1906003052015

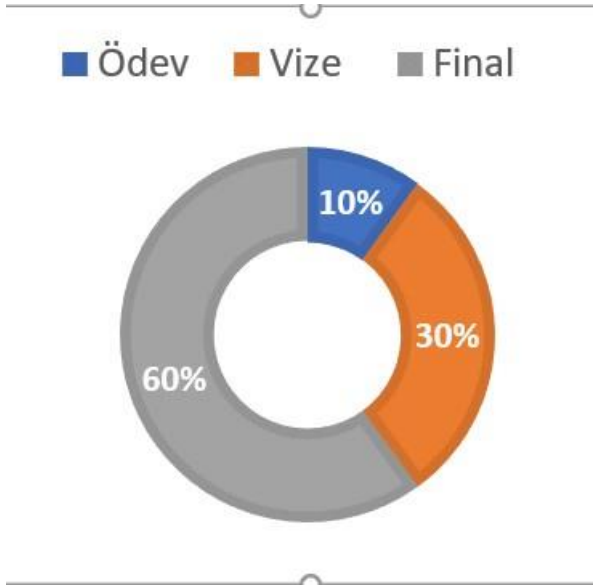
İşletim Sistemleri

Dr. Öğr. Üy. Önder EYECİOĞLU
Bilgisayar Mühendisliği

Giriş

Ders Günü ve Saati:
Çarşamba: 13:00-16:00

- Uygulama Unix (Linux) İşletim sistemi
- Devam zorunluluğu %70
- Uygulamalar C programlama dili üzerinde gerçekleştirilecektir. Öğrencilerden programlama bilgisi beklenmektedir.



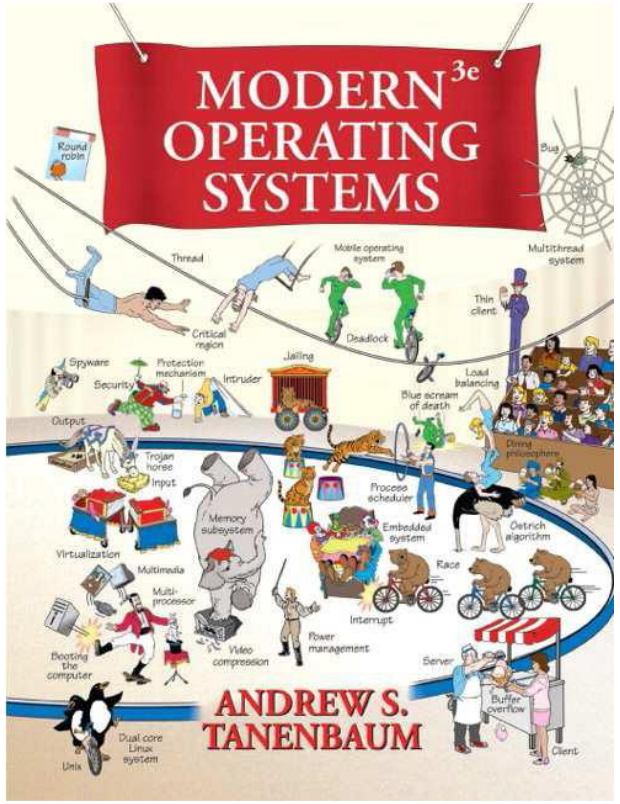
HAFTA	KONULAR
Hafta 1	: İşletim sistemlerine giriş, İşletim sistemi stratejileri
Hafta 2	: Sistem çağrıları
Hafta 3	: Görev, görev yönetimi
Hafta 4	: İplikler
Hafta 5	: İş sıralama algoritmaları
Hafta 6	: Görevler arası iletişim ve senkronizasyon
Hafta 7	: Semaforlar, Monitörler ve uygulamaları
Hafta 8	: Vize
Hafta 9	: Kritik Bölge Problemleri
Hafta 10	: Kilitlenme Problemleri
Hafta 11	: Bellek Yönetimi
Hafta 12	: Sayfalama, Segmentasyon
Hafta 13	: Sanal Bellek
Hafta 14	: Dosya sistemi, erişim ve koruma mekanizmaları, Disk planlaması ve Yönetimi
Hafta 15	: Final

Giriş

Kaynaklar:

– Modern Operating Systems, 3rd Edition by Andrew S. Tanenbaum, Prentice Hall, 2008.

– Bilgisayar İşletim Sistemleri (BIS), Ali Saatçi, 2. Baskı, Bıçaklar Kitabevi.



Görev Çizelgeleme

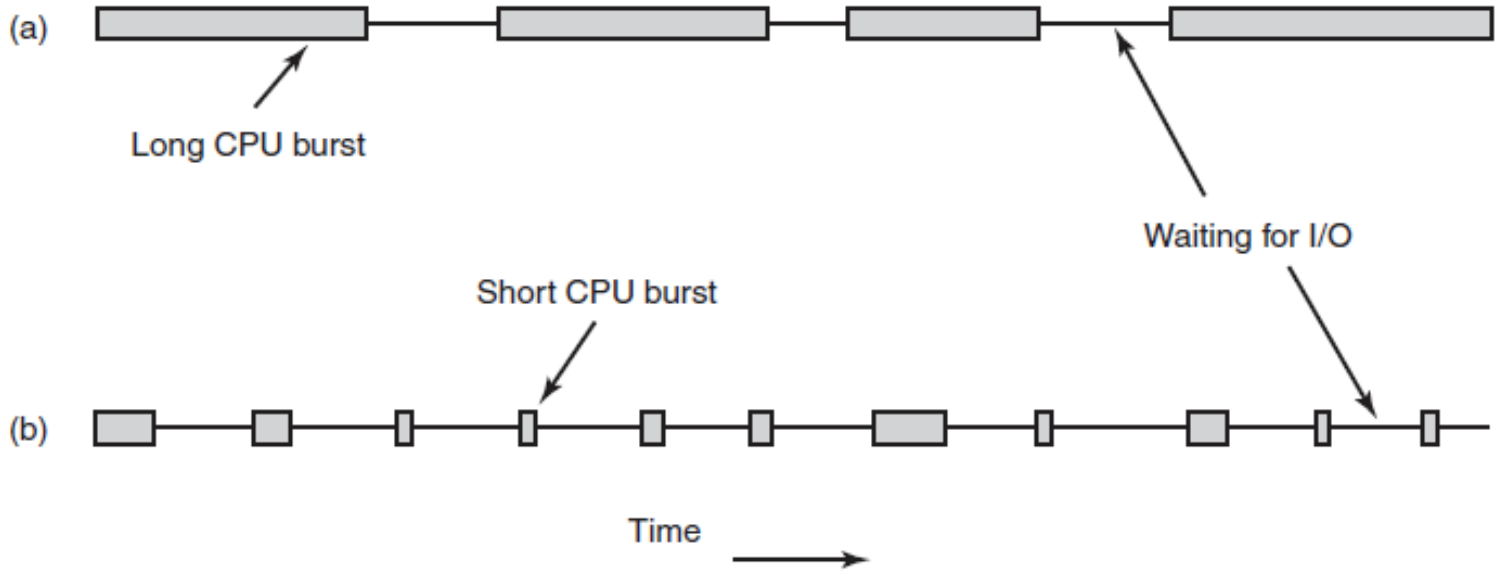
Süreç çizelgeleme, çalışan sürecin CPU'dan çıkarılmasını ve belirli bir strateji temelinde başka bir sürecin seçilmesini yöneten süreç yöneticisinin etkinliğidir.

Süreç çizelgeleme, Çoklu Programlama işletim sistemlerinin önemli bir parçasıdır. Bu tür işletim sistemleri, aynı anda birden fazla işlemin yürütülebilir belleğe yüklenmesine izin verir ve yüklenen işlem, zaman çoklama (time multiplexing) kullanarak CPU'yu paylaşır

Tek bir CPU çekirdeğine sahip bir sistemde, aynı anda yalnızca bir işlem çalışabilir. Diğerleri, CPU'nun çekirdeği boşalana ve yeniden planlanabilene kadar beklemek zorundadır. Çoklu programlamanın amacı, CPU kullanımını en üst düzeye çıkarmak için bazı işlemlerin her zaman çalışmasını sağlamaktır.
Tipik olarak bazı G/Ç isteklerinin tamamlanması için beklemesi gerekene kadar bir işlem yürütülür. Basit bir bilgisayar sisteminde, CPU boşta kalır.
Çoklu programlama ile bu zamanı verimli kullanmaya çalışılır. Aynı anda birkaç işlem bellekte tutulur. Bir işlemin beklemesi gerektiğinde, işletim sistemi CPU'yu bu işlemden alır ve CPU'yu başka bir işleme verir. Bu desen devam eder. Bir işlemin beklemesi gerektiğinde, başka bir işlem CPU'nun kullanımını devralabilir. Çok çekirdekli bir sistemde, CPU'yu meşgul tutma kavramı, sistemdeki tüm işlem çekirdeklerine genişletilir.

Görev Çizelgeleme

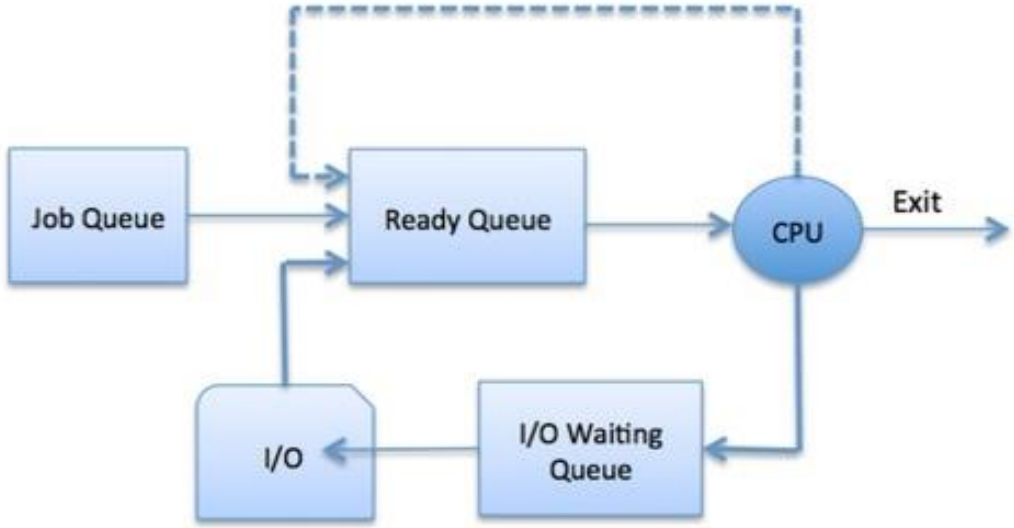
Bu tür zamanlama, temel bir işletim sistemi işlevidir. Hemen hemen tüm bilgisayar kaynakları kullanımdan önce planlanır. CPU, elbette, birincil bilgisayar kaynaklarından biridir. Bu nedenle, zamanlaması işletim sistemi tasarımının merkezinde yer alır.



Görev Çizelgeleme

İşletim sistemi, tüm PCB'leri İşlem Çizelgeleme Kuyruklarında tutar. İşletim sistemi, işlem durumlarının her biri için ayrı bir sıra tutar ve aynı yürütme durumundaki tüm işlemlerin PCB'leri aynı sıraya yerleştirilir. Bir işlemin durumu değiştirildiğinde, PCB'si mevcut kuyruğundan ayrılır ve yeni durum kuyruğuna taşınır.

- **İş kuyruğu** - Bu kuyruk, sistemdeki tüm işlemleri tutar.
- **Hazır kuyruk** - Bu kuyruk, ana bellekte bulunan tüm işlemlerin bir dizisini hazır ve yürütülmeyi bekleyen tutar. Bu kuyruğa her zaman yeni bir işlem konur.
- **I/O kuyrukları** - Bir G/Ç cihazının mevcut olmaması nedeniyle bloke edilen işlemler bu kuyruğu oluşturur.



Görev Çizelgeleme

Programlama ile ilgili önemli bir konu, zamanlama kararlarının ne zaman verileceğidir.

- İlk olarak, yeni bir süreç oluşturulduğunda, ana sürecin mi yoksa alt sürecin mi çalıştırılacağına karar verilmesi gerekir. Her iki süreç de hazır durumda olduğundan, bu normal bir zamanlama kararıdır ve her iki yöne de gidebilir, yani zamanlayıcı meşru olarak bir sonraki ebeveyni veya çocuğu çalıştırmayı seçebilir.

- İkincisi, bir süreç çıktığında bir zamanlama kararı verilmelidir. Bu süreç artık çalışamaz (çünkü artık mevcut değildir), bu nedenle hazır süreçler kümesinden başka bir süreç seçilmelidir. Hiçbir işlem hazır değilse, sistem tarafından sağlanan boşta bir işlem normal olarak çalıştırılır.

Görev Çizelgeleme

- Üçüncüsü, bir işlem G/Ç'de, bir semaforda veya başka bir nedenle bloke ettiğinde, çalıştırmak için başka bir işlemin seçilmesi gerekir. Bazen engelleme nedeni seçimde rol oynayabilir. Örneğin, A önemli bir süreçse ve B'nin kritik bölgesinden çıkmasını bekliyorsa, B'nin daha sonra çalışmasına izin vermek, kritik bölgesinden çıkmasına ve böylece A'nın devam etmesine izin verecektir. Ancak sorun, zamanlayıcının genellikle bu bağımlılığı hesaba katacak gerekli bilgilere sahip olmamasıdır.

- Dördüncüsü, bir G/Ç kesintisi meydana geldiğinde, bir zamanlama kararı verilebilir. Kesinti, işini tamamlamış bir G/Ç aygıtından geldiyse, G/Ç'yi beklerken engellenen bazı işlemler artık çalışmaya hazır olabilir. Yeni hazır işlemin mi, kesinti anında çalışmakta olan işlemin mi yoksa üçüncü bir işlemin mi çalıştırılacağına karar vermek zamanlayıcıya bağlıdır.

Görev Çizelgeleme

Bir donanım saati, belirli bir frekansta periyodik kesintiler sağlıyorsa, her saat kesintisinde (interrupt) veya her k. saat kesintisinde bir zamanlama kararı verilebilir. Zamanlama algoritmaları, saat kesintileriyle nasıl başa çıktıklarına göre iki kategoriye ayrılabilir.

Önleyici olmayan (nonpreemptive) bir zamanlama algoritması, çalıştırılacak bir işlem seçer ve ardından bloke olana (G/Ç'de veya başka bir işlem için beklerken) veya gönüllü olarak CPU'yu serbest bırakana kadar çalışmasına izin verir. Saatlerce çalışsa bile, zorla askıya alınmayacaktır. Gerçekte, saat kesintileri sırasında hiçbir zamanlama kararı verilmez. Saat kesme işlemi tamamlandıktan sonra, daha yüksek öncelikli bir işlem artık tatmin edilmiş bir zaman aşımını beklemiyorsa, kesintiden önce çalışmakta olan işlem devam ettirilir.

Buna karşılık, **önleyici (preemptive)** bir zamanlama algoritması bir süreç seçer ve en fazla sabit bir süre boyunca çalışmasına izin verir. Zaman aralığının sonunda hala çalışıyorsa, askıya alınır ve zamanlayıcı çalıştırmak için başka bir işlem seçer (eğer varsa). Önleyici programlama yapmak, CPU'nun kontrolünü zamanlayıcıya geri vermek için zaman aralığının sonunda bir saat kesintisinin olmasını gerektirir. Herhangi bir saat mevcut değilse, tek seçenek önleyici olmayan zamanlamadır.

Görev Çizelgeleme

Önleyici ve Önleyici Olmayan Zamanlama

CPU zamanlama kararları aşağıdaki dört koşul altında gerçekleşebilir:

1. Bir işlem, çalışan durumdan bekleme durumuna geçtiğinde (örneğin, bir G/Ç isteğinin veya bir alt işlemin sonlandırılması için bir wait() çağrısının sonucu olarak)
2. Bir işlem çalışır durumdayken hazır duruma geçtiğinde (örneğin, bir kesinti meydana geldiğinde)
3. Bir işlem bekleme durumundan hazır duruma geçtiğinde (örneğin, G/Ç tamamlandığında)
4. Bir süreç sona erdiğinde

Durum 1 ve 4 için, zamanlama açısından bir seçenek yoktur. Yürütme için yeni bir işlem (hazır kuyruğunda varsa) seçilmelidir. Ancak 2. ve 3. durumlar için bir seçenek vardır.

Programlama yalnızca 1. ve 4. koşullar altında gerçekleştiğinde, zamanlama şemasının önleyici (nonpreemptive) veya işbirlikçi (cooperative) olmadığını söyleriz. Aksi takdirde, önleyicidir (preemptive). Önleyici olmayan zamanlama altında, CPU bir işleme tahsis edildikten sonra, işlem sonlandırarak veya bekleme durumuna geçerek CPU'yu serbest bırakana kadar tutar.

Windows, macOS, Linux ve UNIX dahil olmak üzere neredeyse tüm modern işletim sistemleri, önleyici zamanlama algoritmaları kullanır.

Görev Çizelgeleme

İŞLEMCİ Çizelgeleme TÜRLERİ

İşlemci çizelgelemenin amacı, işlemci veya işlemciler tarafından zaman içinde yürütülecek süreçleri, yanıt süresi, çıktı ve işlemci verimliliği gibi sistem hedeflerini karşılayacak şekilde atamaktır. Birçok sistemde, bu çizelgeleme etkinliği üç ayrı işleve ayrılır: uzun, orta ve kısa dönemli çizelgeleme. Adlar, bu işlevlerin gerçekleştirildiği göreli zaman ölçeklerini önerir.

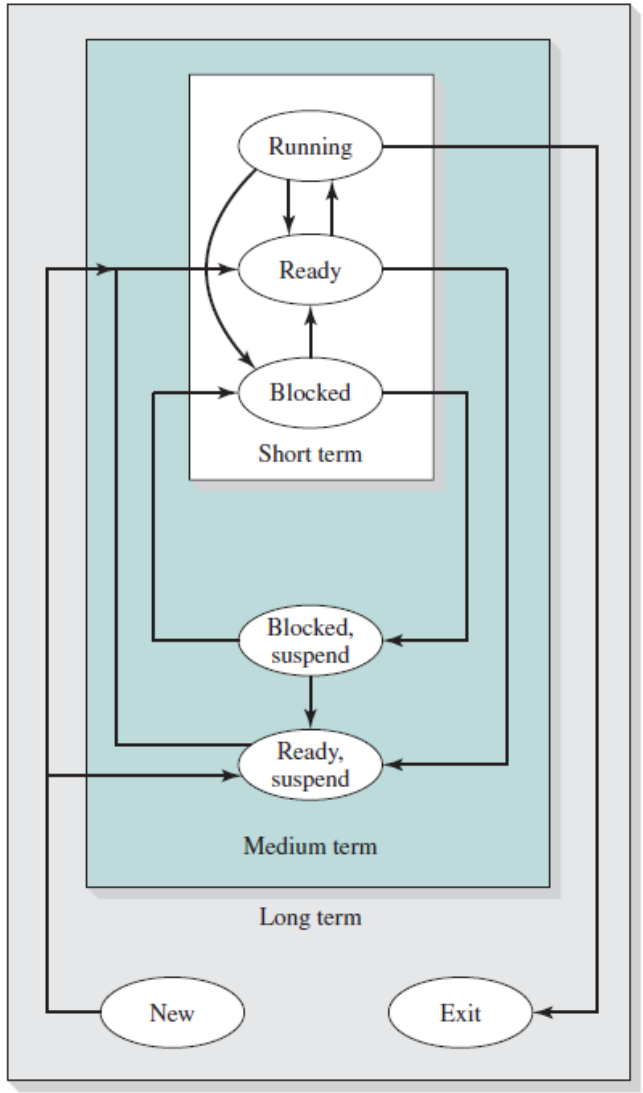


Figure 9.2 Levels of Scheduling

Görev Çizelgeleme

İŞLEMCİ Çizelgeleme TÜRLERİ

Uzun Vadeli Zamanlayıcı

Aynı zamanda iş zamanlayıcı (**job scheduler)** olarak da adlandırılır . Uzun vadeli bir zamanlayıcı, işleme için sisteme hangi programların kabul edildiğini belirler. Kuyruktan süreçleri seçer ve bunları yürütülmek üzere belleğe yükler. İşlem, CPU zamanlaması için belleğe yüklenir.

İş planlayıcının birincil amacı, G/Ç'ye bağlı ve işlemciye bağlı gibi işlerin dengeli bir karışımını sağlamaktır.

Bazı sistemlerde, uzun vadeli planlayıcı mevcut olmayabilir veya minimum düzeyde olabilir. Zaman paylaşımlı işletim sistemlerinin uzun vadeli zamanlayıcıları yoktur. Bir süreç, durumu yeniden hazıra değiştirdiğinde, uzun vadeli zamanlayıcı kullanılır.

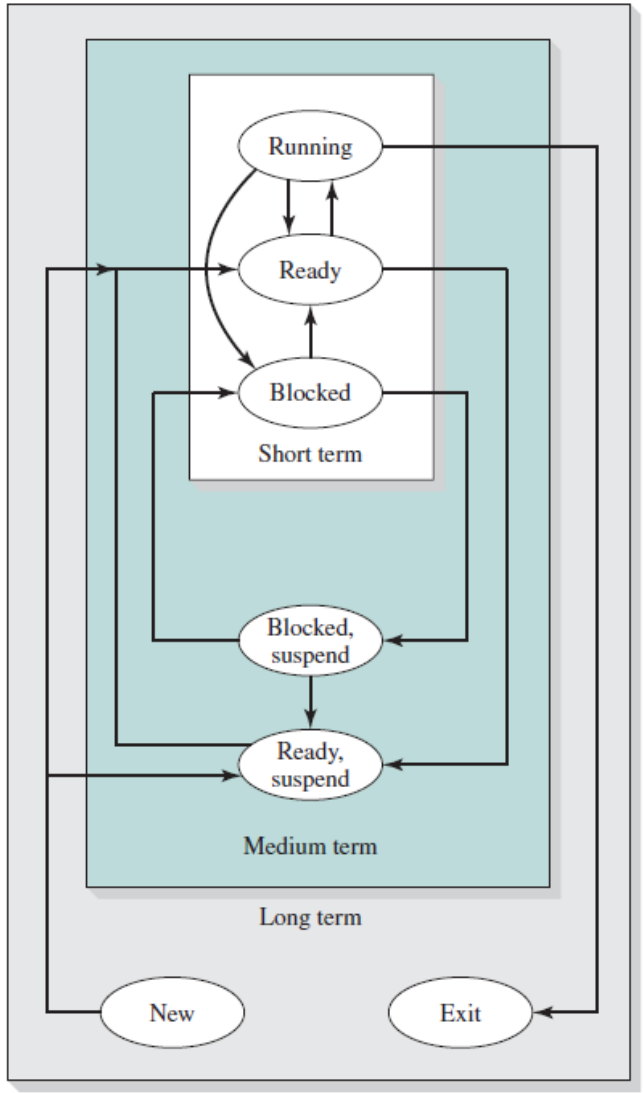


Figure 9.2 Levels of Scheduling

Görev Çizelgeleme

İŞLEMCİ Çizelgeleme TÜRLERİ

Orta Vadeli Zamanlayıcı

Orta vadeli planlama, takasın bir parçasıdır . İşlemleri bellekten kaldırır. Çoklu programlamanın derecesini azaltır. Orta vadeli planlayıcı, değiştirilen süreçleri yönetmekten sorumludur.

Çalışan bir işlem, bir G/Ç isteğinde bulunursa askıya alınabilir. Askıya alınmış bir süreç, tamamlanma yönünde herhangi bir ilerleme kaydedemez. Bu durumda, işlemi bellekten kaldırmak ve diğer işlemlere yer açmak için askıya alınan işlem ikincil depolamaya taşınır. Bu sürece takas denir ve sürecin takas edildiği veya kullanıma sunulduğu söylenir. Proses karışımını iyileştirmek için takas gerekli olabilir.

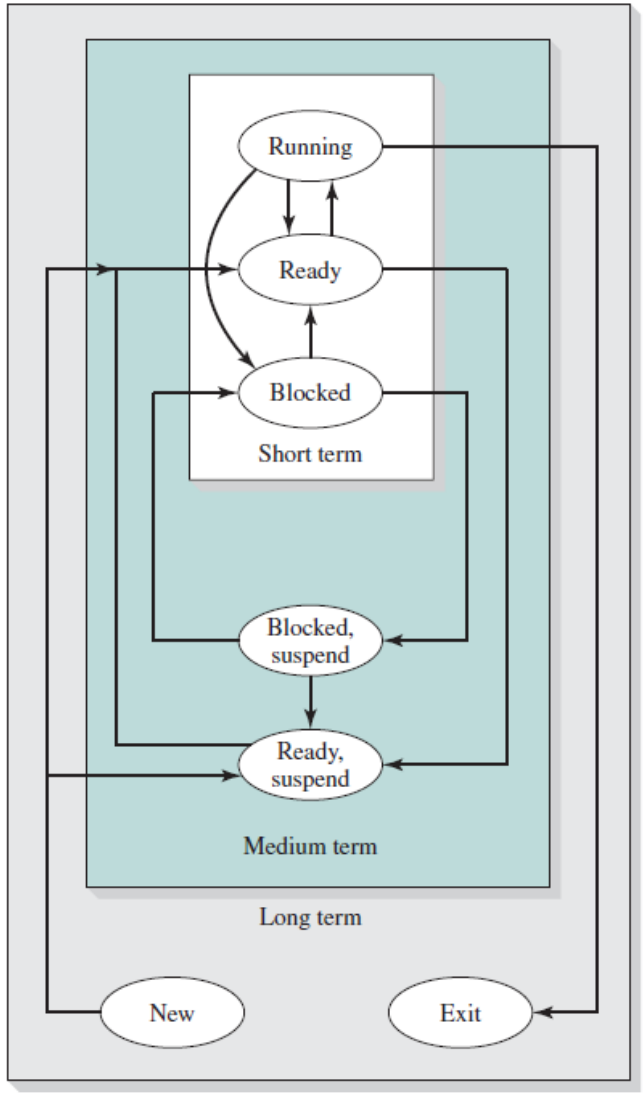


Figure 9.2 **Levels of Scheduling**

Görev Çizelgeleme

İŞLEMCİ Çizelgeleme TÜRLERİ

Kısa Vadeli Zamanlayıcı

Ayrıca CPU zamanlayıcı olarak da adlandırılır . Temel amacı, seçilen kriterler kümesine göre sistem performansını artırmaktır. İşlemin hazır durumundan çalışır duruma geçmesidir. CPU zamanlayıcı, yürütmeye hazır işlemler arasından bir işlem seçer ve CPU'yu bunlardan birine tahsis eder.

Göndericiler olarak da bilinen kısa vadeli planlayıcılar, bir sonraki işlemin yürütüleceğine karar verir. Kısa vadeli planlayıcılar, uzun vadeli planlayıcılardan daha hızlıdır.

Kısa vadeli çizelgelemenin temel amacı, sistem davranışının bir veya daha fazla yönünü optimize edecek şekilde işlemci zamanını tahsis etmektir. Genel olarak, çeşitli çizelgeleme politikalarının değerlendirilebileceği bir dizi kriter belirlenir.

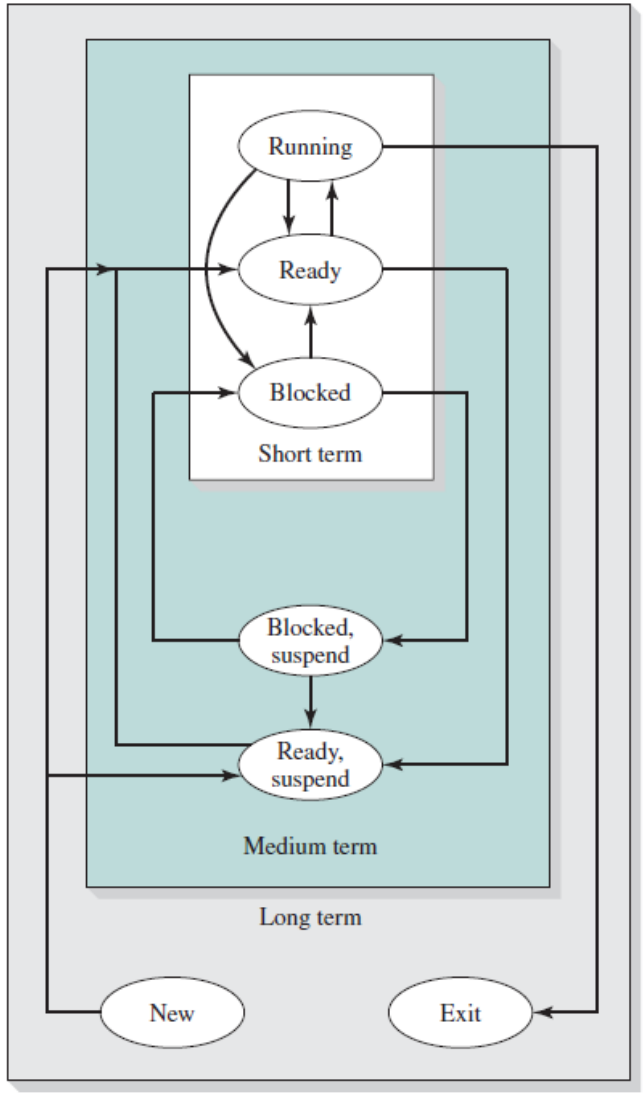


Figure 9.2 Levels of Scheduling

Zamanlama Algoritmaları KategorileriŞaşırtıcı

Farklı ortamlarda farklı zamanlama algoritmalarına ihtiyaç vardır. Bu durum, farklı uygulama alanlarının (ve farklı türdeki işletim sistemlerinin) farklı hedefleri olması nedeniyle ortaya çıkar.

1 . Toplu (Batch) Sitemler.

2. Etkileşimli (İnteractive) Sistemler.

3. Gerçek zamanlı sistemler.

Görev Çizelgeleme

Zamanlama Kriterleri

Farklı CPU zamanlama algoritmaları farklı özelliklere sahiptir ve belirli bir algoritmanın seçimi, bir süreç sınıfını diğerine tercih edebilir. Belirli bir durumda hangi algoritmanın kullanılacağını seçerken, çeşitli algoritmaların özelliklerini dikkate almalıyız.

CPU zamanlama algoritmalarını karşılaştırmak için birçok kriter önerilmiştir. Karşılaştırma için hangi özelliklerin kullanıldığı, hangi algoritmanın en iyi olduğuna karar verilmesinde önemli bir fark yaratabilir.

Görev Çizelgeleme

Zamanlama Kriterleri

CPU kullanımı. CPU'yu mümkün olduğunca meşgul tutmak istiyoruz. Kavramsal olarak, CPU kullanımı yüzde 0 ila 100 arasında değişebilir. Gerçek bir sistemde, yüzde 40 (hafif yüklü bir sistem için) ile yüzde 90 (ağır yüklü bir sistem için) arasında değişmelidir. (CPU kullanımı, Linux, macOS ve UNIX sistemlerinde top komutu kullanılarak elde edilebilir.)

Verim. CPU işlemleri yürütmekle meşgulse, iş yapılıyor demektir. İşin bir ölçüsü, verim adı verilen zaman birimi başına tamamlanan işlemlerin sayısıdır. Uzun işlemler için bu oran, birkaç saniye boyunca bir işlem olabilir; kısa işlemler için saniyede onlarca işlem olabilir.

Görev Çizelgeleme

Zamanlama Kriterleri

Geri dönüş süresi. Belirli bir sürecin bakış açısından,önemli kriter, bu işlemin ne kadar sürede gerçekleştirildiğidir. Bir sürecin sunulmasından tamamlanma zamanına kadar geçen süre, geri dönüş süresidir. Geri dönüş süresi, hazır kuyruğunda beklemek, CPU üzerinde yürütmek ve G/Ç yapmak için harcanan sürelerin toplamıdır.

Bekleme süresi. CPU zamanlama algoritması, bir işlemin yürütüldüğü veya G/Ç yaptığı süreyi etkilemez. Yalnızca bir işlemin hazır kuyruğunda beklerken harcadığı süreyi etkiler. Bekleme süresi, hazır kuyruğunda beklemek için harcanan sürelerin toplamıdır.

Görev Çizelgeleme

Zamanlama Kriterleri

Tepki Süresi. Etkileşimli bir sistemde, geri dönüş süresien iyi kriter. Çoğu zaman, bir süreç oldukça erken bir çıktı üretebilir ve önceki sonuçlar kullanıcıya verilirken yeni sonuçları hesaplamaya devam edebilir. Bu nedenle, başka bir ölçü, bir talebin sunulmasından ilk yanıtın üretilmesine kadar geçen süredir. Tepki süresi adı verilen bu ölçü, yanıtın çıktısını almak için geçen süre değil, yanıt vermeye başlamak için geçen süredir.

Görev Çizelgeleme

Zamanlama Kriterleri

Tepki Süresi. Etkileşimli bir sistemde, geri dönüş süresien iyi kriter. Çoğu zaman, bir süreç oldukça erken bir çıktı üretebilir ve önceki sonuçlar kullanıcıya verilirken yeni sonuçları hesaplamaya devam edebilir. Bu nedenle, başka bir ölçü, bir talebin sunulmasından ilk yanıtın üretilmesine kadar geçen süredir. Tepki süresi adı verilen bu ölçü, yanıtın çıktısını almak için geçen süre değil, yanıt vermeye başlamak için geçen süredir.

Görev Denetimi

Planlama Algoritmaları

1. FCFS (First Come First Served – İlk Gelen Önce) Algoritması:

Bu algoritmaya göre; AIB yi ilk talep eden görev, ilk olarak işlemciyi kullanır. FIFO kuyruğu ile çalıştırılabilir. Görev hazır görevler kuyruğuna sunulduktan sonra, onun görev denetim bloğu (PCB) kuyruğun sonuna ilave edilir. AIB boş olduğu zaman kuyruğun başındaki görev çalışması için AIB ye sunulur ve kuyruktan silinir. Bu algoritmada görevlerin bekleme süresi yüksek olur.

Örnek: P1, P2, P3 görevlerinin sırasıyla kuyrukta yerleştiklerini kabul edelim:

Görev	Çalışma Zamanı (sn)
P1	24
P2	3
P3	3

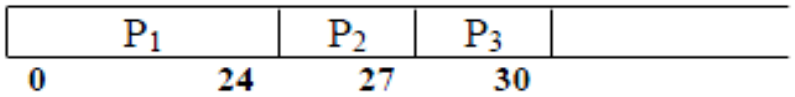
Görev Denetimi

1. FCFS (First Come First Served – İlk Gelen Önce) Algoritması:

Örnek: P1, P2, P3 görevlerinin sırasıyla kuyrukta yerleştiklerini kabul edelim:

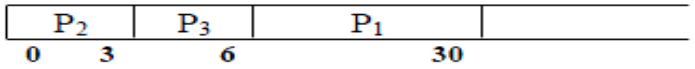
Görev	**Çalışma Zamanı (sn)**
P1	24
P2	3
P3	3

1.Görevler P1, P2, P3 ardışıklığı ile sunulmuş olduğunu varsayalım. Buna göre planlama:



Ortalama bekleme süresi : (24+27+0) / 3 = 17 msn 'dir.

2.Eğer görevlerin gelme P2, P3, P1 şeklinde sıralanırsa, planlama:



Ortalama bekleme süresi : (3+6+0) / 3 = 3 msn'dir.

Görev Denetimi

2.SJF (Shortest Job First – En Kısa İşletim Süresi Olan Önce) Algoritması:

Bu algoritmada CPU boş olduğunda, kalan görevler içinde çalışma süresi en küçük olan görev, çalışması için işlemciye sunulur. Eğer iki görevin kalan süreleri aynı ise o zaman FCFS algoritması uygulanır. Bu algoritmada: Her görev, o görevin bir sonraki MİB işlem zamanı ile değerlendirilir. Bu, en kısa zamanlı işin bulunması için kullanılır.

SJF Türleri:

1.Kesilmesiz SJF : Eğer AİB bir göreve tahsis edilmişse, AİB işlem zamanı bitmeyince görev kesilemez.

2.Kesilmeli SJF : Eğer AİB işlem zamanı, şu anda çalışan görevin kalan işlem zamanından küçük olan yeni bir görev sisteme sunulmuşsa, eski görev kesilecek. Bu yönteme, SRTF(Shortest Remaining Time First – En kısa işlem zamanı kalan, birinci) yöntem denir.

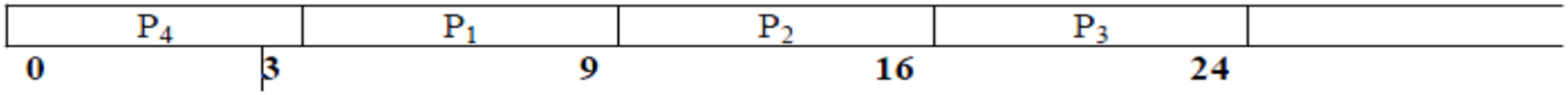
SJF verilmiş görevler kümesi için en küçük ortalama bekleme zamanı oluşması için optimizasyon yapar.

Görev Denetimi

2.SJF (Shortest Job First – En Kısa İşletim Süresi Olan Önce) Algoritması:

Örnek: : P1, P2, P3, P4 görevleri aşağıdaki ardışıklık ile sunulmuş olduğunu varsayalım. Buna göre kesilmesiz SJF yöntemine göre ortalama bekleme süresini bulalım:

Görev	Çalışma Süresi
P_1	6
P_2	7
P_3	8
P_4	3

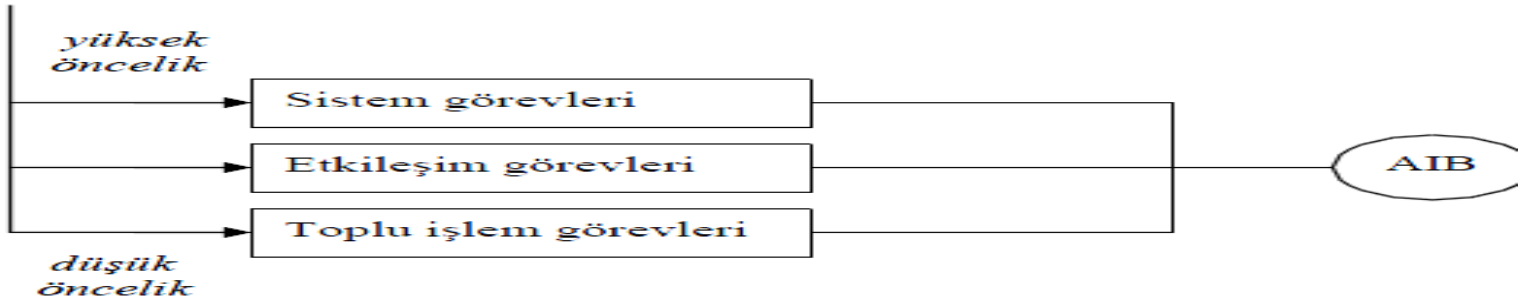


- **SJF :** tob = tP1 + tP2 + tP3 + tP4 = (3+9+16+0) / 4 = 7 msn
- **FCFS :** tob = tP1 + tP2 + tP3 + tP4 = (0+6+13+21) / 4 =10.75 msn

Görev Denetimi

3.Çok Kuyruklu Planlama Algoritması:

Bu algoritmaya göre görevler belli sınıflara ayrılır ve her sınıf görev kendi kuyruğunu oluşturur. Başka deyişle hazır görevler çok seviyeli kuyruğa dönüştürülür. Görevin türüne önceliğine, bellek durumuna veya başka özelliklerine göre görevler belli kuyruğa yerleştirilirler. Her kuyruk için planlama algoritması farklı olabilir. Bununla birlikte görevlerin bir kuyruktan diğerine aktarılmasını sağlayan algoritmada oluşturulur.



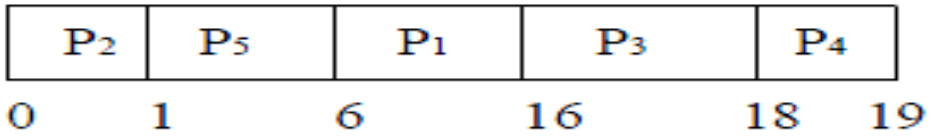
Bu algoritmaya göre yüksek öncelikli kuyruktaki görevler önce işlenir. Eğer bu kaynak boş ise ondan aşağı seviyedeki görevler çalıştırılabilir.

Görev Denetimi

4.Öncelikli Planlama Algoritması:

Bu algoritmaya göre her bir göreve öncelik değeri atanır ve görevler öncelik sırasına göre işlemciyi kullanırlar. Aynı öncelikli görevler FCFS algoritması ile çalıştırılırlar.

Görev	Öncelik	Çalışma Süresi (msn)
P_1	3	10
P_2	1	1
P_3	3	2
P_4	4	1
P_5	2	5



$t_{ob} = (0 + 1 + 6 + 16 + 18) / 5 = 8.2$ msn

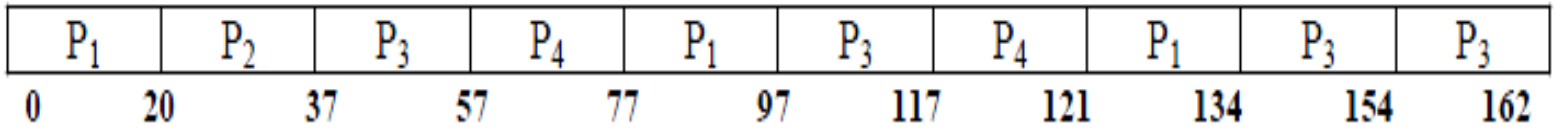
Görev Denetimi

5.Döngülü Planlama (Round Robin - RR) Algoritması:

Her görev küçük bir AİB zaman dilimini alır. Bu zaman bittiğinde, görev kesilir ve hazır görevler kuyruğunun sonuna eklenir.

- **Örnek:** P1, P2, P3, P4 görevleri aşağıdaki ardışıklık ile sunulmuş olduğunu varsayalım. Zaman dilimi 20 msn ise buna göre:

Görev	İşlem Zamanı (msn)
P_1	53
P_2	17
P_3	68
P_4	24



Görev Denetimi

Table 9.4 Process Scheduling Example

Process	Arrival Time	Service Time
A	0	3
B	2	6
C	4	4
D	6	5
E	8	2