OPERATING SYSTEM

Name: Keerthan P.V

SRN: PES2UG23CS272

Unit #3

1)Write a program to simulate Segmentation. Compute the physical address
Take as input:

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

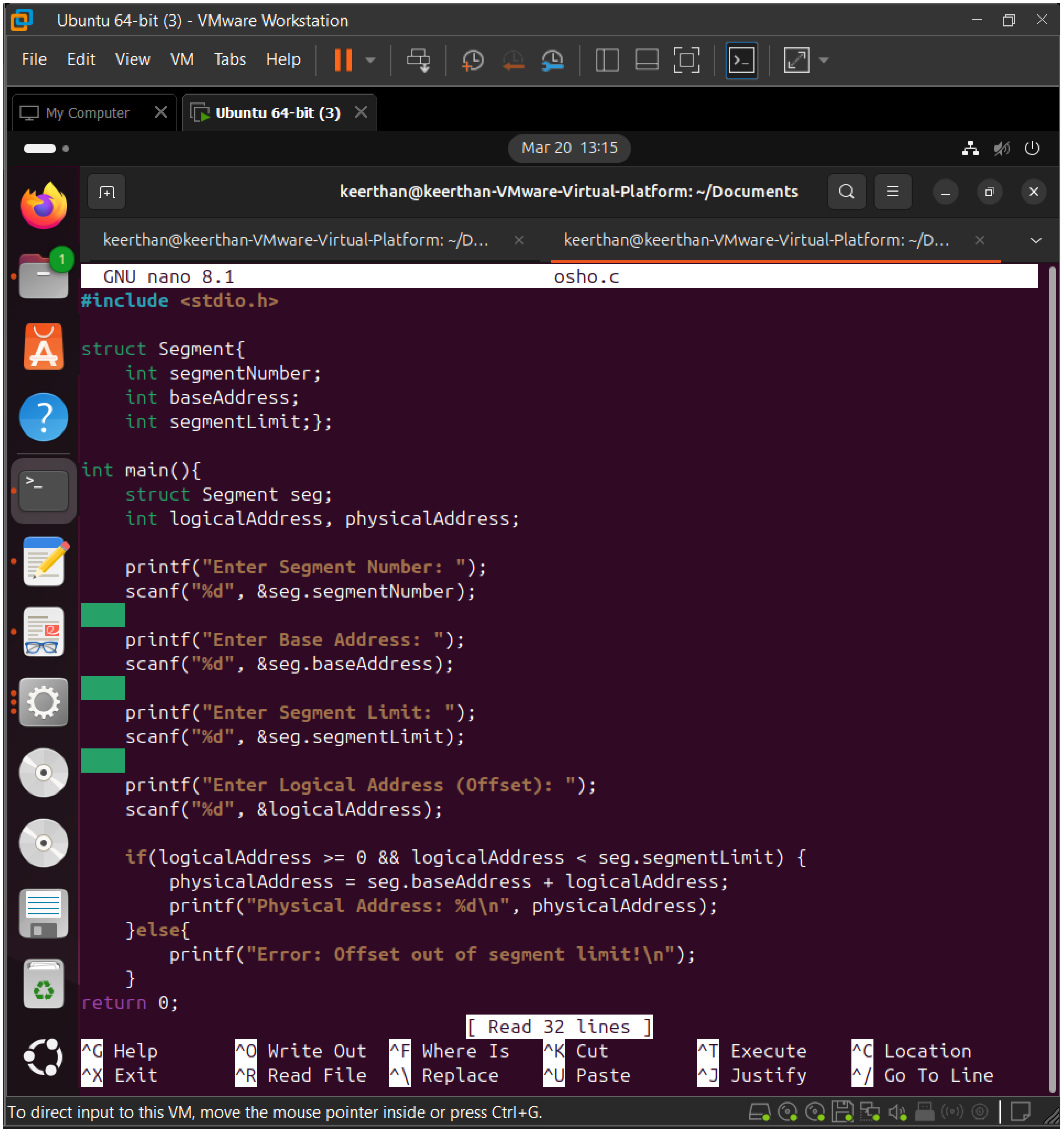
OUTPUT:

```
#include <stdio.h>


struct Segment{

   int segmentNumber;

   int baseAddress;

   int segmentLimit;};


int main(){

   struct Segment seg;
```

```
int logicalAddress, physicalAddress;


printf("Enter Segment Number: ");

scanf("%d", &seg.segmentNumber);


printf("Enter Base Address: ");

scanf("%d", &seg.baseAddress);


printf("Enter Segment Limit: ");

scanf("%d", &seg.segmentLimit);


printf("Enter Logical Address (Offset): ");

scanf("%d", &logicalAddress);


if(logicalAddress >= 0 && logicalAddress < seg.segmentLimit) {

    physicalAddress = seg.baseAddress + logicalAddress;

    printf("Physical Address: %d\n", physicalAddress);

}else{

    printf("Error: Offset out of segment limit!\n");

}
```

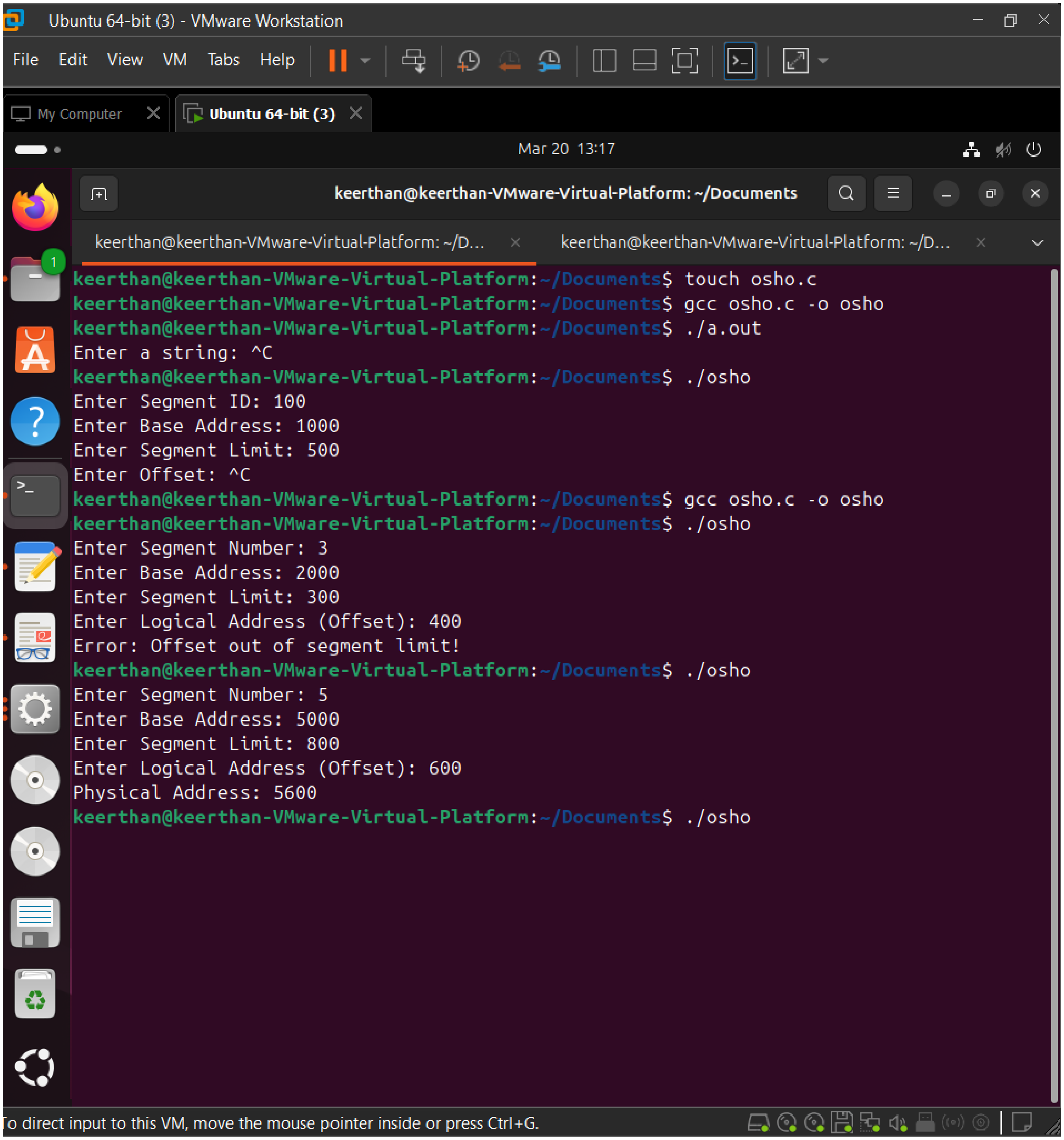
return 0;

}


Ubuntu 64-bit (3) - VMware Workstation
File Edit View VM Tabs Help
My Computer
Ubuntu 64-bit (3)
Mar 20 13:15
keerthan@keerthan-VMware-Virtual-Platform: ~/Documents
keerthan@keerthan-VMware-Virtual-Platform: ~/D...
keerthan@keerthan-VMware-Virtual-Platform: ~/D...

```
  GNU nano 8.1                          osho.c
#include <stdio.h>

struct Segment{
    int segmentNumber;
    int baseAddress;
    int segmentLimit;};

int main(){
    struct Segment seg;
    int logicalAddress, physicalAddress;

    printf("Enter Segment Number: ");
    scanf("%d", &seg.segmentNumber);
    
    printf("Enter Base Address: ");
    scanf("%d", &seg.baseAddress);
    
    printf("Enter Segment Limit: ");
    scanf("%d", &seg.segmentLimit);
    
    printf("Enter Logical Address (Offset): ");
    scanf("%d", &logicalAddress);

    if(logicalAddress >= 0 && logicalAddress < seg.segmentLimit) {
        physicalAddress = seg.baseAddress + logicalAddress;
        printf("Physical Address: %d\n", physicalAddress);
    }else{
        printf("Error: Offset out of segment limit!\n");
    }
return 0;
                                              [ Read 32 lines ]
^G Help         ^O Write Out    ^F Where Is     ^K Cut          ^T Execute      ^C Location
^X Exit         ^R Read File    ^\ Replace      ^U Paste        ^J Justify      ^/ Go To Line
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


Ubuntu 64-bit (3) - VMware Workstation
File Edit View VM Tabs Help
My Computer
Ubuntu 64-bit (3)
Mar 20 13:17
keerthan@keerthan-VMware-Virtual-Platform: ~/Documents
keerthan@keerthan-VMware-Virtual-Platform: ~/D...
keerthan@keerthan-VMware-Virtual-Platform: ~/D...

```
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ touch osho.c
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ gcc osho.c -o osho
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ ./a.out
Enter a string: ^C
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ ./osho
Enter Segment ID: 100
Enter Base Address: 1000
Enter Segment Limit: 500
Enter Offset: ^C
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ gcc osho.c -o osho
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ ./osho
Enter Segment Number: 3
Enter Base Address: 2000
Enter Segment Limit: 300
Enter Logical Address (Offset): 400
Error: Offset out of segment limit!
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ ./osho
Enter Segment Number: 5
Enter Base Address: 5000
Enter Segment Limit: 800
Enter Logical Address (Offset): 600
Physical Address: 5600
keerthan@keerthan-VMware-Virtual-Platform:~/Documents$ ./osho
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

2) Compare virtual memory in Windows and Linux

Comparison of Virtual Memory in Windows and Linux

Paging Mechanism

Windows uses demand paging and a pagefile stored on disk to manage virtual memory.

Linux also uses demand paging but relies on a swap partition or swap file for handling virtual memory.

Swap Management

Windows stores its swap data in a file called pagefile.sys, which is dynamically managed by the system.

Linux allows both swap partitions and swap files, giving users more flexibility in configuring virtual memory.

Page Replacement Algorithm

Windows employs the Least Recently Used (LRU) algorithm along with optimizations like SuperFetch to improve performance.

Linux primarily uses LRU, but also supports other algorithms like NRU (Not Recently Used) and Clock Algorithm for better memory management.

Memory Overcommit

Windows avoids memory overcommitment, meaning applications may fail to allocate memory if there isn't enough available.

Linux allows overcommitment, meaning it may allocate more memory than is physically available. However, if memory runs out, the OOM Killer (Out of Memory Killer) terminates processes to free up space.

Memory Allocation

In Windows, applications allocate memory using VirtualAlloc and free it using VirtualFree.

In Linux, memory is allocated using system calls like mmap(), sbrk(), and functions like malloc() and free().

Address Translation

Windows manages virtual memory using Page Table Entries (PTEs) and optimizes translation through a Translation Lookaside Buffer (TLB).

Linux also relies on PTEs and TLB, but includes additional features like HugePages for better efficiency with large memory allocations.

Performance Optimization

Windows uses SuperFetch and ReadyBoost to preload frequently used applications into memory.

Linux improves performance using swapiness, Transparent Huge Pages (THP), and Kernel Same-page Merging (KSM) to optimize memory usage.

Customization & Control

Windows provides limited user control over virtual memory settings, mainly through the control panel options.