Siri N Shetty

PES2UG22CS556

Semester 4 Section J

Programming Exercise-3 (UE22CS242B – Operating Systems)

Exercise #3

Write a program to simulate Segmentation. Compute the physical address

Take as input:

1. Segment number
2. Base address
3. Segment limit

(Create a segment table with the necessary fields with values.)

Segment Number	Base Address	Segment Limit
10	12	84
13	2	55
32	66	800
10	24	200
5	12	95

Final Calculation:

When the offset 9 and the segmentation number 5 are entered, the program computes the physical address.

The base address for segment number 5 is 12.

The offset given is 9.

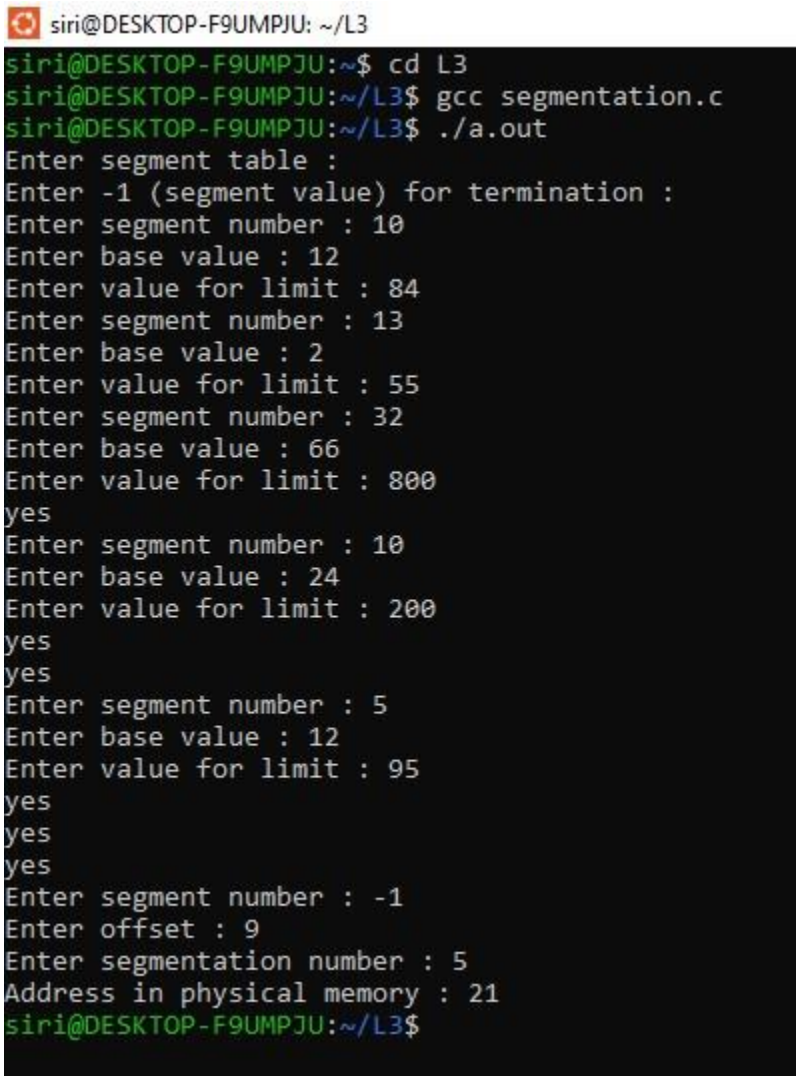
Therefore, the physical address = base address + offset = 12 + 9 = 21.

Program:

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

struct list {
    int seg_num;
    int base_addr;
    int seg_limit;
    struct list *next;
} *p;

void insert(struct list *q, int base_addr, int seg_limit, int seg_num) {
    if (p == NULL) {
        p = malloc(sizeof(struct list));
        p->seg_limit = seg_limit;
        p->base_addr = base_addr;
        p->seg_num = seg_num;
        p->next = NULL; }
    else {
        while (q->next != NULL) {
            q = q->next;
            printf("yes\n");
        }
        q->next = malloc(sizeof(struct list));
        q->next->seg_limit = seg_limit;
        q->next->base_addr = base_addr;
        q->next->seg_num = seg_num;
        q->next->next = NULL;
    }
}

int find(struct list *q, int seg_num) {
    while (q->seg_num != seg_num) {
        q = q->next;
    }
    return q->seg_limit;
}

int search(struct list *q, int seg_num) {
    while (q->seg_num != seg_num) {
        q = q->next;
    }
    return q->base_addr;
}
```

```c
int main() {
    p = NULL;
    int seg, offset, limit, base, c, s, physical;
    printf("Enter segment table : \n");
    printf("Enter -1 (segment value) for termination : \n");
    do {
        printf("Enter segment number : ");
        scanf("%d", &seg);
        if (seg != -1) {
            printf("Enter base value : ");
            scanf("%d", &base);
            printf("Enter value for limit : ");
            scanf("%d", &limit);
            insert(p, base, limit, seg);
        }
        } while (seg != -1);

        printf("Enter offset : ");
        scanf("%d", &offset);
        printf("Enter segmentation number : ");
        scanf("%d", &seg); c = find(p, seg);
        s = search(p, seg);

        if (offset < c) {
            physical = s + offset;
            printf("Address in physical memory : %d\n", physical);
        }
        else {
            printf("error\n");
        }
}
```

Output:


siri@DESKTOP-F9UMPJU: ~/L3
siri@DESKTOP-F9UMPJU:~$ cd L3
siri@DESKTOP-F9UMPJU:~/L3$ gcc segmentation.c
siri@DESKTOP-F9UMPJU:~/L3$ ./a.out
Enter segment table :
Enter -1 (segment value) for termination :
Enter segment number : 10
Enter base value : 12
Enter value for limit : 84
Enter segment number : 13
Enter base value : 2
Enter value for limit : 55
Enter segment number : 32
Enter base value : 66
Enter value for limit : 800
yes
Enter segment number : 10
Enter base value : 24
Enter value for limit : 200
yes
yes
Enter segment number : 5
Enter base value : 12
Enter value for limit : 95
yes
yes
yes
Enter segment number : -1
Enter offset : 9
Enter segmentation number : 5
Address in physical memory : 21
siri@DESKTOP-F9UMPJU:~/L3$