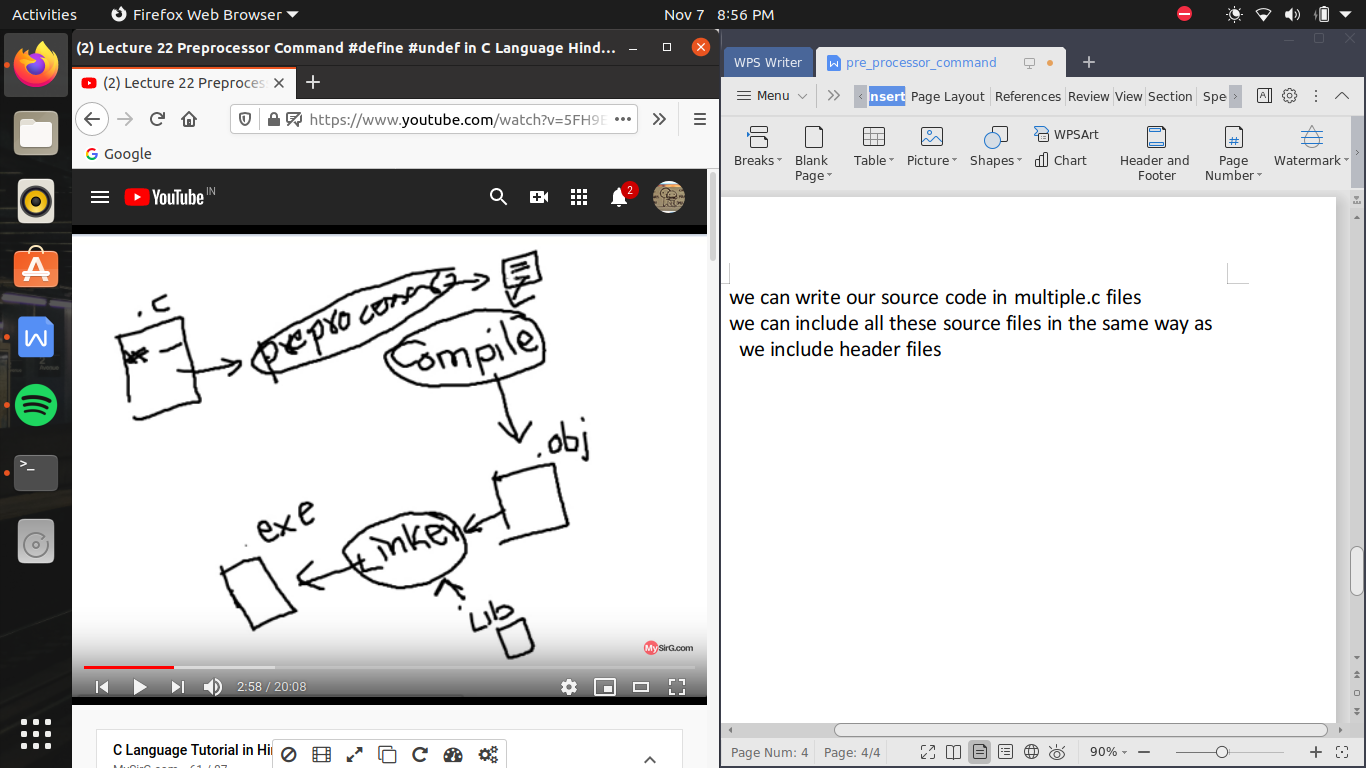
pre-processor command



#include in c language ->

1. pre-processor is a program just like a compiler is a program
2. pre-processor always runs before the compiler

when we start compiling .c file so before compiler, preprocessor will run and consider only those lines which include (#)

or baki lines ko wo untouched rakhata hai us me koe change nhi karata, or # symbol wali lines ki jagah ab kuch or a chuka hota hai

ab compiler es file ko padata hai na ki hamari original .c file ko or fir compiler bana deta hai .obj file

then we need linker software ki , linker software . obj file ko library files ke sath jodata hai jis me sare predefined function ki coding maujut hoti hai or bana deta hai ek nayi file jo hoti hai .exe file

line which include # is knowns as pre-processor directive

because # symbol wali line preprocessor ko direction provide krate hai h

toh ek line me ek se jayada # laga kr hum ek se jayada direction provide nhi kr sakate

the word after # is called pre-processor command

#include ->

include is one of the most popular pre processor command

jo kisi external file ka content ko hamare program me include karwana

we can write it in two ways

1. #include<file\_name>
2. #include “file\_name”

ex-

#include<stdio.h>

toh es ke jariye stdio.h wali jo alag file hogi us ke pure content ko hamare program me copy kr dega

jaise hamane program ko do part me divide kr diya hai or hum janate hai dono mil kr ke hi program complete karege

so hum

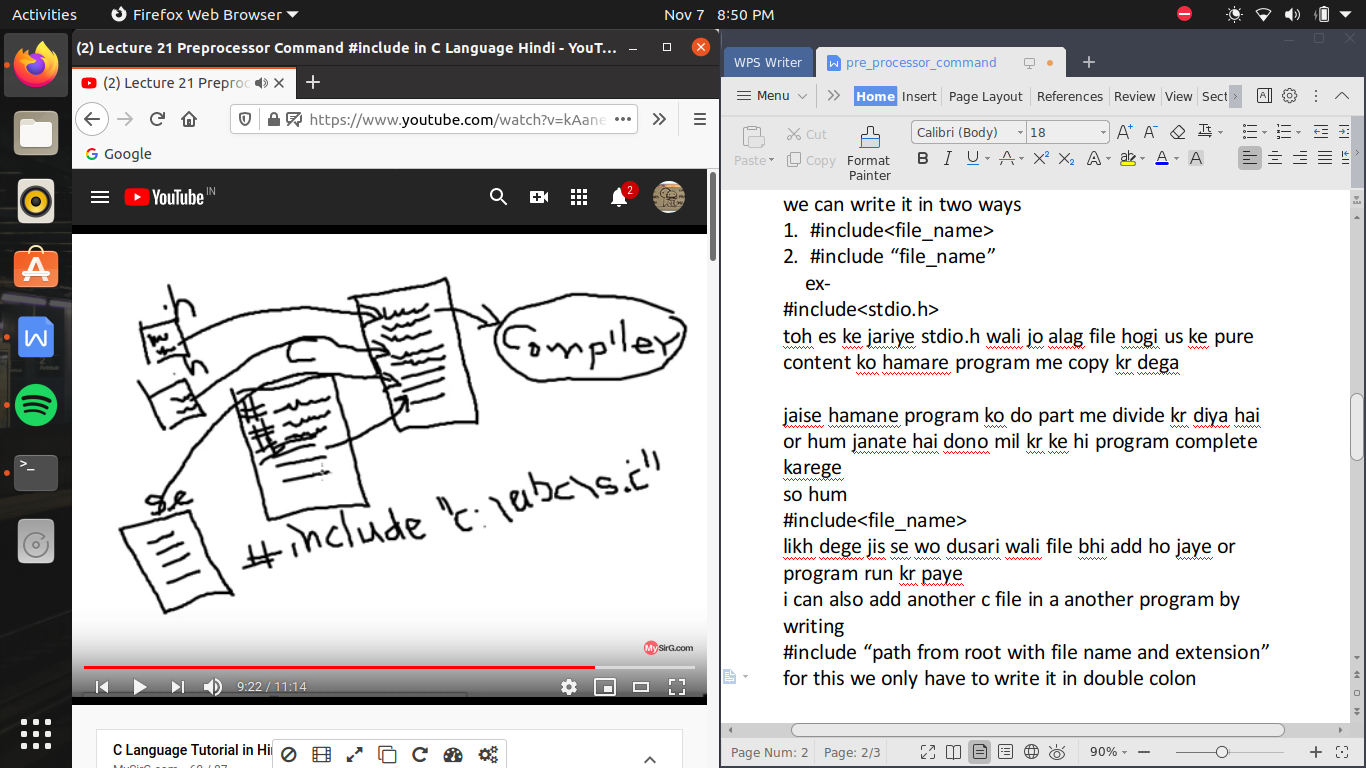
#include<file\_name>

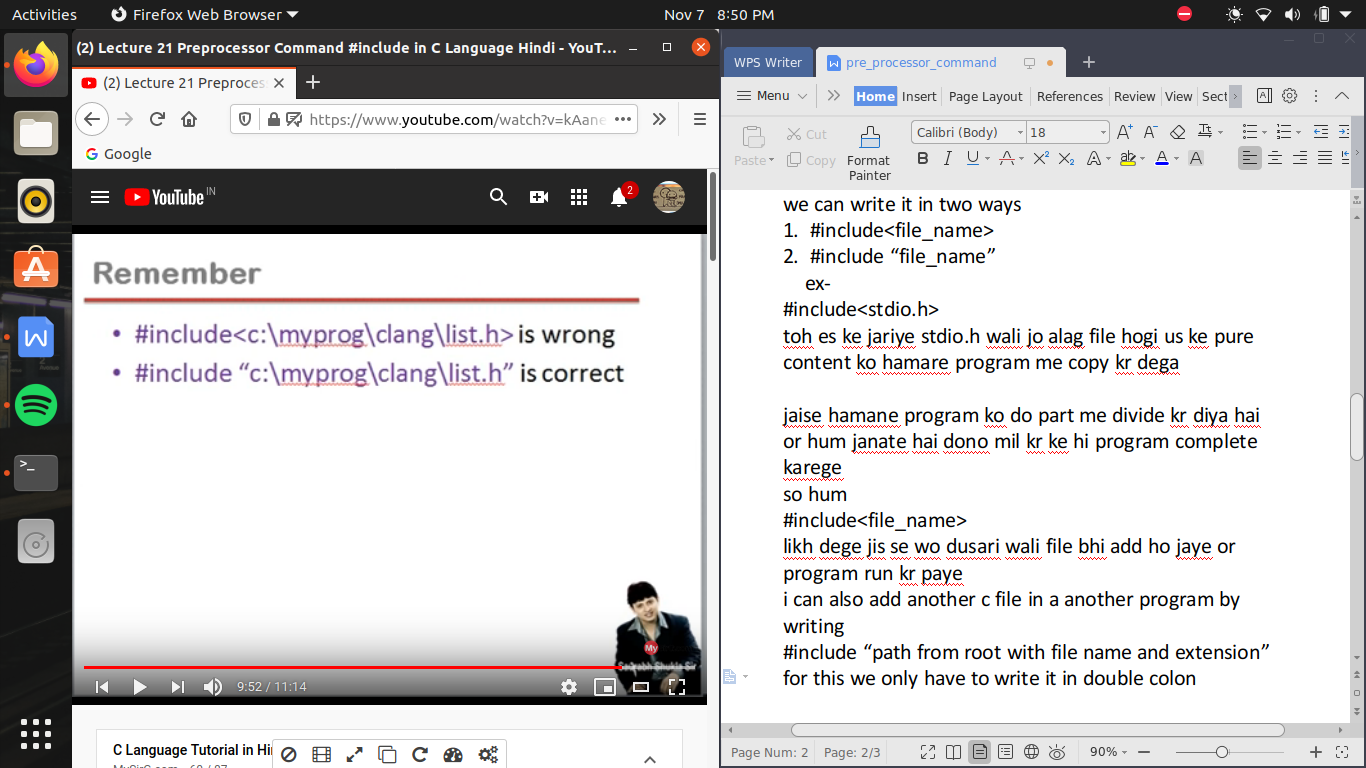
likh dege jis se wo dusari wali file bhi add ho jaye or program run kr paye

i can also add another c file in a another program by writing

#include “path from root with file name and extension”

for this we only have to write it in double colon





inclusion of another source file ->

we can write our source code in multiple.c files

we can include all these source files in the same way as we include header files

#define ->

#defined ke turant bad ek word likhate hai jise bolate hai macro ,es me kuch bhi likh sakate hai toh

for ex -

#define PI 3.14 or

#defined pi 3.14

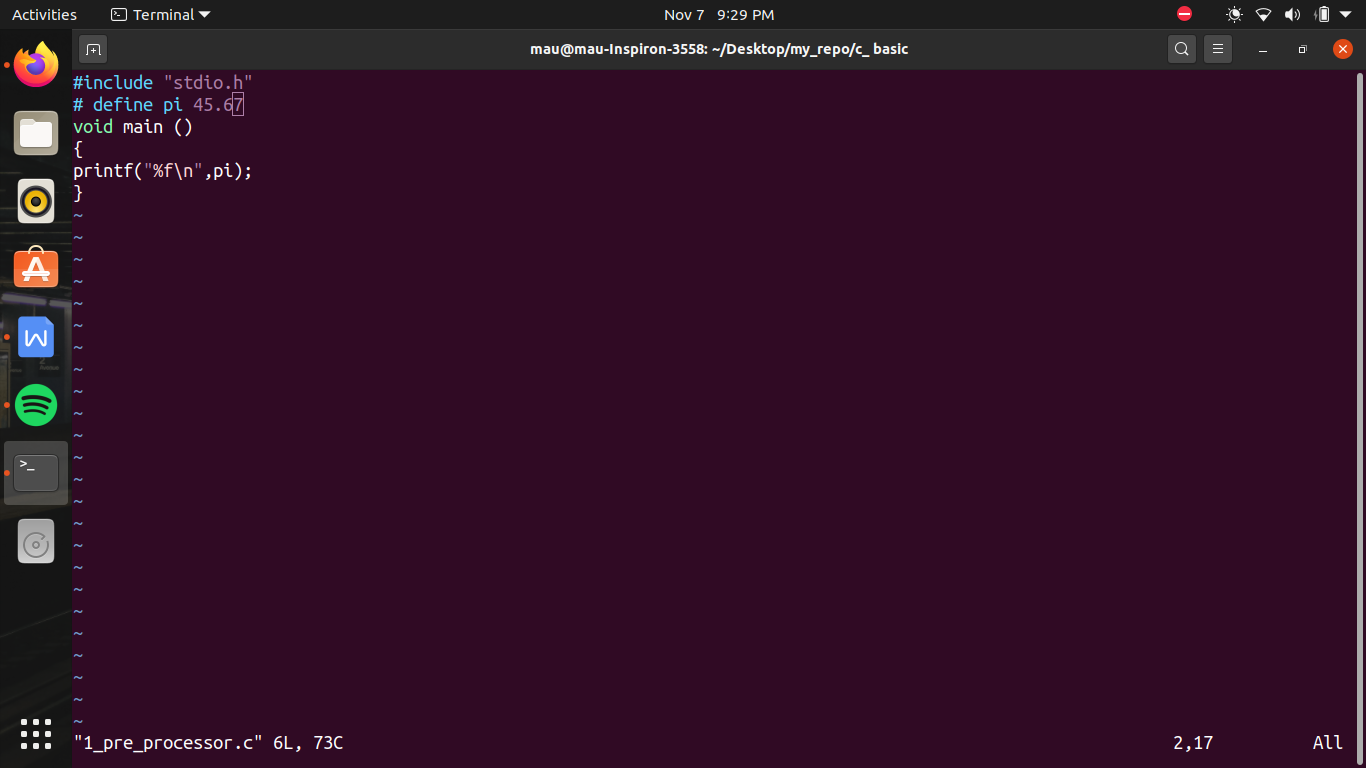
syntax->

#define marco\_name char-sequence

ya toh upper case or lower case dono me likh sakate hai or us ke bad us ka replace character sequence likh diya hai jaise 3.14 here ab pure program me kahi pr bhi likha hoga pi toh pre-processor kya karega pahale toh # wali line padega fir use samjh a jayega ki

pi matalb 3.14 hai toh jaha pi milega toh wo us ko replace kr ke kr dega 3.14

toh compiler jab file ko dhek raha hoga toh wo pi ko kahi nhi dekhega us ko waha 3.14 milega



defining macro like a function ->

#include "stdio.h"

#define SUM(a,b) a+b //a time of making a function we can also write an argument pr argument me koe data type nhi hoga and the character sequence which is going to be replace by is a+b

void main()

{

printf("sum of 3 and 4 is %d \n", SUM(3,4)); //here 3 or 4 takes the place of a,b respectively and replace by a+b i.e 3+4

//SUM(3,4) koe function nhi hai yeh toh bs deekhane me lag raha hai kyu ki compiler se pahale yeh replace ho jayega 3+4 me

}

/\*

# define pi 45.67

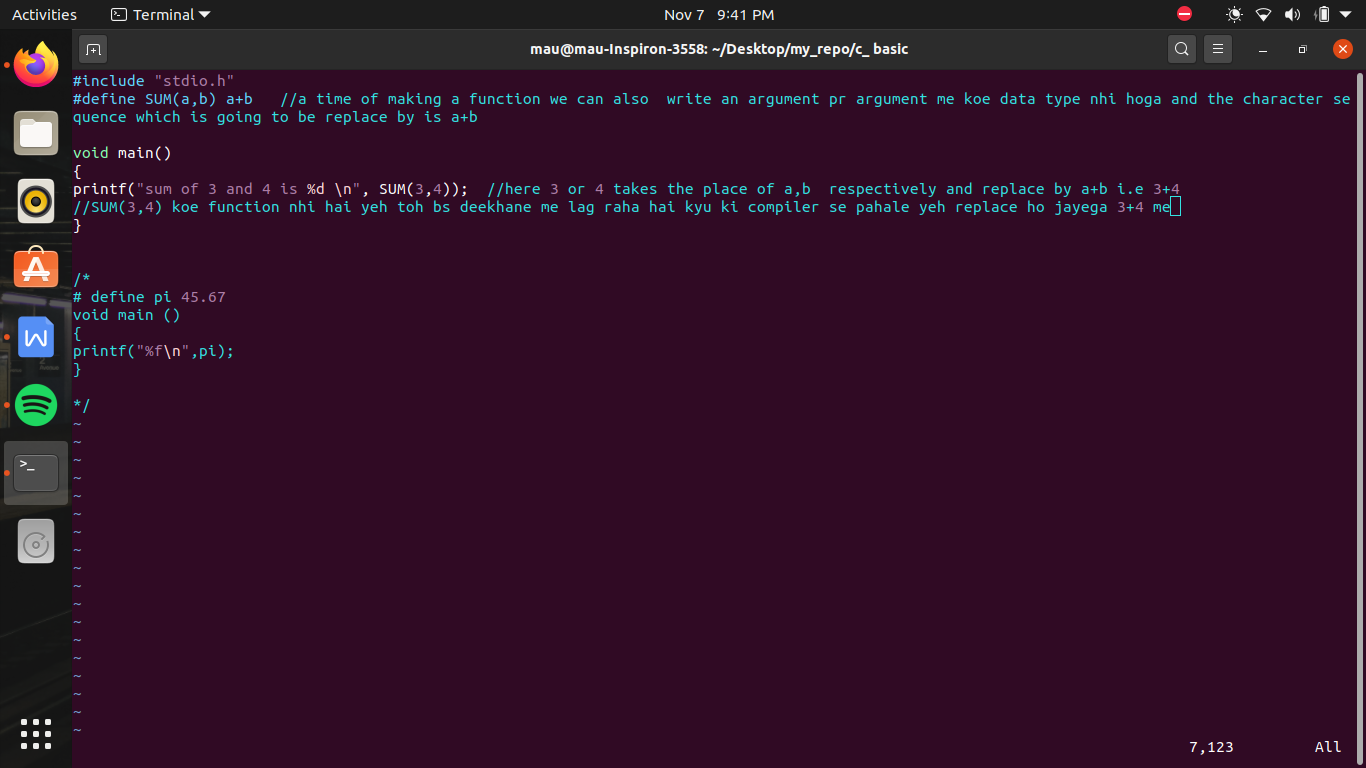
void main ()

{

printf("%f\n",pi);

}

\*/



#include "stdio.h"

#define product(a,b) a\*b

void main()

{

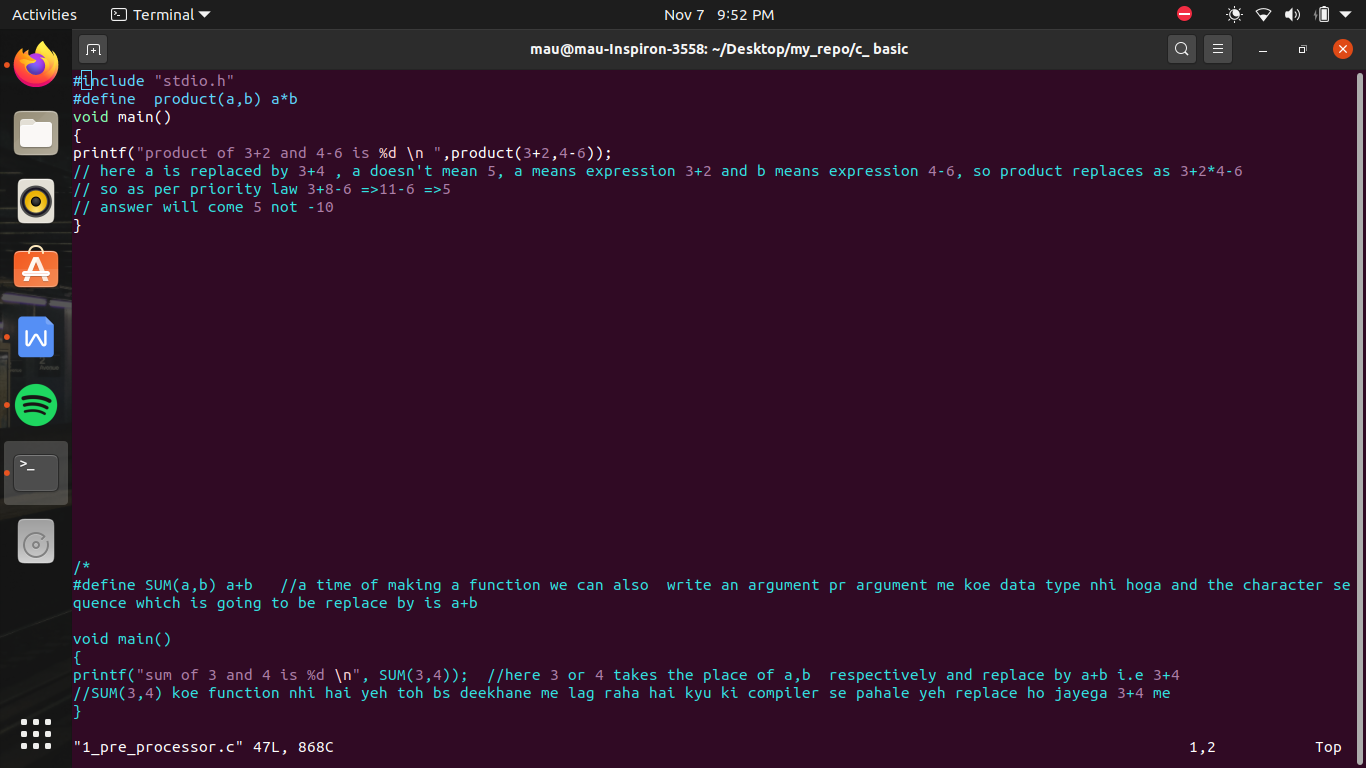
printf("product of 3+2 and 4-6 is %d \n ",product(3+2,4-6));

// here a is replaced by 3+4 , a doesn't mean 5, a means expression 3+2 and b means expression 4-6, so product replaces as 3+2\*4-6

// so as per priority law 3+8-6 =>11-6 =>5

// answer will come 5 not -10

}



#undef() ->

jab hum koe #define ki madat se macro banate hai or jab hame lagata hai ki hame ab es macro ki jarurat nhi hai toh hum

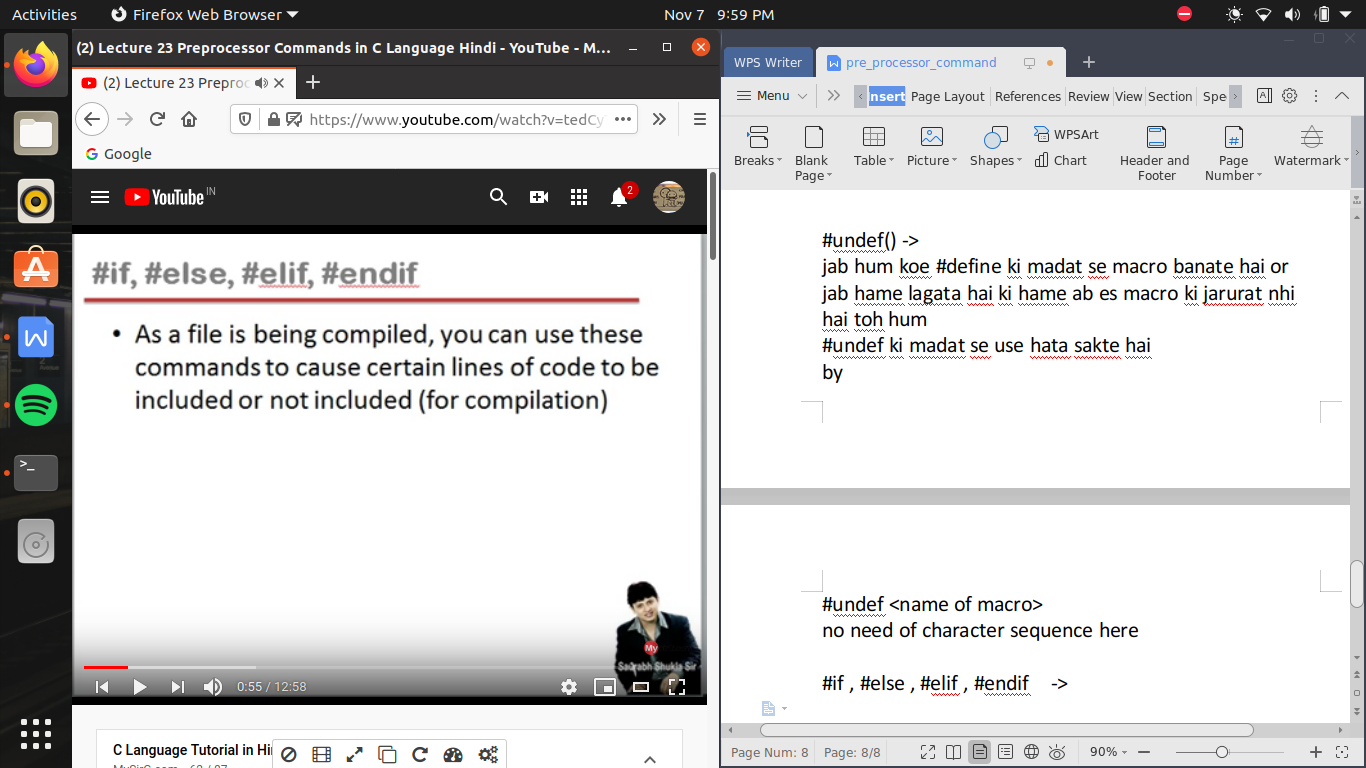
#undef ki madat se use hata sakte hai

by

#undef <name of macro>

no need of character sequence here

#if , #else , #elif , #endif ->



#if.......

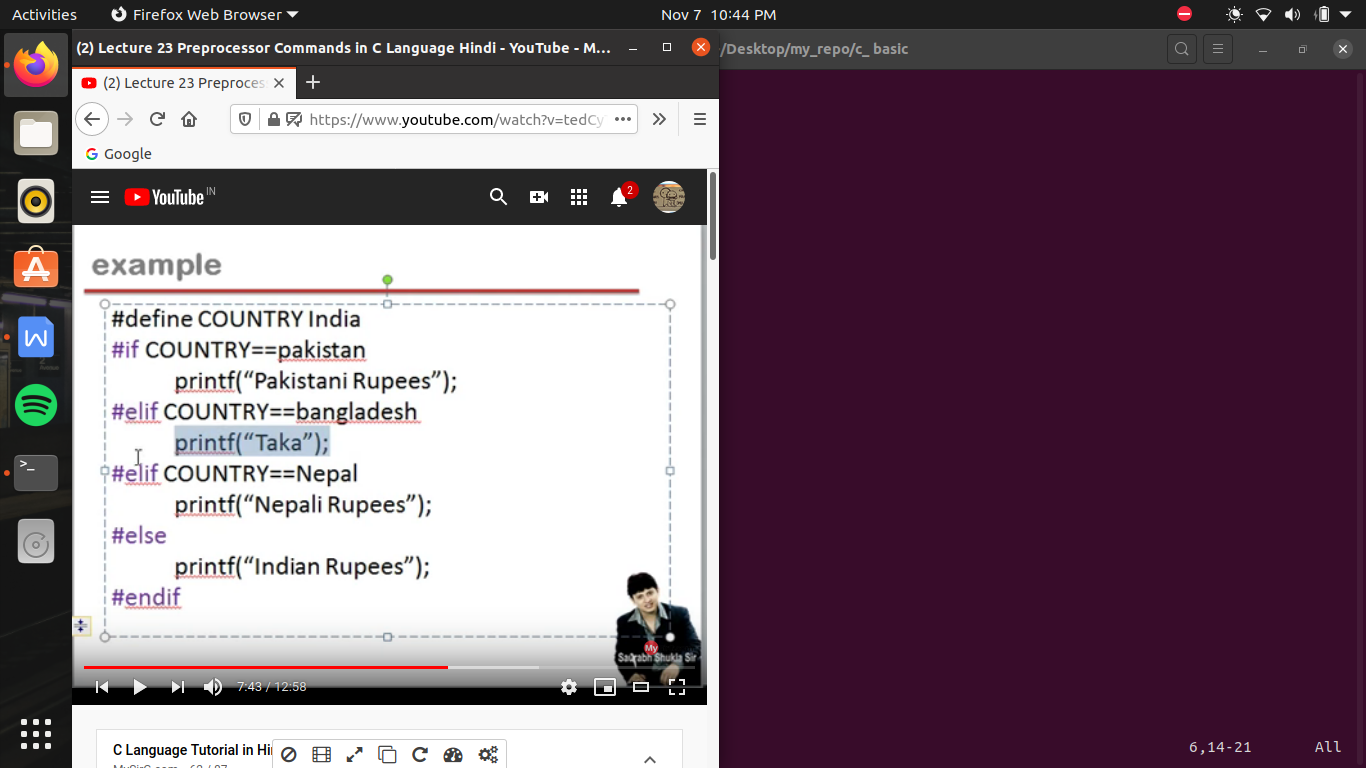
....

....

...

#endif

#if command preprocessor ke leye hoti hai yani ki #if ke age wali agar condition sahi hogi toh #if and #endif ke beech me likha code compilation ke leye eligible hai agar condition false hoti toh us ke beech wala code compiler ke leye hide ho jayega , yani compiler ko beech wala code deekhega ya nhi yeh depend karega ki #if condition sahi hai ya nhi



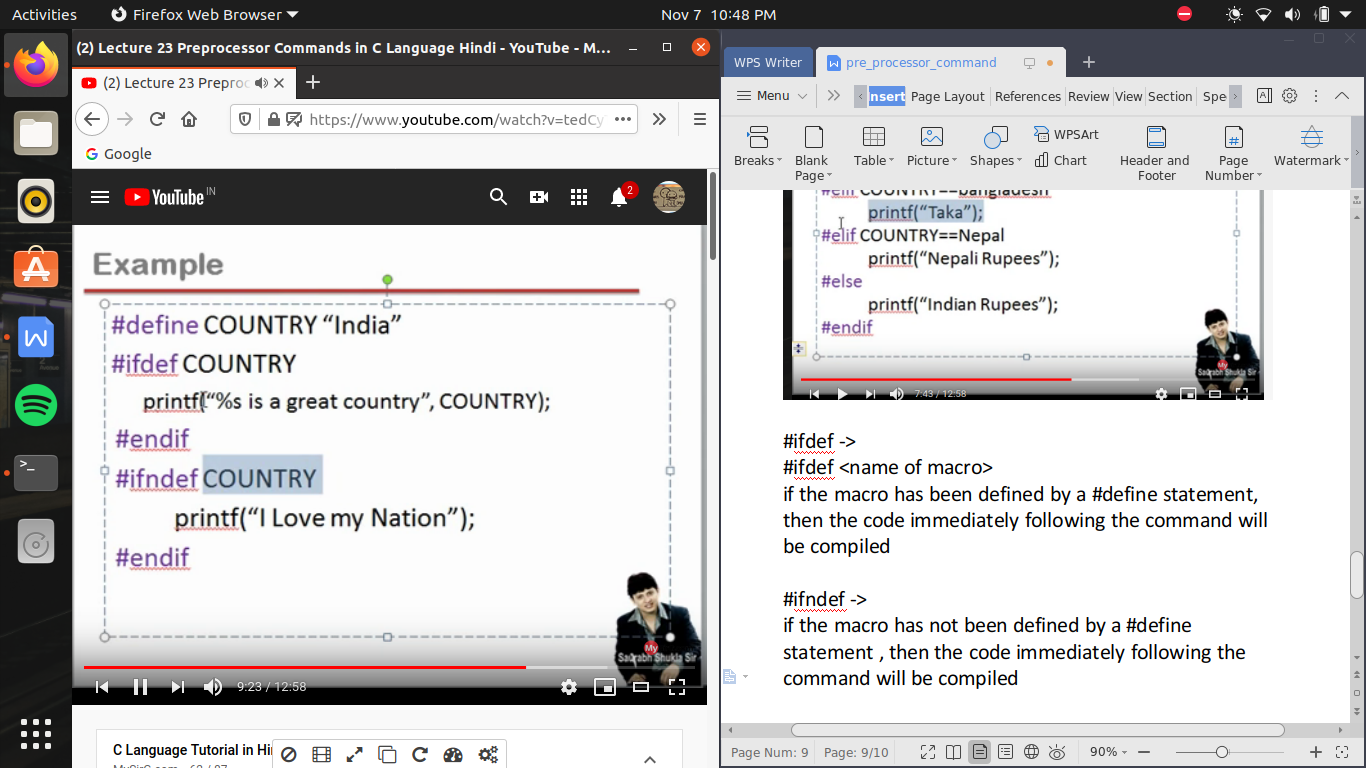
#ifdef ->

#ifdef <name of macro>

if the macro has been defined by a #define statement, then the code immediately following the command will be compiled

#ifndef ->

if the macro has not been defined by a #define statement , then the code immediately following the command will be compiled



## ->

the ## operator is used with the #define macro.

using ## concatenates(link togeather) what,s before the ## with what’s after it.

ex->

#include "stdio.h"

#define action(a,b) a##b+a\*b

void main()

{

printf("%d \n",action(3,4));

//here a means 3 and b means 4 ,, and a##b means 34 ,so a##b+a\*b =>34+3\*4 => 34+12 =>46

}

