

İŞLETİM SİSTEMLERİ

DERS 4 PROSES

PROSES

İşletim sistemlerindeki en önemli kavramlardan biri de **proses** 'tir. Proses çalışmakta olan programlara verilen genel addır.

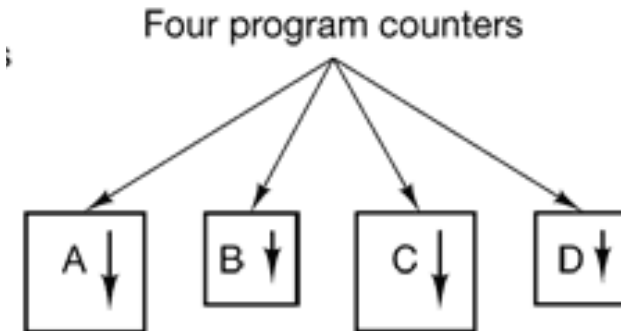
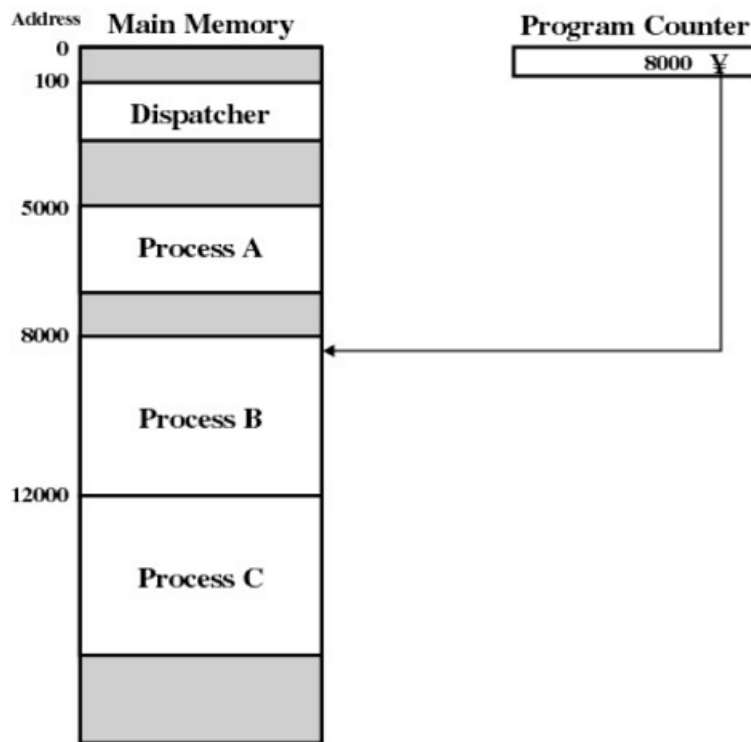
Günümüzdeki bilgisayarlarda aynı anda birden fazla programı çalıştırmak mümkündür. Bir program bir taraftan diskten veri okurken, diğer taraftan bu verileri ekrana ya da yazıcıya gönderebilir.

Multiprogramming bir sistemde CPU, çalışmakta olan programlar arasında anahtarlama yapar ve işlemci her bir proses için saniyenin yüzde biri ya da binde biri kadar süre ayırır.

PROSES MODEL

- Çalışmakta olan bütün prosesler ardışık bir şekilde bellekte beklerler.
- Bir proses program sayacı, yazmaçlar ve çeşitli değişken değerlerini tutan ve icra edilmekte olan program olarak tanımlanabilir.
- Kavramsal olarak bakıldığında, sanki her bir proses kendi sanal işlemcisini kullanıyormuşçasına hareket eder. Ama gerçekte, gerçek CPU prosesler arasında anahtarlanır. Bu hızlı anahtarlama işlemine **multiprogramming** denir.

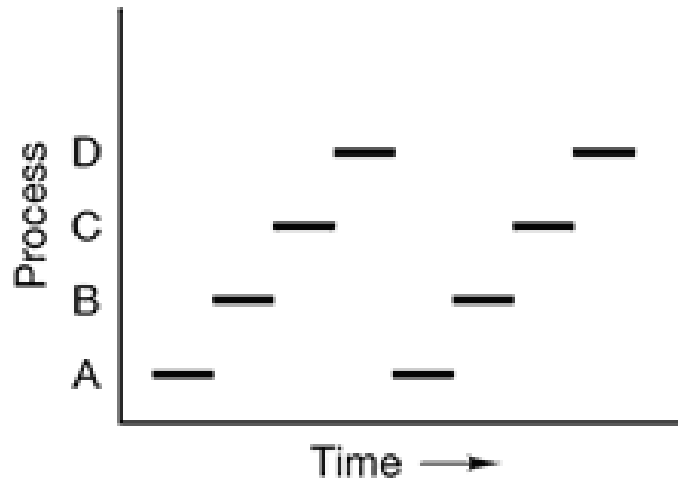
- Şekil a'da 4 programa sahip bir multiprogramming örneği görülmektedir.
- Şekil b'de her bir proses kendine ait program sayacı değerini ayrı ayrı tutmaktadır.



(b)

- Gerçekte fiziksel olarak sadece bir tane program sayacı vardır. O da işlemcinin içinde bir yazmaçtır.
- Her bir proses çalışmaya başladığında kendi adres alanında sakladığı program sayacı değeri bu işlemci üzerindeki fiziksel program sayacı yazmacına yükler. Proses işlemciden çıkarken fiziksel program sayaç değeri bellek üzerindeki o prosese ait mantıksal program sayaç alanına yazılır.

- Aşağıdaki şekilde çalışmakta olan birden fazla prosesi görmekteyiz. Fakat her hangi bir anda işlemci içerisinde çalışan yalnız bir proses mevcuttur.



(c)

BİR ÖRNEK

- Program ile proses arasındaki farkı anlamak için şu örneği inceleyelim. Bir baba kızının doğum günü için pasta hazırlıyor olsun. Öncelikle bu adam bir pasta tarifine ihtiyaç duyar. Daha sonra un, yumurta, şeker, vanilya vb. malzemelerin olduğu bir mutfağa gereksinim duyar.
- Bu örnekte, pasta tarifi bilgisayardaki programı, adam işlemciyi (CPU), kullanacağı mutfaktaki malzemeler ise bilgisayardaki giriş verilerini temsil eder. Adamın tarifi okuyup ordaki talimatlara göre malzemeleri kullanıp, pasta yapması işlemlerinin hepsi ise proses olarak adlandırılır.

- Bu adam pastayı yaparken, birden içeriye ağlayarak oğlu gelir. Çocuk kendisini arı soktuğunu söyler. Bu durumda adam pasta yapma işlemine ara vermek zorundadır. Tarif üzerinde kaldığı en son noktayı işaretler ve daha acil durumda olan oğlu ile ilgilenmeye başlar. Bunun içinden raftan ilk yardım kitabını alır ve oradaki talimatlara göre çocuğunu tedavi eder. Çocuk iyileştikten sonra tekrar kaldığı noktadan pasta yapmaya devam eder.
- Bu örnekle ise işlemcinin bir prosesten daha öncelikli bir prosese anahtarlanması söz konusudur.
- Sonuç olarak proses, bir programa, giriş verilerine, çıkış verilerine, ayrıca bir duruma sahip bir aktivitedir.

PROSES OLUŞTURMA

Temel olarak prosesin oluşturulduğu dört durum vardır.

- İşletim sistemi başlatıldığında,
- Çalışan bir proses tarafından oluşturulması,
- Kullanıcı isteği ile,
- Batch job ile

- İlk bilgisayar açıldığında işletim sistemine ait bazı prosesler otomatikman oluşturulur. Bunlardan bazıları ön planda bazıları ise arka planda çalışır. Arka planda çalışan proseslerin kullanıcılar ile direk teması söz konusu değildir. Bu tip proseslerin güvenlik, e-mail kontrol, güncelleme gibi özel görevleri mevcuttur. Arka planda çalışan bu tip proseslere **daemon** adı verilir.
- UNIX'de, *ps* programı ile Windows ise , CTRL-ALT-DEL tuşlarına basılarak bilgisayarda çalışmakta olan prosesler listelenebilir.

- Yeni bir proses oluşturma konusunda ikinci seçenek olarak, çalışmakta olan proses bir sistem çağrısı yaparak yeni prosesleri oluşturup, kendisine yardımcı prosesler oluşturabilir.
- Örneğin büyük miktarda verinin bir ağ üzerinden transfer edilip işlendiği bir sistemde, bu işlem tek bir proses üzerinden icra edilebileceği gibi, iki farklı proses tarafından da icra edilebilir. Birinci proses ağ üzerinden veriyi alıp tampon bellekte depolarken, diğeri ise bu tampon üzerindeki verileri alıp işleyebilir.
- Proses oluşturma konusunda üçüncü örnek ise, etkileşimli sistemlerde kullanıcılar bir komut vererek yada bir ikona tıklayarak yeni bir programı çalıştırabilir dolayısı ile yeni bir prosesi oluşturulmuş olur.

- En son seçenek ise özellikle mainframeler üzerinde icra edilen batch işlemlerdir. İşletim sistemi ardışık şekilde olan bu proseslerden biri bittiğinde diğerini oluşturup çalışmasına sağlar.
- Teknik olarak, proses oluşturulan bu dört durumda da, işletim sistemine yeni bir proses oluşturmak için sistem çağrısı yapılır. Sistem çağrısı ile işletim sistemi yeni bir proses oluşturarak, bu prosesi çağrıda bulunan programla ilişkilendirir.
- UNIX, işletim sisteminde yeni bir proses oluşturmak için kullanılan sistem çağrısı “**fork**” tur. Bu çağrı ile çağrıda bulunan prosesin tam olarak kopyası olan başka bir proses oluşturulur.
- Windows’ ta ise, yeni bir proses oluşturup bunun ilgili programla ilişkisini kuran Win32 API fonksiyonu ise “**CreateProcess**” dir.

- Hem UNIX hem de Windows'ta, bir proses oluşturulduktan sonra, ana proses ile çocuk proses için bellek üzerinde birbirinden bağımsız farklı alanlar ayrılır. Ana prosesin kendi alanında yaptığı değişiklikler çocuk proses tarafından, çocuk prosesin kendi alanında yaptığı değişiklikler ise ana proses tarafından görülemez.
- Fakat UNIX'de, çocuk prosesin başlangıçta adres alanı, ana prosesinkinin bir kopyasıdır. Fakat farklı adreslere sahiptirler ve ana proses ile çocuk prosesin ortak yazılabilir bir alanı yoktur. Sadece çocuk proses ana prosesin başlangıçtan gelen bazı verilerini paylaşabilir. Örneğin açık olan dosyaları okumak gibi.
- Windows'da ise ana proses ile çocuk prosesin paylaştığı bir alan yoktur.

PROSES SONLANDIRMA

Prosesler genel olarak aşağıdaki belirtilen dört durumdan dolayı sonlandırılır.

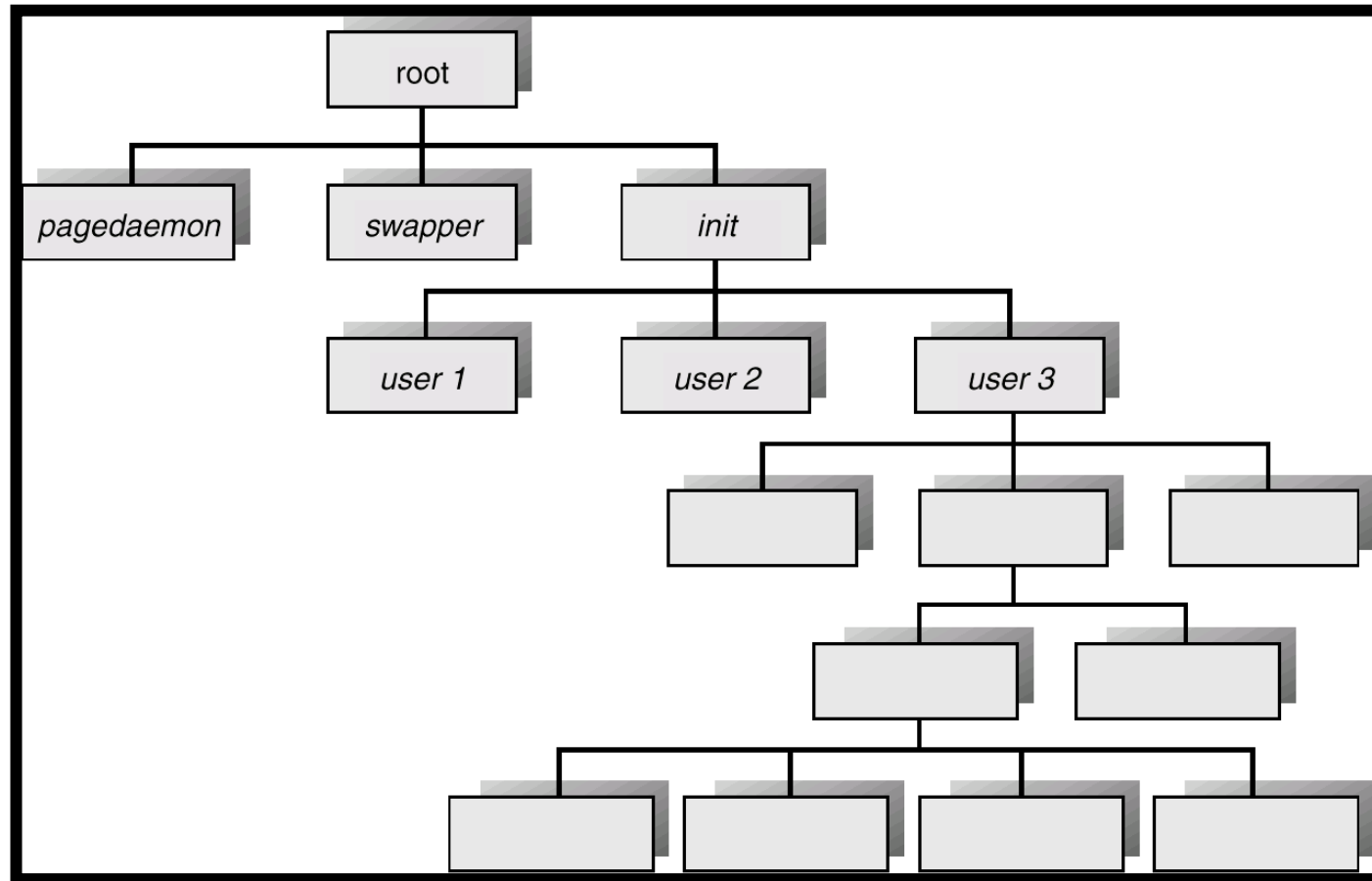
- Normal Çıkış (Gönüllü).
- Hatadan Dolayı Çıkış (Gönüllü).
- Onarılamaz (Fatal) Hatadan Dolayı Çıkış (Gönülsüz).
- Diğer Bir Proses Tarafından Sonlandırma (Gönülsüz).

1. Birçok proses görevlerini bitirdikten sonra sonlandırılır. Derleyici bir programı derlediğinde son satıra işletim sistemine, prosesi sonlandırmak için bir sistem çağrısı ekler. Bu çağrı UNIX’de “**exit**” ve Windows’ta ise “**ExitProcess**” dir.
2. İkinci sebep ise, program kodlarındaki bir hatadan dolayı proses sonlandırılır. Örneğin sıfıra bölünme hatası, yanlış bir komut icrası, olmayan bellek bölgesini referans etme gibi.

- Proses sonlandırmaya neden olan üçüncü durum ise programın ölümcül hata (fatal error) ile karşı karşıya kalmasıdır. Örneğin, kullanıcının olmayan bir dosyayı açmak için komut vermesi durumunda derleyici hemen prosesin sonlandırması için çağrı yapar.
- Dördüncü neden ise, bir proses işletim sistemine diğer bir prosesi sonlandırmak için çağrıda bulunur. UNIX’de bu çağrı “**kill**”, Windows’da ise “**TerminateProcess**” çağrısıdır. Yanlız bir prosesin diğer bir prosesi sonlandırabilmesi için gerekli yetkilere sahip olması gerekir.

PROSES HİYERARŞİSİ

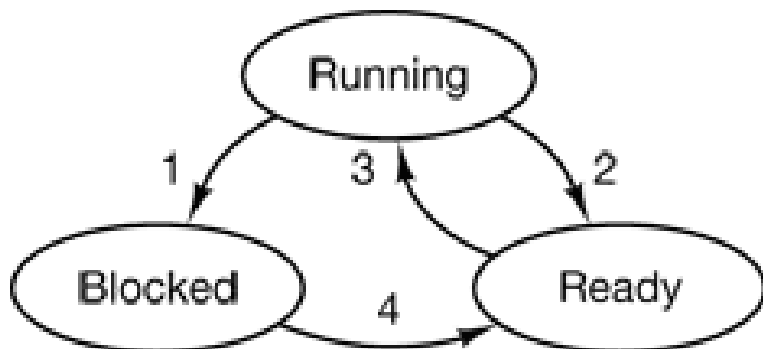
- Bazı işletim sistemlerinde, bir proses yeni bir proses oluşturduğu zaman, ana proses ile çocuk proses arasındaki ilişki belirli ölçüde devam eder.
- UNIX’de, bir proses, bu procesten oluşturulan çocuk proses ve bu çocuk proseslerin oluşturduğu diğer proseslerin hepsi bir proses grubu oluşturur. Kullanıcı klavyeden bir sinyal gönderdiğinde, bu sinyal gruba bağlı bütün proseslere iletilir. Gruptaki ilgili proses ya da prosesler bu sinyalleri dikkate alırken diğer prosesler ise bu sinyali yok sayarlar.
- Buna karşılık, Windows ‘da proses hiyerarşisi yoktur ve bütün prosesler eşit durumdadır.



PROSES DURUMLARI

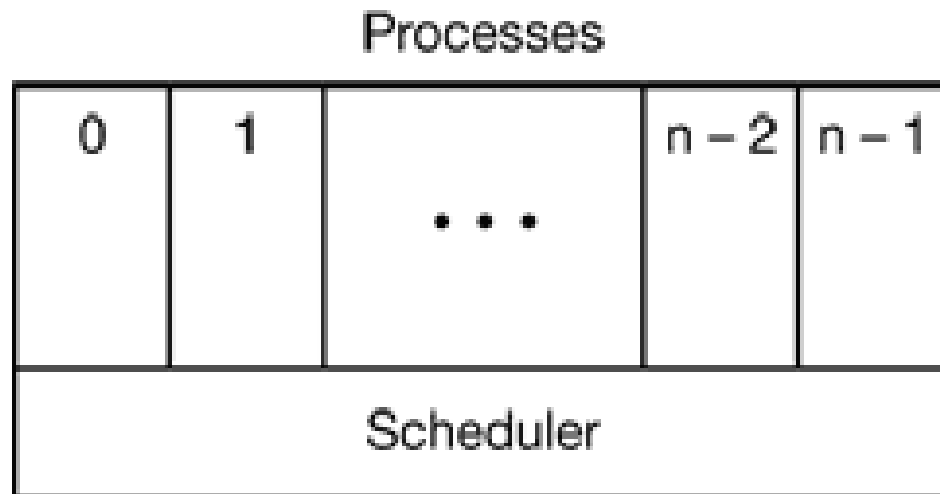
- Eğer bir proses işlemcide çalışırken, dış kaynaklı bir giriş verisine ihtiyaç duyarsa, ihtiyaç duyduğu veri gelinceye kadar bloklanır.
- Bazen de çalışmakta olan bir prosesin işleme devam etmesine herhangi bir engel olmasa işletim sistemi tarafından durdurulup, bellekte hazır konumda beklemeye alınabilir. Çünkü işletim sistemi diğer bir prosesin işlemciyi kullanmasına karar vermiştir.
- Bu iki durum tamamıyla farklı durumlardır. Birincisinde proses kesinlikle askıya alınmalıdır. İkincisinde ise işlemcinin farklı prosesler arasında anahtarlanması gerektiği gibi bir teknik ayrıntıdan dolayı çalışabilecek durumda olan proses belirli bir müddet bellekte beklemeye alınmıştır.

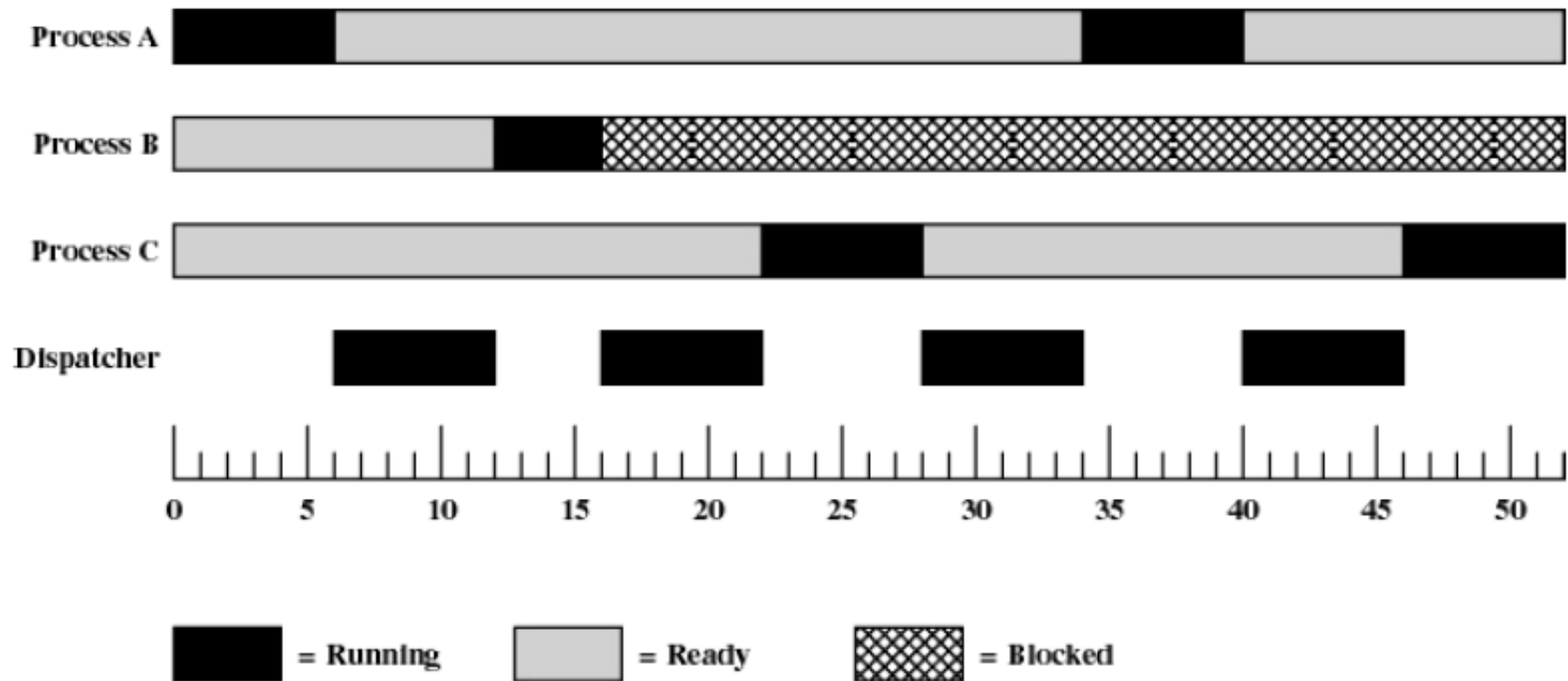
- Aşağıdaki şekilde Proses-Durum Diyagramı gösterilmektedir.
- 1. **Running (Çalışıyor):** Proses şu anda işlemci içinde çalıştırılıyor.
- 2. **Ready (Hazır):** Proses çalışabilecek durumda, fakat diğer proseslerde işlemciden faydalanması için geçici olarak durduruldu.
- 3. **Blocked (Bloklanmış):** Proses dış kaynaklı bir olay gerçekleşinceye kadar çalışamaz.



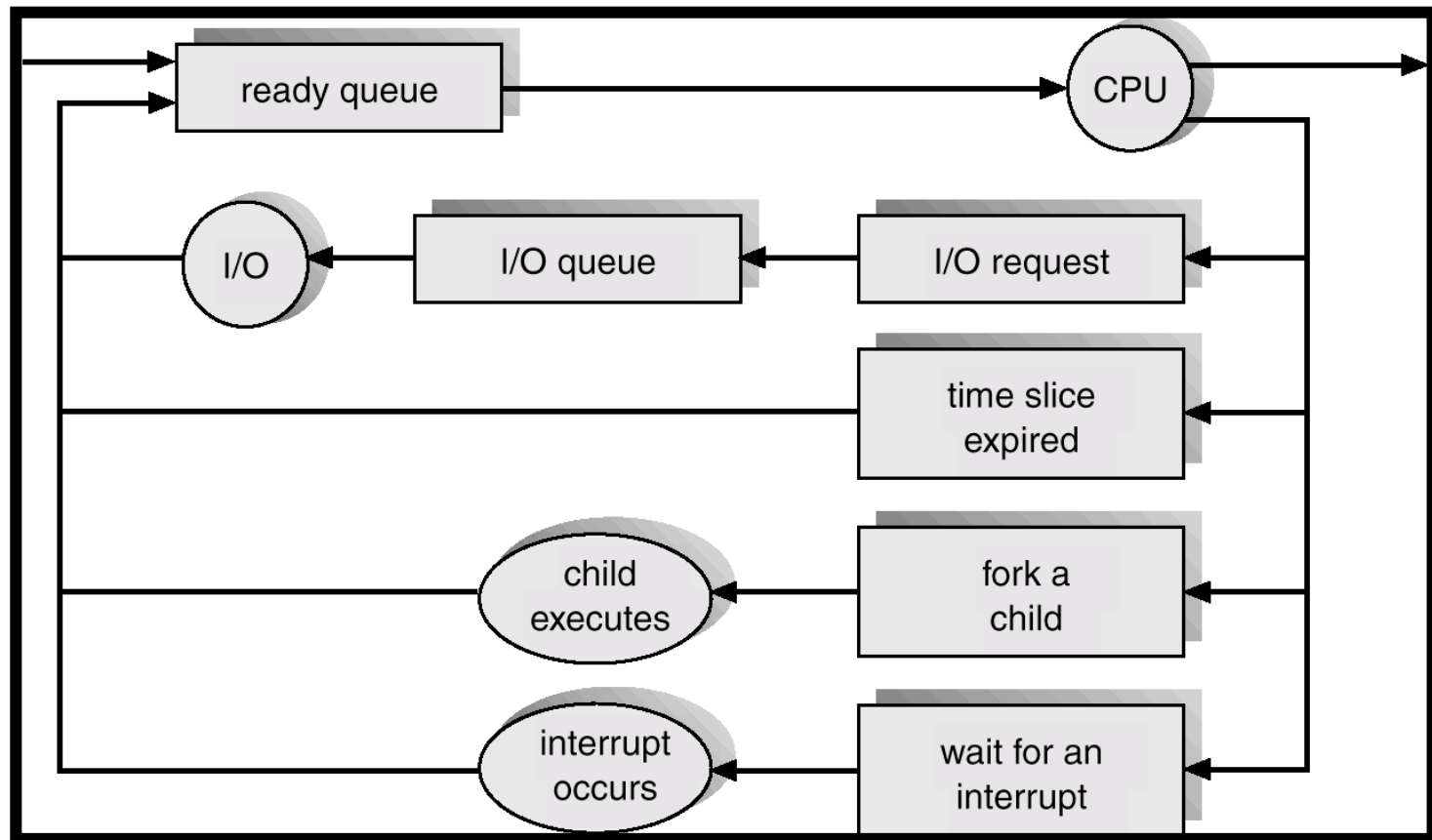
1. Process blocks for input
2. Scheduler picks another process
3. Scheduler picks this process
4. Input becomes available

- 2 ve 3 nolu ok yönündeki geçişler işletim sisteminin bir parçası olan Scheduler tarafından sağlanmıştır.
- Aşağıdaki şekilde Scheduler'ın yapısı gösterilmiştir. Scheduler işlemciye girecek ve işlemciden çıkacak prosesi belirleyen kodlara sahip bir işletim sistemi programıdır.





Şekil: Görevlerin durumları



Aşağıdaki tabloda bir işletim sisteminin proses kontrolü için saklaması gereken bazı önemli veriler gösterilmiştir:

Process management	Memory management	File management
Registers Program counter Program status word Stack pointer Process state Priority Scheduling parameters Process ID Parent process Process group Signals Time when process started CPU time used Children's CPU time Time of next alarm	Pointer to text segment Pointer to data segment Pointer to stack segment	Root directory Working directory File descriptors User ID Group ID

Süreç Tablosu

A süreci
B süreci
C süreci
.
.
.
n. süreç



Yazmaçlar(registers)
Program sayacı
Program durum kelimesi
Yığın işaretçisi
Süreç durumu
Öncelik(priority)
Zamanlama parametreleri
Süreç numarası
Ana süreç
Süreç grubu
Sinyaller
Başlama zamanı
Kullanılan işlemci zamanı
...

PROSESLERİN İCRASI

- İşletim sistemleri, proseslerin kullanılabilmesi için Proses Tablosu (ya da Proses Kontrol Bloğu) adı verilen bir tabloyu kullanır. Bu tablo bir dizi yapısındadır ve bu tabloda her bir proses için ayrı bir kayıt tutulur.
- Her bir kayıt, ilgili prosesin durumu, program sayacı, yığın göstergesi, bellek alanı, açık dosyaları, zamanlama parametreleri gibi bir prosesin çalışıyor durumdan hazır ya da bloklanmış duruma geçerken saklanması gereken bütün verilerini tutar. Böylece bu proses işlemciye bir daha girdiğinde sanki hiç çalışmasına ara vermemiş gibi kaldığı yerden devam edebilir.

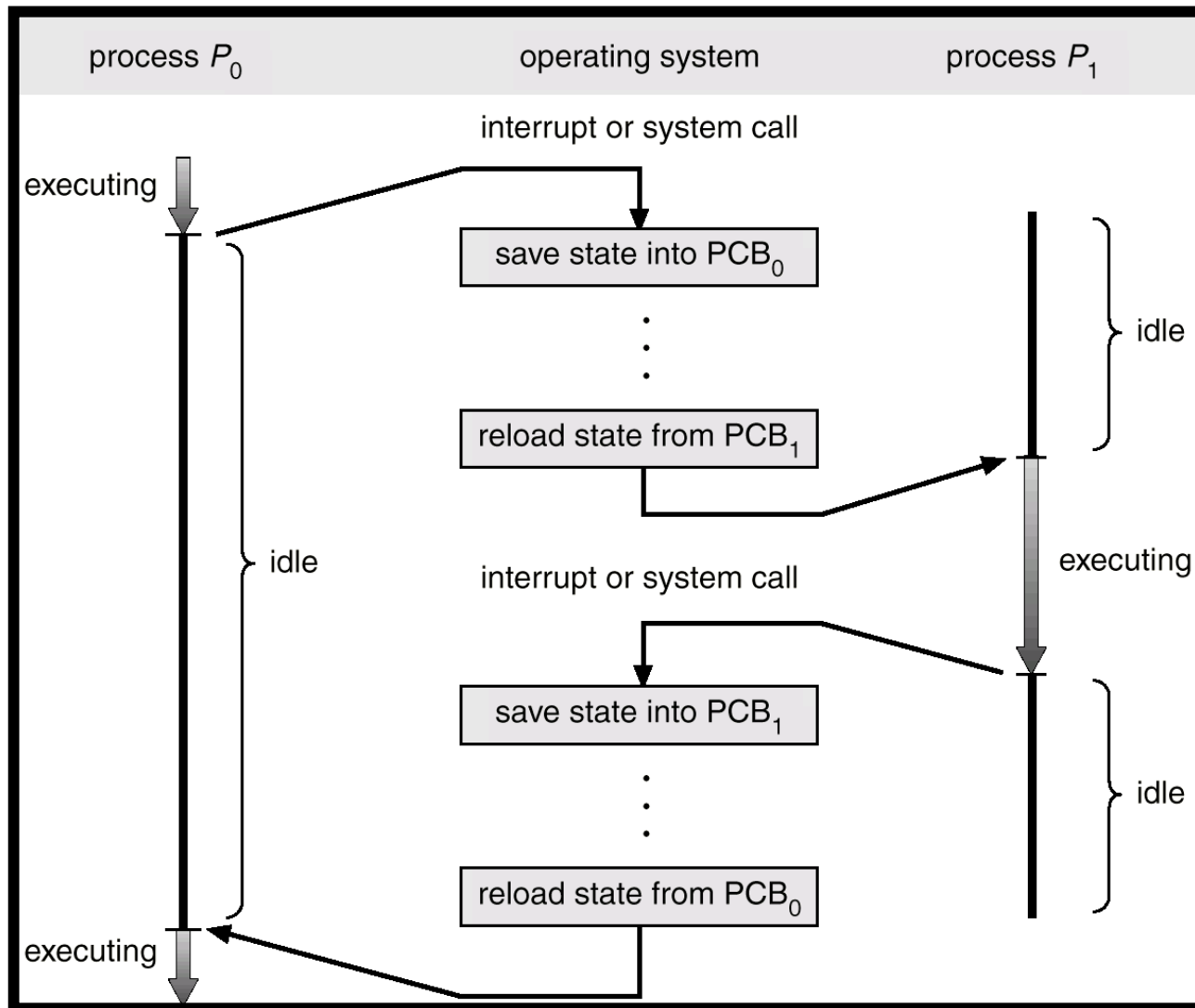
PROCESS TABLE (PROCESS CONTROL BLOCK)

PROCESS ID	PC	SP	PSW	PRIORITY	PARENT	USERID
------------	----	----	-----	----------	--------	--------	------

234							
-----	--	--	--	--	--	--	--

546							
-----	--	--	--	--	--	--	--

XXX							
-----	--	--	--	--	--	--	--



SORULAR ???