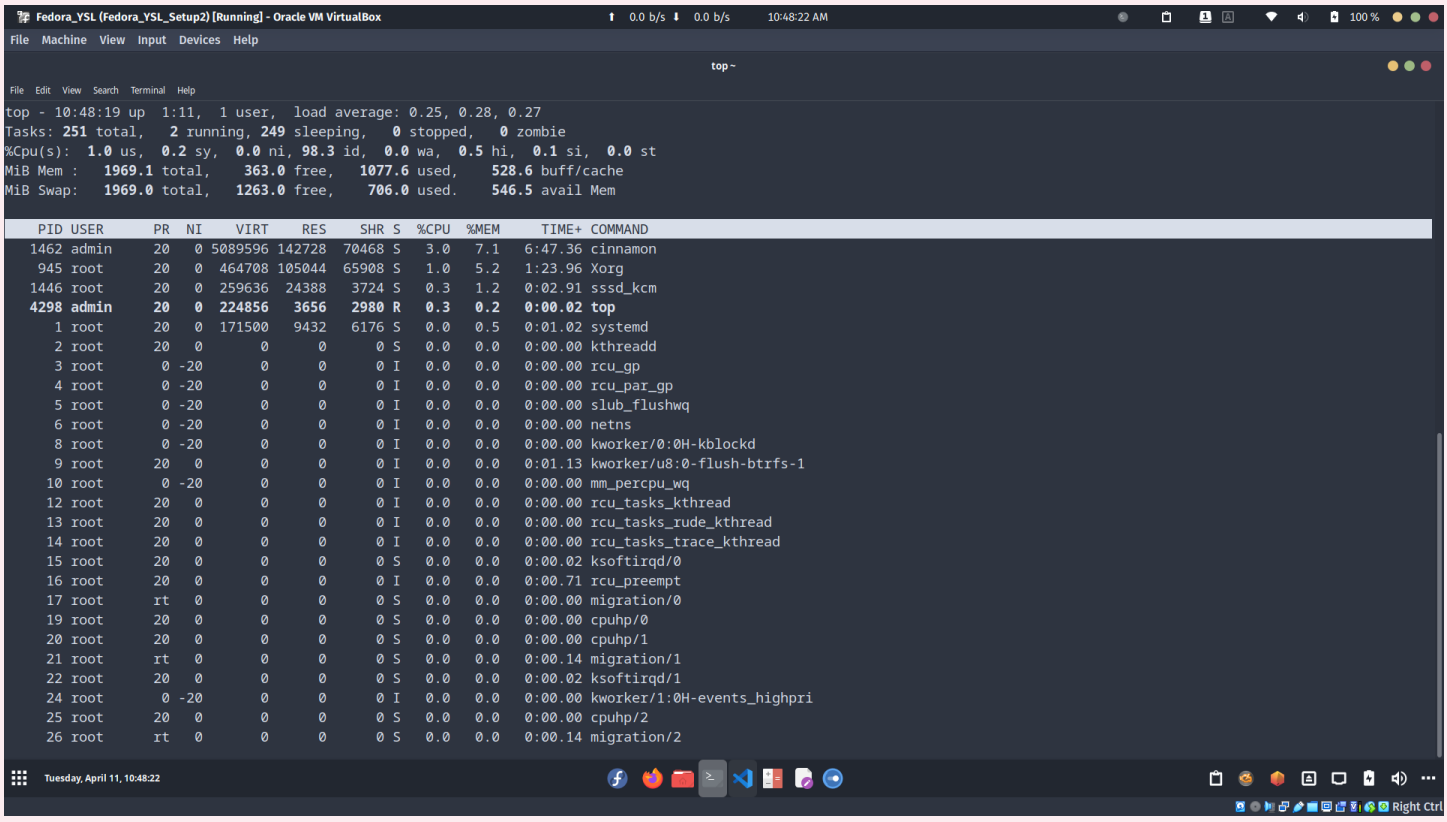
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 41
OS Practical 9

1. How to check/See Memory Allocation in your system. Also check that which users have consumed how much main and virtual memory? Also verify that:
   A. List of users who have started a process and tried to consume memory.
   B. Total size of the process in Virtual Memory in KB as a decimal integer.
   C. The Resident Set Size of the process, in kilobytes as a decimal integer.
   D. The ratio of the process's resident set size to the physical memory on the machine, expressed as a percentage.
   E. Name of the process

Command :

```
top (or htop can be used if installed)
```

Output :



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 41
OS Practical 9

2. **For a given Problem -1 , Design the solution after applying Banker Algorithm using C program.**

Code :

```
#include"fcntl.h"
#include<unistd.h>
#include<sys/stat.h>
#include"stdio.h"
#include"stdlib.h"
#include<sys/mman.h>
int main()
{
   int fd;
   char buffer[100];
   fd = open ("ICTGUNI. txt", O_CREAT | O_RDWR, 0666) ;
   if(fd≠-1)
   {
       char a[]= "I am student of ICT!!";
       write(fd,a,sizeof(a));
       struct stat st;
       stat("ICTGUNI. txt", &st) ;
       printf("File Size: %lu\n",st.st_size);
       int *value = mmap(NULL, 4*sizeof (int), PROT_READ|
PROT_WRITE, MAP_PRIVATE | MAP_ANONYMOUS, 0, 0);
       if (value==MAP_FAILED)
```

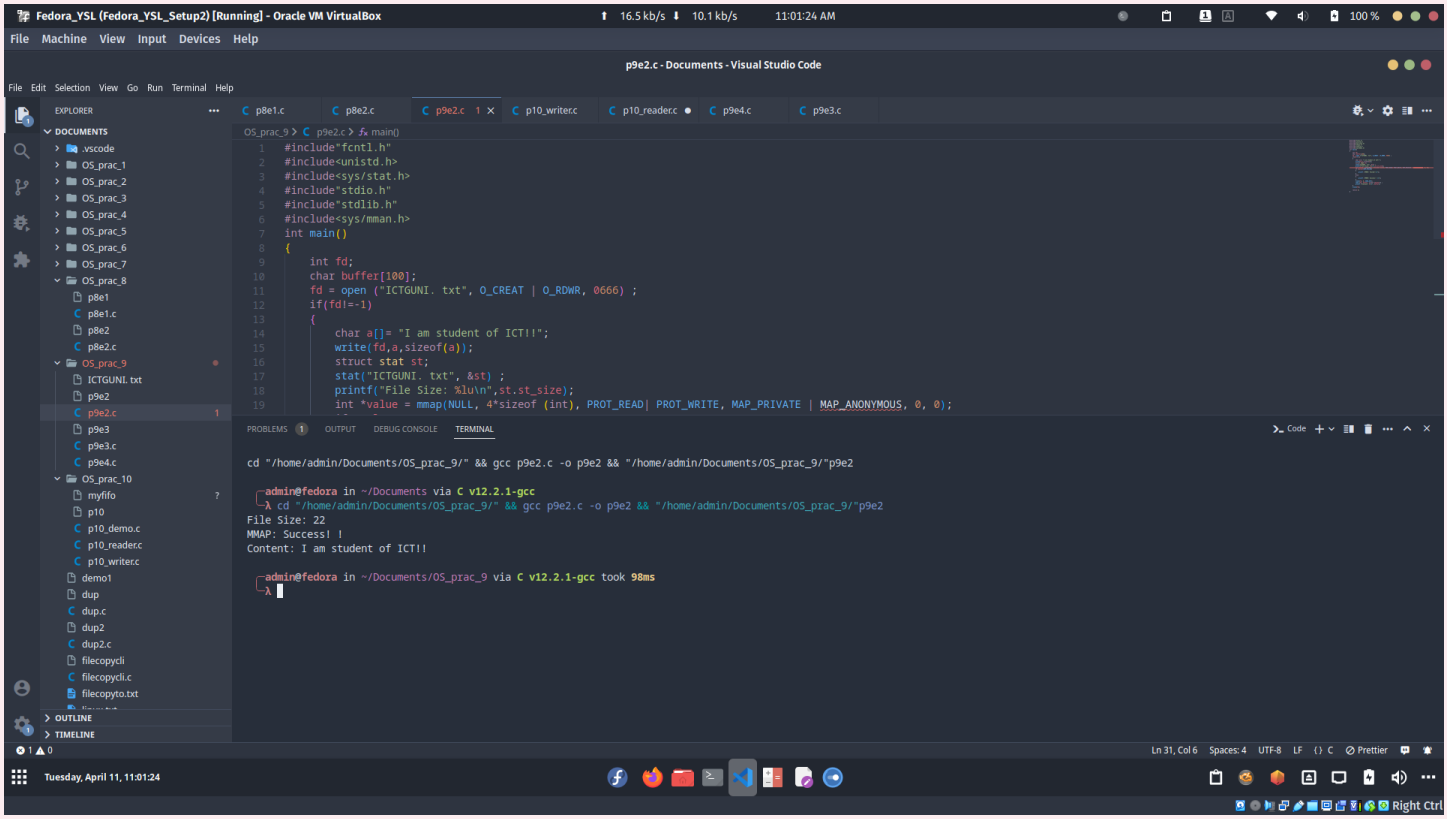
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 41
OS Practical 9

```
        {
            printf ("MMAP: Failed!!\n");
        }
        else
        {
            printf ("MMAP: Success! !\n");
        }
        lseek(fd, 0, SEEK_SET);
        read (fd, buffer, sizeof (buffer)) ;
        printf ("Content: %s\n", buffer);
    }
    close(fd);

    return 0;
}
```

Output :

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 41
OS Practical 9



3. **Using memchr() system call, perform memory searches (scanning of memory) for the first occurrence of the character given by setting the first n bytes of the string pointed to.**

Code :

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


int main()
{
   char a[] = "linuxkernel.com";
   char searchObj = '.';
```

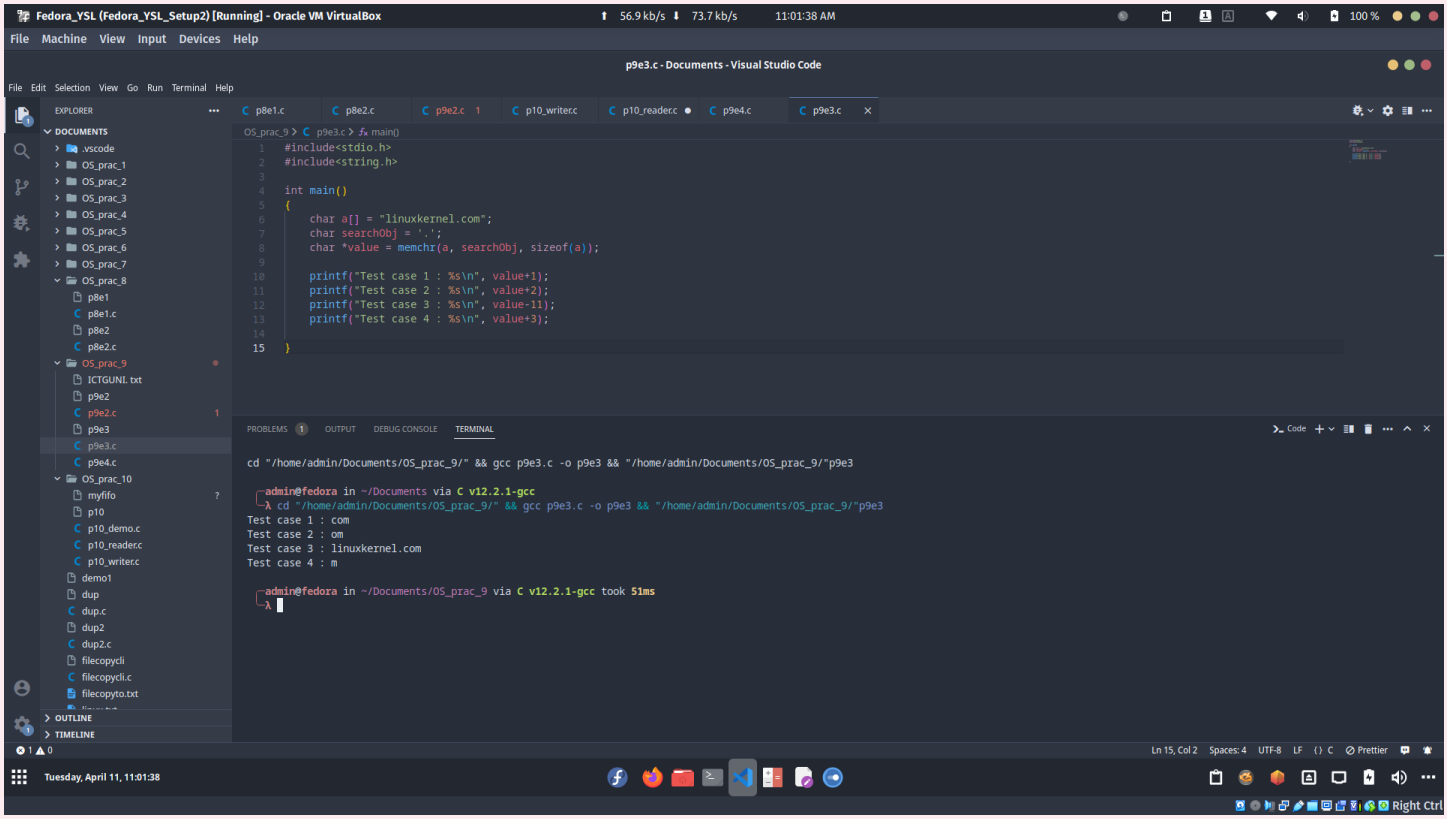
Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 41
OS Practical 9

```
    char *value = memchr(a, searchObj, sizeof(a));

    printf("Test case 1 : %s\n", value+1);
    printf("Test case 2 : %s\n", value+2);
    printf("Test case 3 : %s\n", value-11);
    printf("Test case 4 : %s\n", value+3);

}
```

Output :



Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 41
OS Practical 9

4. **The system functions brk(), and sbrk() is utilized to regulate the amount of memory allotted towards the process's data segment. Usually, these functions are invoked from a larger memory management library function like malloc. The program break, which identifies the end of the process's heap section, is moved about with brk() and sbrk(). brk() assigns the value of addr to the ending of the heap segment. sbrk() increases the heap space of the program by increment bytes. It takes you back to the earlier program break. The present location of the program break can be found by calling sbrk() with just a raise of 0. Implement brk() and SBRK() system call in C**

Code :

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

int main()
{
   void* old_brk = sbrk(0); // get current end of heap memory segment
   printf("Current end of heap : %p\n", old_brk);

   int* p= (int*)sbrk(4); // allocate 4 bytes on the loop
   *p = 42;

   void *new_brk = sbrk(0); // get new end of heap memory segment
   printf("New end of heap : %p\n", new_brk);
```

Name - Yash Lakhtariya
Enrollment number - 21162101012
Branch - CBA Batch - 41
OS Practical 9

```
    printf("Value of p : %d\n", *p);

    brk(old_brk);

    return 0;
}
```

Output :