

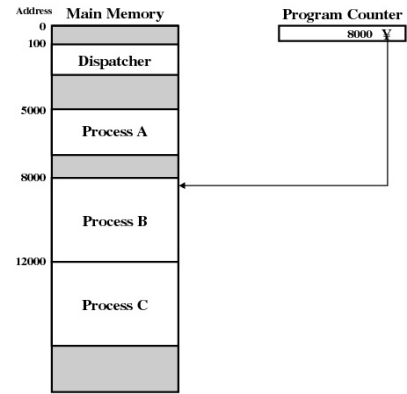
2 PROSESLER

Proses

- Bir işlevi gerçeklemek üzere ardışıl bir program parçasının yürütülmesiyle ortaya çıkan işlemler dizisi
⇒ Programın koşturma halı
- 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	8000	12000
5001	8001	12001
5002	8002	12002
5003	8003	12003
5004		12004
5005		12005
5006		12006
5007		12007
5008		12008
5009		12009
5010		12010
5011		12011

(a) Trace of Process A

(b) Trace of Process B

(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

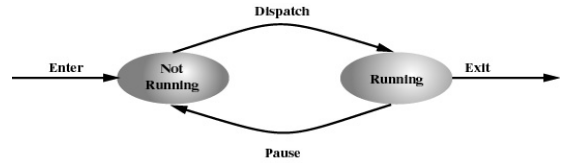
1	5000	27	12004
2	5001	28	12005
3	5002		Time out
4	5003	29	100
5	5004	30	101
6	5005	31	102
	Time out	32	103
7	100	33	104
8	101	34	105
9	102	35	5006
10	103	36	5007
11	104	37	5008
12	105	38	5009
13	8000	39	5010
14	8001	40	5011
15	8002		Time out
16	8003	41	100
	I/O request	42	101
17	100	43	102
18	101	44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

Proses

- Proseslerin işlemciye sahip olma sıraları kestirilemez ⇒ program kodunda zamanlamaya dayalı işlem olmamalı

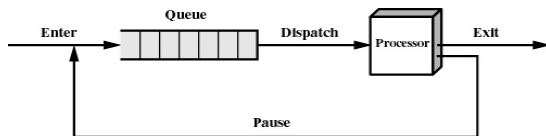
İki Durumlu Proses Modeli

- Proses iki durumdan birinde olabilir:
 - Koşuyor
 - Koşmuyor



Proses Kuyruğu

O anda çalışmayan proses sırasını bir kuyruқта bekler:

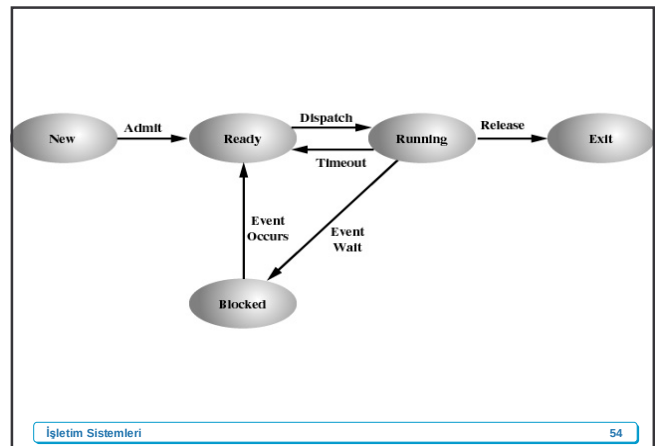


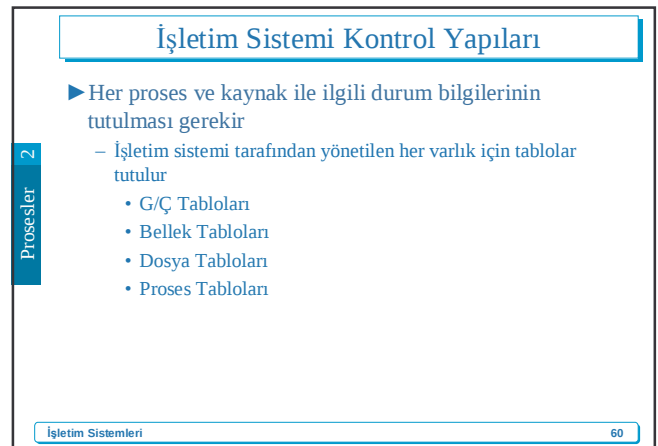
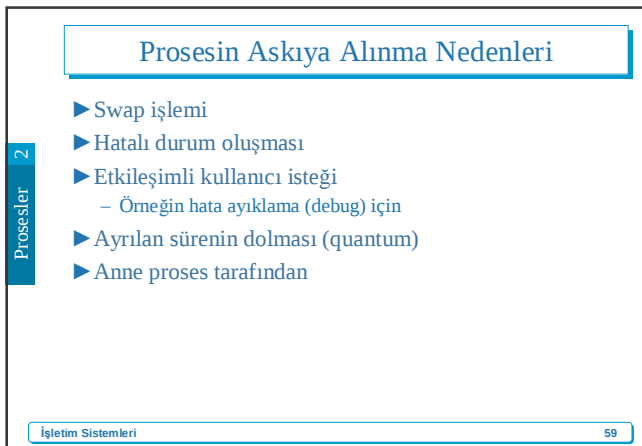
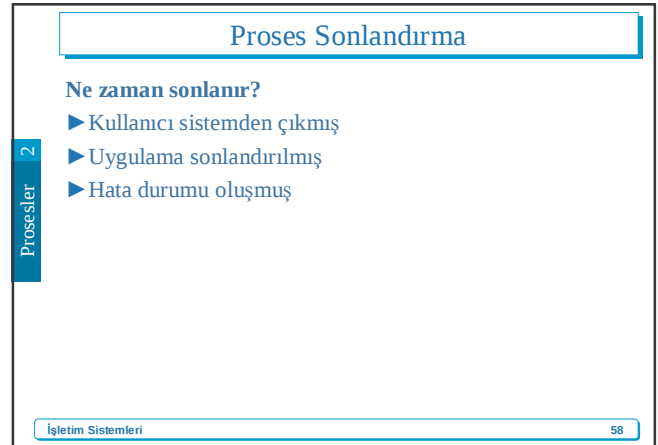
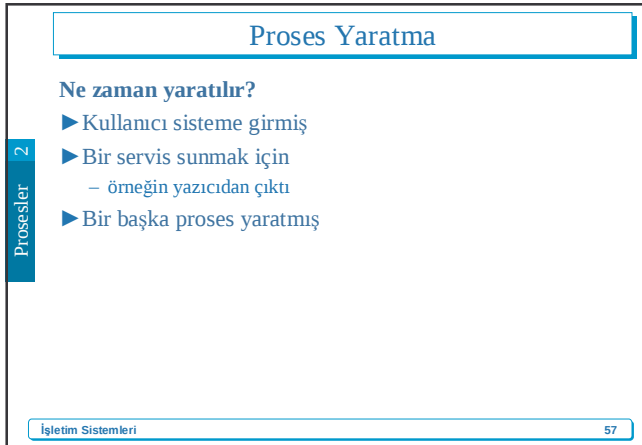
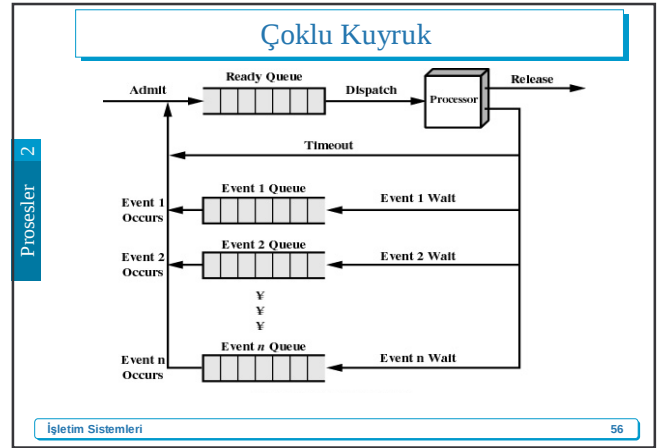
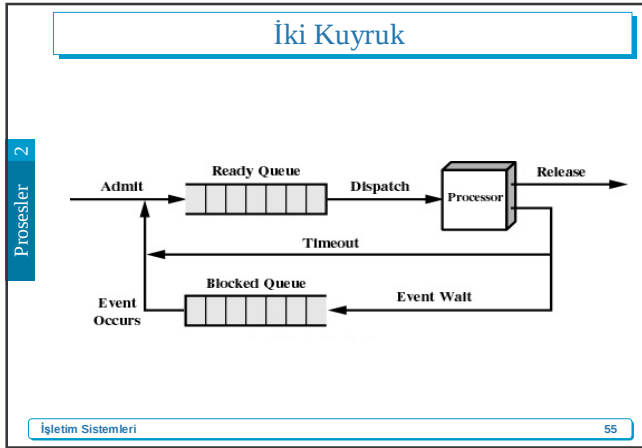
Proses

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

Beş - Durumlu Model

- Koşuyor
- Hazır
- Bloke
- Yeni
- Sonlanıyor



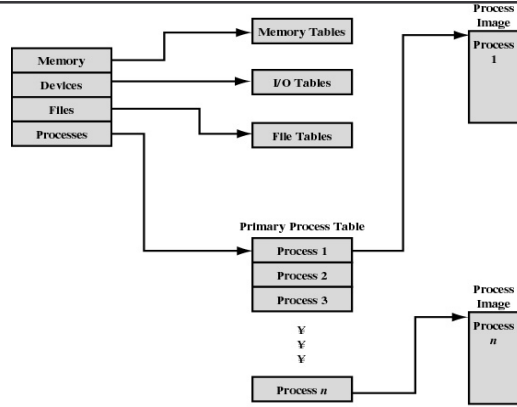


Proses Tablosu

- 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

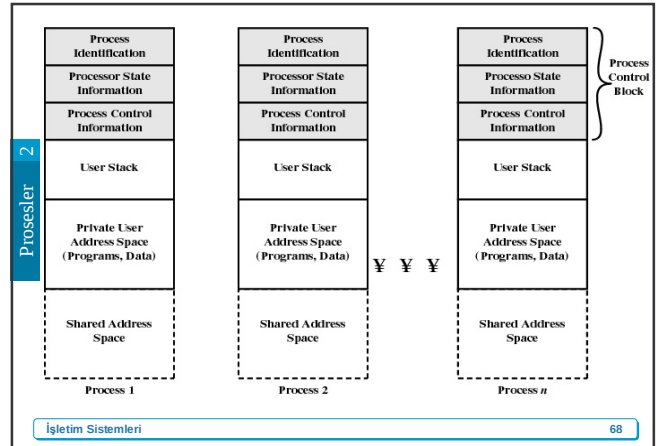
- İşlemci Durum Bilgisi
 - Kullanıcıya açık saklayıcılar
 - İşlemcinin makina dili kullanılarak erişilebilen saklayıcıları.
 - Kontrol ve Durum saklayıcıları
 - Program sayacı
 - Durum saklayıcısı
 - Yığın işaretçileri
 - Program durum sözcüğü (çalışma modu biti var)

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 Ayrıcılığı
 - Bellek erişimi, kullanılabilir komutlar ve sistem kaynak ve servislerinin kullanımı ile ilgili haklar
- Bellek yönetimi
 - Prosese 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 çağırısı

Proseslerin Durum Değiştirmesi

- İşlemci bağlamının saklanması (program sayacı ve diğer saklayıcılar dahil)
- O anda koşturmakta olan prosesin proses kontrol bloğunun güncellenmesi
- Prosese ilişkin proses kontrol bloğunun uygun kuyruğa yerleştirilmesi: hazır / bloke
- Koşacak yeni prosesin belirlenmesi

Proseslerin Durum Değiřtirmesi

- Seçilen prosesin proses kontrol bloğunun güncellenmesi
- Bellek yönetimi ile ilgili bilgilerin güncellenmesi
- Seçilen prosesin bağlamının yüklenmesi

Prosesler 2

UNIX'te Proses Durumları

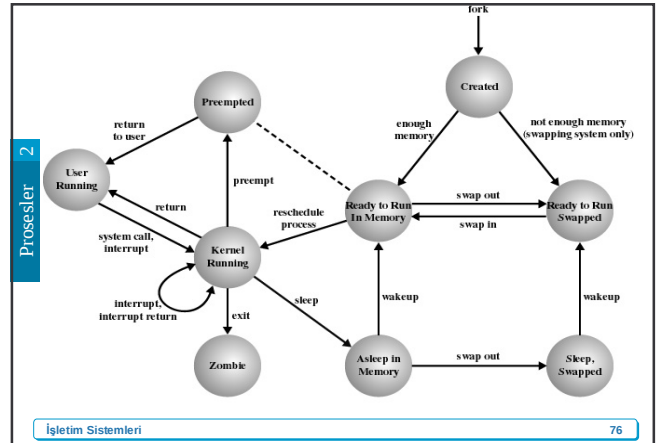
- Kullanıcı modunda koşuyor
- Çekirdek modunda koşuyor
- Bellekte ve kořmaya hazır
- Bellekte uyuyor
- İkincil bellekte ve kořmaya hazır
- İkincil bellekte uyuyor

Prosesler 2

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 kořmaya hazır deęil
- Zombie (proses sonlanmış ancak anne prosesin kullanabilmesi için bazı kayıtları hala tutulmakta, ilgili kaynaklar henüz geri verilmemiş)

Prosesler 2



Prosesler 2

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 açılıřta çekirdek yaratılır; fork ile yaratılmayan tek prosestir

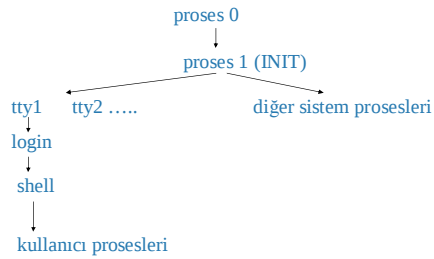
Prosesler 2

UNIX'de Proses Yaratma

- fork sistem çağrısı yapıldığında çekirdeğin yürüttüęü işlemler:
 - proses tablosunda (varsa) yer ayrılır (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

Prosesler 2

UNIX’de fork Sistem Çağrısı ile Proses Yaratılma Hiyerarşisi



UNIX’de Proses Sonlanması

- exit sistem çağrısı ile
- sentaksı: `exit(status)`
 - “status” değeri anne procese 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 kimliği=%d\n", getppid());
    sleep(2);
    exit(0);
}
else /* anne */
{
    printf("\nBen anne. Kimlik= %d\n", getpid());
    printf("Annemin kimliği=%d\n", getppid());
    printf("Cocugumun kimliği=%d\n", f);
    sleep(2);
    exit(0);
}
return(0);
}
```