file handling in c++

data persistence ->

data persistence ka mtlb hota hai data ka existence

yani life of the data , hum abhi tak program me data , variable ke rakhate hai

toh man le ki hame apane program me kuch data rakhana hai

toh abhi situation kuch aisi hai , ki hamara ek program hai jo run hua , toh ram me aya , toh ram me kuch memory mili usi ke andar kuch variables hai jis ke andar data rakha hua hai

pr yeh data program me kab tak safe hai , may be jab tak yeh program chl raha hai jayda se jayada , or agar program agar end hoga toh yeh variable ho end ho hi jayega,

toh yeh variables ki life , program ki life se jayada nhi ho sakati , toh kayi bar hame data ko variable me store karana hota hai ki , us data pe hame kuch operation perform karane hai , mtlb kayi sari line usme likhani hai

jis me hame us data ko use karana hai , toh hum us data ko kisi variable store kr ke jaha , use karana hota hai hum kr sakate hai

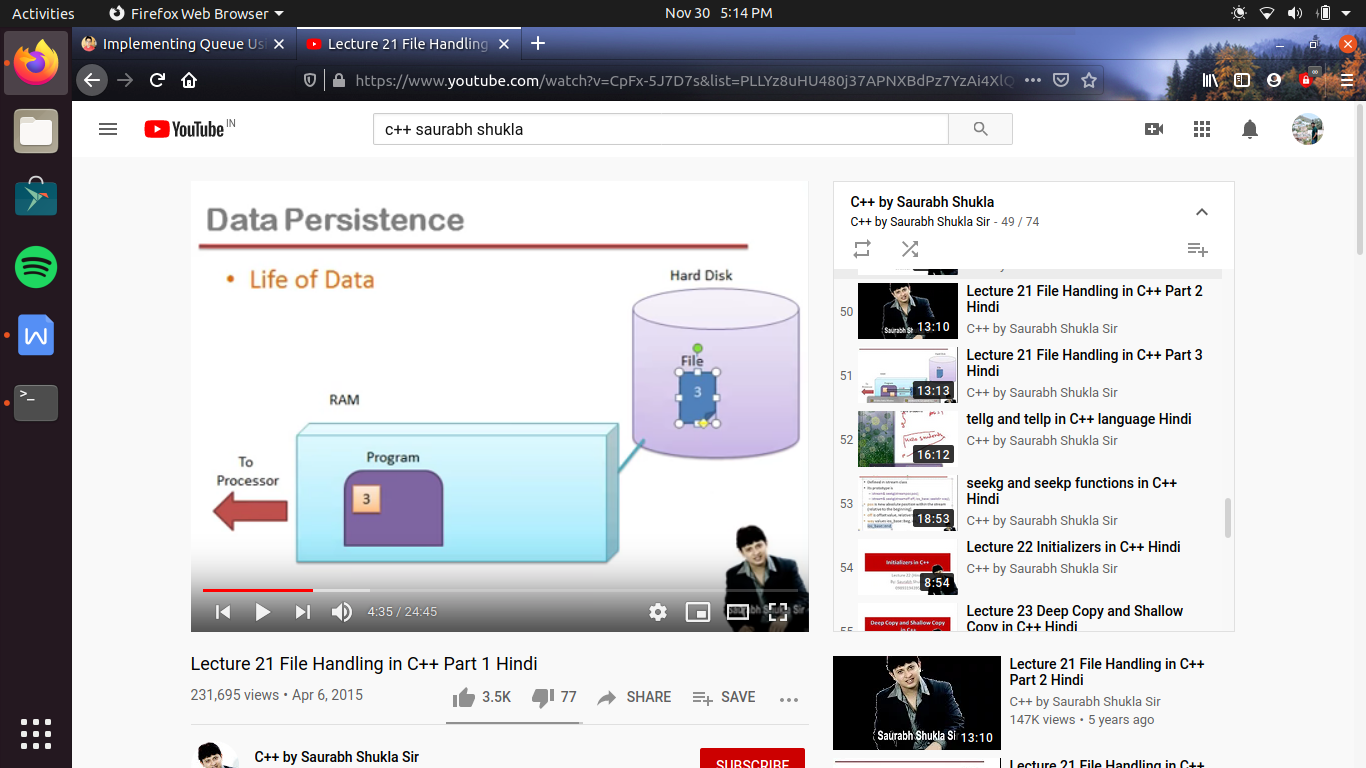
or agar hum chahate hai ki hamara yeh data program end hone ke bad bhi persist kare , toh hame wo program end hone se pahale us data ko permanent storage device me store karana hoga , pr secondary device me jo data store hota hai wo file ko roop me hota hai , toh hum chahate hai ki jo bhi data variable me rakha hai wo store ho jaye file ke andar , toh agar bad me program end bhi ho jata hai toh bhi file me data store hai

toh agar hum chahate hai ki jab bhi yeh program dobara run kare toh wo jo data store hai , ab dobara use data ko use karana hai

toh hamare program me aisa bhi code likha hoga ki hard disk me rakha data wapas hamari program me a sake

toh hame program me aisi coding karani hogi ki program se hard disk me data store ho , or use data ko wapas program me le kr use kiya ja sake

yeh dono kam karane ke leye hi hum bolate hai file handling



stream ->

jab bhi hamare program ko run hona hota hai wo ram me ata hai , ram me usko space milati hai , kuch variable banaye hote hai us program ke andar , toh ab hame kya karana hai es program ke andar , variable me store data ko file me store karana hai

toh file me rakhane ke leye pahale file ko open karana padata hai , mtlb koe funtion ki help se file ko call karege toh wo file ram me a jayegi , ab hame kya karana hai ki es variable ka data file ke andar pahuchana hai , toh es ke lye hame chahiye ke channel jis ke through hum data program se file me transfer kar sake ,

toh yeh sarika important nhi hai kyu ki wo tarika pahale se bana hua hai hame bs es tarike ko estamal karna hai

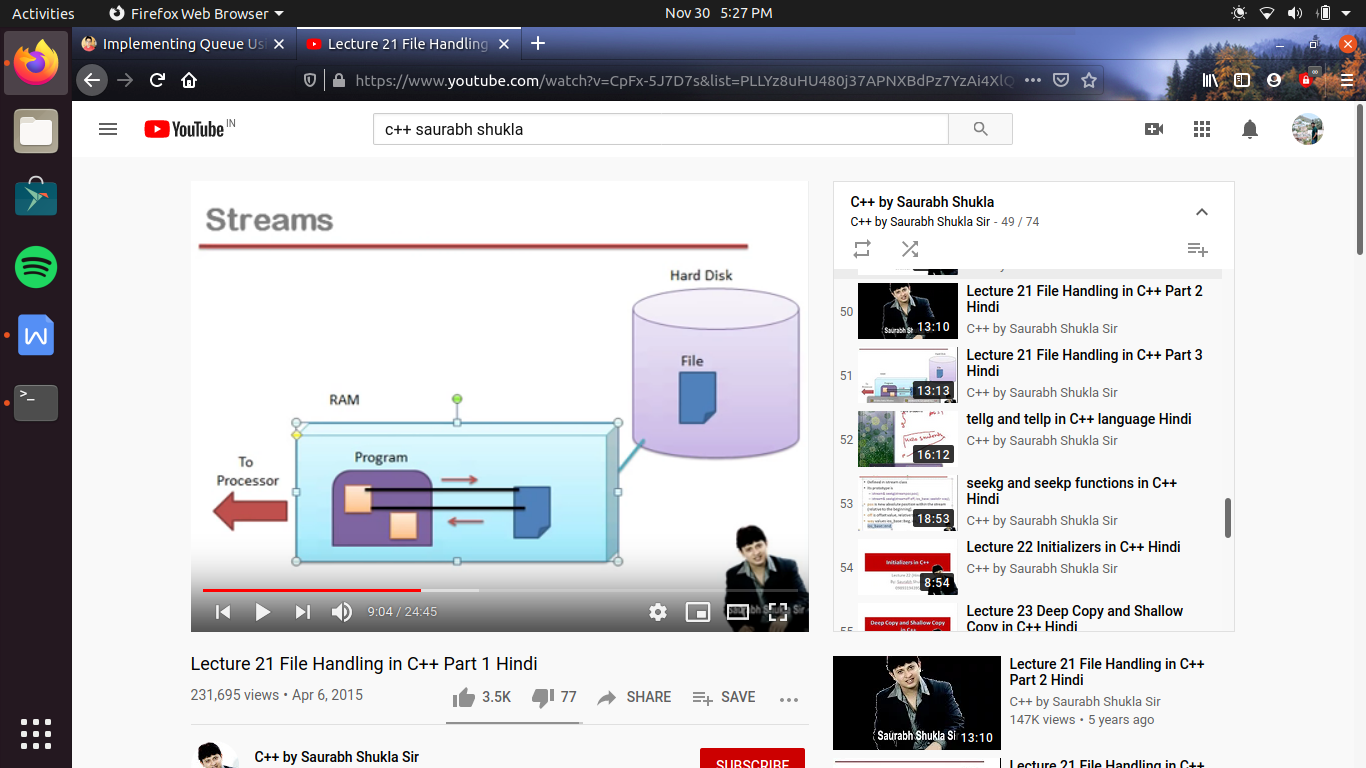
toh jo data bahar ja raha hai use stream kahate hai, or yeh kis direction me ja rahi hai, hamare program ke bahar toh ese bolate hai output stream , yani ab ek or chij hogi jis ko hum bolege input stream

toh input stream wo hota hai jis ki help se file ka data , program me jata hai

toh hame ek function chahiye jo output stream ko manage kare or ek dusara function chahiye jo input stream ko manage kare

jo ki c++ me pahale se hi aisi classes hai , jin ke object bana kr hum , input or output stream ko kr skate hai represent

ab agar yeh object bhi toh class bhi hogi , pr yeh class hamane nhi bane , jo ki hogi predefined



waise toh stream se related bahot sari classes hoti hai , pr hum un me se kuch important samjh rahe hai

es me ek class hoti hai jo es pure hirarcy me sabse upar hoti hai , i.e parent most class hai , jis ka nam hai

ios (i.e input output stream)



es me istream or ostream ki ek common child class hai iostream

or jis ki bhi ek child class hai fstream

ab file handling me hame 3 classe ki jarurat hai

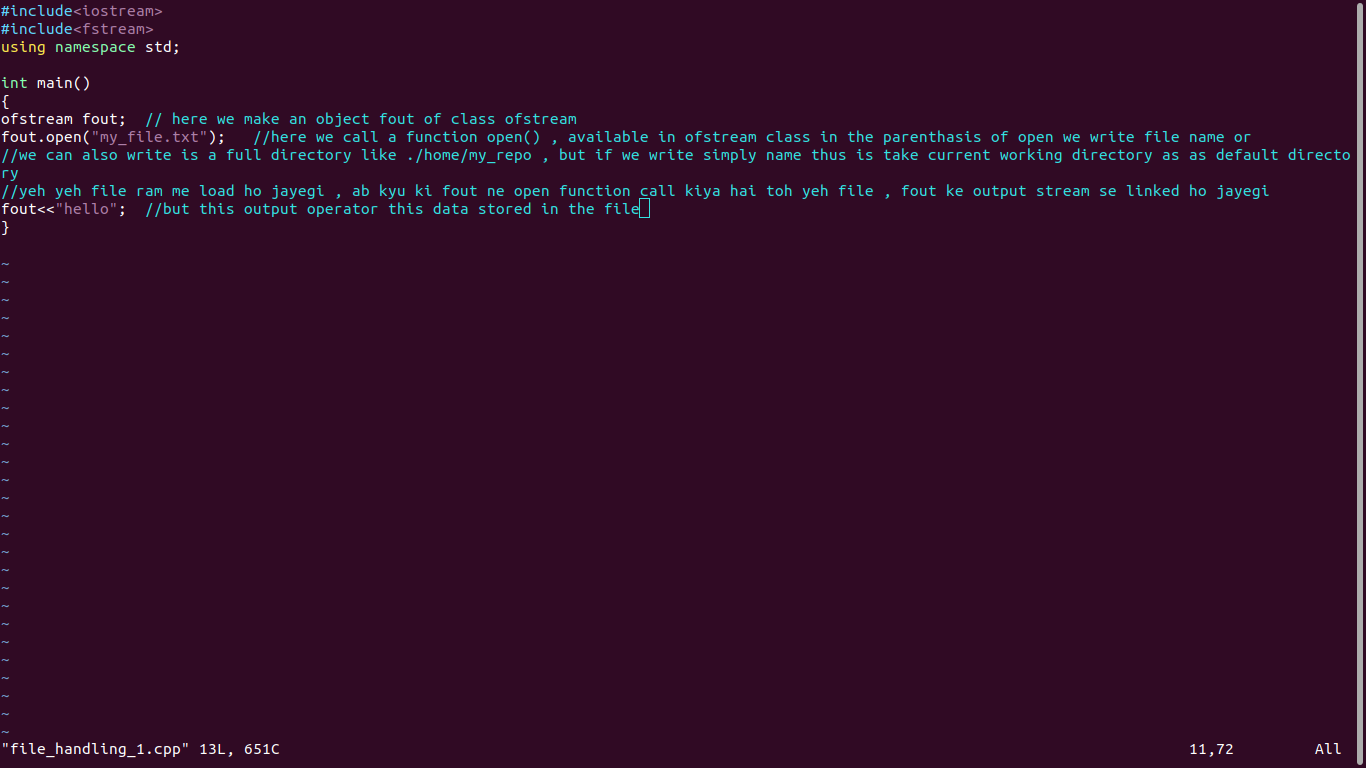
ifstream-> agra sirf input lena hai toh

fstream -> agar hame input or output dono karana hai toh

ofstream ->aagr sirf writing karana chahate hai toh hame ofstream ka object banana padega

toh pahale hum output stream banate hai fir file ko open karate hai

how to use output stream



reading a file

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ifstream fin; //hamane yaha pr ek upstream object banaya fin nam ka i.e input stream ban chuki hai

fin.open("my\_file.txt"); //yaha pr jo open ho kr , input stream se link ho chuki hai

//ab hame reading karani hai

/\*

int a;

cin>>a; //ab hum jab keyboad sedata read karate hai toh hum cin ki help lete hai , ese se yeh hota hai ki ek value enter kaarate hai hum keyboard se or wo a me store ho jata hai

//yeh value keyword se hi kyu ata hai kyu ki cin keyword keyboad ke sath associated hai

//toh jab yeh object ho jo input stream ko keyboad se associated karata hai toh data keyboard se lega

//or ese input stream se hote hue kaha se laya jayega , input stream me laya jayega

//ab hamane kya kiya hai usi class ka ek object banaya hai ,bus difference bs etana hai ki cin keyboad se associated hai pr fin file se associated hai

\*/

char ch;

ch =fin.get();

// fin>>ch;

//ab ese se kya hoga kyu ki fin file se related hai toh jo pahala character hoga file ne , wo ch me a jayega

// toh hum kya kare kyu ki hae ek character nhi kayi sare charecter read karane hai file se

// toh ese ke leye hum loop lagayege jo tab tak chale jab tak file ka end na a jaye toh ese ke leye hum likege (fin.eof())

//eof() ek function hai jis ka matlb hai end of file , yeh function pata lagata hai ki file ka end aya kya or agar end a jata hai toh yeh return kr deha 1 or jab file ka end nhi ayega toh yeh return kr dega 0;

//toh hum chahate hai ki yeh loop tab tak chale jab tak file ka end nhi yaha ho toh es ke leye hum ! (not ) laga dete hai condition me

while(!fin.eof())

{

cout<<ch;

ch=fin.get(); //ab ch me aagala charecter a jayega

}

fin.close();

printf("\n");

}

//ab yaha pr jo extraction operator hai >>jis ki help hum fin se file ka data ch me la rahe hai yeh space ko ek delimiter manata hai i.e data saperator manata hai

//yani space , tab or next linea gaye toh yeh en thino ko ek aisa special charecter manata hai ki , ki agar yeh data a gaye toh yeh data nhi hai, balaki data ko saperate karane wale special symbols hai

//toh es ko toh es ne read hi nhi kiya , ek tarah se un se yeh la kr ch me raha hi nhi hai toh , jo chij yeh laya hi nhi to show kaise karega

//ese wajah se space show nhi ho raha hai

//ab es ka solution bhi hai pr pahale hum samajhate hai ki fin>>ch; es ka mtlb kya hai

// fin.opearator >>(ch); i.e fin ne apane extraction operator ko call kiya tha or us me ch ka reference pass kiya tha

// toh aise jiase ki yeh extraction operator space ko data saperator manata hai toh kyu na hum ese call hi na kare

// toh es ki jagaha kyu na hum get() function ko call kare jo return karege us charecter ko jo us ne kiya hai read

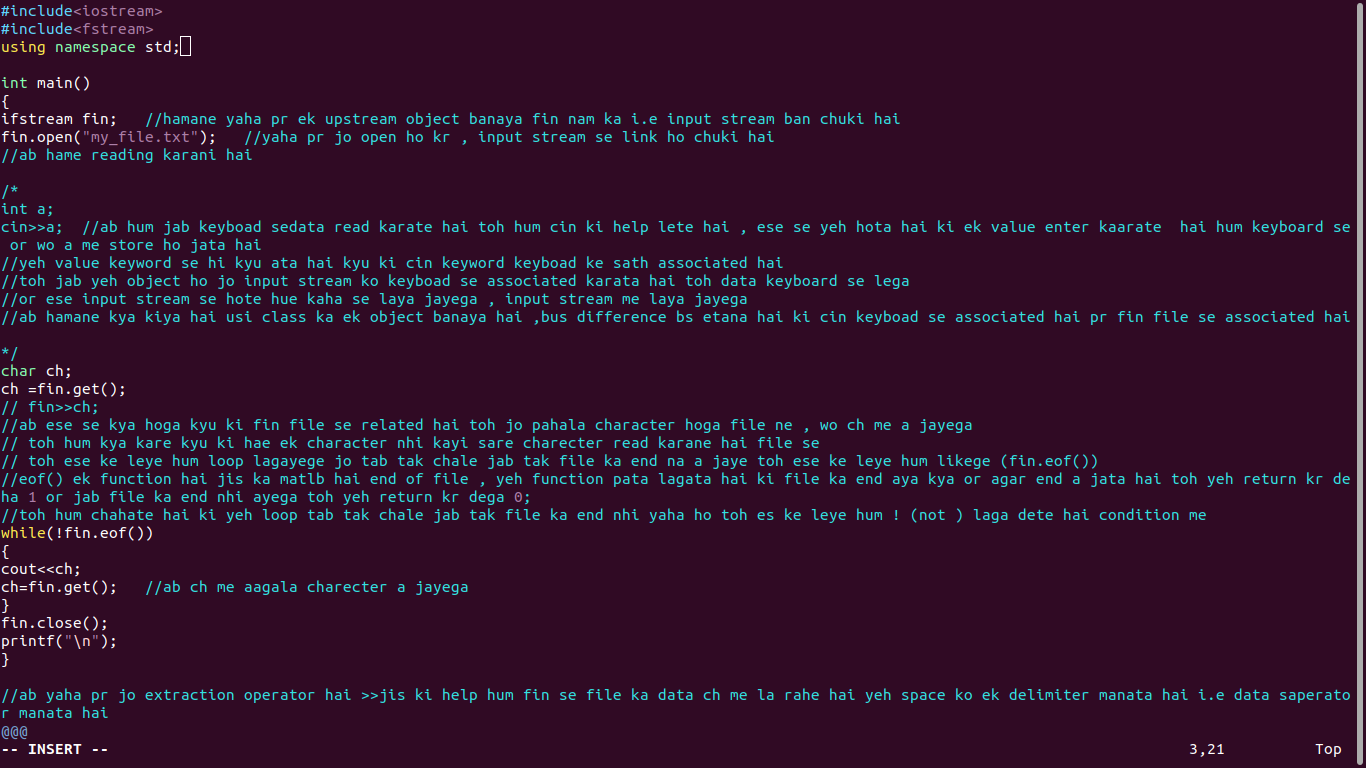
//fayada yeh hai ki get() function space ko data saperator nhi manata

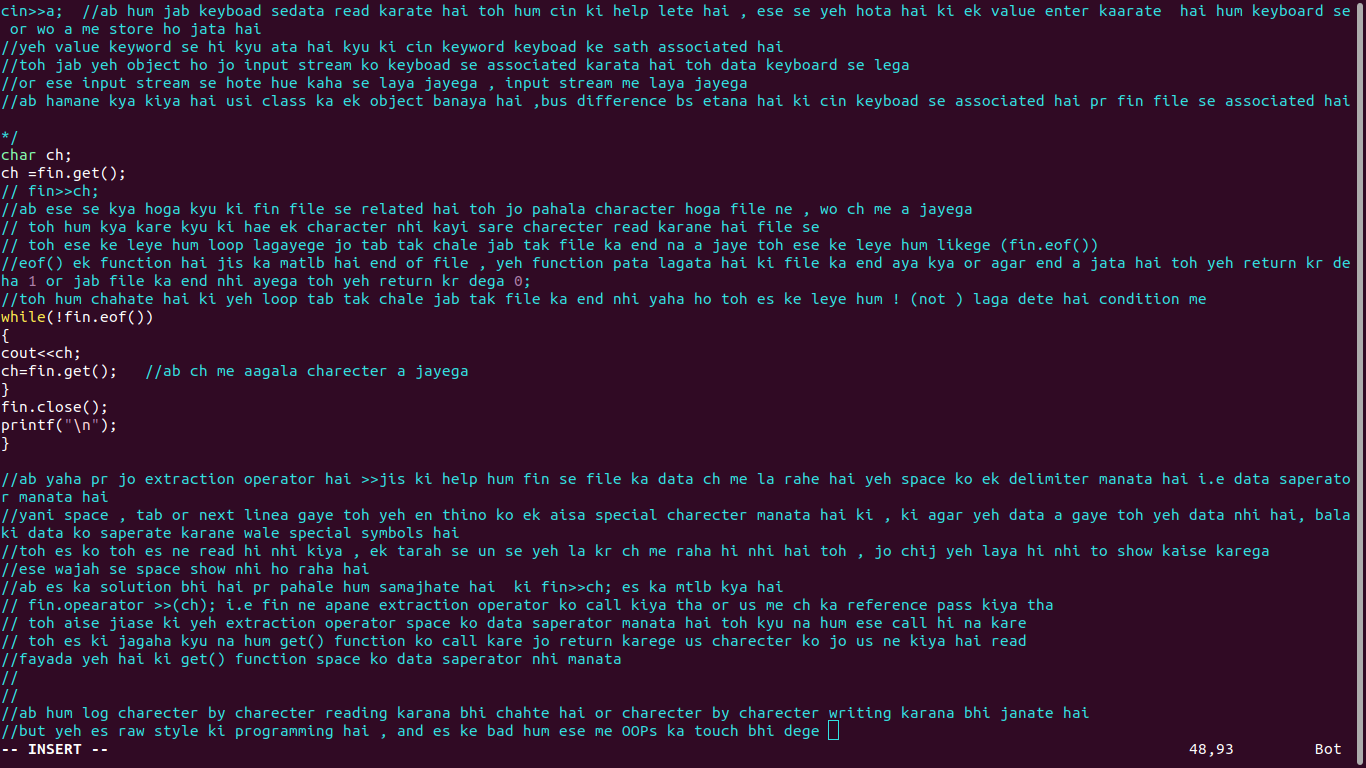
//

//

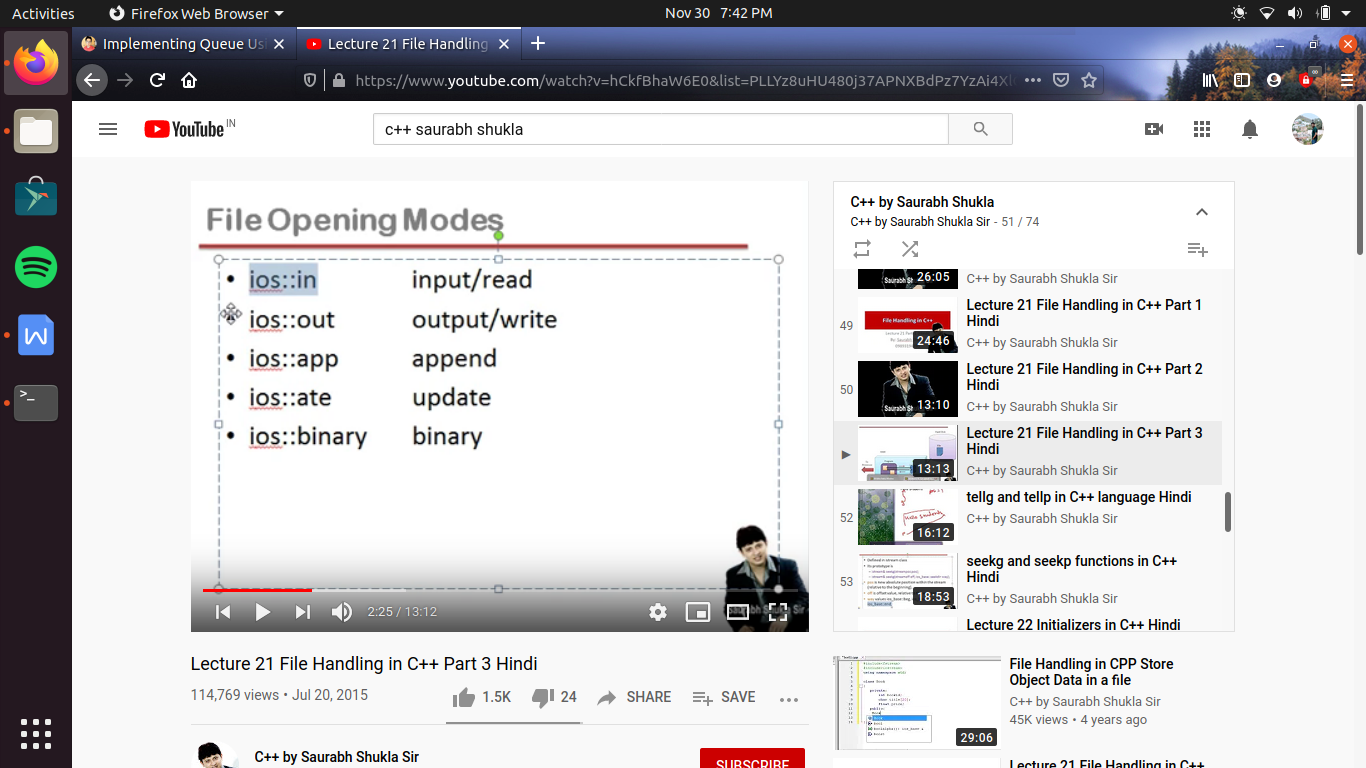
//ab hum log charecter by charecter reading karana bhi chahte hai or charecter by charecter writing karana bhi janate hai

//but yeh es raw style ki programming hai , and es ke bad hum ese me OOPs ka touch bhi dege





file opening modes ->



yeh as an argument pass karane padate hai hame open() function me

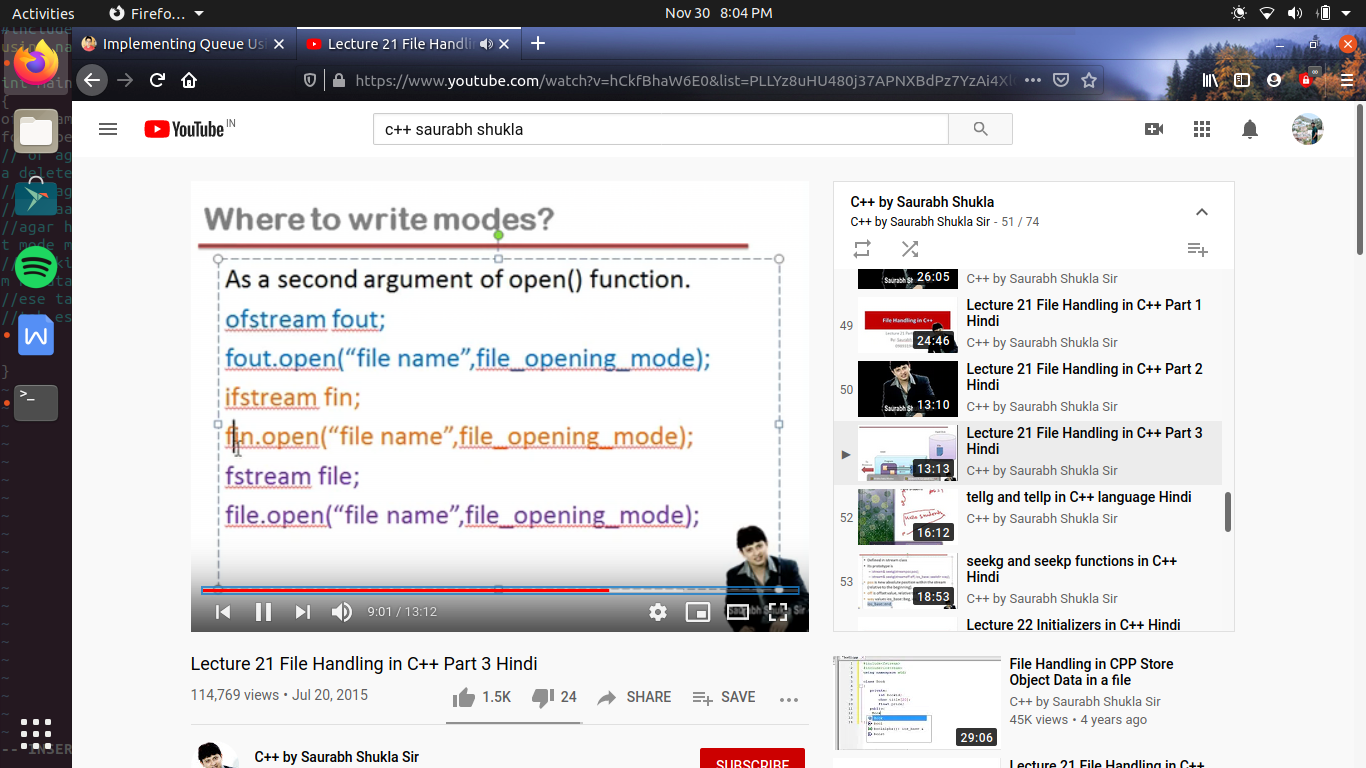
ios::in -> ios scope resolution in ->yeh ek contant value hai jis se hum , file ko open kr rahe hai read karane ke leye

ios::out -> for writing pr dhayan me yeh rakha jaye hi jab bhi hum file ko open karege es mode me toh , file ka purana content erase ho jata hai , or fir naye tarike se file edit hoti hai

ios::app -> (append)agar hum chahate hai ki file ka purana content erase na ho balaki jo pahale se store hai use age edit karana chahate hai

ios::ate -> (update ) jis ka mtlb hai file me hame randomly koe operation perform karana hai ya koe modification karana hai toh hum es mode ka use kare

ios::binary-> age padege



text mode v/s binary mode :-

jab hum file open karate time argument me ios::binary likh dete hai toh es ka mtlb hai ki wo file binary mode me open ho rahi hai

text mode me open karane ke leye bs hame yeh nhi likhana hai , agar hum ios:: binary nhi likhate hai toh es ka mtlb hai ki yeh file text mode me open hogi

if hame koe message ko print karwana hai let say

cout<<”my name is \nshashank”;

toh yeh output screen me print karata hai , es me

pahale my name is likh kr sayega fir line change ho jayegi

then next line me shashank likh kr ayega

es ka mtlb hai \n hai ya koe or operation en ke meaning consider hote hai

or jab output ata hai toh es meaning ka ashar deekahe deta hai

toh mtlb \n as it consider nhi hua es ka mtlb consider hua

ab yahi chij hum file me write kare toh kya hoga

fout<<”my name is \n shashank”;

fout likhana bhi ek output hi hai , bas jab cout likh rahe the screen me output a raha tha and jab fout likh rahe hai toh file me output a raha hai

lekin \n write hoga file me as it is , ya file me bhi line change ho jayegi

yeh depend kaarta hai ki hum ne file ko text mode me open kiya hai ya binary mode me

toh agar hum ne file ko binary mode me kiya hai open toh file me as it is print ho jata hai , but agar hum ne binary mode me open nhi kiya hai file ko toh \n ka mening consider hoga or file me bhi line change ho jayegi

#include<iostream>

#include<fstream>

using namespace std;

int main()

{

ofstream fout; //this object represent output stream

fout.open("my\_file.txt", ios::app);

fout<<"this is the second line \n this line i add by opening this file again in append mode an use operation to change the line"<<endl;

//fout.open("my\_file.txt", ios::app | ios ::binary ); //es me hum pahala argument file ka nam pass karate hai

// or agar hum es me dusara arguement pass nhi bhi karate hai toh file output mode me open ho jati hai, yani agar pahale se file hue toh us ka saradata delete ho jayega , or new file me file se data edit kar sakate hai

// pr agar chahe toh ese me 2nd argument ke roop me file opening mode mention kr sakte hai

// or aagr hum chahe toh ek se jayada mode mebhi es file ko open kr sakate hai

//agar hum ese aise likhate hai fout.open(".....", ios::app |ios::binary | ios::in); ab yaha hum bol rahe hai ki hum es file ko open kr rahe hai input mode me toh yeh logically galat hai , yaha pe koe error nhi ayegi kyu ki hum yaha pr koe bhi mode likh sakte hai

//kyu ki yaha pr read out mode likane ka mtlb hi nhi nikale kyu ki hum ne fout ek ofstream ka object banaya hai , jo ki sirf writting operation perform karata hai

//ese tarah se agar hum ifstream ka object banate hai or hum sochate hai ki ese ios:: out ya ios::app; toh es ka bhi koe meaning nhi nikalega

//toh es tarah se hume ek se jayada mode me file ko open kr sakate hai

}

