

## Virtual Memory

Virtual memory:

1. NOT virtual add

virtual add : issued by cpu

virtual mem : memory management technique employed by OS

**Problems in memory schemes used:**

1. code has to be in mem. **but entire prog. RARELY used.**

Error code, large ds rares

2. Entire code not needed at same time

3. **Partially loaded prog.**

**Advantage :**

- prog not constrain by phy mem can be large than total phy mem

**Virtual memory:**

separate user logical mem from physical mem.

part of prog in mem for execution

logical address >>>> phy add

**addr. space shared by several process.**

More effic. pro. creation

more prog. run concurrently.

Less I/O as we don't have to read whole data at load or swap proc

Methods:

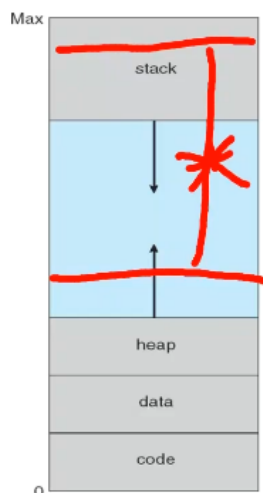
1. Demand Paging

2. Demand Segment

Virtual Address Space:

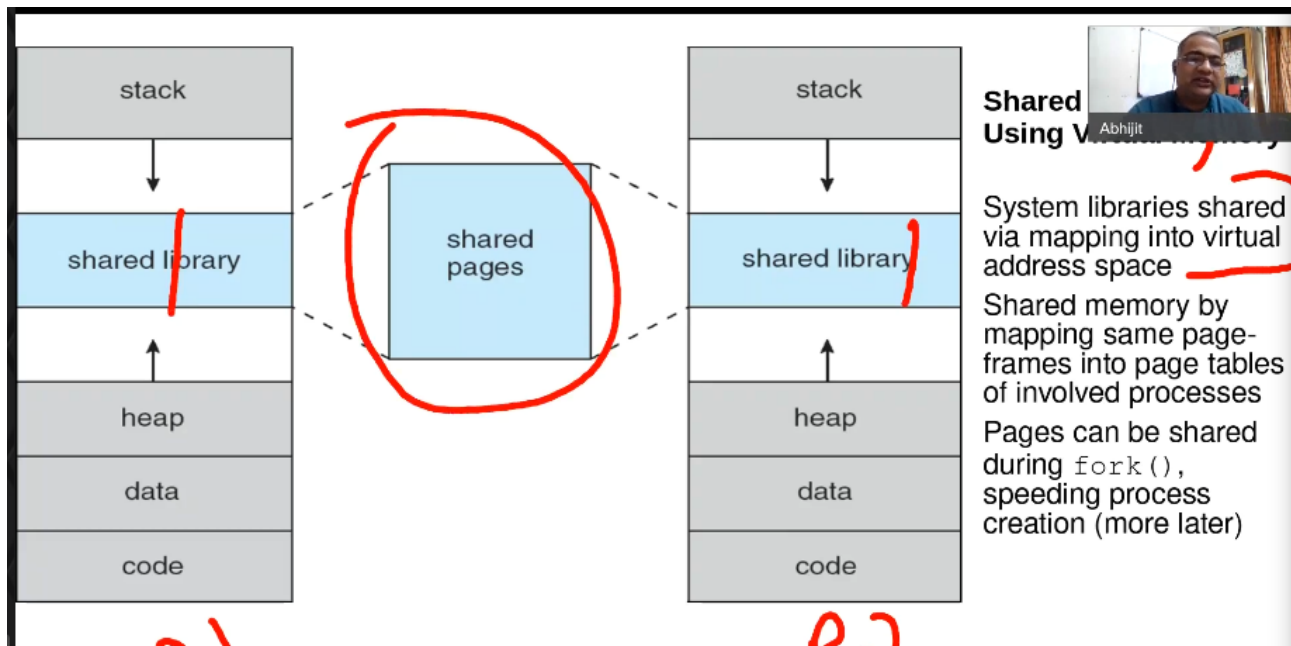
enable sparse addr space with holes left for growth, dynamically linked lib.

## Virtual Address space



Enables **sparse** address spaces  
with holes left for growth,  
dynamically linked libraries, etc

If there wasn't virtual addr, we have to allocated all seg ata start.  
In virtual mem, we can only assign page for code and data.



### Demand Paging:

load a "page" to mem when its needed  
less i/o , no unnecessary  
less mem needed  
faster response

New meaning in Valid/Invalid bit :

## New meaning for valid/invalid bits in page table

Frame #	valid-invalid bit
	v
	v
	v
	v
	i
....	
	i
	i

- With each page table entry a valid-invalid bit is associated
  - v: in-memory - memory resident
  - i : not-in-memory or illegal
- During address translation, if valid-invalid bit in page table entry is I : **raises trap called page fault**

Now if invalid then it should be in the backing store, else it would be illegal

Page Fault :

h/w interr

**occurs when page table entry is “i”**

All kernel action are taken on h/w

change to kernel stack

save context

switch to kernel stack

**On page fault :**

1. OS looks at data struc in PCB to determine  
if invalid reference --> abort the proc (**segfault**)  
just not in memory --> need to get the page in memory
2. Get empty frame
3. Swap page into frame via scheduled IO operation