**Tasks Assigned:-  
  
  
1) How can we open and switch between multiple files on linux terminal without using vi/vim?**

>> I have googled **“how to split terminal in ubuntu using command”** and I followed 1st link: <https://linuxhint.com/split-linux-terminal/>

**>> Using tmux command for split terminals:**

**$ sudo apt install tmux -y**

Ctrl + b + % for vertical and ctrl +b+” for horizontal.

Ctrl+b " Split current pane horizontally into two panes

Ctrl+b % Split current pane vertically into two panes

Ctrl+b o Go to the next pane

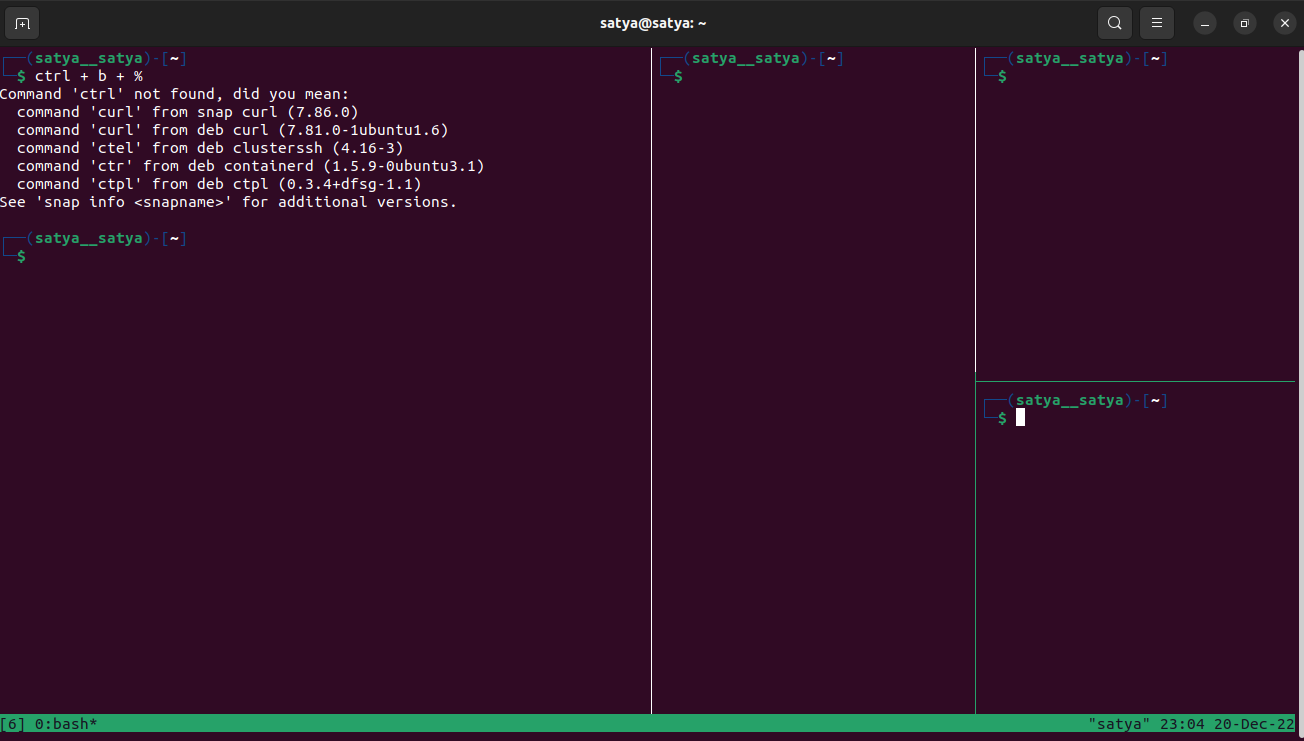
Ctrl+b ; Toggle between the current and previous pane

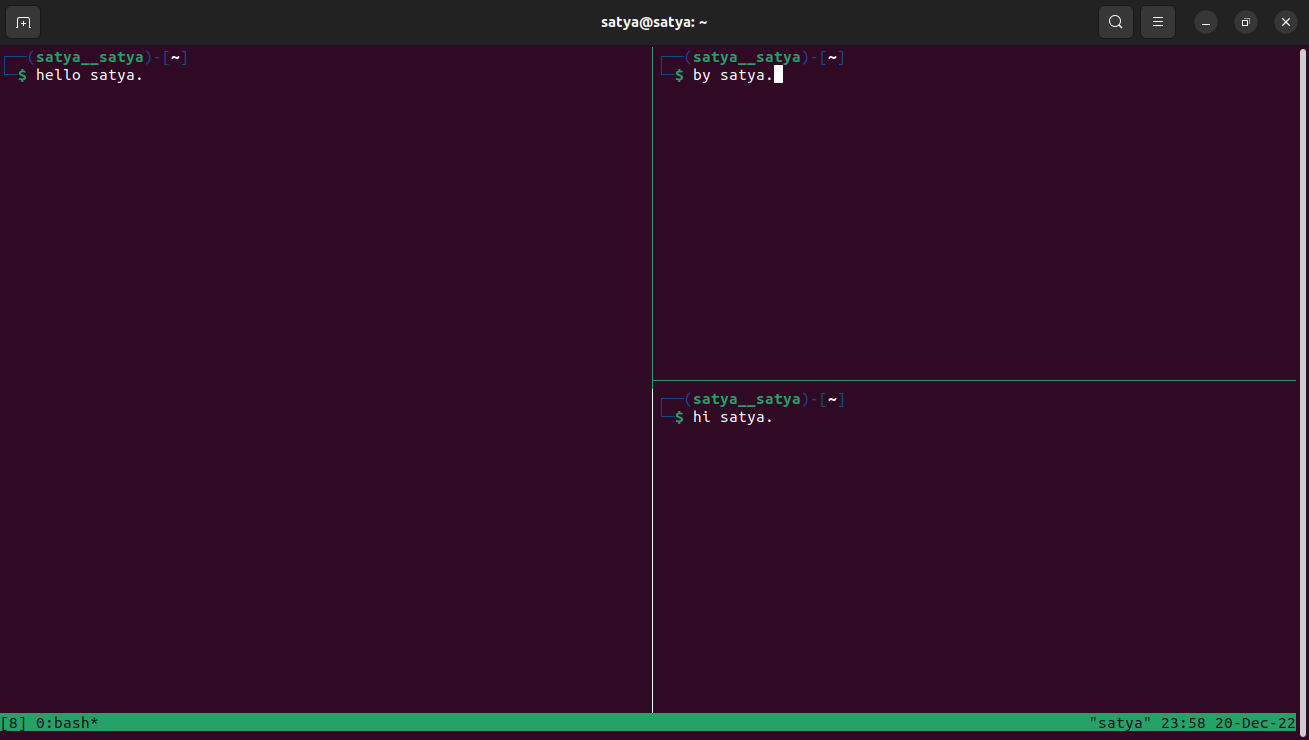
Ctrl+b x Close the current pane

Ctrl+b d keep your terminal running in background (you can close the terminal even if a socket is running.

Ctrl+b w list your panes (e.g. after reopening the terminal)

Ctrl+b [ Activate scroll. Use ESC to exit.





**2) To run a program and visualise memory leaks in Java and C language using Plumbr.**

I have googled **“Memory leak in Java program”** and I followed 2nd link: <https://www.javatpoint.com/memory-leak-in-java>

* **memory leaks in C**

┌──(satya㉿satya)-[~/tester\_program]

└─$ cat leak.c

#include <stdio.h>

#include <stdlib.h>

int main()

{

int \*ptr, i;

ptr = (int \*)malloc(sizeof(int) \* 5);

for (i = 0; i < 5; i++)

{

ptr[i] = i;

}

for (i = 0; i < 5; i++)

{

printf("%d\n", ptr[i]);

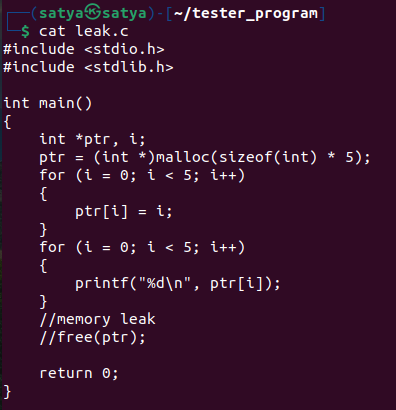
}

//memory leak

//free(ptr);

return 0;

}



┌──(satya㉿satya)-[~/tester\_program]

└─$ ./leak

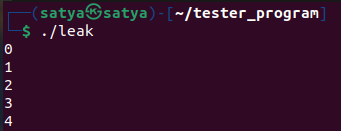
0

1

2

3

4



installed valgrind for checking the memory leak by executing the following command:

**sudo apt install valgrind**  => for Ubuntu, Debian, etc.

**sudo yum install valgrind**  => for RHEL, CentOS, Fedora, etc

In case, If the above mentioned commands don't work, try the following command:

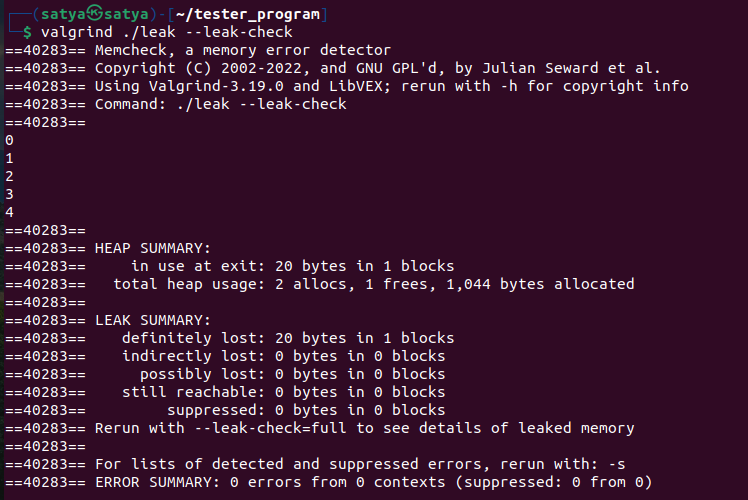
**sudo snap install valgrind --classic**

Then, Run the following command to check the memory leak (for C programs)

valgrind ./<c\_program\_file\_name>

In my case, it is **valgrind ./leak**

Here is the output of 'valgrind ./leak' command:



* **memory leaks in java**

┌──(satya㉿satya)-[~/tester\_program]

└─$ cat Leakmemory.java

// Java Program to illustrate memory leaks

import java.util.Vector;

public class Leakmemory

{

public static void main(String[] args)

{

Vector v = new Vector(214444);

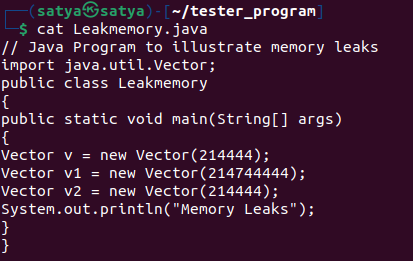
Vector v1 = new Vector(214744444);

Vector v2 = new Vector(214444);

System.out.println("Memory Leaks");

}

}



I have compiled the program.

┌──(satya㉿satya)-[~/tester\_program]

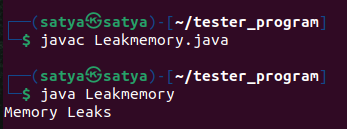
└─$ javac Leakmemory.java

* **Output.**

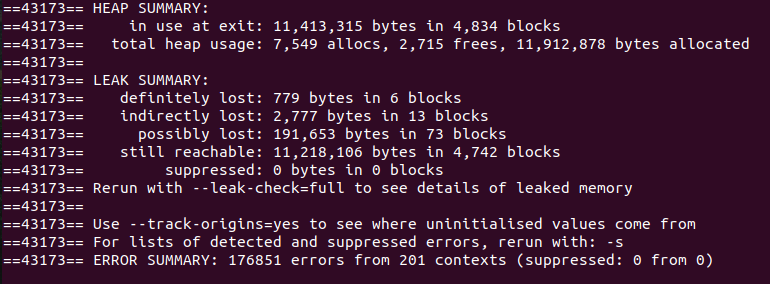
┌──(satya㉿satya)-[~/tester\_program]

└─$ java Leakmemory

Memory Leaks



Output of leak memory for java program.



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*