extended\_pointer

wild pointer ->

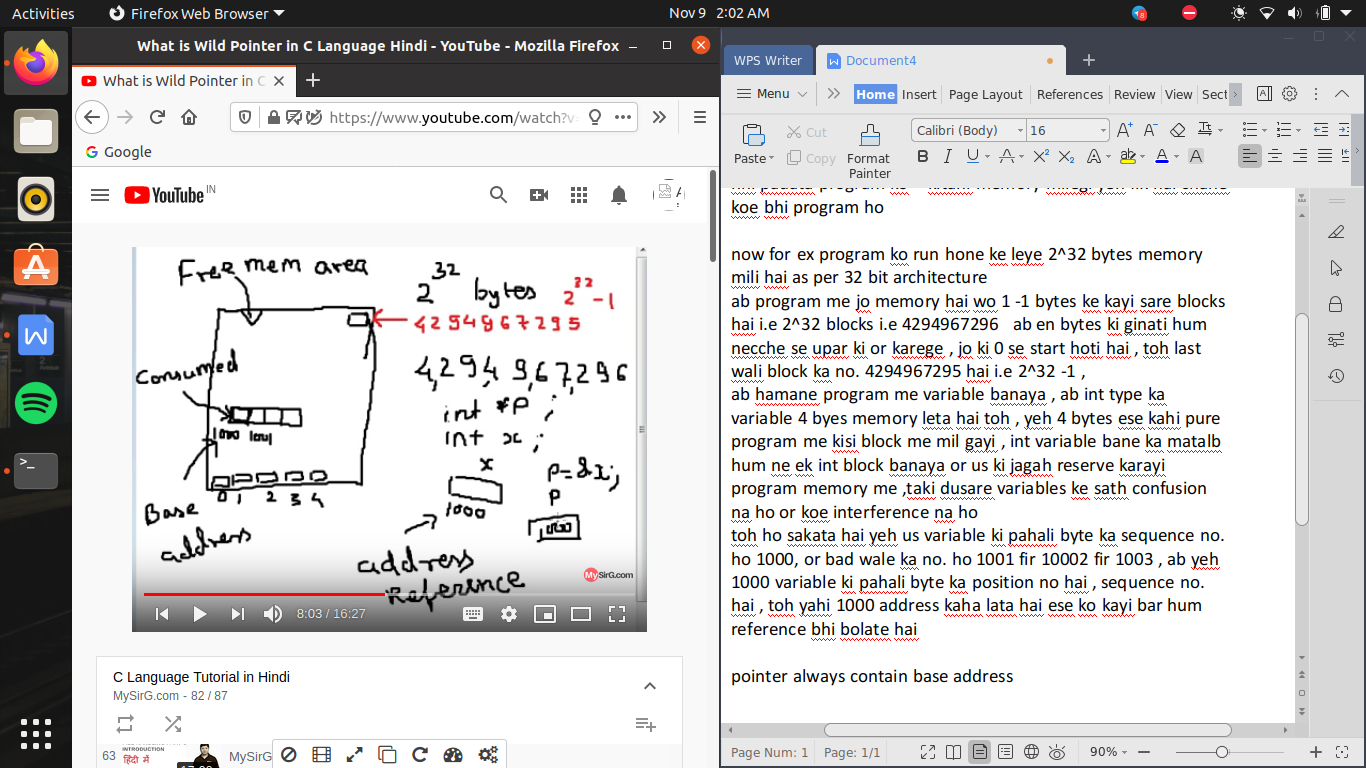
jab bhi koe program run hota hai use koe memory allocate hoti hai ab es program me memory kitani hai yeh architecture pe depend karta hai toh program kitana bada ho ya chota es se frk nhi padata program ko kitani memory milegi yeh fix hai chahe koe bhi program ho

now for ex program ko run hone ke leye 2^32 bytes memory mili hai as per 32 bit architecture

ab program me jo memory hai wo 1 -1 bytes ke kayi sare blocks hai i.e 2^32 blocks i.e 4294967296 ab en bytes ki ginati hum necche se upar ki or karege , jo ki 0 se start hoti hai , toh last wali block ka no. 4294967295 hai i.e 2^32 -1 ,

ab hamane program me variable banaya , ab int type ka variable 4 byes memory leta hai toh , yeh 4 bytes ese kahi pure program me kisi block me mil gayi , int variable bane ka matalb hum ne ek int block banaya or us ki jagah reserve karayi program memory me ,taki dusare variables ke sath confusion na ho or koe interference na ho

toh ho sakata hai yeh us variable ki pahali byte ka sequence no. ho 1000, or bad wale ka no. ho 1001 fir 10002 fir 1003 , ab yeh 1000 variable ki pahali byte ka position no hai , sequence no. hai , toh yahi 1000 address kaha lata hai ese ko kayi bar hum reference bhi bolate hai



pointer always contain base address

ab yeh program me jo memory hai wo 2 tarah ki hai ya toh free memory area or consumed memory area

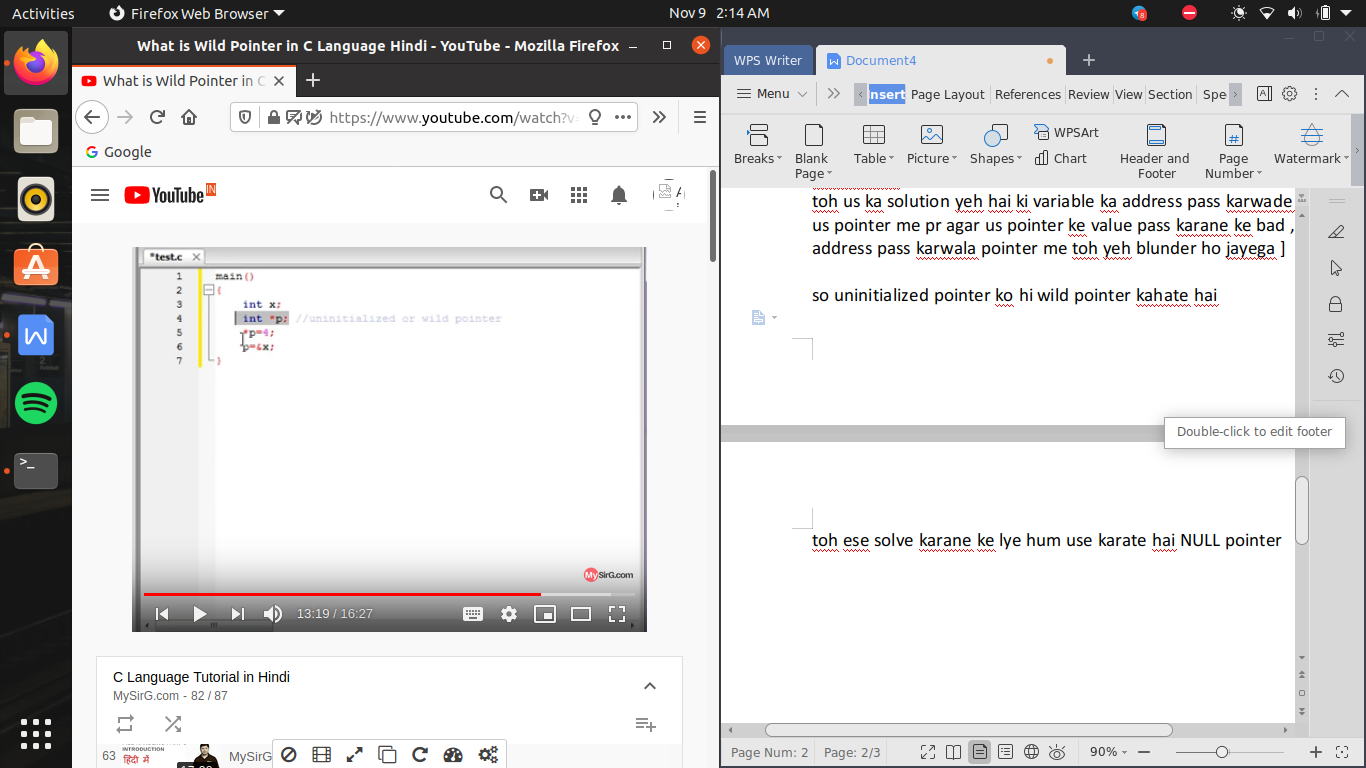
or yeh dono ka ratio badalata rahata hai jaise jaise program shuru or end hote jate hai

ab agar hum ne ek pointer banaya or us ko defining ke time initialize nhi kiya toh yeh pointer carry karege garbage value jo ki ek address hi hoga kisi block ka , or wo block khali hai bhara kuch bhi ho sakata hai

toh yeh pointer jis me garbage value rakhi ho toh use kahate hai wild pointer ab agar hamne galati se es pointer me jaha yeh point kar raha hai koe value pass karawa di toh ek unknown location me data access ho jayega or yeh illegal memory access kahalayega

toh us ka solution yeh hai ki variable ka address pass karwade us pointer me pr agar us pointer ke value pass karane ke bad , address pass karwala pointer me toh yeh blunder ho jayega ]

so uninitialized pointer ko hi wild pointer kahate hai



toh ese solve karane ke lye hum use karate hai NULL pointer

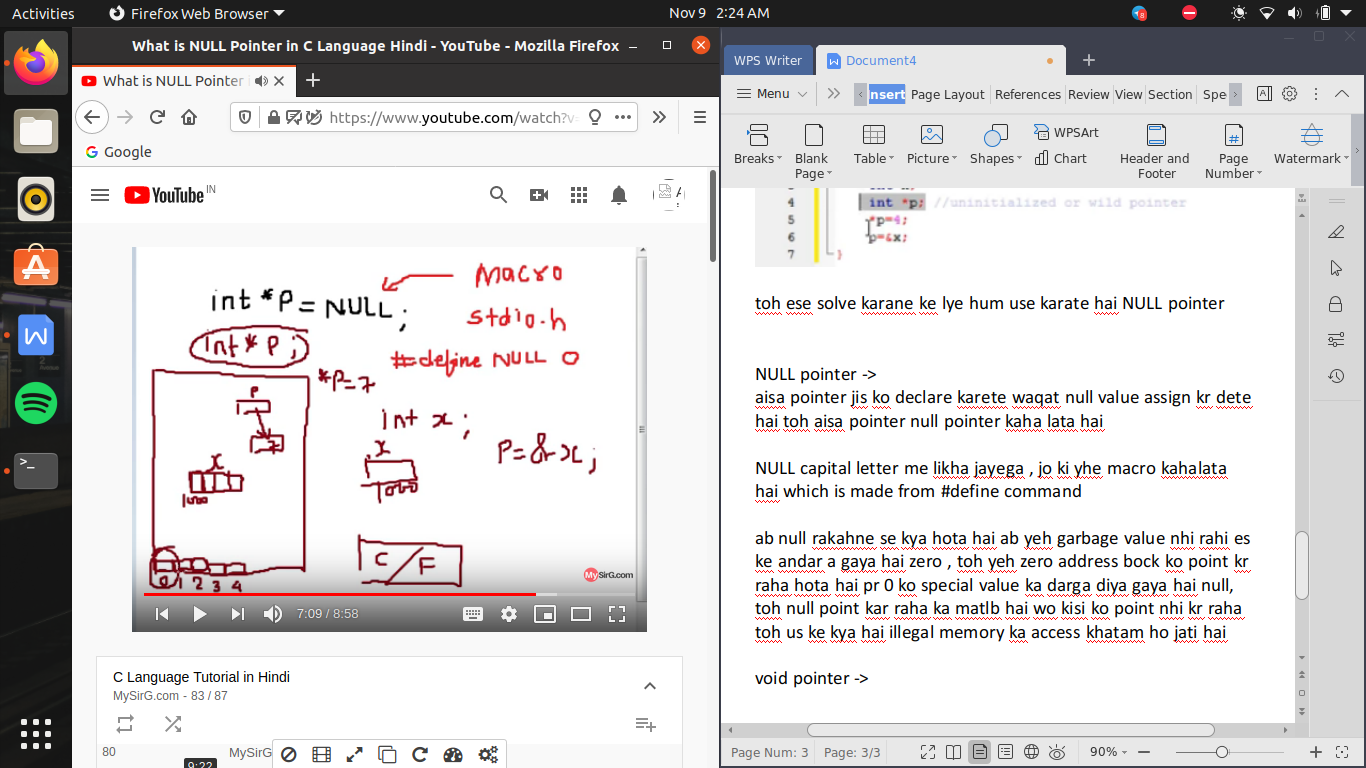
NULL pointer ->

aisa pointer jis ko declare karete waqat null value assign kr dete hai toh aisa pointer null pointer kaha lata hai

NULL capital letter me likha jayega , jo ki yhe macro kahalata hai which is made from #define command

ab null rakahne se kya hota hai ab yeh garbage value nhi rahi es ke andar a gaya hai zero , toh yeh zero address bock ko point kr raha hota hai pr 0 ko special value ka darga diya gaya hai null, toh null point kar raha ka matlb hai wo kisi ko point nhi kr raha

toh us ke kya hai illegal memory ka access khatam ho jati hai



void pointer ->

int pointer ->carry address of int type variable

float pointer -> carry address of float type variable

void pointer -> it can carry all type of pointer es ko data types se kuch mtlb nhi hai or ese hum generic pointer bhi kahate hai

let say

int a;

float b;

void \*p;

p=&a and p=&b

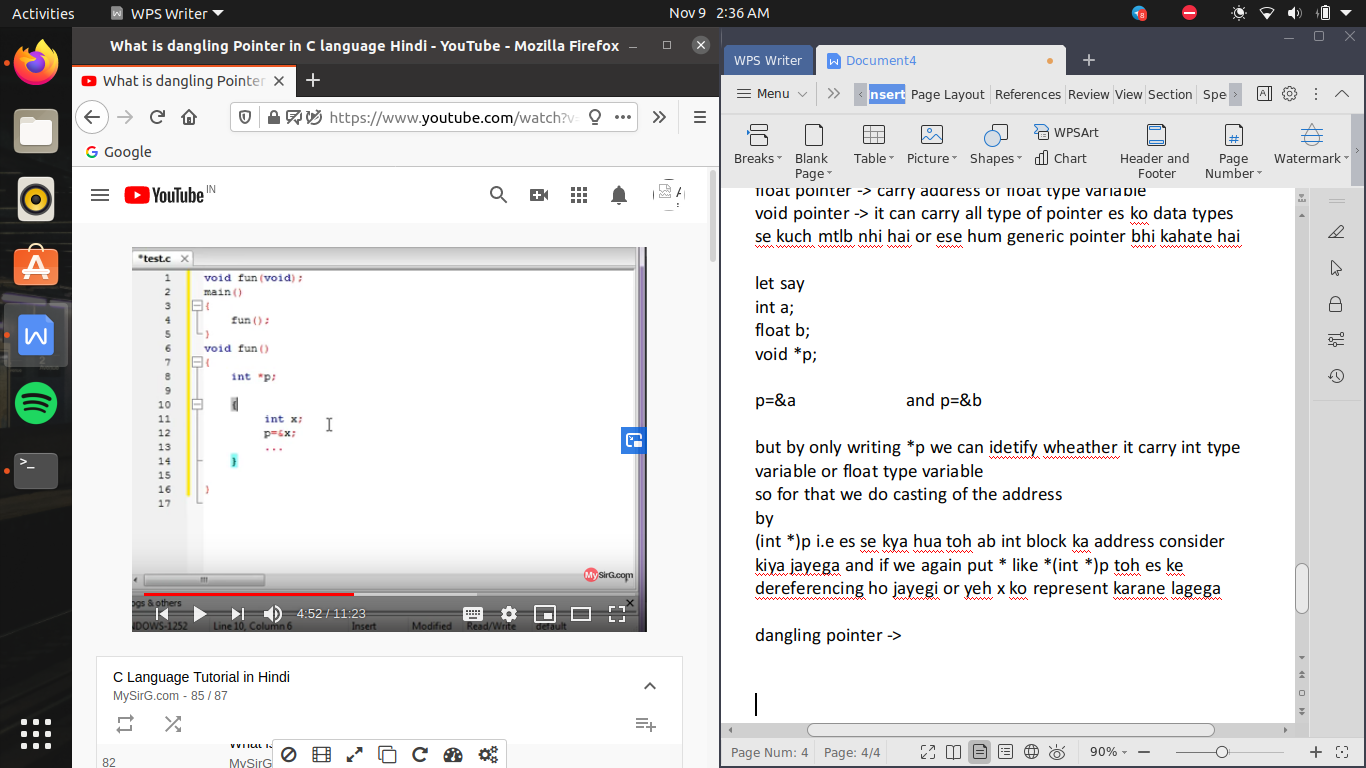
but by only writing \*p we can idetify wheather it carry int type variable or float type variable

so for that we do casting of the address

by

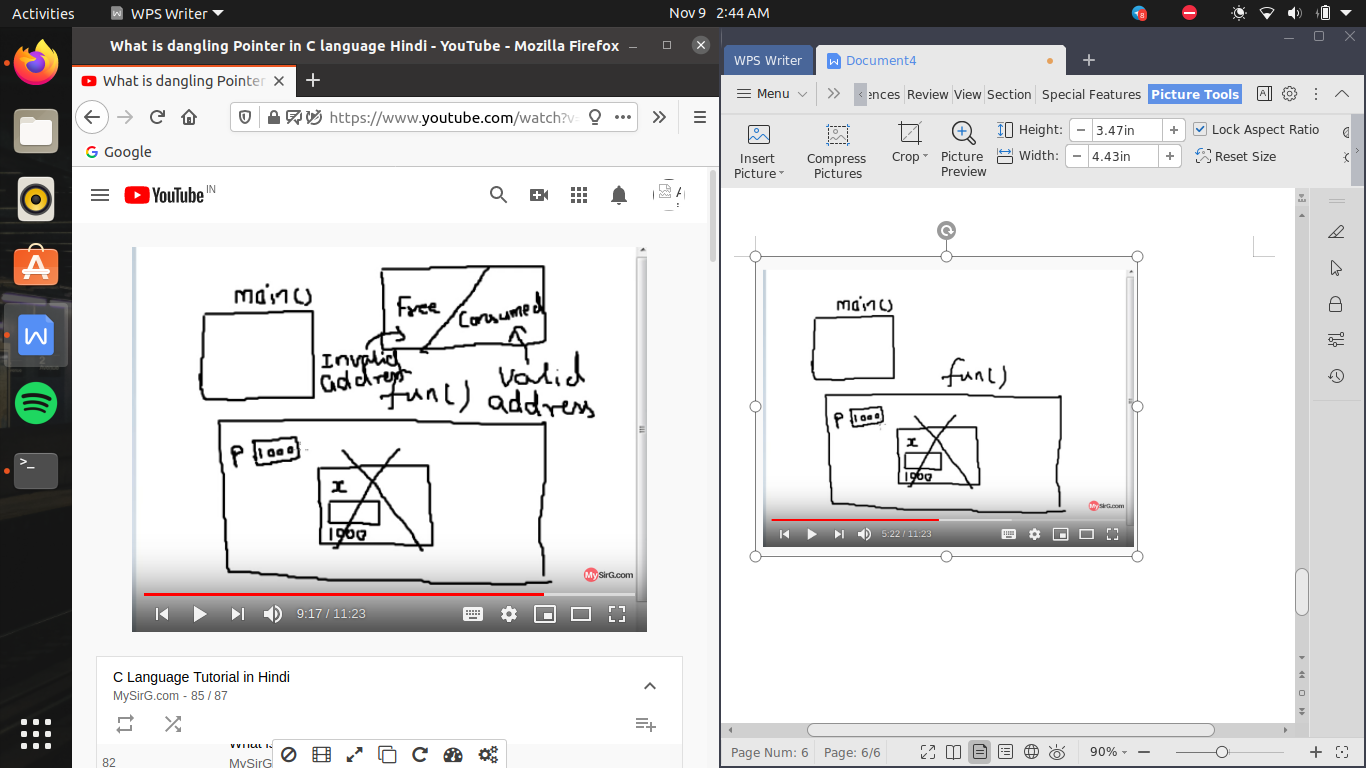
(int \*)p i.e es se kya hua toh ab int block ka address consider kiya jayega and if we again put \* like \*(int \*)p toh es ke dereferencing ho jayegi or yeh x ko represent karane lagega

dangling pointer ->



yaha x ka address p ko assign toh ho gaya but , jabwo function end hon jauyega ki memory release ho jayegi pr p me abhi bhi wohi address contain kr raha hai leking ab p kis ko point kr raha hai wo valid address nhi hai toh p yaha pr kahalane lagega dangling pointer kyu ki wo jis ko point kr raha tha ab wo jahah ab valid nhi rah gayi hai jab ki abhi bhi usi ko point kr raha hai toh ab yeh p ke rakhi value valid se invalid address kaha lane lagegi

toh hame yeh ensure karana chahiye ki jab program end hone wala ho toh us me p=NULL rkh de jis se us me dangling error na bane



function pointer ->

pointer ka data type yeh decide nhi karata ki pointer kitane size ka hai balaki yeh decide karata hai ki pointer konse data type ko point karega

ab function ko point karane wala pointer , function pointer kahalata hai

decleration of function pointer ->

pointer ka data type kya hai yeh depend karega ki function ka data type kya hai toh sab se pahale jo function ka return type hoga wohi pointer ka data type hoga ,

ab function pointer banate waqat hame ke kam karana hi padega jo hi es ke upar paranthsis lagana , yeh bahot important hai , if hamane wo paranthis nhi lagaya toh wo pointer nhi ban payega

or us ke bad hame deekhana hoga ki kya wo function koe argument bhi lega toh hame es ke age paranthsis bhi laga dena hai bs

i.e

void f1();

void(\*p)();

p=f1; //(assigning address of function)

p(); //call the function

