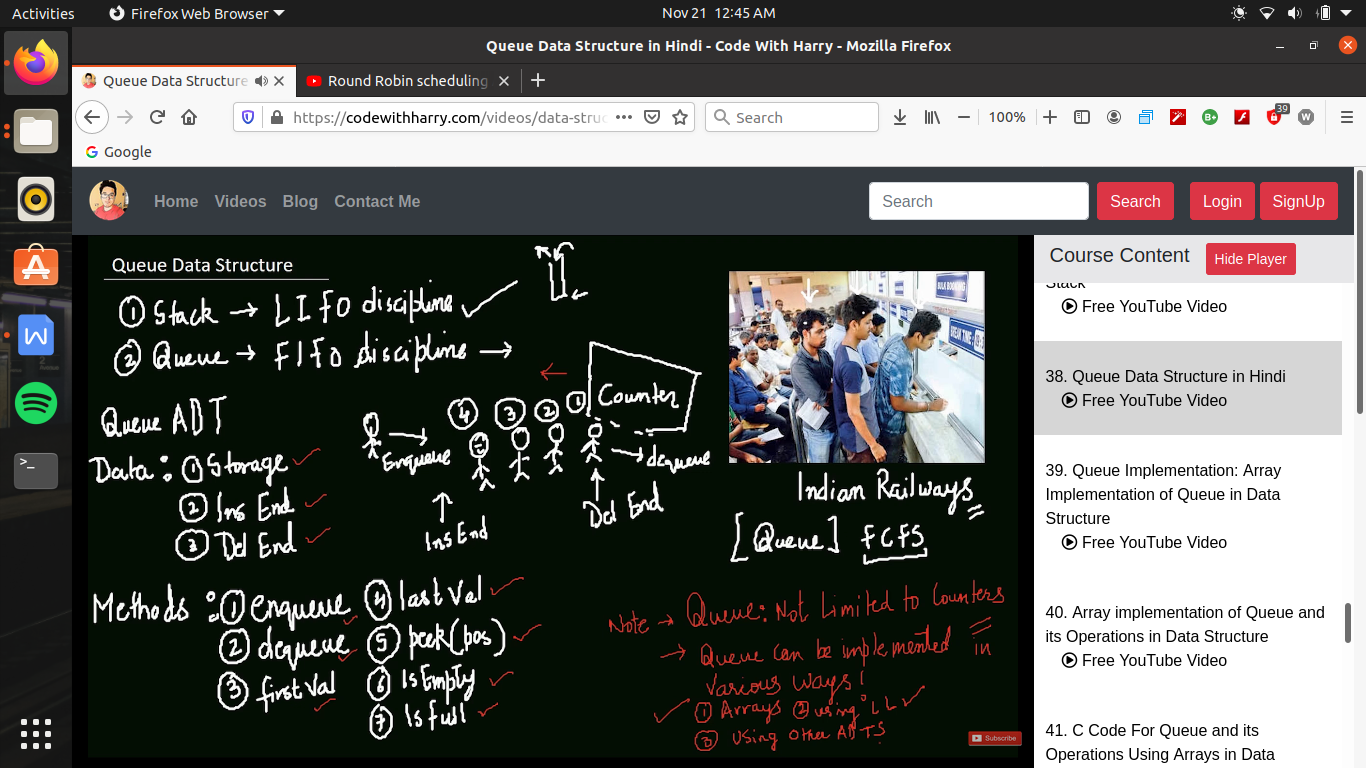
queue

stack ->LIFO (last in first out )

queue ->FIFO (first in first out)



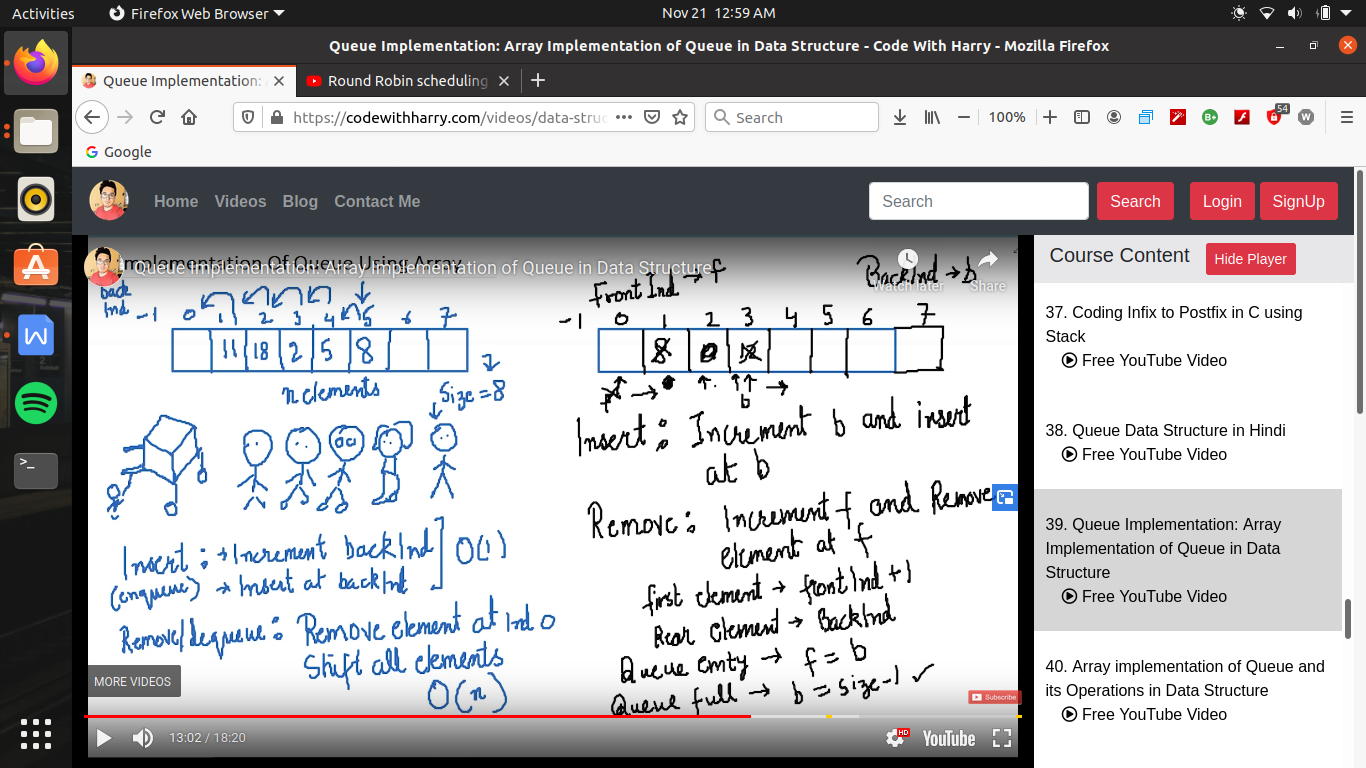
queue using array :-

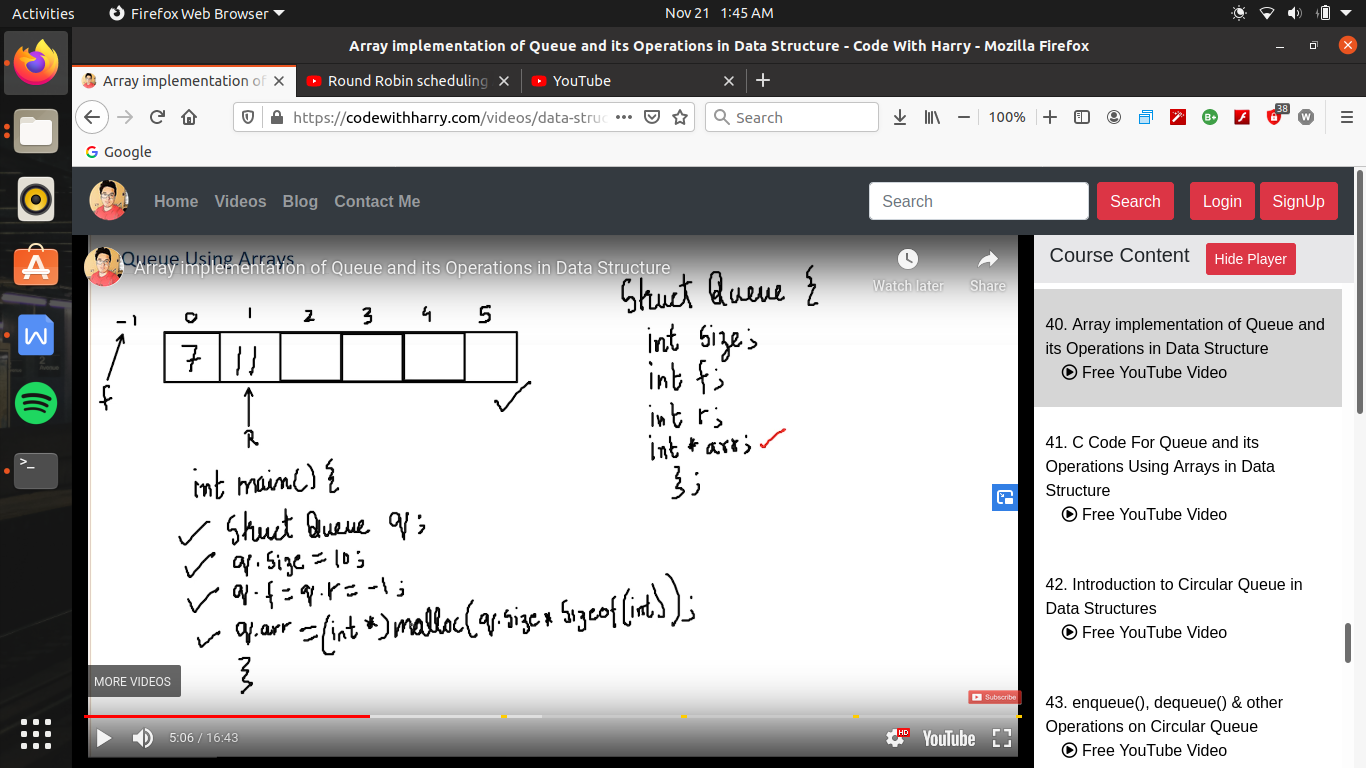
in queue using array if we have to remove(dequeue) so after removal we have to swift all elements one by one

toh es problem ko sovle karane ke leye hum 2 index lete hai ex-> fornt\_index and back\_index(es me pahle front end ko age bada dege fir remove karege )

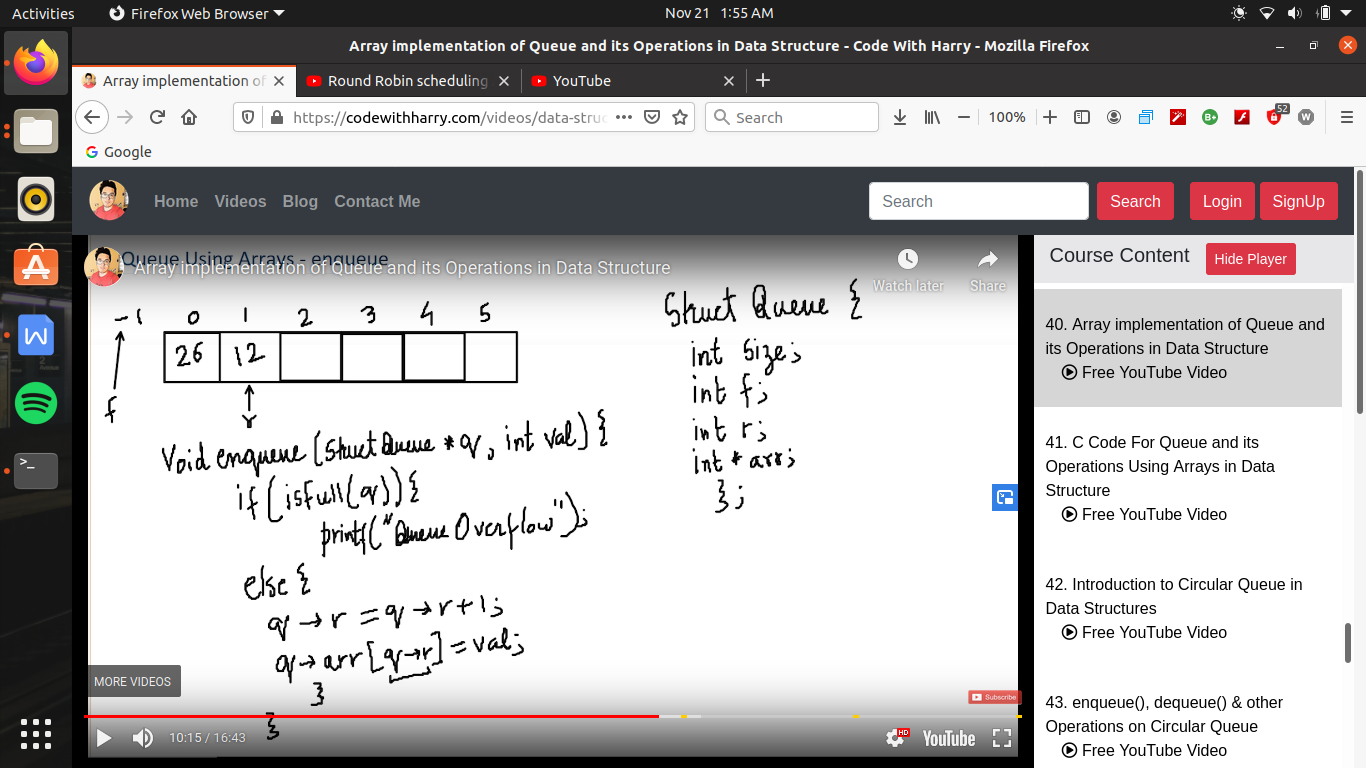
and id front index =back index then it is a queue empty condition

queue full condition when back\_index = size -1;





enqueue operation ->



Drawbacks of queue using array :-

1. space is not used efficiently (as because array me already space allocated hai so jaise jaise front\_index age badata jayega waise waise age wali index khali hoti jayegi , but we can not use this space again)

so to resolve this issue we again set the back\_index on the front

to this we will use circular increment to do that

here

i= i+1; //this is linear increment

now for circular increment we use modulus operator i.e(%)

by i=(i+1)%size

we made circular queue becaue we don’t want ot waste your elements

but es me bhi ek front wali space use nhi kr pa rahe hai

