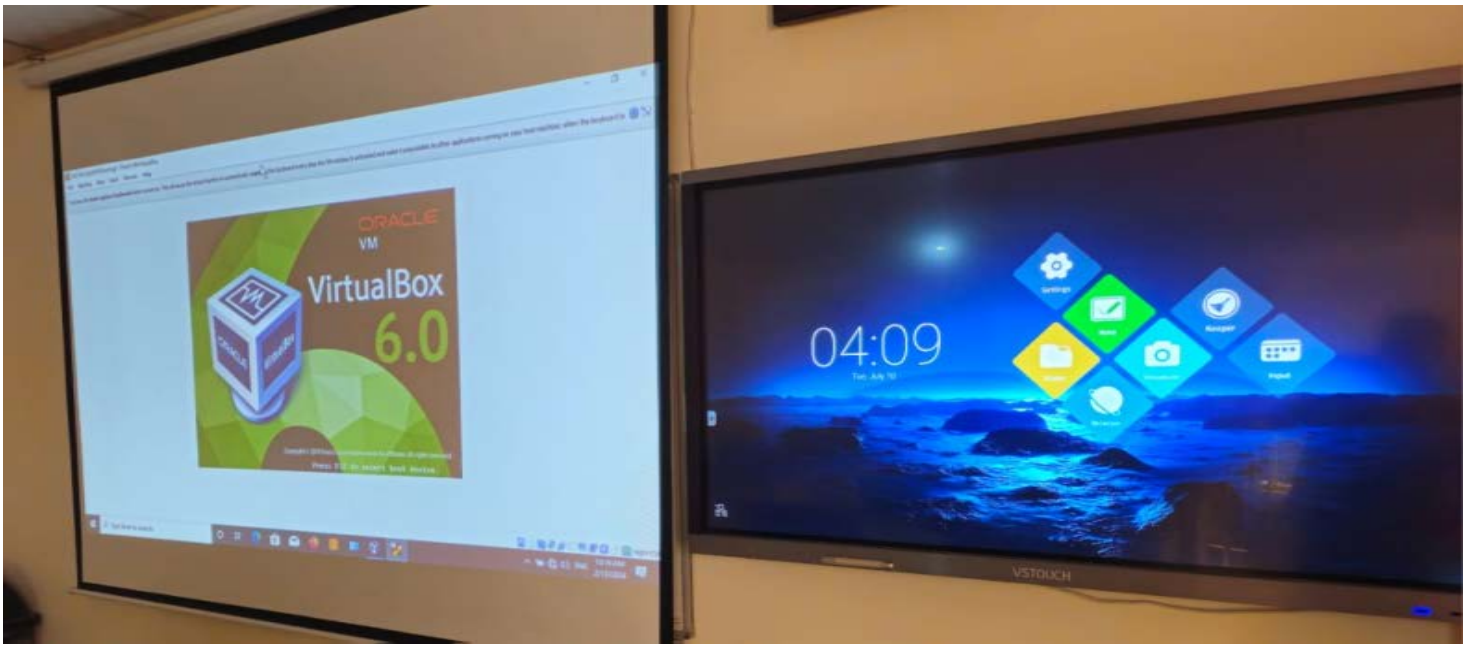
- Operating Systems Concepts, 10th Edition Silberschatz, Abraham, Galvin, Peter B., and Gagne, Greg JohnWiley&Sons.,Inc. ISBN: 9781119320913.
- An Introduction to GCC: For the GNU Compilers GCC and G++, Brian J. Gough, Richard M. Stallman, Network Theory Ltd, ISBN : 978-095416179.
- Lectures notes at www.tutorial.com.

  Other lectures notes on the Internet network.

10. Attachments:

• Experimental Report Form.

• Weekly Work Plan.


ORACLE
VM
VirtualBox
6.0
04:09
VSTOUCH