OPERATING SYSTEM ASSIGNMENT

DATE : 25-03-2024

Name : ANKITH GOWDA B S

SRN : PES2UG22CS077

QUESTION :

```
Write a Program To Simulate Segmentaion. Compute The Physical Address
    Take A Input
        1. Segment Number
        2. Base Address
        3. Segment Limit
```

CODE :

```c
#include <stdio.h>
#include <stdlib.h>
typedef struct{
    int seg_num;
    int base_addr;
    int seg_limit;
    int *pageTable;
} segment;
int phys_addr(segment segment, int offset){
    if (offset > segment.seg_limit){
        printf("Offset is out of bounds.\n");
        return -1;
    }
    return segment.base_addr + offset;
}
void matrix_table(segment *matrix, int size){
    printf("\nMatrix Table:\n");
    printf("Segment Number | Base Address | Segment Limit\n");
    for (int i = 0; i < size; i++)
    {
        printf("%5d %12d %12d\n", matrix[i].seg_num, matrix[i].base_addr,
                matrix[i].seg_limit);
    }
}
```

```c
int main(){
    int size;
    printf("Enter the number of segments: ");
    scanf("%d", &size);
    segment *matrix = (segment *)malloc(size * sizeof(segment));
    for (int i = 0; i < size; i++){
        printf("Enter details for segment %d:\n", i);
        printf("Segment Number: ");
        scanf("%d", &matrix[i].seg_num);
        printf("Base Address: ");
        scanf("%d", &matrix[i].base_addr);
        printf("segment Limit: ");
        scanf("%d", &matrix[i].seg_limit);
        matrix[i].pageTable = (int *)malloc(sizeof(int) * 1);
        matrix[i].pageTable[0] = matrix[i].base_addr;
    }
    matrix_table(matrix, size);
    for (int i = 0; i < size; i++){
        int offset;
        printf("Enter offset for segment %d: ", i);
        scanf("%d", &offset);
        int physicalAddress = phys_addr(matrix[i], offset);
        if (physicalAddress != -1)
        {
            printf("Physical Address for segment %d: %d\n", i,
physicalAddress);
        }
    }
    for (int i = 0; i < size; i++)
    {
        free(matrix[i].pageTable);
    }
    free(matrix);
    return 0;
}
```

OUTPUT :

```
● PS C:\Users\ankit\OneDrive\Documents\GitHub\SEM-4\OS\ASSIGNMENT-3> gcc -o q1 q1.c
● PS C:\Users\ankit\OneDrive\Documents\GitHub\SEM-4\OS\ASSIGNMENT-3> ./q1
 Enter the number of segments: 3
 Enter details for segment 0:
 Segment Number: 0
 Base Address: 3221
 segment Limit: 700
 Enter details for segment 1:
 Segment Number: 1
 Base Address:
 245
 segment Limit: 300
 Enter details for segment 2:
 Segment Number: 2
 Base Address:
 2789
 segment Limit: 500
```

```
Matrix Table:
Segment Number | Base Address | Segment Limit
    0             3221             700
    1              245             300
    2             2789             500
Enter offset for segment 0: 345
Physical Address for segment 0: 3566
Enter offset for segment 1: 762
Offset is out of bounds.
Enter offset for segment 2: 231
Physical Address for segment 2: 3020
```