İşletim Sistemlerine Giriş

Bellek Yönetimi
(Memory Management)

SANAL BELLEK(Virtual Memory)

Yıllar önce insanlar kullanılabilir olan belleğe sığmayan programlar ile karşılaştılar.

Bu programların çalışabilmesi problemini programı ***yer paylaşan(overlay)*** adı verilen parçalara bölerek çözdüler.

0. overlay ilk çalışmaya başlayacaktı ve bittiğinde diğer overlay i çağıracaktı.

Bazen birden fazla overlay in aynı anda bellekte olması gerekebilir.

Overlay ler diskte bulunurlar ve işletim sistemince dinamik olarak takaslanırlar.

SANAL BELLEK(Virtual Memory)

Programın overlay lere parçalanması işlemini programcı yapmalıydı. Bu işlem sıkıcı ve uzun süre gerektiren bir iştir.

Bu problemi ortadan kaldırmak için ***sanal bellek(virtual memory)*** fikri ortaya çıkmıştır.

Temel fikir; programın büyüklüğü, verisi ve yığın(stack) toplamı fiziksel belleği aşabilir.

İşletim sistemi programın çalışan parçasını bellekte, diğer kısmını diskte tutar.

SANAL BELLEK(Virtual Memory)

Örneğin; 16 MB boyutundaki bir program 4 MB lık bir bellekte çalışabilir.

Bu işlem için programın hangi parçasının şu anda bellekte bulunması gerektiğine iyi karar verilmesi gerekmektedir.

İstek yapıldıkça parçalar disk ver bellek arasında değiş tokuş yapılır.

Sayfalama (Paging)

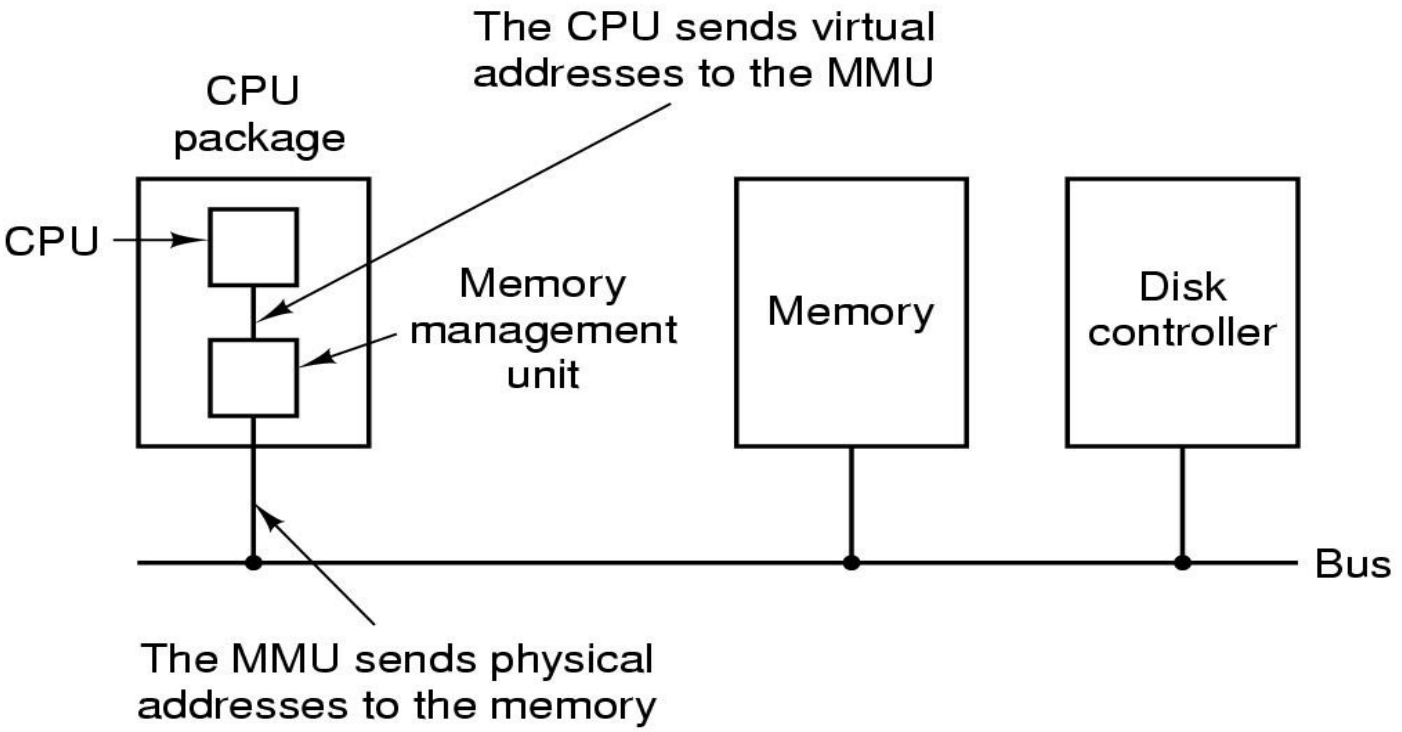
Sanal bellek sistemleri genellikle ***sayfalama (paging)*** adı verilen bir teknik kullanır. Bilgisayarda her bellek bölgesinin bir adresi vardır. Programda şu şekilde bir komut (instruction) olabilir:

mov eax,1000
Bu komut 1000 adresinin içerisindeki veriyi eax yazmacına (register) kopyalar (ya da tam tersi).

Adresler indeksleme (indexing), taban yazmacı(base register) ve parça yazmaçları (segment register) kullanılarak oluşturulabilir.

Sayfalama (Paging)

Program tarafından oluşturulan bu adreslere sanal adresler (virtual address) denilir. Sanal adres MMU (Memory Manegement Unit) tarafından fiziksel adreslere dönüştürülür.



Sayfalama (Paging)

Basit bir örneği inceleyelim. Bilgisayarımız 16 bitlik adres üretsin. (0-64 KB) adres olabilir. Bu adresler sanal adreslerdir.
Bilgisayarımızın 32 KB fiziksel belleği olduğunu düşünelim.

Tüm sanal adres uzayı(virtual memory space =0-64 KB) ***sayfa (page)*** adı verilen birimlere bölünür.

Her sanal adres sayfasına karşılık gelmesi için fiziksel bellek de aynı boyuttaki sayfalara bölünür.

Fiziksel bellekte bulunan birimlere ***sayfa çerçeveleri (page frames)*** denilir.

Sayfalama (Paging)

Sayfalar(sanal sayfa) ve sayfa çerçeveleri(fiziksel sayfa) her zaman aynı boyuttadır. Örneğimizde bu değer 4KB (4096 byte= 2^{12}) olsun.

Gerçek sistemlerde bu değer 512 byte-64 KB değişebilir.

64 KB sanal bellek = 4kb x 16 sayfadır.
32 KB fiziksel bellek= 4kb x 8 sayfa çerçevesidir.

Fiziksel bellek(RAM) ile disk arasında her zaman bir sayfa değiş tokuş yapılır.

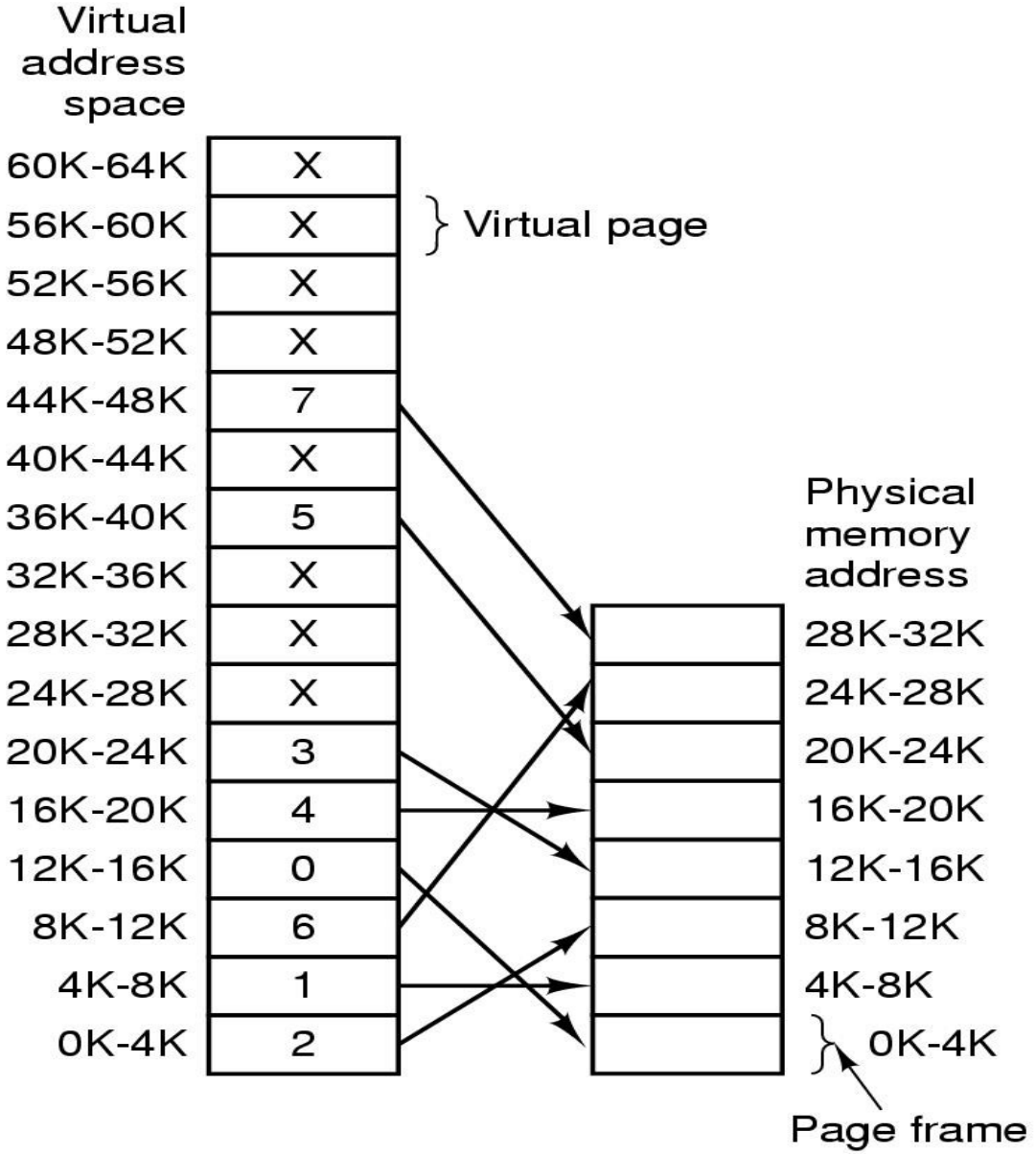
Sayfalama (Paging)

Program örneğin aşağıdaki komut ile 0. sayfaya erişmek istesin.

mov eax,0

0 sanal adresi MMU ya gönderilir. MMU bu adresin 0. sayfada olduğunu görür (0-4095).

Sayfalama (Paging)



0 sanal adresi MMU ya gönderilir.

MMU bu adresin 0. sayfada olduğunu görür (0-4095).

Şekilden görüldüğü gibi bu adres 2. sayfa çerçevesine karşılık gelmektedir.

Böylece 0 adresini 8192 ye çevirir ve yola (bus) bu adres bilgisini koyar.

Sayfalama (Paging)

16 sanal sayfanın 8 sayfa çerçevesine dönüştürülmesi, sanal adres uzayının fiziksel bellekten büyük olması problemini çözmez.

Şekildeki gibi X ile gösterilen sanal sayfalar fiziksel bellek ile eşleştirilmemiştir. Donanımda hangi sayfaların bellekte olup olmadığını tutan bir bit mevcuttur.

Program eşlenmemiş bir belleğe erişmek isterse ?

mov eax,32780 { 8.sayfanın başı = 32768 +12. byte}

8. sanal sayfanın 12. byte na erişmek istesin.

Sayfalama (Paging)

MMU bu sayfanın eşlenmediğini fark ettiği zaman işletim sistemine uyarı verir(trap).

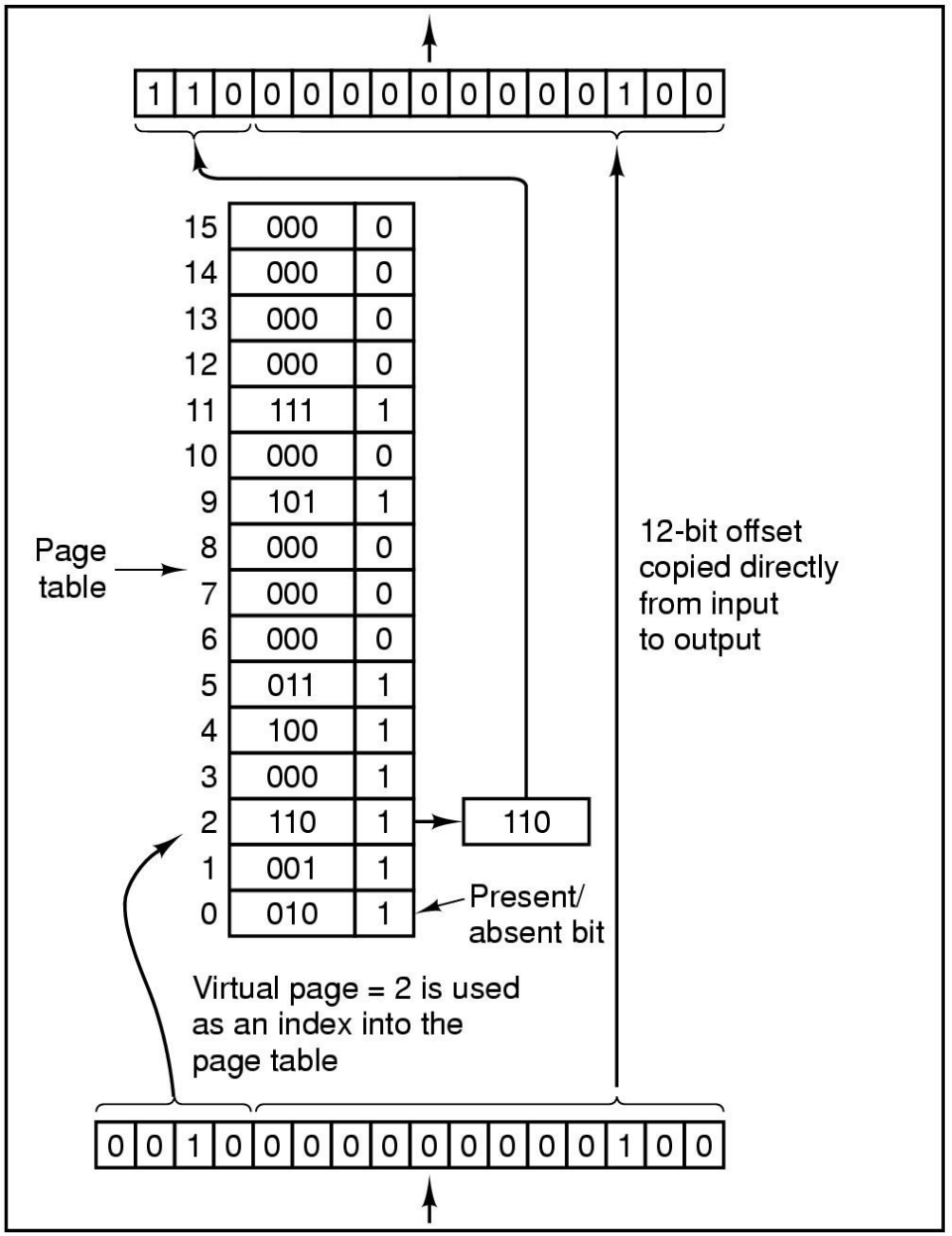
İşletim sistemine gönderilen bu hataya ***sayfa hatası (page fault)*** denilir.

İşletim sistemi mevcut bellekteki sayfalardan az kullanılan bir tanesini seçer ve diske yazar.

Boşalan yere istenilen sayfayı getirir ve sayfa hatasına neden olan komutu tekrar çalıştırır.

İşletim sistemi ayrıca MMU haritasında yeni getirilen sayfanın sanal sayfasını işaretler ve fiziksel sayfa ile eşleştirir.

Sayfalama (Paging)



8196 sanal adresi MMU ya gelsin.

Gelen 16 bitlik sanal adres,

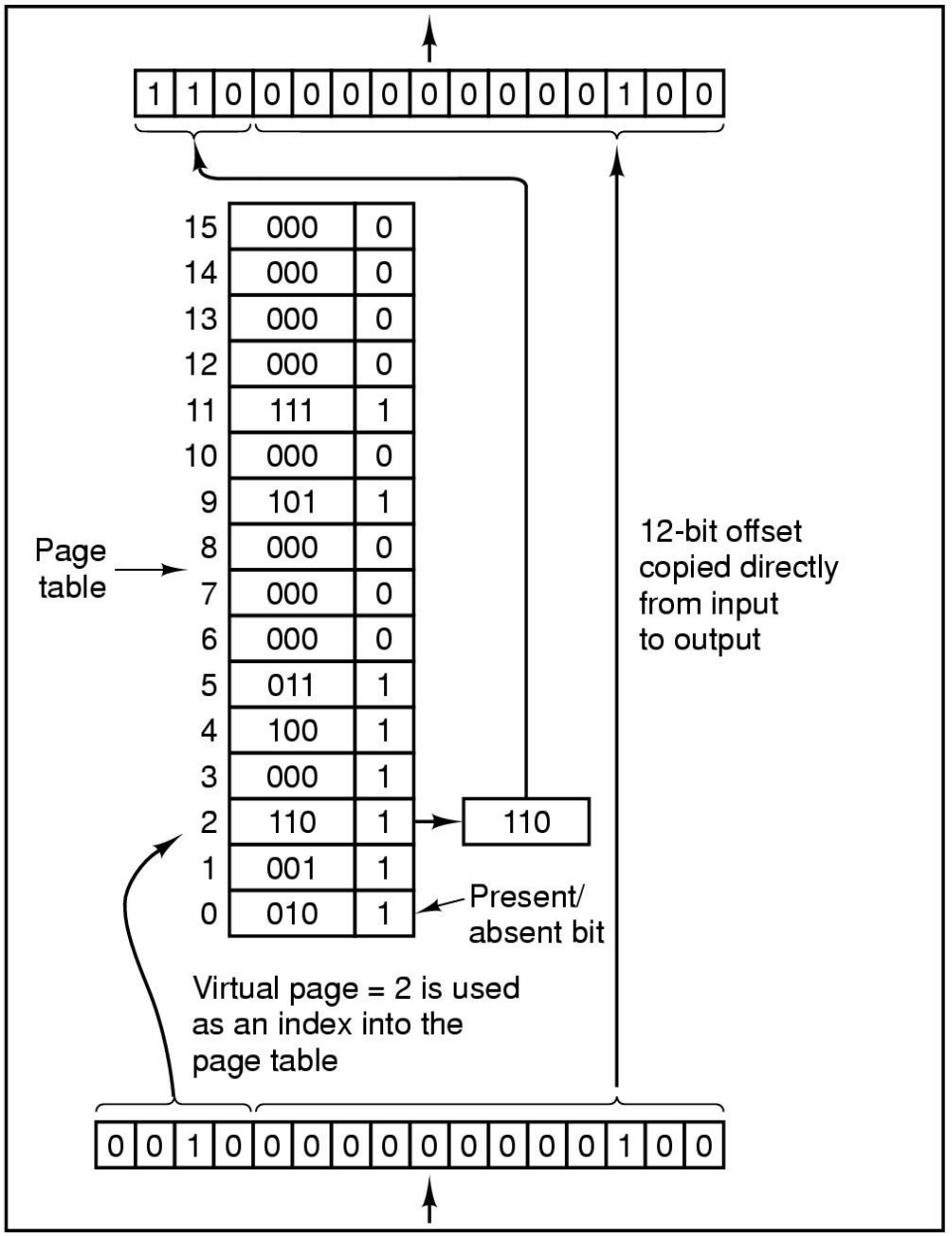
4 bit **sayfa numarasına (page number) ve** ***12 bit*** **göreli konuma (offset)** ayrılır.

12 bitlik göreli konum(offset) ile

2^{12}=4096 byte lık olan sayfanın içerisindeki adres gösterilir.

2^{4}=16 ile sanal sayfalar gösterilir.

Sayfalama (Paging)



Sayfa numarası (page number) ***sayfa tablosu (page table)*** içerinseki indeks numrasıdır.

Sayfa tablosu sanal tabloya karşılık gelen fiziksel sayfa çerçevesinin bulunması için kullanılır.

Eğer sayfa tablosunda sayfa eşlemesi yok ise ***(bu durum present/absent biti ile bilinir)*** işletim sistemine sayfa hatası (page fault) gönderilir.

Eşleme var ise; sayfa çerçevesi bulunur, ***bu değerin sağına göreli konum(offset) eklenir*** ve bu değer çıktı yazmacı vasıtasıyla yola (bus) konulur.

Sayfa Tabloları (Page Tables)

Sanal adresi (virtual address) fiziksel adrese çevirebilmek için sanal adres, sanal sayfa numarası(virtual page b-number) ve göreli konum (offset) e parçalanır.

Örneğin; 16 bitlik adres , 4kb lık sayfa boyutu ile

16 bit adres =4 bit sanal sayfa numarası + 12 bit göreli konuma

parçalanır. Sayfa boyutuna göre farklı parçalama yapılabilir.

Sanal sayfa numarası sayfa tablolarında indeks değeri olarak kullanılır.

Sayfa Tabloları (Page Tables)

Sayfa tablosundan sanal sayfa numarasına karşılık gelen sayfa çerçevesi bulunur. Bulunan adres göreli konum(offset) ile birleştirilir ve fiziksel adres oluşturulur. Bu adres belleğe gönderilir.
Bu yaklaşımda iki temel problem ortaya çıkabilir :

1. Sayfa tabloları oldukça büyük olabilir.
2. Eşleme işlemi çok hızlı yapılmalıdır.

Modern bilgisayarlarda 32 bit sanal adres kullanırlar. Eğer sayfa boyutunu 4 kb seçersek, 1 milyon sanal sayfa olmalıdır. Sayfa tablosunda 1 milyon girdi bulunmalıdır. Her sürecin kendi sanal adres uzayı olduğu için kendi sayfa tablosu da olmalıdır.

Çok seviyeli sayfa tabloları (multi level page tables)

Büyük sayfa tablolarını bellekte tutma problemini çözmek için çok seviyeli sayfa tabloları (multilevel page tables) kullanılmaktadır.

32 bitlik sanal adres = 10 bit PT1 + 10 bit PT2 +12 bit göreli konum

a parçalanır. Sayfa boyutları 4 kb olduğu için sayfa tablosunda 2^{20} adet girdi vardır.

Çok seviyeli sayfa tablolarında tüm sayfalar aynı anda bellekte bulunmaz.

Çok seviyeli sayfa tabloları (multi level page tables)

Sanal adres MMU ya verildiğinde, PT1 ayrılır. Bu değer ilk seviyeli tabloda indeks olarak kullanılır.İlk seviyedeki tablo girdisi ikinci seviyedeki tablo indeksini ya da adresini verir.

İlk tablodaki 1024 girdiden her biri 4MB lık sanal belleğe karşılık gelmektedir.

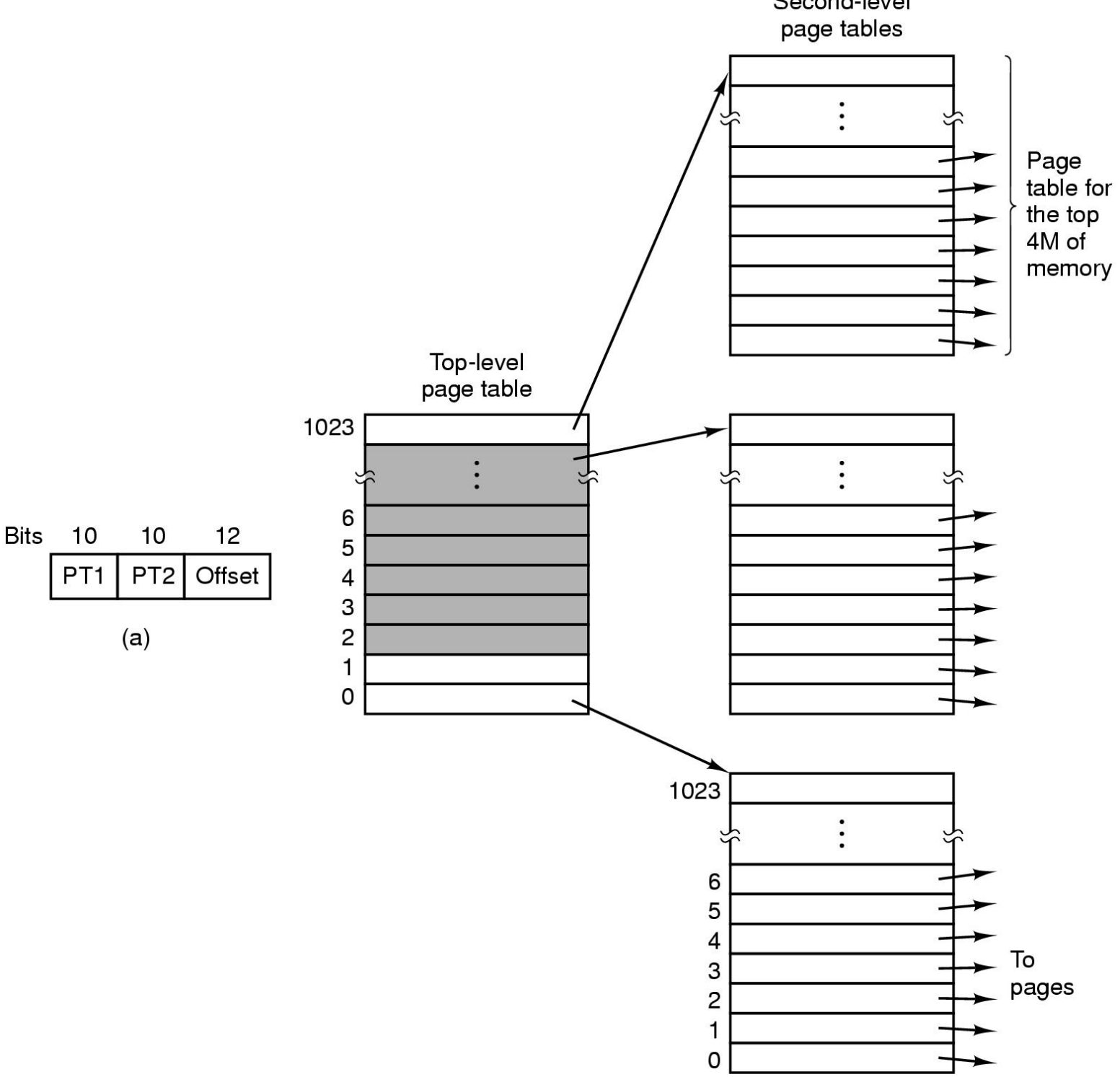
PT2 2. seviyedeki tablonun indeks değeri olarak kullanılır ve sayfa çerçevesi numarasını bulmak için kullanılır.

Çok seviyeli sayfa tabloları (multi level page tables)

Sanal adres MMU ya verildiğinde, PT1 ayrılır. Bu değer ilk seviyeli tabloda indeks olarak kullanılır.İlk seviyedeki tablo girdisi ikinci seviyedeki tablo indeksini ya da adresini verir.

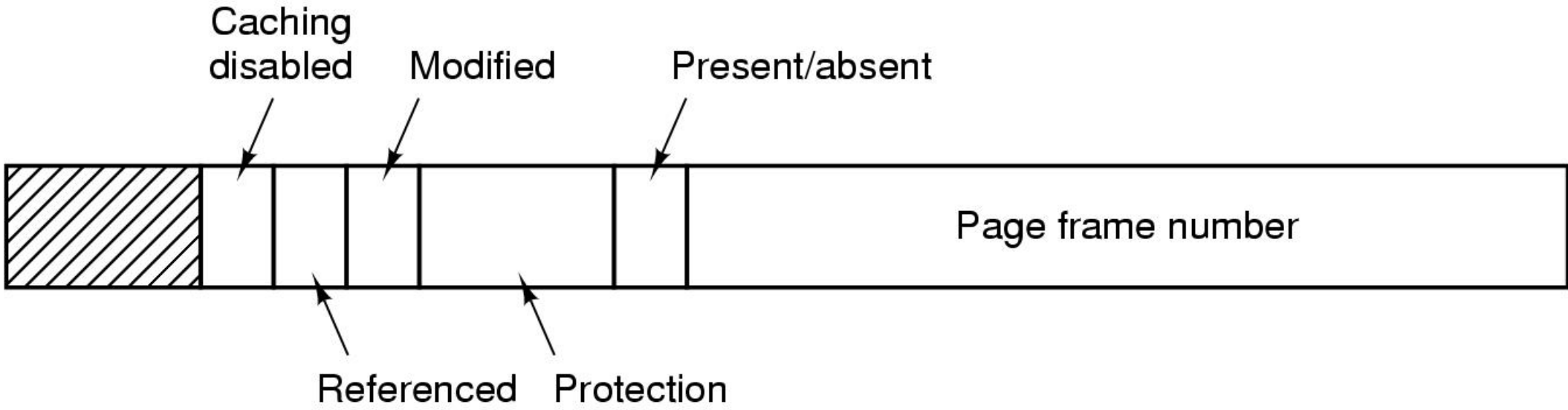
İlk tablodaki 1024 girdiden her biri 4MB lık sanal belleğe karşılık gelmektedir.

PT2 2. seviyedeki tablonun indeks değeri olarak kullanılır ve sayfa çerçevesi numarasını bulmak için kullanılır.


Second-level
page tables
Page
table for
the top
4M of
memory
Top-level
page table
1023
6
5
4
3
2
1
0
Bits 10 10 12
PT1
PT2
Offset
(a)
1023
6
5
4
3
2
1
0
To
pages

Sayfa tablosu girdilerinin yapısı

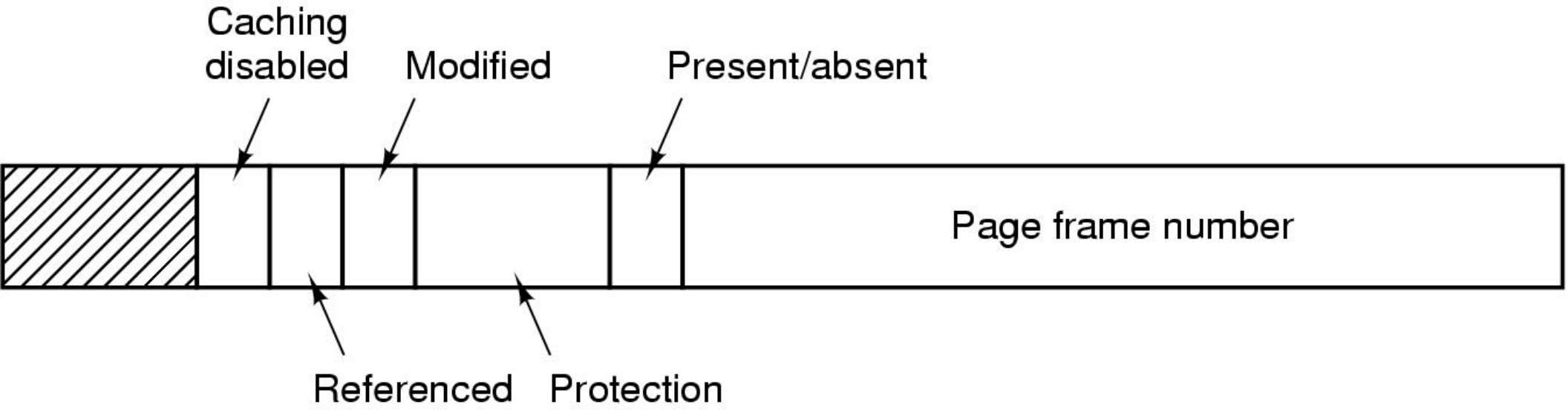
Bu girdinin tam şekli makineye göre değişir fakat kabaca içerisindeki bilgiler aynıdır. Boyutu değişebilir ama genellikle 32 bit dir.
Sağdan sola doğru sayfa çerçeve numarası, eşleme var ya da yok, koruma bitleri (RWX) | (R / W), M biti sayfaya yazma olduğunda 1 lenir(dirty biti), R biti sayfaya okuma ya da yazma olduğunda 1 yapılır.



Sayfa tablosu girdilerinin yapısı

Caching disabled biti cache(ön bellekleme) işleminin sayfa için yapılmasını engeller. Bu özellik bellek yerine aygıt yazmaçlarına eşlenmiş olan sayfalar için önemlidir.

Eğer bellek ile I/O eşlemesi yok ise bu bit kullanılmaz.



İşletim Sistemlerine Giriş

Bellek Yönetimi
(Memory Management)