

# **OPERATING SYSTEM LABORATORY MANUAL**



**UNIVERSITY OF THE PUNJAB**

**FACULTY OF COMPUTING & INFORMATION TECHNOLOGY, LAHORE  
DEPARTMENT OF COMPUTER SCIENCE**

<b>Course:</b>	<b>Operating System Lab</b>	<b>Date:</b>
<b>Course Code:</b>	<b>CC-217-3L</b>	<b>Max Marks: 40</b>
<b>Faculty/Instructor's Name &amp; Email:</b>	<b>Dr. Ahmad Hassan Butt (ahmad.hassan@pucit.edu.pk)</b>	

**LAB MANUAL # 13  
(SPRING 2023)**

---

---

---

Name: \_\_\_\_\_ Enroll No: \_\_\_\_\_

---

**Objective(s) :**

To understand memory management using paging.

**Lab Tasks :**

**Task 01:** Explain memory management.

**Task 02:** Explain the process of paging.

**Task 03:** Differentiate between logical and physical address.

**Task 04:** Write the output of the program given in Lab Manual.

**Lab Grading Sheet :**

Task	Max Marks	Obtained Marks	Comments( <i>if any</i> )
1.	10		
2.	10		
3.	10		
4.	10		
<b>Total</b>	<b>40</b>		<b>Signature</b>

**Note : Attempt all tasks and get them checked by your Instructor**

## Lab 13: Memory Management Scheme-Paging

### Objective(s):

To write a C program to implement memory management using paging technique.

### Tool(s) used:

Ubuntu, VIM Editor

### Paging:

Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non – contiguous.

- Logical Address or Virtual Address (represented in bits): An address generated by the CPU
- Logical Address Space or Virtual Address Space( represented in words or bytes): The set of all logical addresses generated by a program
- Physical Address (represented in bits): An address actually available on memory unit
- Physical Address Space (represented in words or bytes): The set of all physical addresses corresponding to the logical addresses

Read the base address, page size, number of pages and memory unit. If the memory limit is less than the base address display the memory limit is less than limit. Create the page table with the number of pages and page address. Read the page number and displacement value. If the page number and displacement value is valid, add the displacement value with the address corresponding to the page number and display the result. Display the page is not found or displacement should be less than page size. Stop the program.

## **PROGRAM**

```
#include<stdio.h>
#include<unistd.h>
void main(){

    int b[20],n,i,pa,p,a,d;
    printf("\nProgram for
paging"); scanf("%d",&n);
    printf("\nEnter the base
address:");
    for(i=0;i<n;i++){
        scanf("%d",&b[i]);
    }
    printf("\nEnter the logical
address:"); scanf("%d",&p);
    for(i=0;i<n;i++){
        if(i==p){
            pa=b[i]+d;
            a=b[i];
            printf("\n\tPageNo.\t BaseAdd. PhysicalAdd. \n\t %d \t
            %d \t %d \t
            ",p,a,pa);
        }
    }
    printf("\nInvalid page");
}
```

**Sample Input 1**

**Sample Output 1**

**Sample Input 2**

**Sample Output 2**