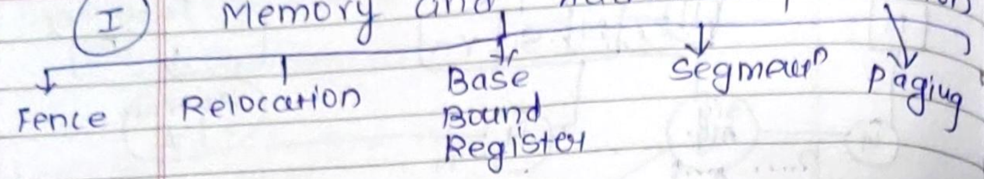
System Security

Chapter - 5

I Memory and Address protection

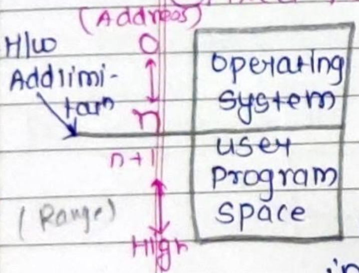


I Fence. → fence is nothing but 'boundary' / boundary wall which keep our Resources From outside of the wall.

* fencing ~~our~~ arround our Resource.

e.g our society having boundary to protect from outsider's

1 Fixed Fence Mechanism. -

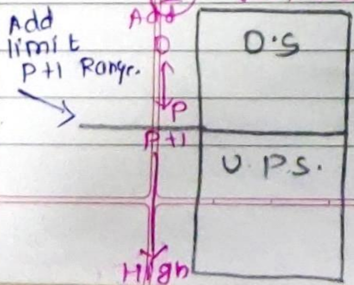


1 with help of h/w Address limitatⁿ we are limiting the area ~~that~~ How much O.S & user have to use.

2 H. A. limit divide memory into 2 part i.e 0 → n for O.S & n+1 → High for user p. space

3 If user goes beyound the limit (OR <n) O.S reject the request ~~and~~ And tell tham to access only (n+1 to Higuy Range.

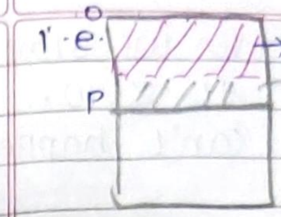
2 Variable Fence mechanism.



1 In this O.S having Range 0 to p & UPS having p+1 & High.

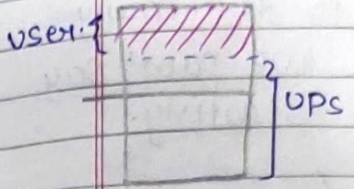
2 if O.S want only small

Range out of Range.



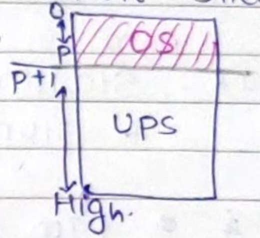
wow to

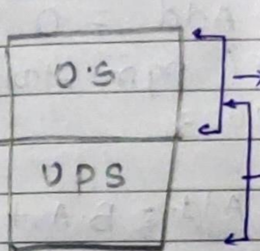
}OS used only these Range than this empty space is slide up + give to U.P.S.



** O.S give empty space to the UPS

Now limit slide up. like.





→ If OS having this much part and ups want → this much part (extra

O.S will not give to O.S How much part ~~O.S~~ U.P.S want it Required. it totally depend on O.S

- When extra part will remaining after O.S use than & than O.s provide to U.ps.

Disad ⇒ In Fixed ~~remaining~~ remaining part OF O.S is wasted

Ad of Var → Hence we use Varible when extra portain remain O.S provide to U.P.S or |slide moves up|

⇒ Relocatn ⇒ whenever we start pgm it alway's begain to zero,
(2) bt every time it can't happen.

(3) In memory there is too many sectn & pagging so we can't' say from where it's starting.

IF pgm starting /from/ ~~Q~~ = 0 m. Add.
& stop at 1003

& pgm Base Add = 0
& Offset add (pgm starting add)
= 1003

than Actual Add. = B. A + Off. A
= 0 + 1003.

So Actually starting from 1003.

e.g. Mem. Add → 1000
(Base) ⇒ pgm locatn → 0, 1, 2 ...

(Offset) ⇒ OFF = 1000 , Base ad = 0, 1, 2
p+1

Actual Ad = 1001, 1002, ...

(III) Base & Bound Register.

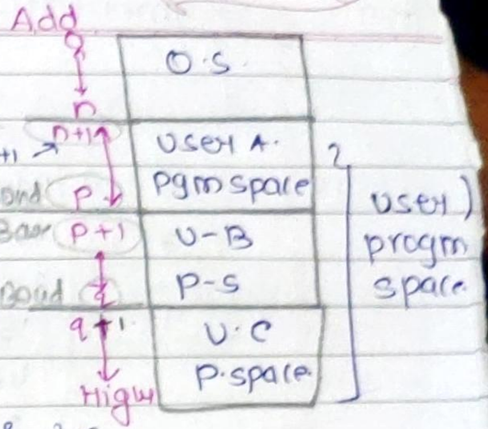
In fence fixed & varible we have only 1 Register but here 2
1 for B-Register
1 for Bound R.

Actual user

Base R – ↑starting ~~pt~~ Add of pertical (offset) user.

Here user started from n+1 → to high i.e nth

Bound R → if user B want to use space than Base R → p+1 &
Bound R → q.



Simllarly for U-A & U-C.

All this thing will decided by O.S if U-B want to ~~write~~ Read pgm in main memory & compile pgm in m mem then O.S check whether their is ~~p~~ space to fit in main memory, it will

Base & Bo→ check U-A space, if not avalble than check, U-B, & then check U-C. if no one having free space than it will wait to free

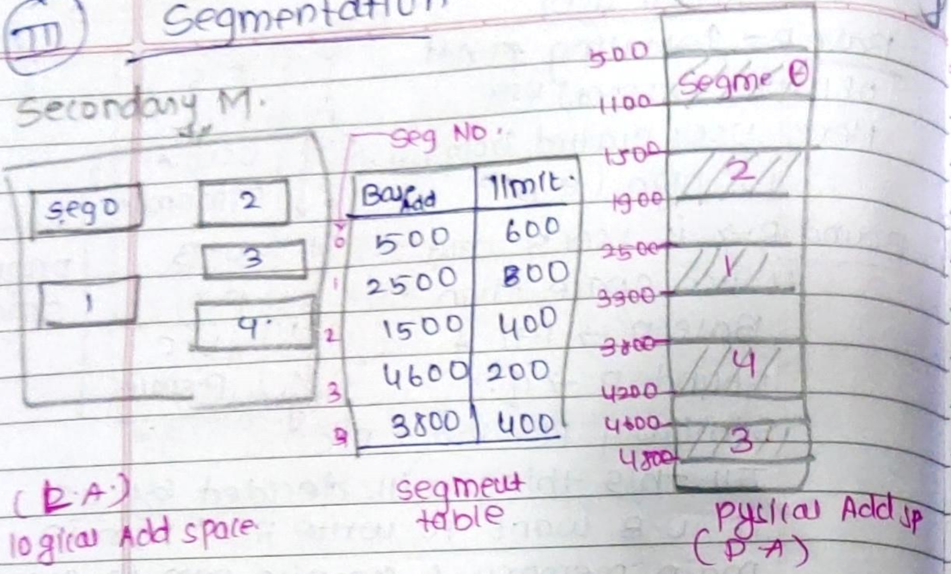
Base – R – It's starting Add of perticular user ~~or~~ space where all executn is started.

B-R ⇒ n+1 is starting Add of user A.

Bound R = 'p' is the upper add limit of the user area (or end Add) of user space Area.

non contigeneous— seg

(III) Segmentation.



	Base Add	limit.
0	500	600
1	2500	800
2	1500	400
3	4600	200
4	3800	400

(1) In P Add segments are arrange in very Random way.

(2) limit is a Offset of perticula segmt.

(3) For seg 0 → Base R = 500 we need to Add • Base Regis + limit = Actual Add • i-e startlug Ad.

seg 0 = 500+600 = 1100

so seg 0 started from 1100 till 15 600 mean give idea that 600 Add space are availble in seg 0.

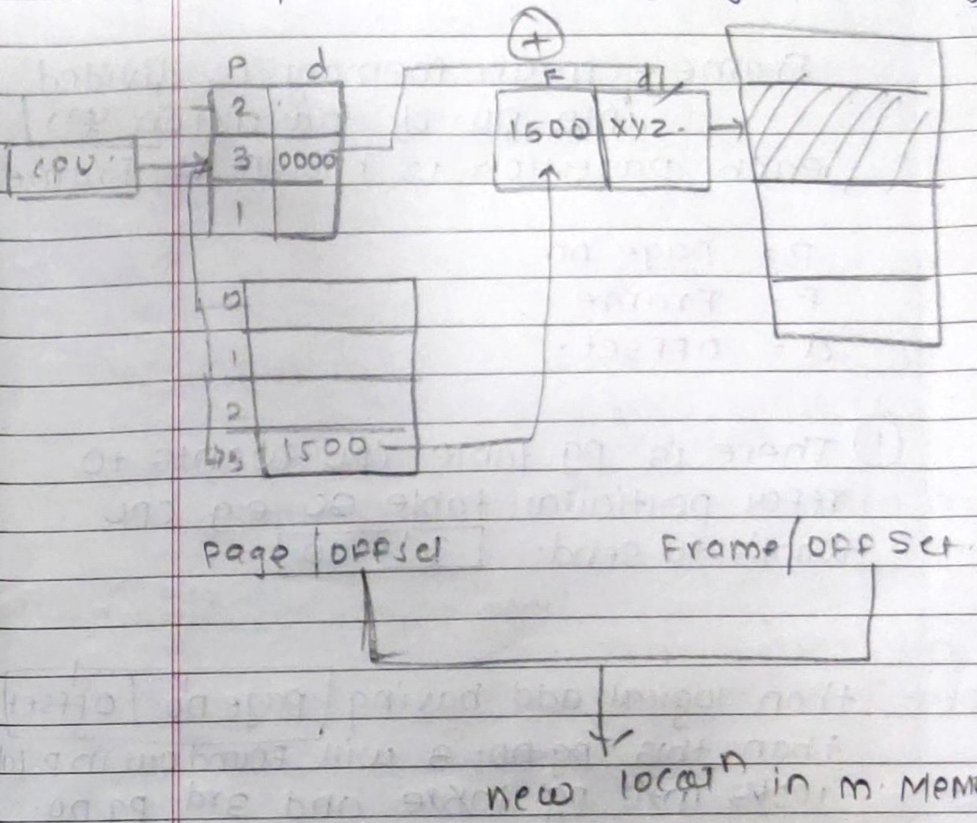
< segment 0, 600> or < name, Offse

e.g

from • LA seg 0 will point out to seg table which is Base Add is 500 & 600 is avilble space & (500+600 = 1100) upto till 1100 seg 0 with can use the space.

It's a memory mngmnt tech. in which the memory is divided into variable size parts Each part is known as segment which can be

whichever locatn comes out as a result, then their you will get a perticulor result. (physical Memory



6th Chapter.

Date

Page

(1) File protection Mechanism

- we have to ~~ex~~ protect our created file using file protectn

method. (1) - apply pswd on desk | Lptp but if no: of user uses system (pc,) than How to protect.

Access → In Direct Acc. user can directly access the file which is not good

so Access types are (operatn)

(1) Read → user can only Read file

(2) write - user can only write or ~~w~~ rewrite file.

(3) Execute - loading the file, after loading executn process will start ~~process~~

(4) Append → Already existing file user can add editing at the end of file (write, add annother file)

(5) Delete → If file covring more space than user can dlt the file

(6) list - user can list the name of file and list the attribute of file

All these are operatn that user can do, make protectn on the file.

Remane - user can't Rename the file.

editing → copying, these can also be controlled.

e.g drive Access

Accesing is method to allowing another user to do anything in file.

Access c: Classified by 3 way.

① Owner - O. is user who created the file.

(what's grp) ② Group - is set of member who need the same things & sharing same file.

③ universe - In the system, all other user are under the category

In Solaris is O.S these 3 cate. are bydefalu

User Authenticatn.

① Single factor authenticatn.

user enter psw / login to system
If pswd / id wrong system will not open

② Two-factor Aunticatn.

e.g OTP, email ~~alters~~ alerts.

③ multi-factor Auticatn.

- ① Bank → user enterd in bank once enterd debit card Atm ask pin, etc.

② Online Banking → user enterd their deta
① bank send code
② user enter code & continue process

① Somthing you have. - phone where Receive OTP or code

② Something you are - Fingerprint, face, voice - As pswd.

③ Something you know → pswd.

④ Access control → All 3 categories