Practical No. : 8

Name: Hajare Vinay Arjun **Branch:** Computer Engineering

Div: CS-A **Roll No.:** 82 **P.R.N. No.:** 12320035

Course: Operating System

Aim: Write a C / C++/ Java program to convert given virtual/ logical address in physical address using segmentation and paging

Theory:

Code –

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define SEGMENT_SIZE 1024 // Size of each segment
#define PAGE_SIZE 256     // Size of each page

// Segment Table Entry
typedef struct {
    int segmentNumber;
    int baseAddress;
    int limit;
} SegmentTableEntry;

// Page Table Entry
typedef struct {
    int pageNumber;
    int frameNumber;
} PageTableEntry;

// Function to perform address translation using segmentation
int translateAddressSegmentation(int logicalAddress, SegmentTableEntry
segmentTable[], int segmentCount) {
    int segmentNumber = logicalAddress / SEGMENT_SIZE; // Calculate segment
number
    int offset = logicalAddress % SEGMENT_SIZE;       // Calculate offset

    if (segmentNumber < segmentCount) {
        if (offset < segmentTable[segmentNumber].limit) {
            int physicalAddress = segmentTable[segmentNumber].baseAddress +
offset;
            return physicalAddress;
```

```
        } else {
            printf("Segmentation Fault: Offset exceeds segment limit\n");
            return -1;
        }
    } else {
        printf("Segmentation Fault: Segment number exceeds segment count\n");
        return -1;
    }
}

// Function to perform address translation using paging
int translateAddressPaging(int logicalAddress, PageTableEntry pageTable[], int
pageCount) {
    int pageNumber = logicalAddress / PAGE_SIZE; // Calculate page number
    int offset = logicalAddress % PAGE_SIZE;    // Calculate offset

    if (pageNumber < pageCount) {
        int frameNumber = pageTable[pageNumber].frameNumber;
        int physicalAddress = (frameNumber * PAGE_SIZE) + offset;
        return physicalAddress;
    } else {
        printf("Page Fault: Page number exceeds page count\n");
        return -1;
    }
}

int main() {
    // Segment Table
    SegmentTableEntry segmentTable[4] = {
        {0, 0, 1023},   // Segment 0
        {1, 1024, 511}, // Segment 1
        {2, 1536, 255}, // Segment 2
        {3, 1792, 511}  // Segment 3
    };

    // Page Table
    PageTableEntry pageTable[4] = {
        {0, 2}, // Page 0
        {1, 3}, // Page 1
        {2, 1}, // Page 2
        {3, 0}  // Page 3
    };

    // Logical address to be translated
    int logicalAddress = 883;

    // Translate logical address to physical address using segmentation
```

```
    int physicalAddressSegmentation =
translateAddressSegmentation(logicalAddress, segmentTable, 4);

    if (physicalAddressSegmentation != -1) {
        printf("Physical Address (Segmentation): %d\n",
physicalAddressSegmentation);
    }

    // Translate logical address to physical address using paging
    int physicalAddressPaging = translateAddressPaging(logicalAddress,
pageTable, 4);

    if (physicalAddressPaging != -1) {
        printf("Physical Address (Paging): %d\n", physicalAddressPaging);
    }

    return 0;
}
```

Output:

Physical Address (Segmentation): 883

Physical Address (Paging): 115