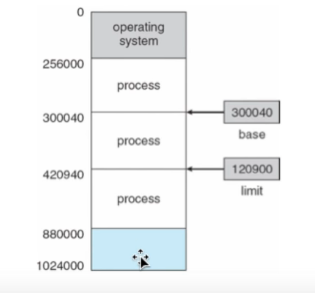
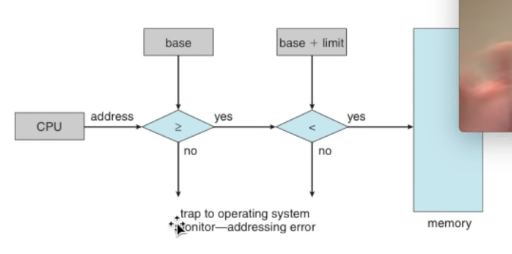
RAM’den bahsediyoruz aslinda.

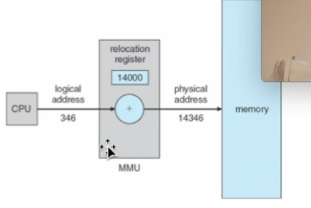
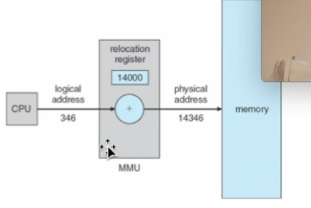
Isletim sisteminin kendisi de bir processtir. Isletim sisteminin kendisi de disk’e yukludur. Bilgisayar ilk acildiginda diskten RAM’e yuklenir isletim sistemi. Her program, process boyle calisiyor!

Processler kendi adres uzaylarinda yasarlar. Yani isletim sistemi her process’e sadece o process varmis gibi hissettiriyor. Process zannediyor ki atiyorum 0. adrese erisiyor ama aslinda process oyle zannediyor halbuki iste 80. adrese erisiyor falan.



120 binden fazlasina erismeye calisirsa process isletim sistemi bir hata vererek bunu engelliyor. Process icin ayrilmis alan sadece gosterildigi gibidir.



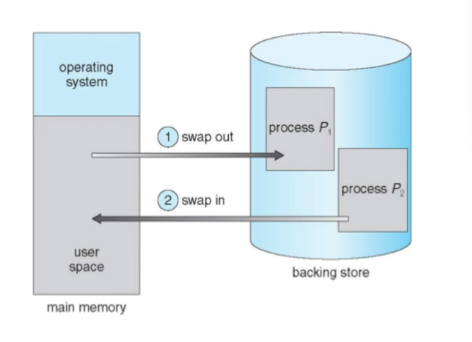


Process kendi dunyasinda 346’ya eristigini zannediyor fakat aslinda 14346 adresine erisiyor. (Bellekte) Relocation register processin bellekteki baslangic adresini verir. Bu durumda eger kendi dunyasinda 346’da calisiyorsa bellekte aslinda processin baslangic adresinden itibaren 346 ekledigimizde olusan adreste calisiyor demektir ;)

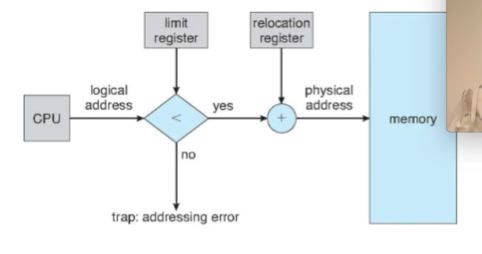
Swapping = Hafizadaki bir processin degistirilmesi. Processin RAM’de kapladigi yerin iste mesela diske atilmasi ve bosalan yere baska bir process in gelmesi.

Iste mesela benim bellegim 8GB hepsi de processler tarafindan doldurulmus bu durumda yeni bir process geldiginde ben out of memory diyip calistirmayacak miyim? Hayir tabiki calistirmayi deneyecegiz virtual memory ve swapping ile.

Virtual Memory = Diskin bir kismini RAM olarak kullaniyor.



Hic diskin olmadigi ve sadece RAM ile calisan isletim sistemleri vardir. Cloud computing’de falan her sey RAM’de tutuluyor aslinda.



Her process’e bellekte sabit yer ayrilamaz. Yani mesela bir process daha buyuktur oteki daha kucuktur falan. Dolayisiyla processlere dinamik olarak bellekte yer ayrilir. Process bitip bellekten ciktiginda bosluklar olusuyor. (hole)

Bosluga yeni bir process yerlestirirken uygulanan yontemler:

- **First-fit** = Sigabilen ilk bosluga koy

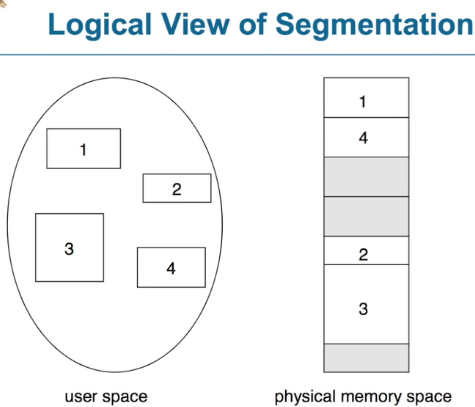
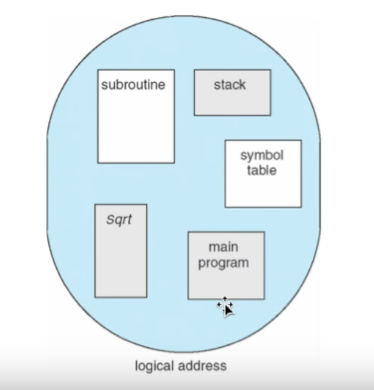
- **Best-fit** = Yerlestirdikten sonra bosluk miktari en az olacak bosluga koy. Yani artik umidimiz kalmadi bu processi nereye soksak da kurtulsak diye,

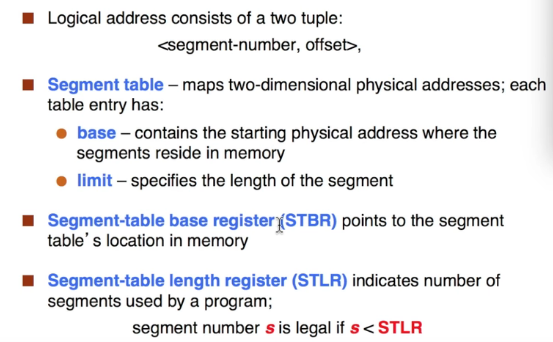
- **Worst-fit** = Processi en buyuk bosluga yerlestirir. Hizli.

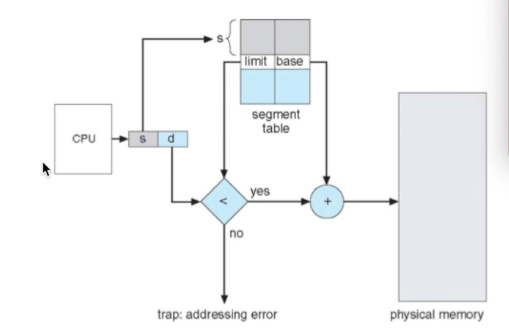
Processler calisirken de farkli hafiza degerlerine sahip oluyor. Illa process’e bellekte ayirdigimiz yer process tarafindan kullanilacak diye bir kaide yok. Process’e bellekte ayrilan yerde bosluk varsa yani process kendine ayrilan kismin hepsini kullanmiyorsa (olabilir) bu duruma **internal fragmentation** denir. Bir processin bitmesinden sonra bellekten cikiyor ve iste yerine baska bir process geliyor o sirada bosluk olusuyor falan bu duruma ise **external fragmentation** denir.

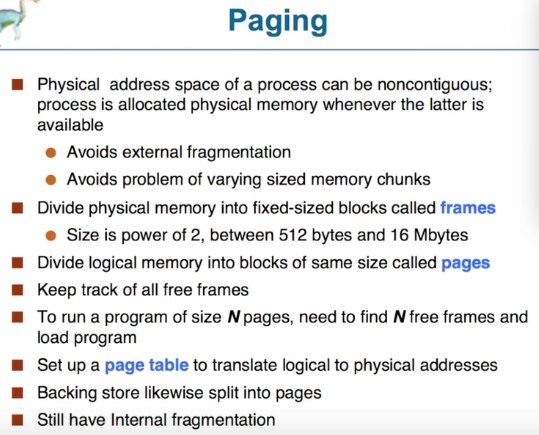
**Compaction (Defragmantation)** = Fragmentation’u azaltmak icin bosluklarin hepsini alip bir yere topluyor ve processleri arka arkaya gelecek sekilde diziyor.

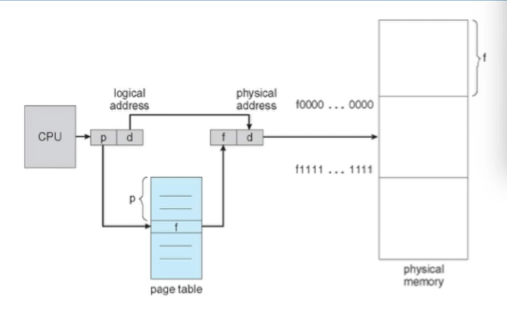
Segmentation



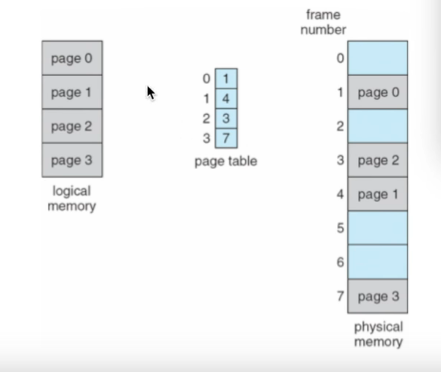


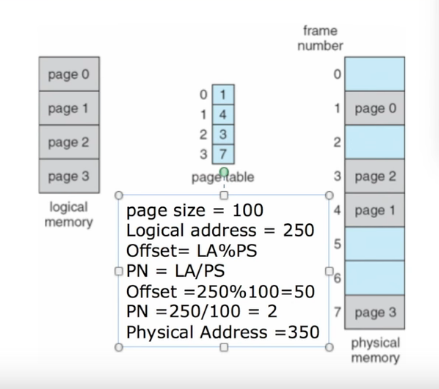


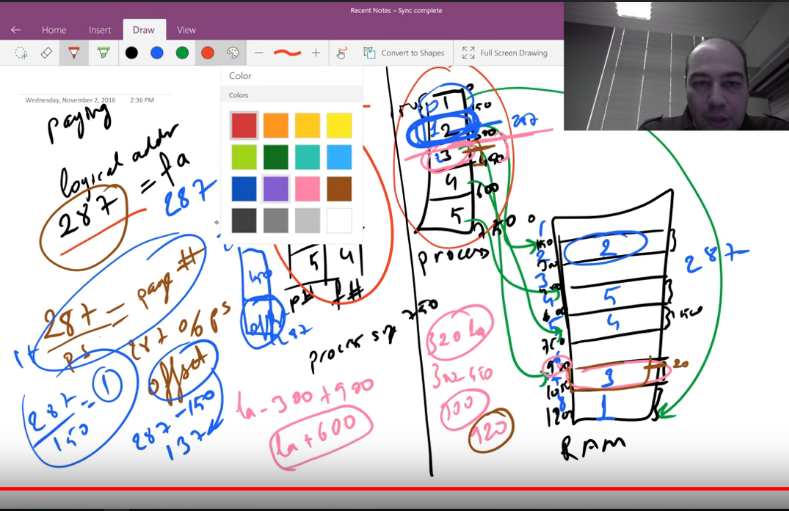


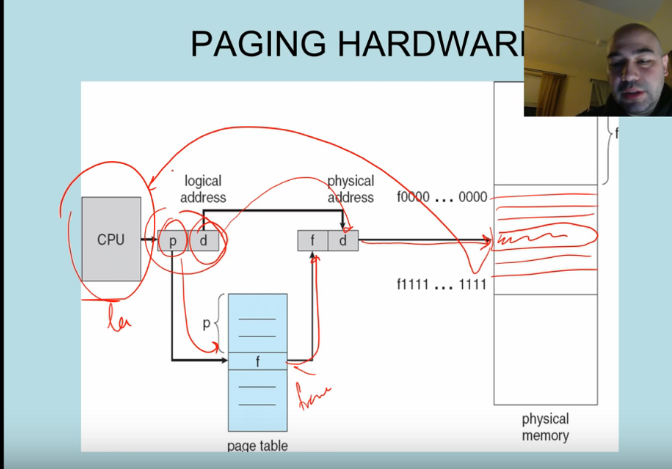


Segment boyutlari farkli olabilirken page boyutlari aynidir. Dolayisiyla pagelerde boyutu kontrol etmeye gerek yoktur zaten hepsinin boyutu ayni.

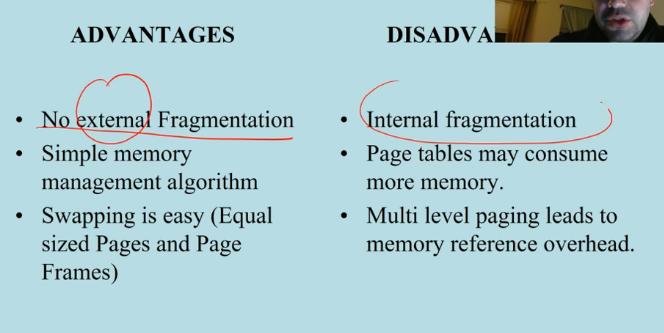








Offset degeri page size olur. Burda offset 10 bit oldugu icin 1024 bittir page size. Bundan buyuk page olusturulamaz.

For segmentation and paging = (Advantages like segmantation)

