

İŞLETİM SİSTEMLERİ

(HAFTA 6)

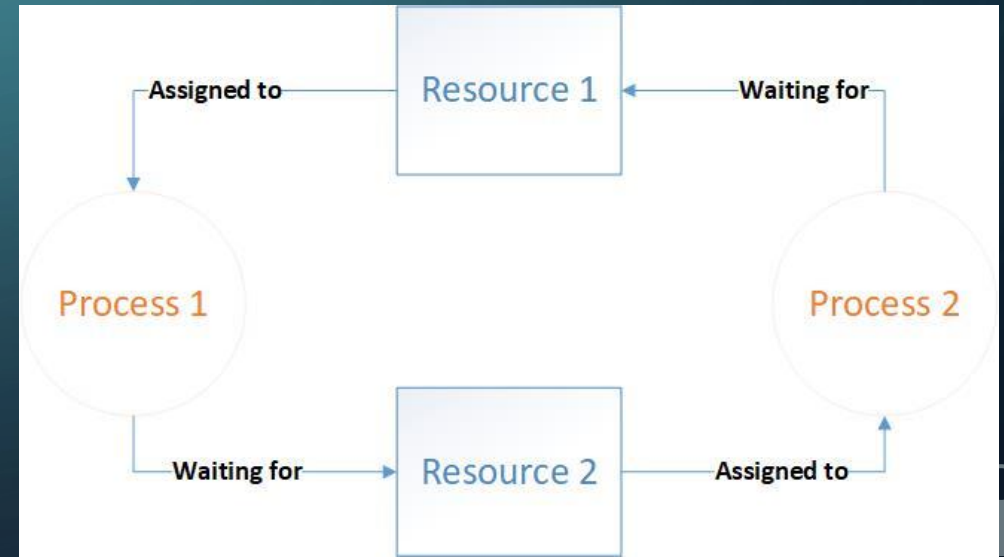
DR. ÖĞR. ÜYESİ TAYYİP ÖZCAN

ÇIKTILAR

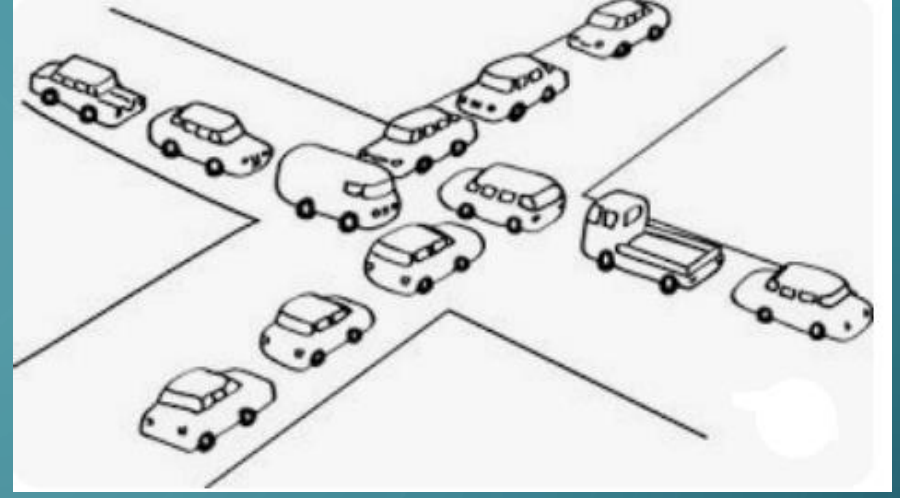
- Ölümcül Kilitlenme

ÖLÜMCÜL KİLİTLENME

- Nedir?
 - Ortak kaynağı kullanan ve birbiri ile haberleşen eş zamanlı çalışan işlemlerin birinin veya birkaçının kalıcı olarak bloke olma durumu deadlock'tır.
 - En bilinen sebepleri:
 - Bir işlemin başka işlem tarafından tutulan kaynağı talep etmesi
 - Kendi sahip olduğu kaynağı serbest bırakmaması
 - Şekilde:
 - İşlem 1, Kaynak 1'e sahip Kaynak 2'yi talep ediyor
 - İşlem 2, Kaynak 2'ye sahip Kaynak 1'i talep ediyor



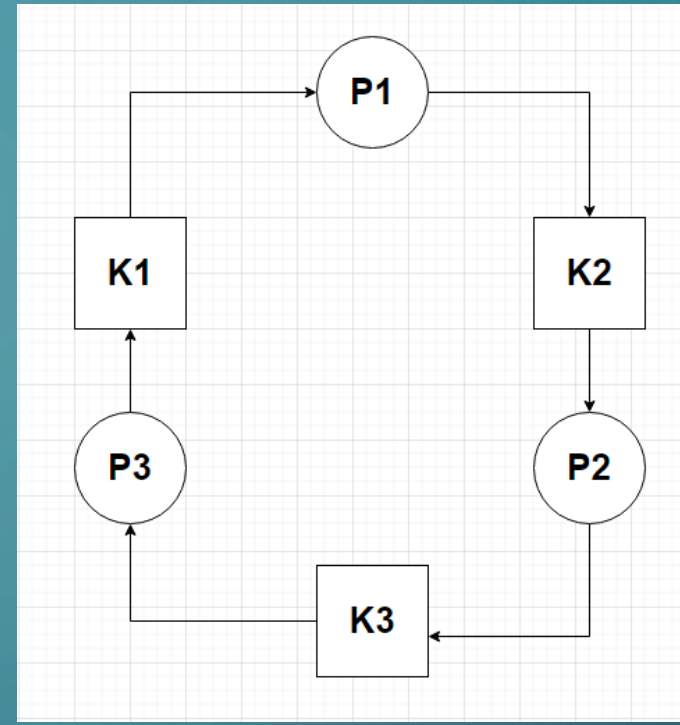
ÖLÜMCÜL KİLİTLENME



- Nedir?
 - Eş zamanlı iki veya daha fazla işlemden kaynaklanmaktadır.
 - Şekilde:
 - Araçlar orta noktada kesişiyor ve hepsi birbirini bekliyor.
 - Birbirlerinin kullandıkları yolu talep ediyorlar.
 - Yol önceliği durumuna bağlı kalırsa ölümcül kilitleme örneği olan trafik kilitlenmesi yaşanmayabilirdi

ÖLÜMCÜL KİLİTLENME

- Gösterim (Graph)
 - Her bir daire bir işlemi ifade ediyor
 - Her bir kare bir kaynağı ifade ediyor.
 - Kaynaktan işleme yönlü ok kaynağın ilgili işleme aktarıldığını ifade eder.
 - İşlemden kaynağa yönlü ok işlemin ilgili kaynağı talep ettiğini ifade eder.
 - P1 işlemi K1'i kullanırken K2'yi talep etmekte
 - P2 işlemi K2'yi kullanırken K3'ü talep etmekte
 - P3 işlemi K3'ü kullanırken K1'i talep etmekte



ÖLÜMCÜL KİLİTLENME

P1	P2
iste (A);	iste (T);
kilitle (A); <<1>>	kilitle (T); <<2>>
iste (T);	iste (A);
kilitle (T); <<3>>	kilitle (A); <<4>>
<işlem yap>	<işlem yap>
kilit_ac (T);	kilit_ac (A);
kilit_ac (A)	kilit_ac (T)
Eş zamanlı işlemler arasında ölümcül kilitlenme gösterimi	

- A ve T kaynakları, P1 ve P2 işlemleri ifade etsin.
- 1 numaralı işlem adımıyla P1 A kaynağını bloke ediyor
- 2 numaralı işlem ile P2 T kaynağını bloke ediyor
- 3 numaralı işlem adımıyla P2 ye tahsis edilmiş olan T kaynağını P1 işlemi,
- 4 numaralı işlem adımıyla P1 e tahsis edilmiş A kaynağını P2 işlemi talep etmektedir.
- Birbirini bekleyen işlemlerden dolayı ve sistemde çalışacak işlemlerin zamanları, öncelikleri, ya da nerde ne zaman kesilecekleri bilinmediğinden dolayı deadlock potansiyeli bulunmaktadır.

ÖLÜMCÜL KİLİTLENME

	P1	P2	
(1)	iste (60K);	iste (50K);	(2)
	
(3)	iste (40K);	iste (60K);	(4)
	
Kaynak kısıtı nedeni ile ölümcül kilitlenme			

- Eş zamanlı işlemler arasında kaynak kısıtı nedeniyle ölümcül kilitlenme durumu söz konusu olabilir.
- Bu örnek için sistemde 140 Kb lık bellek bölgesinin işlemlere ayrıldığını varsayalım.
- Numaralandırılmış alanlar şekildeki gibi tahsis edilip kilitlenecekse 1 ve 2. Adımlar gerçekleşince 110 Kb lık alan kullanılmış olacaktır.
- Geriye $140 - 110 = 30$ kb lık alan kaldığından dolayı 3. Adım için gerekli olan 40 kb lık alan kullanılamayacak ve ölümcül kilitlenme durumu gerçekleşme potansiyeli yine bulunmaktadır.

ÖLÜMCÜL KİLİTLENME

	P1	P2	
↓	mesajAl(P2);	mesajAl(P1);	↑
	
	mesajGonder(P2);	mesajGonder(P1);	
	
Mesajlaşma nedeni ile ölümcül kilitlenme			

- Eş zamanlı işlemler arasında mesajlaşma nedeniyle ölümcül kilitlenme oluşabilmesi durumu bulunmaktadır.
- Burada P1 işlemi mesajAl(P2) işlemi sonuçlanmadan işlevlerine devam edememekte ve askıya alınmaktadır.
- Aynı durum P2 için de geçerli olsun.
- Bu durumda P2 işlemi P1 den P1 işlemi de P2 den mesaj bekleme durumuna geçtiklerinde işlemler askıya alınacaktır.

ÖLÜMCÜL KİLİTLENME

- Deadlock olası sebepler koşulları
 - Karşılıklı dışlama koşulu
 - Paylaşılan kaynaklar üzerinde sadece bir işlem çalışabilecekken işlemler birbirini karşılıklı dışlayabilirler.
 - Sahiplenme ve bekleme koşulu
 - Bir işlem hem elindeki kaynağı bırakmayıp hem de başka kaynak isteyip alamayınca da beklemeye geçebilir.

ÖLÜMCÜL KİLİTLENME

- Deadlock olası sebepler koşulları
 - Geri alınamaz kaynak koşulu
 - İşlem için atanan kaynaklar işlemin isteği dışında geri alınamıyorsa geri alınamaz kaynak koşulu oluşur.
 - Çevrimsel bekleme koşulu
 - Her biri zincirin bir sonraki üyesi tarafından tutulan kaynağı talep eden iki veya daha fazla işlemden oluşan dairesel bir zincir durumu varsa çevrimsel bekleme koşulu oluşmaktadır.

ÖLÜMCÜL KİLİTLENME

- Deadlock'a karşı kullanılan yaklaşımlar:
 - Sistemin deadlock olmayacak şekilde tasarlanması
 - Ölümcül kilitlenmeye neden olan koşulların en azından 1 tanesinin ortadan kaldırılması ile bu yaklaşım mümkündür.
 - Deadlock ı sezme (detection)
 - Deadlock ı önleme (prevention)
 - Ölümcül kilitlenme başlatacak işlemlerin başlatılmaması, deadlock oluşturabilecek isteklerin yerine getirilmemesiyle bu koşul sağlanabilmektedir.

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Önleme
- Banker Algoritması
 - İlk kullanılan algoritmalarından
 - Sistemde sabit sayıda işlem ve kaynak olduğu varsayılır.
 - Burada durum:
 - Kaynak (resource), boş (available), atanmış (allocation) ve max_istek (max_request) matrislerinden oluşmaktadır.

Max. İstek Matrisi				Atanmış Matrisi				Kalan İstek Matrisi			
	K1	K2	K3		K1	K2	K3		K1	K2	K3
P1	3	2	2	P1	1	0	0	P1	2	2	2
P2	6	1	3	P2	6	1	2	P2	0	0	1
P3	3	1	4	P3	2	1	1	P3	1	0	3
P4	4	2	2	P4	0	0	2	P4	4	2	0

Kaynak Vektörü			Boş Kaynak Vektörü		
K1	K2	K3	K1	K2	K3
9	3	6	0	1	1

Banker Algoritması ile Sistem Durumu Gösterimi					
--	--	--	--	--	--

ÖLÜMCÜL KİLİTLENME

Max. İstek Matrisi				Atanmış Matrisi				Kalan İstek Matrisi			
	K1	K2	K3		K1	K2	K3		K1	K2	K3
P1	3	2	2	P1	1	0	0	P1	2	2	2
P2	6	1	3	P2	6	1	2	P2	0	0	1
P3	3	1	4	P3	2	1	1	P3	1	0	3
P4	4	2	2	P4	0	0	2	P4	4	2	0

ÖLÜMCÜL KİLİTLENME

- Banker Algoritması
 - Güvenli Durum
 - Tüm işlemler ölümcül kilitlenme olmadan kaynak atamasıyla sonlanabilir.
 - Güvensiz Durum
 - Kaynak atama sekansı bulunamayan, sonlanamayan işlemler olasılığı bulunma durumunda
 - Banker algoritmasına göre bir işlem bir kaynağa ihtiyaç duyduğunda işlemin sahip olduğu kaynak sayısı ile istediği kaynak sayısının toplamı maksimum istek sayısından küçük eşittir olmalı ve bir kaynak atama sekansı bulunmalıdır.

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Önleme
- Banker Algoritması Örnek Uygulama 1
 - 3 işlem 12 adetten oluşan kaynağı kullanmak istesin. Banker Algoritması kullanarak sistemin güvenli olup olmadığına karar verelim.

		Sahip	Max_İstek						Sahip	Max_İstek	Kalan_İstek
	A	1	4					A	1	4	3
	B	4	6					B	4	6	2
	C	5	8					C	5	8	3
Tek tipte kaynak bulunan sistem durumu								Kalan istek durumu			

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Önleme
- Banker Algoritması Örnek Uygulama 2
 - 3 işlem 12 adetten oluşan kaynak

	Sahip	Max_İstek	Kalan_İstek
A	8	10	2
B	2	5	3
C	1	3	2

Tek tipte kaynak bulunan sistem durumu

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Önleme
- Banker Algoritması Çalışma Adımları
 - 1) kalan_istek_matrisi \leq boş_kaynak_vektörü var mı?
 - 2) 1. adımda bulunan işlemin çalışması, boş_kaynak_vektörünün güncellenmesi, işin bitti olarak işaretlenmesi
 - 3) 1. ve 2. adımlar tüm işlemler bitene kadar tekrarlanır
- Tüm işlemler sonlanıyorsa güvenli durum, sonlanmayan işlem kalırsa güvensiz durum oluşmaktadır.

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Önleme
- Uygulama

- 5 işlem için 3 kaynak tanımlanmış ve maksimum istek, atanmış ve kaynak vektörü matrisleri gösterilmektedir. Sistem bu durumda iken Banker algoritması çalıştığında kalan istek matrisi ve boş kaynak vektörünü oluşturarak güvenli veya güvensiz durum oluşumunu gösteriniz. Şayet güvenli olması mümkünse proseslerin en etkin hangi sıra ile kullanılacağını belirleyiniz.

Max. İstek Matrisi				Atanmış Matrisi			
	K1	K2	K3		K1	K2	K3
P1	3	5	5	P1	2	1	1
P2	5	3	4	P2	2	2	3
P3	4	4	4	P3	3	2	3
P4	2	6	5	P4	0	3	1
P5	4	4	4	P5	4	3	2

Kaynak Vektörü		
K1	K2	K3
12	13	14

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Önleme
- Uygulama (Çözüm)

Max. İstek Matrisi				Atanmış Matrisi				Kalan İstek Matrisi			
	K1	K2	K3		K1	K2	K3		K1	K2	K3
P1	3	5	5	P1	2	1	1	P1	1	4	4
P2	5	3	4	P2	2	2	3	P2	3	1	1
P3	4	4	4	P3	3	2	3	P3	1	2	1
P4	2	6	5	P4	0	3	1	P4	2	3	4
P5	4	4	4	P5	4	3	2	P5	0	1	2

Kaynak Vektörü			Boş Kaynak Vektörü		
K1	K2	K3	K1	K2	K3
12	13	14	1	2	4

İŞLETİM SİSTEMLERİ

ÖNCE				Kalan İstek Matrisi			
Boş Kaynak Vektörü					K1	K2	K3
K1	K2	K3		P1	1	4	4
0	0	3		P2	3	1	1
				P3	0	0	0
				P4	2	3	4
				P5	0	1	2

SONRA >>				Boş Kaynak Vektörü			
					K1	K2	K3
					4	4	7

ÖNCE				Kalan İstek Matrisi			
Boş Kaynak Vektörü					K1	K2	K3
K1	K2	K3		P1	1	4	4
2	1	3		P2	3	1	1
				P3	0	0	0
				P4	0	0	0
				P5	0	1	2

SONRA >>				Boş Kaynak Vektörü			
					K1	K2	K3
					4	7	8

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Önleme
- Uygulama (Çözüm)

Max. İstek Matrisi				Atanmış Matrisi				Kalan İstek Matrisi			
	K1	K2	K3		K1	K2	K3		K1	K2	K3
P1	3	5	5	P1	2	1	1	P1	1	4	4
P2	5	3	4	P2	2	2	3	P2	3	1	1
P3	4	4	4	P3	3	2	3	P3	1	2	1
P4	2	6	5	P4	0	3	1	P4	2	3	4
P5	4	4	4	P5	4	3	2	P5	0	1	2

Kaynak Vektörü			Boş Kaynak Vektörü		
K1	K2	K3	K1	K2	K3
12	13	14	1	2	4

ÖNCE				Kalan İstek Matrisi			
Boş Kaynak Vektörü					K1	K2	K3
K1	K2	K3		P1	1	4	4
4	6	6		P2	3	1	1
				P3	0	0	0
				P4	0	0	0
				P5	0	0	0

SONRA >>				Boş Kaynak Vektörü			
				K1	K2	K3	
				8	10	10	

ÖNCE				Kalan İstek Matrisi			
Boş Kaynak Vektörü					K1	K2	K3
K1	K2	K3		P1	0	0	0
7	6	6		P2	3	1	1
				P3	0	0	0
				P4	0	0	0
				P5	0	0	0

SONRA >>				Boş Kaynak Vektörü			
				K1	K2	K3	
				10	11	11	

ÖNCE				Kalan İstek Matrisi			
Boş Kaynak Vektörü					K1	K2	K3
K1	K2	K3		P1	0	0	0
7	10	10		P2	0	0	0
				P3	0	0	0
				P4	0	0	0
				P5	0	0	0

SONRA >>				Boş Kaynak Vektörü			
				K1	K2	K3	
				12	13	14	

ÖLÜMCÜL KİLİTLENME

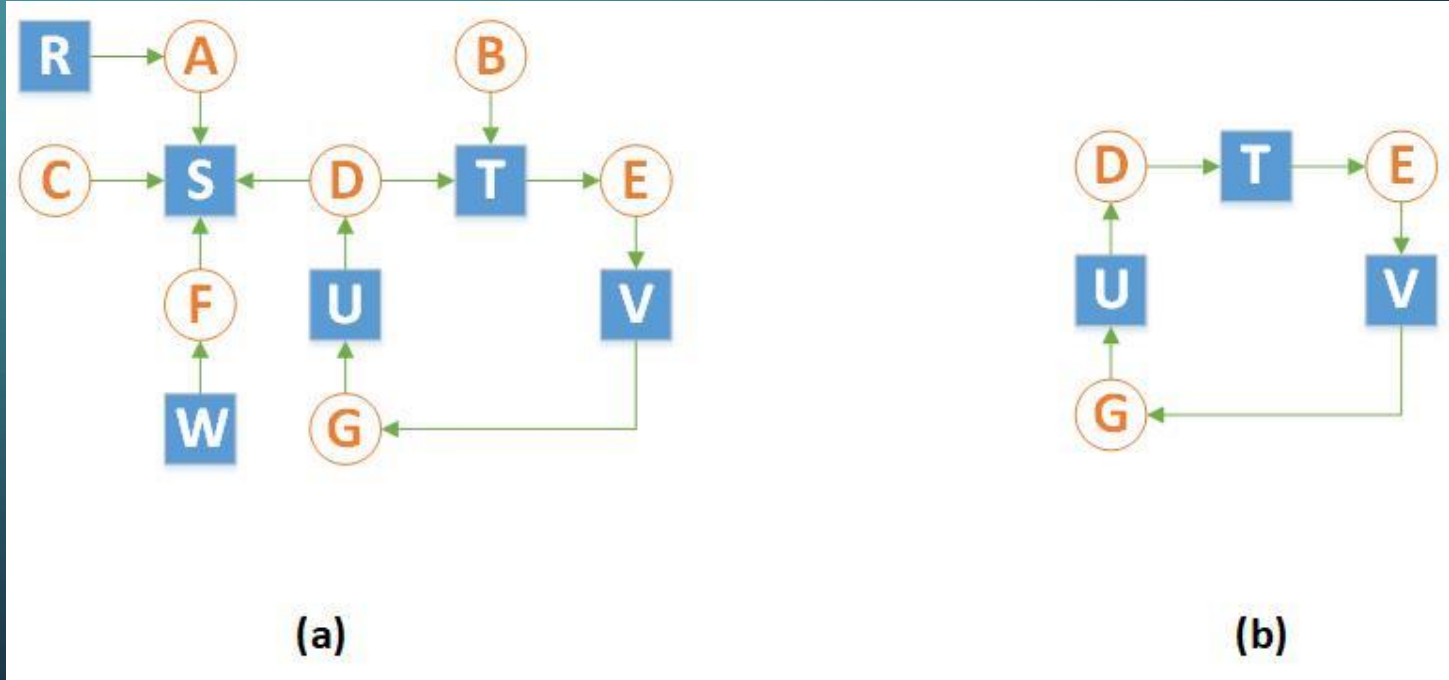
- Ölümcül Kilitlenmeyi Sezme
 - Ölümcül kitlemeye karşı kullanılan yöntemlerden bir diğeridir.
 - Bu yaklaşımda işlemlerin talep ettikleri ve sahip oldukları kaynaklara bakılarak ölümcül kilitlenme durumunun sezilmeye çalışılmasıdır.
 - Sistem durumu graf ile gösterilsin...
 - Kapalı çevrim ölümcül kilitlenmenin göstergesidir.

ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Sezme
- Örnek

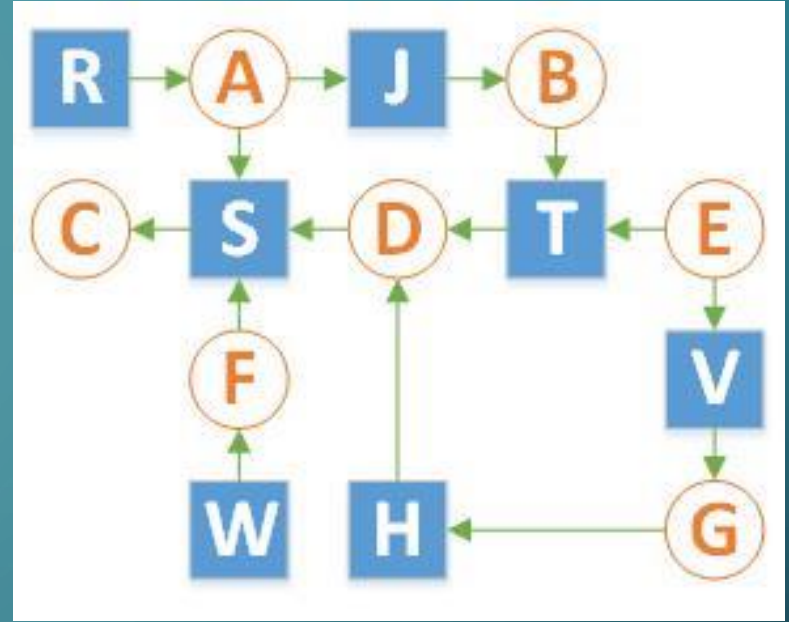
İşlem A, R kaynağına sahip, S'yi istiyor
İşlem B'nin henüz kaynağı yok, T'yi istiyor
İşlem C'nin henüz kaynağı yok, S'yi istiyor
İşlem D, U kaynağına sahip, S ve T'yi istiyor
İşlem E, T kaynağına sahip, V'yi istiyor
İşlem F, W kaynağına sahip, S'yi istiyor
İşlem G, V kaynağına sahip, U'yu istiyor

Çözüm



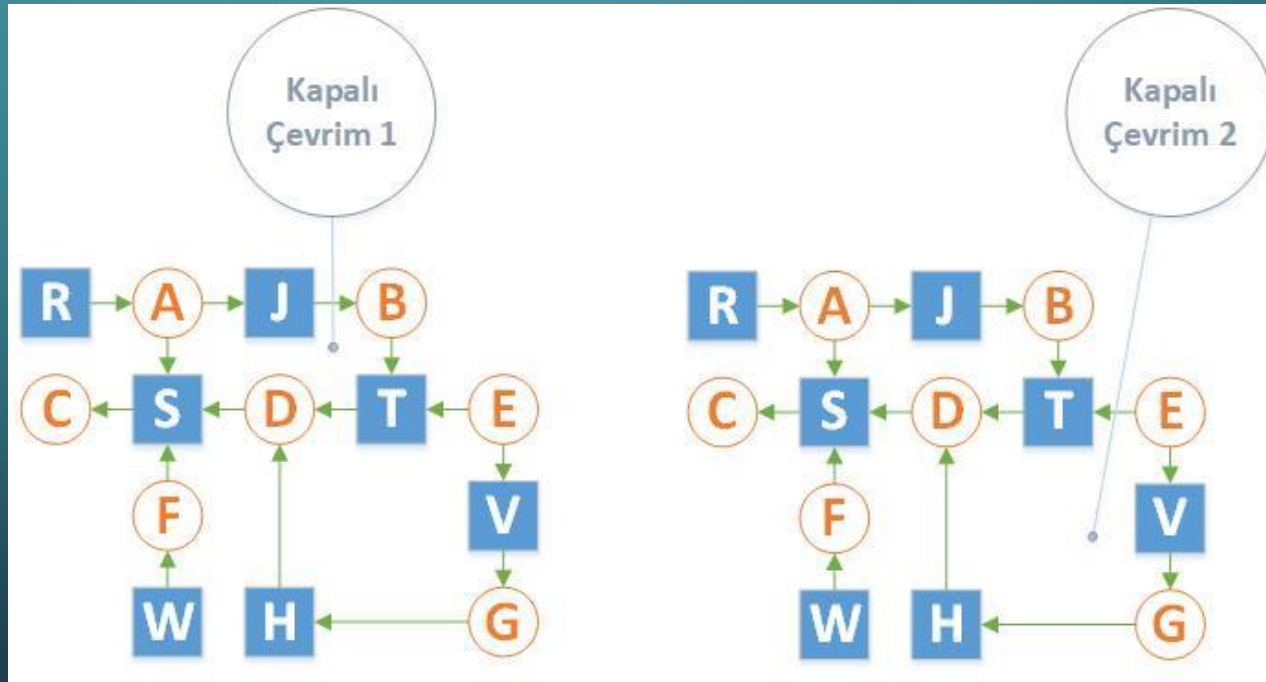
ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Sezme
- Uygulama 1
 - Ölümcül Kilitlenme olup olmayacağını
 - Ölümcül kilitlenme söz konusu ise hangi işlemi sonlandırmak en etkili çözüm olacaktır.



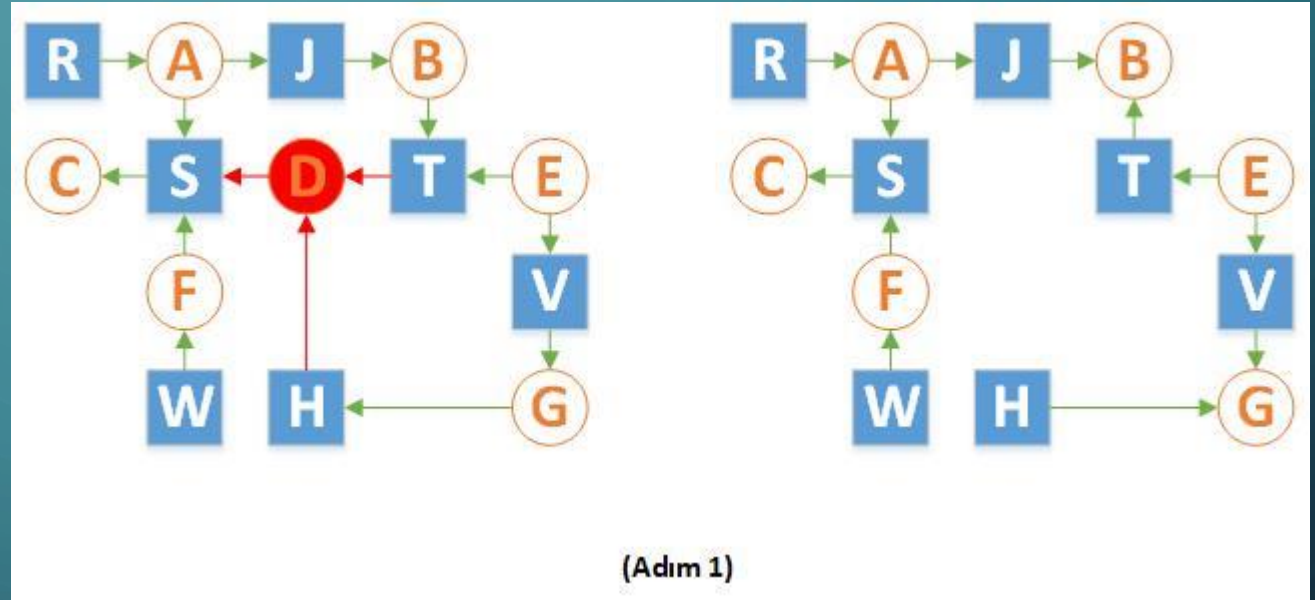
ÖLÜMCÜL KİLİTLENME

- Ölümcül Kilitlenmeyi Sezme
- Uygulama 1
 - Çözüm (α)



ÖLÜMCÜL KİLİTLENME

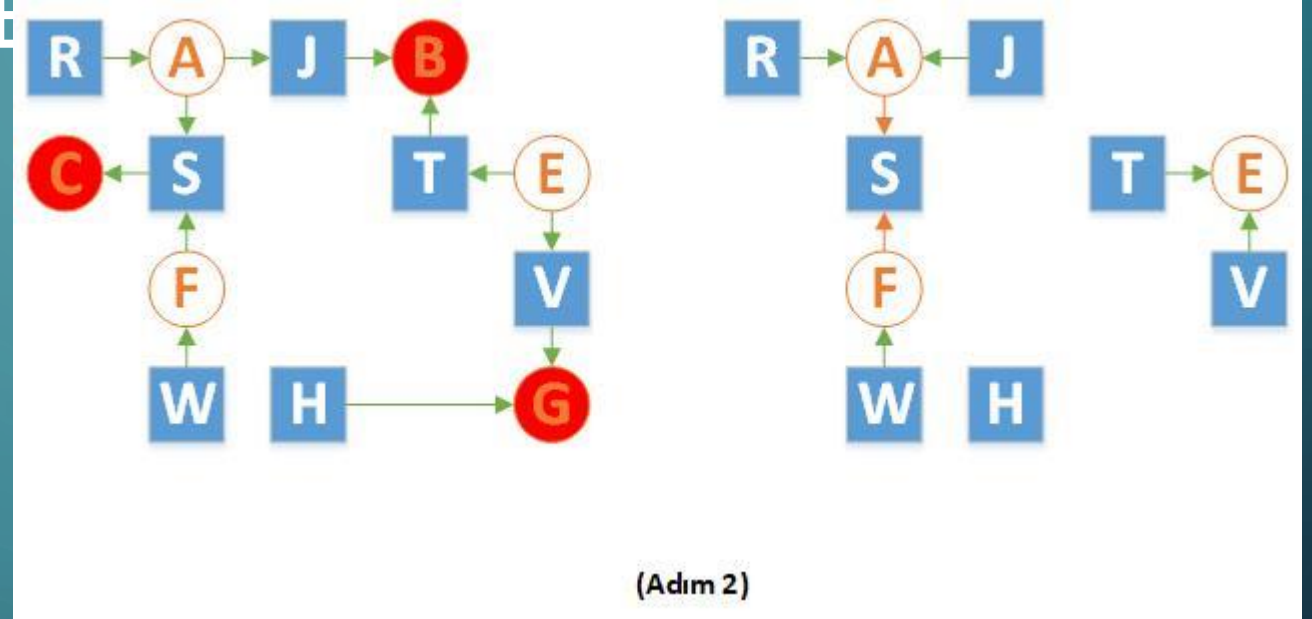
- Ölümcül Kilitlenmeyi Sezme
- Uygulama 1
 - Çözüm (b)



- Adım 1: Bu adım ile D işlemini sonlandırıyoruz. T ve H kaynağı boşa çıkıyor. T ve H kaynağı daha sonra tekrar tahsis ediliyor.

ÖLÜMCÜL KİLİTLENME

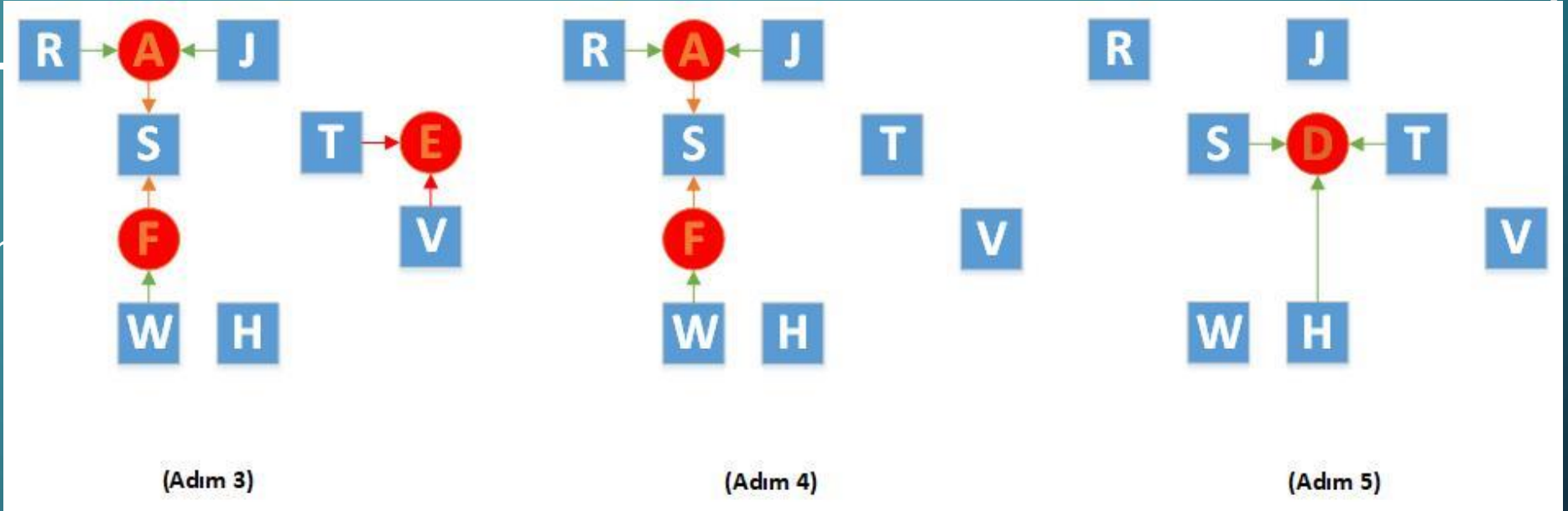
- Ölümcül Kilitlenmeyi Sezme
- Uygulama 1
 - Çözüm (b)



- 2. adımda: B-C ve G işlemleri tamamlanıyor. Dolayısıyla J, T ve V kaynakları yeniden tahsis ediliyor. A ve F işlemleri tarafından da talep edilen S kaynağı boşa çıkıyor.

ÖLÜMCÜL

- Ölümcül Kilitler
- Uygulama 1
 - Çözüm (b)



- Adım 3 te öncelikle A veya F ile E işlemleri tamamlanır. A veya F in hangisinin tamamlanacağı S kaynağını hangisinin kullanacağına bağlıdır.
- Adım 4 te A veya F işleminden hangisi kaldıysa o tamamlanır.
- Adım 5 te D prosesi yeniden canlandırılır ve H , S ve T kaynakları tahsis edilir. İşlem tamamlandırılır ve süreç sonlanır.

HAFTA 7 PLANLAMA

- Bilgisayar Mimarileri
- Planlama Algoritmaları ve Banker Algoritması için Kodlama
- Vmware Workstation Player Kurulumu
- Linux Mint İndirme ve Sanal Makineye Kurulumu
- Temel Linux Komutları

TEŞEKKÜR EDERİM