

2

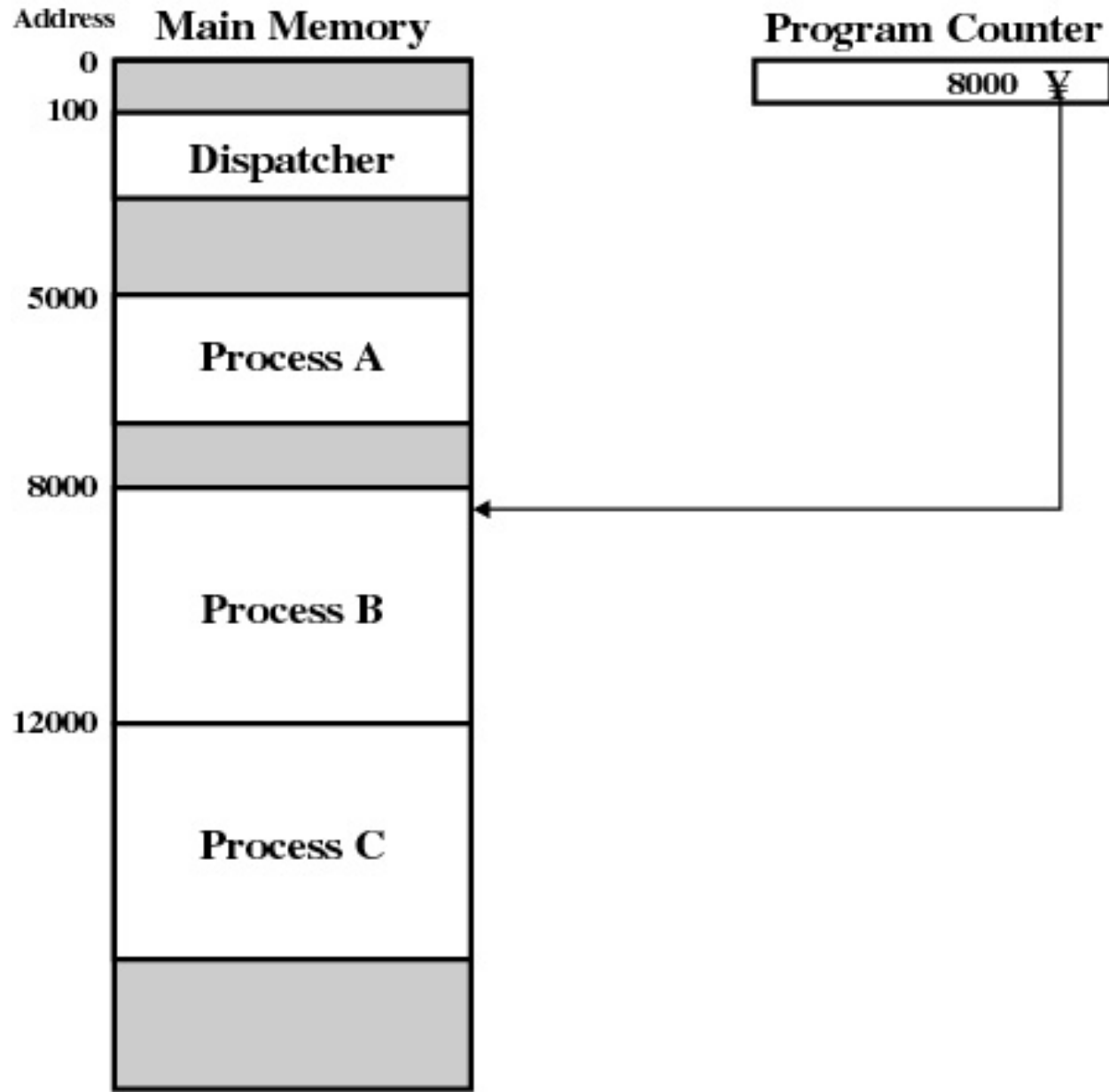
PROSESILER

Proses

- ▶ Bir işlevi gerçeklemek üzere ardışıl bir program parçasının yürütülmesiyle ortaya çıkan işlemler dizisi
⇒ Programın koşmakta olan hali
- ▶ Aynı programa ilişkin birden fazla proses olabilir.
- ▶ Görev (Task) de denir
- ▶ Text, veri ve yığın alanları vardır.

Proses

- ▶ Bazı sistem çağruları ile sistem kaynaklarını kullanırlar.
- ▶ Birbirleri ve dış dünya ile haberleşirler.
- ▶ Davranışını karakterize edebilmek için proses için yürütülen komutların sırası gözlenebilir: prosesin izi (trace)
- ▶ Prosesin ömrü: yaratılması ve sonlanması arasında geçen süre



5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011

(a) Trace of Process A

8000
8001
8002
8003

(b) Trace of Process B

12000
12001
12002
12003
12004
12005
12006
12007
12008
12009
12010
12011

(c) Trace of Process C

5000 = Starting address of program of Process A
8000 = Starting address of program of Process B
12000 = Starting address of program of Process C

```

1    5000
2    5001
3    5002
4    5003
5    5004
6    5005
-----Time out
7    100
8    101
9    102
10   103
11   104
12   105
13   8000
14   8001
15   8002
16   8003
-----I/O request
17   100
18   101
19   102
20   103
21   104
22   105
23   12000
24   12001
25   12002
26   12003
27   12004
28   12005
-----Time out
29   100
30   101
31   102
32   103
33   104
34   105
35   5006
36   5007
37   5008
38   5009
39   5010
40   5011
-----Time out
41   100
42   101
43   102
44   103
45   104
46   105
47   12006
48   12007
49   12008
50   12009
51   12010
52   12011
-----Time out

```

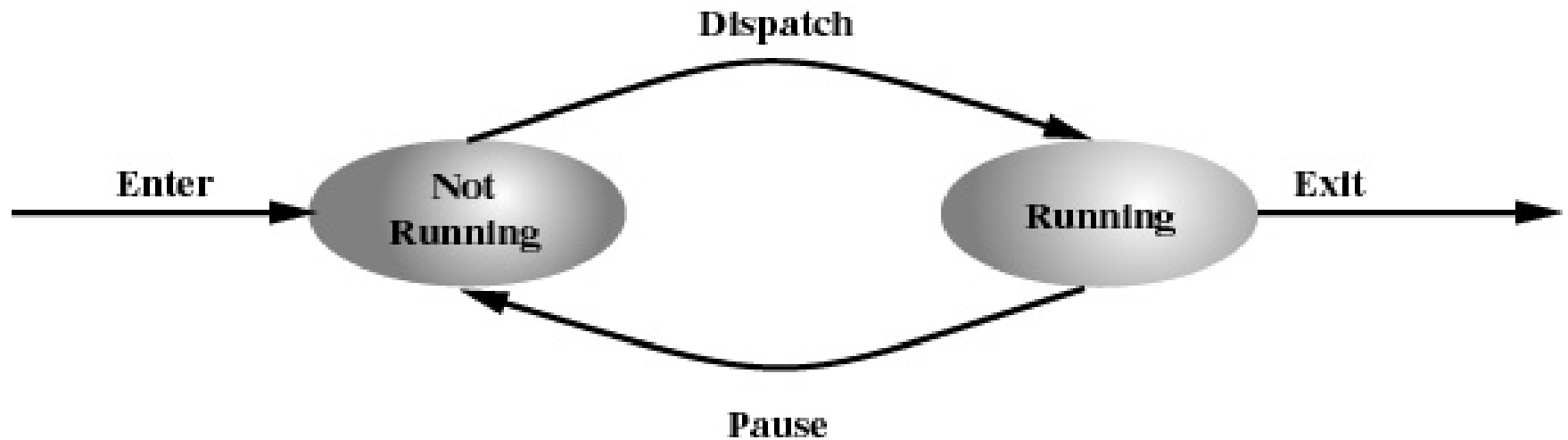
Proses

- ▶ Proseslerin işlemciye sahip olma sıraları kestirilemez \Rightarrow program kodunda zamanlamaya dayalı işlem olmamalı

İki Durumlu Proses Modeli

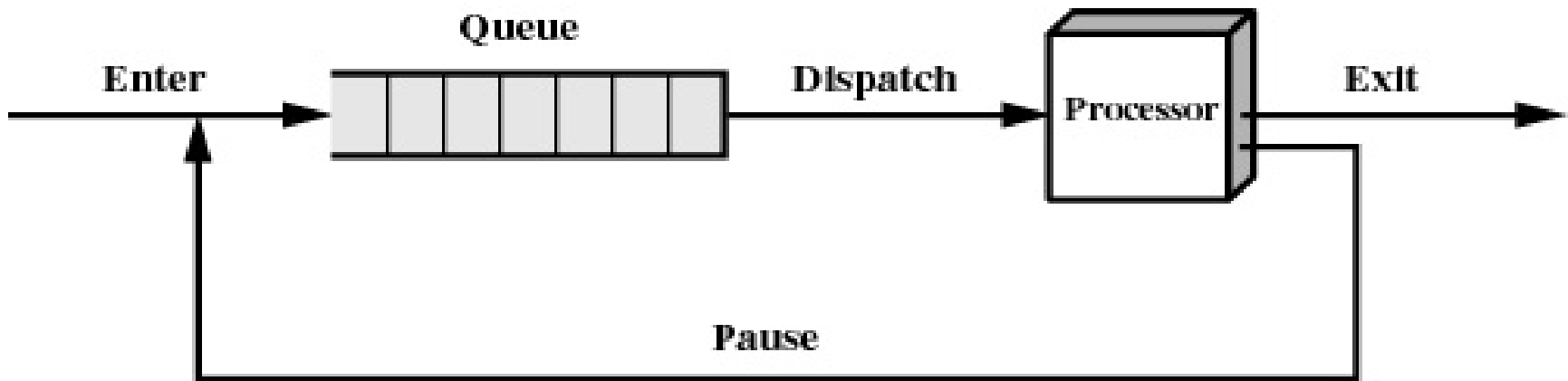
► Proses iki durumdan birinde olabilir:

- Koşuyor
- Koşmuyor



Proses Kuyruğu

O anda çalışmayan proses sırasını bir kuyrukta bekler:

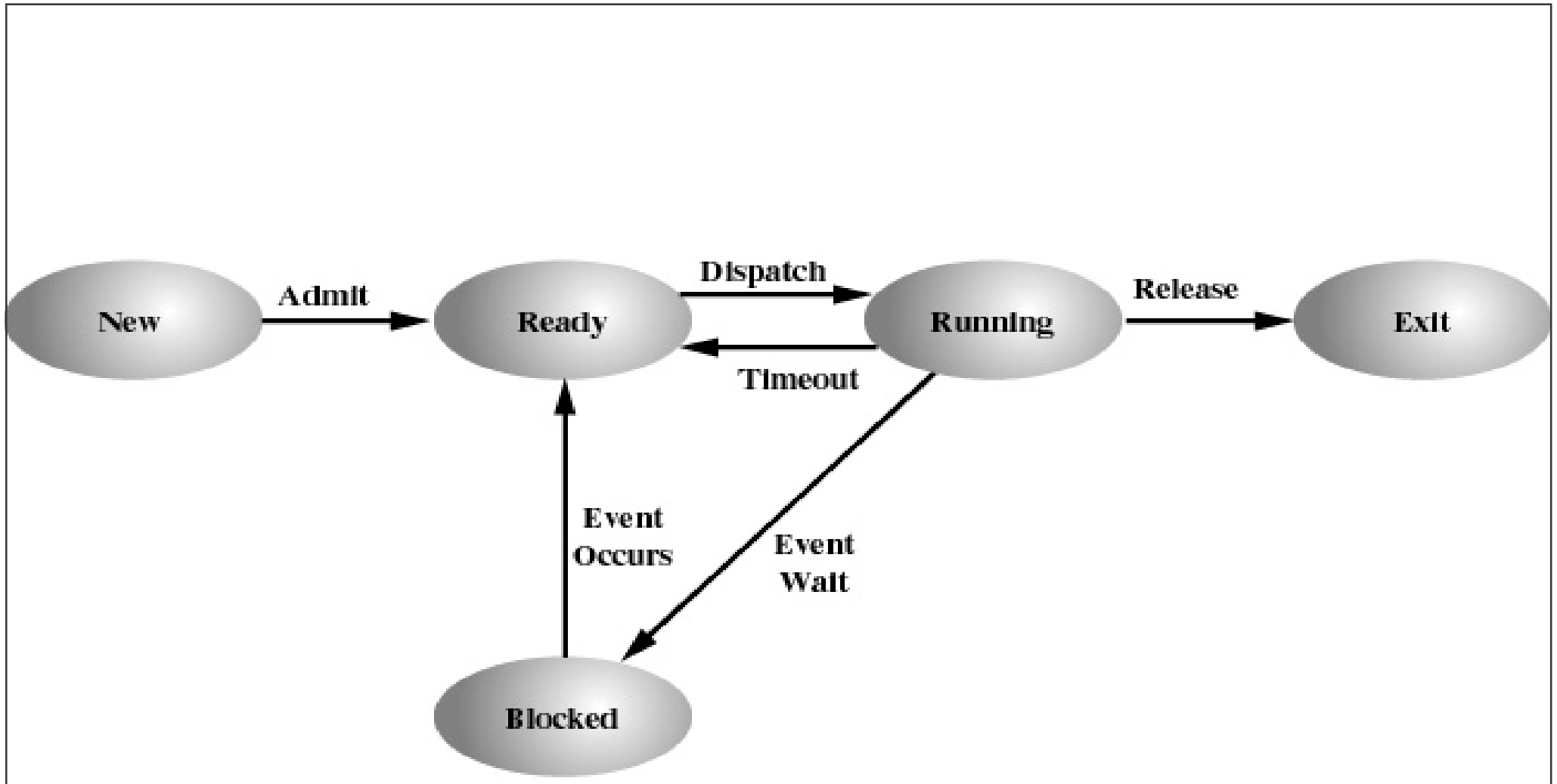


Proses

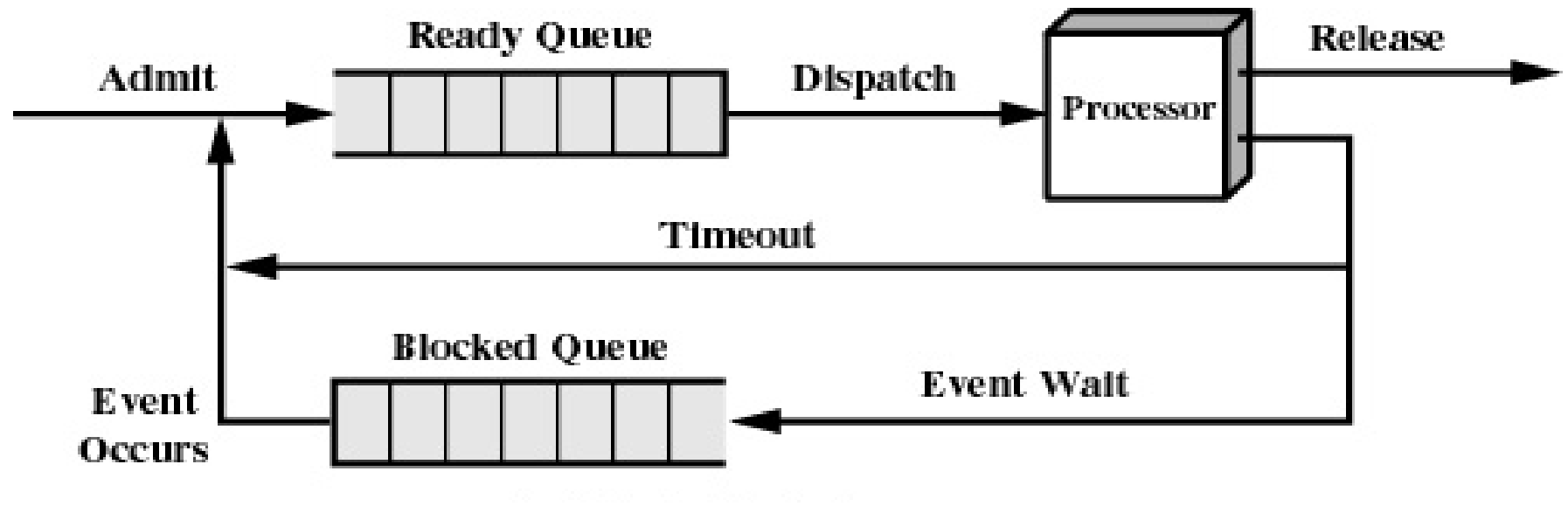
- ▶ Koşmuyor
 - çalışmaya hazır
- ▶ Bloke
 - G/Ç bekliyor
- ▶ Kuyrukta en uzun süre beklemiş prosesin çalıştırılmak üzere seçilmesi doğru olmaz
 - Bloke olabilir

Beş - Durumlu Model

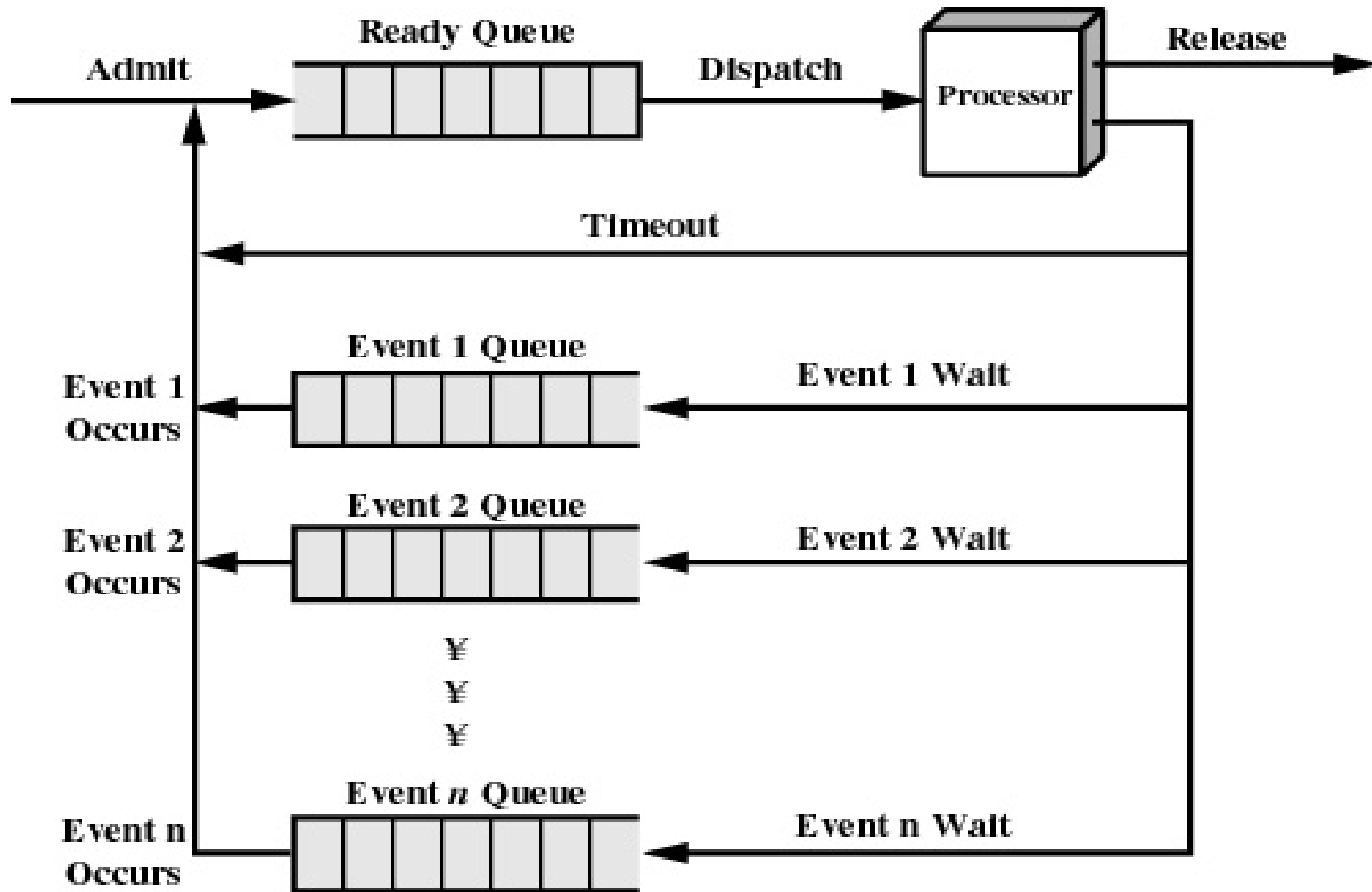
- ▶ Koşuyor
- ▶ Hazır
- ▶ Bloke
- ▶ Yeni
- ▶ Sonlanıyor



İki Kuyruk



Çoklu Kuyruk



Proses Yaratma

Ne zaman yaratılır?

- ▶ Kullanıcı sisteme girmiş
- ▶ Bir servis sunmak için
 - örneğin yazıcıdan çıktı
- ▶ Bir başka proses yaratmış

Proses Sonlandırma

Ne zaman sonlanır?

- ▶ Kullanıcı sistemden çıkmış
- ▶ Uygulama sonlandırılmış
- ▶ Hata durumu oluşmuş

Prosesin Askıya Alınma Nedenleri

- ▶ Swap işlemi
- ▶ Hatalı durum oluşması
- ▶ Etkileşimli kullanıcı isteği
 - Örneğin hata ayıklama (debug) için
- ▶ Ayrılan sürenin dolması (quantum)
- ▶ Anne proses tarafından

İşletim Sistemi Kontrol Yapıları

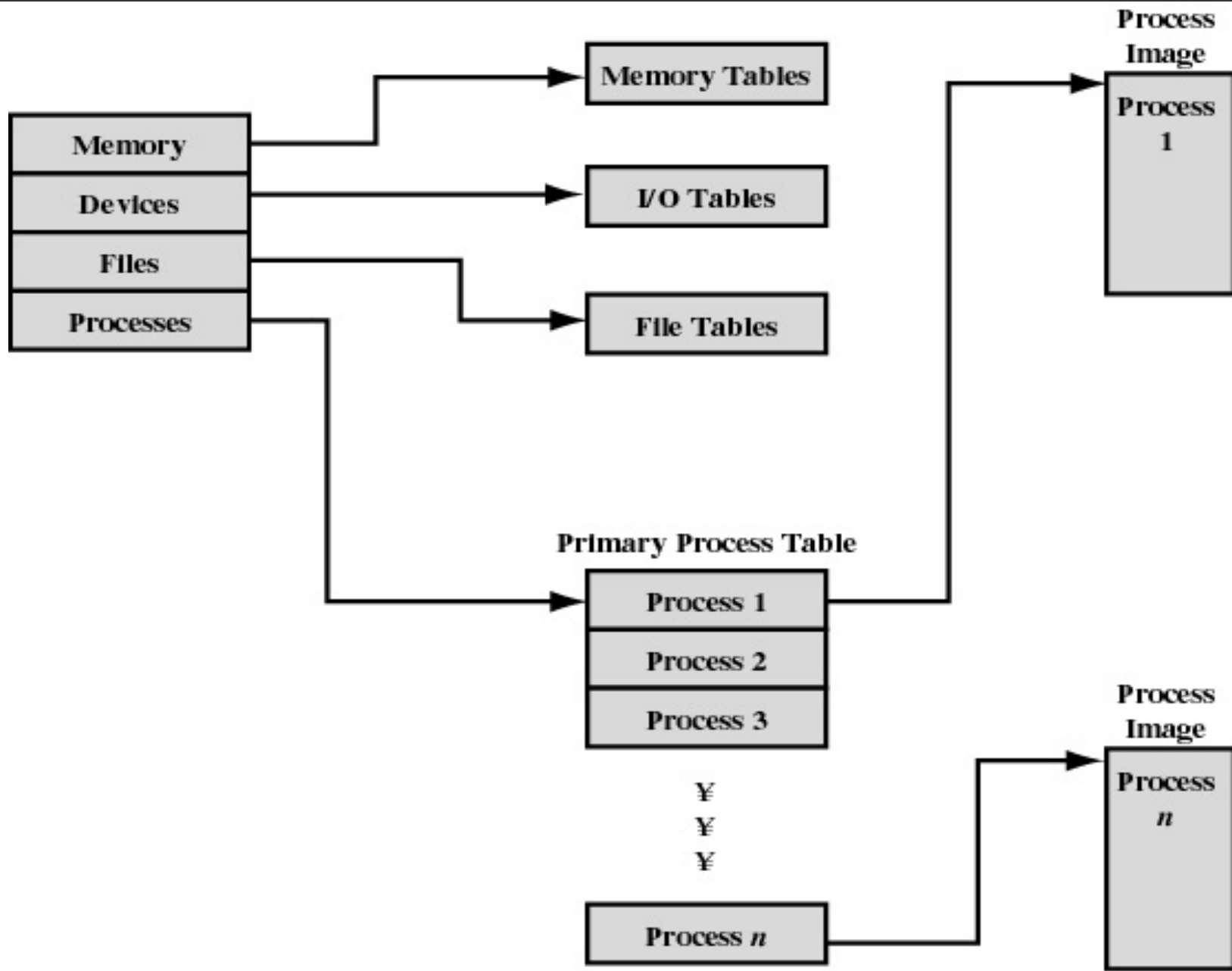
- ▶ Her proses ve kaynak ile ilgili durum bilgilerinin tutulması gerekir
 - İşletim sistemi tarafından yönetilen her varlık için tablolar tutulur
 - G/Ç Tabloları
 - Bellek Tabloları
 - Dosya Tabloları
 - Proses Tabloları

Proses Tablosu

- ▶ Prosesin bileşenleri
- ▶ Yönetilmesi için gerekli özellikleri
 - Kimlik numarası
 - Durumu
 - Bellekteki yeri

Prosesin Bileşenleri

- ▶ Proses birden fazla programdan oluşabilir
 - Yerel ve global değişkenler
 - Sabitler
 - Yığın
- ▶ Proses Kontrol Bloğu
 - Nitelikler (attributes)
- ▶ Prosesin görüntüsü
 - Program, veri, yığın ve niteliklerin tamamı



Proses Kontrol Bloğu

► Proses Kimlik Bilgileri

– Kimlik Bilgileri

- Prosesin kimlik numarası
- Prosesin annesinin kimlik numarası
- Sahibin kullanıcı kimlik bilgisi

Proses Kontrol Bloğu

► İşlemci Durum Bilgisi

- Kullanıcıya açık saklayıcılar
 - İşlemcinin makina dili kullanılarak erişilebilen saklayıcıları.
- Kontrol ve Durum saklayıcıları
 - Program sayacı
 - Durum saklayıcısı
 - Yiğın işaretçileri
 - Program durum sözcüğü (çalışma modu biti var)

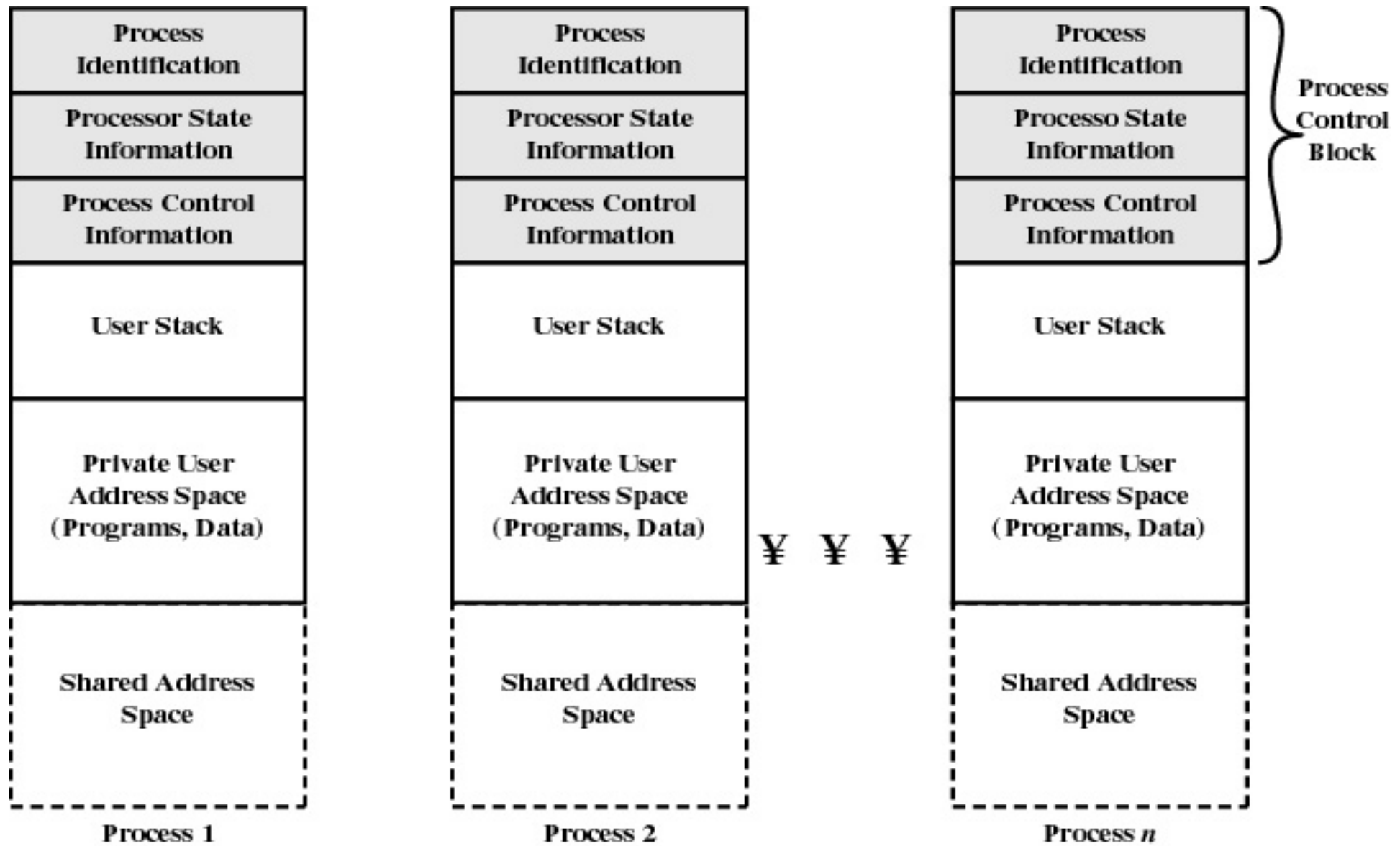
Proses Kontrol Bloğu

► Proses Kontrol Bilgileri

- İş sıralama ve durum bilgileri
 - Prosesin durumu
 - Önceliği
 - İş sıralama ile ilgili bilgiler (Hangi bilgiler olduğu kullanılan iş sıralama algoritmasına bağlı. Örneğin: bekleme süresi, daha önce koştuğu süre)
 - Çalışmak için beklediği olay
- Veri Yapıları
 - Prosesler örneğin bir çevrel kuyruk yapısında birbirlerine bağlı olabilir (örneğin aynı kaynağı bekleyen eş öncelikli prosesler).
 - Prosesler arasında anne-çocuk ilişkisi olabilir

Proses Kontrol Bloğu

- Prosesler arası haberleşme ile ilgili bilgiler
 - Bazı bayrak, sinyal ve mesajlar proses kontrol bloğunda tutulabilir.
- Proses Ayrıcalıkları
 - Bellek erişimi, kullanılacak komutlar ve sistem kaynak ve servislerinin kullanımı ile ilgili haklar
- Bellek yönetimi
 - Prosesse ayrılmış sanal bellek bölgesinin adresi
- Kaynak kullanımı
 - Prosesin kullandığı kaynaklar: örneğin açık dosyalar
 - Prosesin önceki işlemci ve diğer kaynakları kullanımına ilişkin bilgiler



Çalışma Modları

► Kullanıcı modu

- Düşük haklar ve ayrıcalıklar
- Kullanıcı programları genel olarak bu modda çalışır

► Sistem modu / çekirdek modu

- Yüksek haklar ve ayrıcalıklar
- İşletim sistemi çekirdeği prosesleri bu modda çalışır

Proses Yaratılması

- ▶ Proses kimlik bilgisi atanır: sistemde tek
- ▶ Proses için bellekte yer ayrılır
- ▶ Proses kontrol bloğuna ilk değerler yüklenir
- ▶ Gerekli bağlantılar yapılır: Örneğin iş sıralama için kullanılan bağlantılı listeye yeni proses kaydı eklenir.
- ▶ Gerekli veri yapıları yaratılır veya genişletilir: Örneğin istatistik tutma ile ilgili

Prosesler Arası Geçiş Durumu

- ▶ Saat kesmesi
 - proses kendisine ayrılan zaman dilimi kadar çalışmıştır
- ▶ G/Ç kesmesi
- ▶ Bellek hatası
 - erişilen bellek bölgesi ana bellekte yoktur
- ▶ Hata durumu
- ▶ Sistem çağrısı

Proseslerin Durum Deęiřtirmesi

- ▶ İşlemci bağlamının saklanması (program sayacı ve dięer saklayıcılar dahil)
- ▶ O anda kořmakta olan prosesin proses kontrol bloęunun güncellenmesi
- ▶ Prosesle iliřkin proses kontrol bloęunun uygun kuyruęa yerleřtirilmesi: hazır / bloke
- ▶ Kořacak yeni prosesin belirlenmesi

Proseslerin Durum Deęiřtirmesi

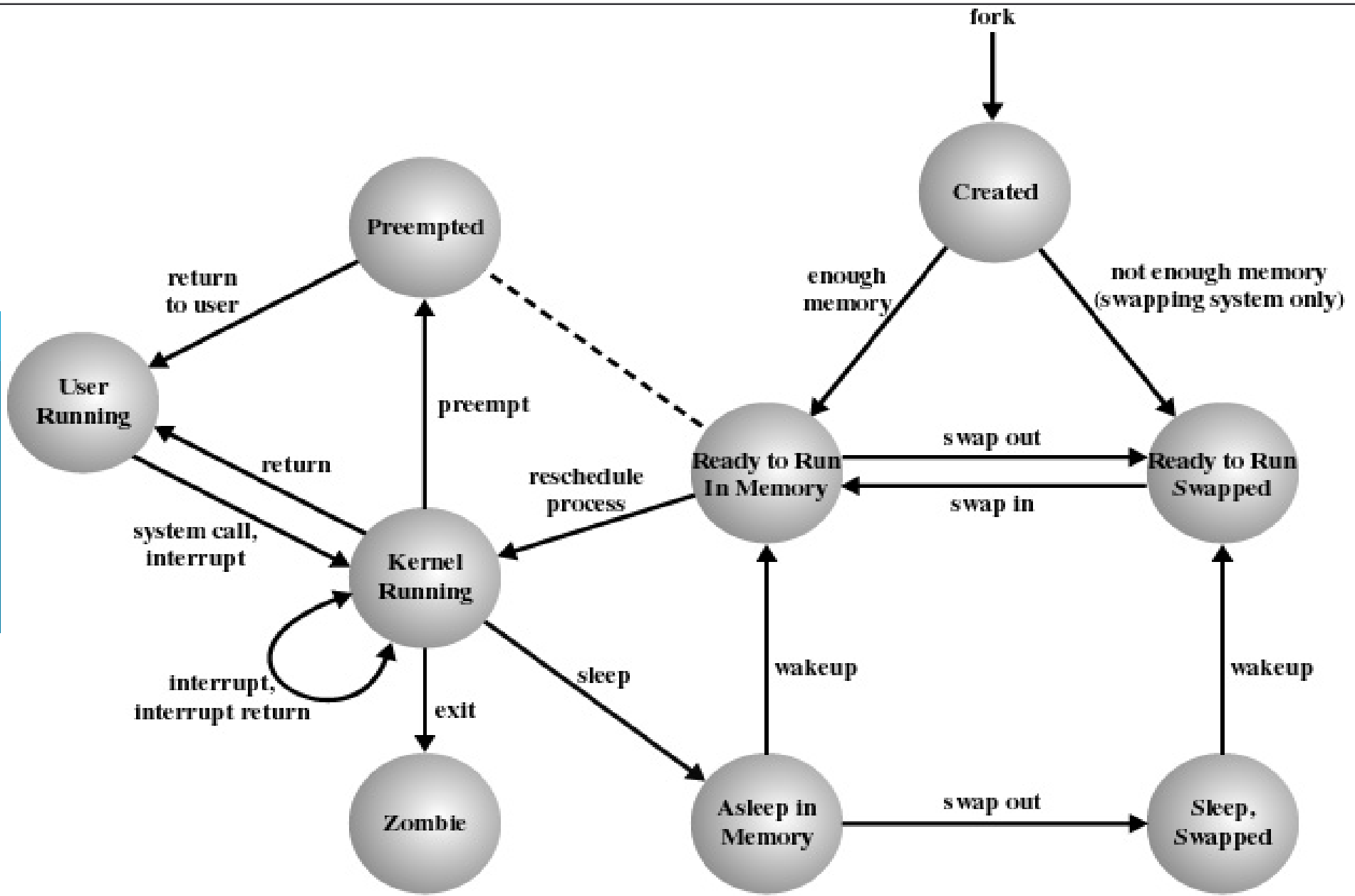
- ▶ Seęilen prosesin proses kontrol bloęunun g¼ncellenmesi
- ▶ Bellek y¼netimi ile ilgili bilgilerin g¼ncellenmesi
- ▶ Seęilen prosesin baęlamının y¼klenmesi

UNIX'te Proses Durumları

- ▶ Kullanıcı modunda koşuyor
- ▶ Çekirdek modunda koşuyor
- ▶ Bellekte ve koşturmaya hazır
- ▶ Bellekte uyuyor
- ▶ İkincil bellekte ve koşturmaya hazır
- ▶ İkincil bellekte uyuyor

UNIX'te Proses Durumları

- ▶ Pre-empt olmuş (çekirdek modundan kullanıcı moduna dönerken iş sıralayıcı prosesi kesip yerine bir başka prosesi çalışacak şekilde belirlemiş)
- ▶ Yaratılmış ama koşturmaya hazır değil
- ▶ Zombie (proses sonlanmış ancak anne prosesin kullanabilmesi için bazı kayıtları hala tutulmakta, ilgili kaynaklar henüz geri verilmemiş)



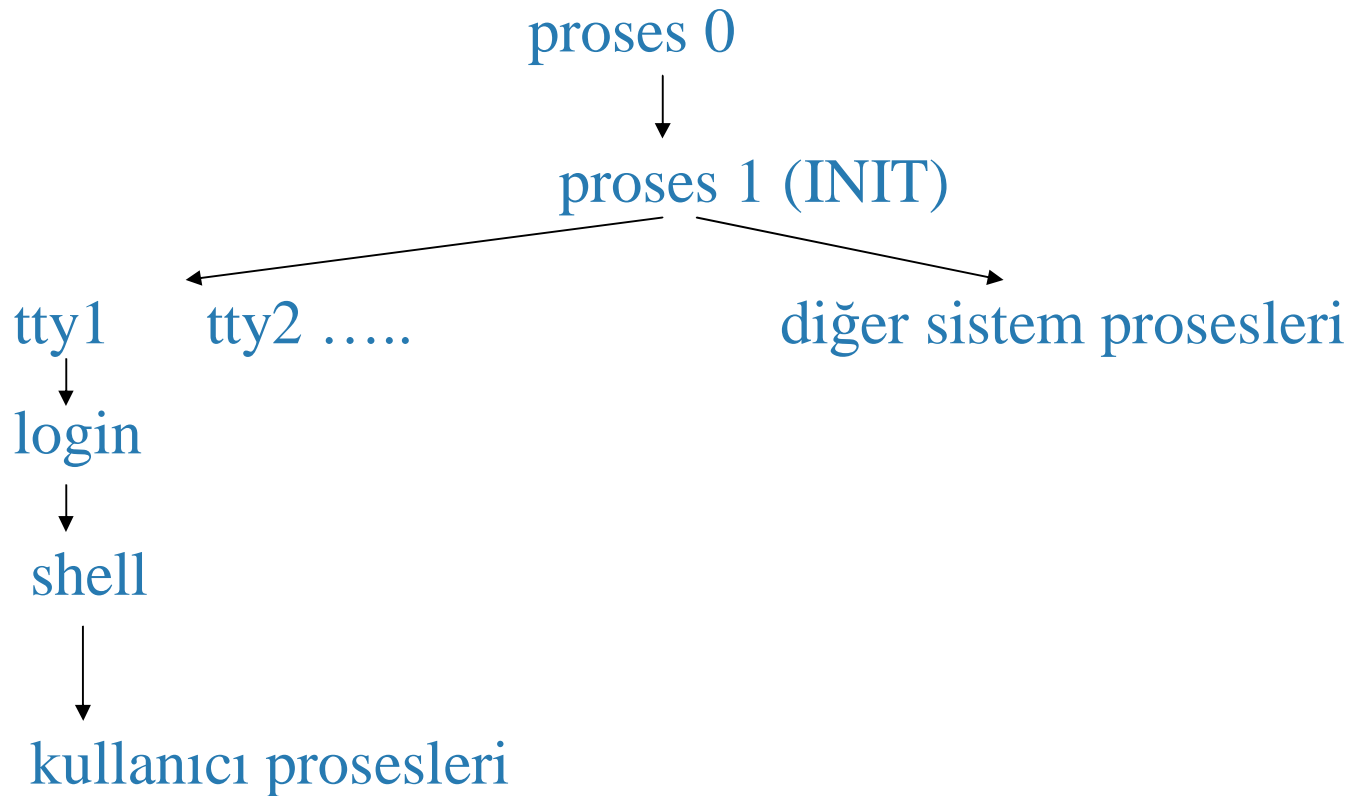
UNIX'de Proses Yaratma

- ▶ fork sistem çağrısı ile yaratılır
 - çağrıyı yapan proses: anne proses
 - Yaratılan proses: çocuk proses
- ▶ sentaksı `pid=fork()`
 - Her iki proses de aynı bağlama sahip
 - Anne prosese çocuğun kimlik değeri döner
 - Çocuk prosese 0 değeri döner
- ▶ 0 numaralı prosesi açılışta çekirdek yaratılır; fork ile yaratılmayan tek procestir

UNIX'de Proses Yaratma

- ▶ fork sistem çağrısı yapıldığında çekirdeğin yürüttüğü işlemler:
 - proses tablosunda (varsa) yer ayırılır (maksimum proses sayısı belli)
 - çocuk prosese yeni bir kimlik numarası atanır (sistemde tek)
 - Anne prosesin bağlamının kopyası çıkarılır.
 - Dosya erişimi ile ilgili sayaçları düzenler
 - anneye çocuğun kimliğini, çocuğa da 0 değerini döndürür

UNIX'de fork Sistem Çağrısı ile Proses Yaratılma Hiyerarşisi



UNIX'de Proses Sonlanması

- ▶ exit sistem çağrısı ile
- ▶ sentaksı: `exit(status)`
 - “status” değeri anne prosese aktarılır
- ▶ Tüm kaynakları geri verilir
- ▶ Dosya erişim sayaçları düzenlenir
- ▶ Proses tablosu kaydı silinir
- ▶ Annesi sonlanan proseslerin annesi olarak init prosesi (1 numaralı proses) atanır

Örnek Program Kodu - 1

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int f;

int main (void)
{
    printf("\n Program calisiyor: PID=%d \n",
           getpid());
    f=fork();
```

Örnek Program Kodu - 2

```
if (f==0) /*cocuk*/
{
    printf("\nBen cocuk. Kimlik= %d\n", getpid());
    printf("Annemin kimliđi=%d\n", getppid());
    sleep(2);
    exit(0);
}
else /* anne */
{
    printf("\nBen anne. Kimlik= %d\n", getpid());
    printf("Annemin kimliđi=%d\n", getppid());
    printf("Cocugumun kimliđi=%d\n", f);
    sleep(2);
    exit(0);
}
return(0);
}
```