



## Filozoflar

Felsefenin bu kadar ölümcül olabileceğini hiç düşünmemiştim.

### Özet:

*Bu projede, bir sürece iş parçacığı eklemenin temellerini öğreneceksiniz.  
İş parçacıklarının nasıl oluşturulacağını öğrenecek ve mutekslerin kullanımını keşfedeceksiniz.*

*Sürüm: 11.2*

# İçindekiler

<b>I</b>	<b>Giriş</b>	<b>2</b>
<b>II</b>	<b>Ortak Talimatlar</b>	<b>3</b>
<b>III</b>	<b>Genel Bakış</b>	<b>5</b>
<b>IV</b>	<b>Küresel kurallar</b>	<b>6</b>
<b>V</b>	<b>Zorunlu kısım</b>	<b>8</b>
<b>VI</b>	<b>Bonus bölüm</b>	<b>9</b>
<b>VII</b>	<b>Sunum ve akran değerlendirmesi</b>	<b>10</b>

# Bölüm I Giriş

Felsefe (Yunanca, *philosophia*, kelimenin tam anlamıyla "bilgelik sevgisi") varlık, bilgi, değerler, akıl, zihin ve dil hakkındaki genel ve temel soruların incelenmesidir. Bu tür sorular genellikle analiz edilmesi veya çözülmesi gereken problemler olarak çerçevelenir. Bu terim muhtemelen Pythagoras (M.Ö. 570-495) tarafından ortaya atılmıştır. Felsefî yöntemler arasında sorgulama, eleştirel tartışma, rasyonel argüman ve sistematik sunum yer alır.

Klasik felsefî sorular şunlardır: Herhangi bir şey gerçekten bilinebilir ve kanıtlanabilir mi? En gerçek olan nedir? Filozoflar ayrıca şu gibi daha pratik ve somut sorular da sorarlar: Yaşamak için en iyi yol var mıdır? Adil olmak mı yoksa adaletsiz olmak mı daha iyidir (eğer bundan kurtulabilirsek)? İnsanların özgür iradesi var mıdır?

Tarihsel olarak, 'felsefe' terimi herhangi bir bilgi bütününe atıfta bulunur. Antik Yunan filozofu Aristoteles'in zamanından 19. yüzyıla kadar "doğa felsefesi" astronomi, tıp ve fiziği de kapsıyordu. Örneğin, Newton'un 1687 tarihli eseri, *Doğa Felsefesinin Matematiksel İlkeleri*, daha sonra bir fizik kitabı olarak sınıflandırılmıştır.

19. yüzyılda modern araştırma üniversitelerinin büyümesi, akademik felsefenin ve diğer disiplinlerin profesyonelleşmesine ve uzmanlaşmasına yol açmıştır. Modern dönemde, geleneksel olarak felsefenin bir parçası olan bazı araştırmalar psikoloji, sosyoloji, dilbilim ve ekonomi gibi ayrı akademik disiplinler haline gelmiştir.

Sanat, bilim, politika veya diğer uğraşlarla yakından ilgili diğer araştırmalar ise felsefenin bir parçası olarak kalmıştır. Örneğin, güzellik nesnel midir yoksa öznel midir? Çok sayıda bilimsel yöntem mi vardır yoksa sadece bir tane mi? Siyasi ütopya umutlu bir rüya mı yoksa umutsuz bir fantezi midir? Akademik felsefenin başlıca alt alanları arasında metafizik ("gerçekliğin ve varlığın temel doğasıyla ilgili"), epistemoloji ("bilginin doğası ve temelleri [ve]... sınırları ve geçerliliği" hakkında), etik, estetik, siyaset felsefesi, mantık ve bilim felsefesi yer alır.

## Bölüm II

### Ortak Talimatlar

- Projeniz C dilinde yazılmış olmalıdır.
- Projeniz Norm'a uygun olarak yazılmış olmalıdır. Eğer bonus dosyalarınız/fonksiyonlarınız varsa bunlar norm kontrolüne dahil edilir ve norm hatası varsa 0 alırsınız.
- Fonksiyonlarınız, tanımlanmamış davranış dışında beklenmedik bir şekilde (segmentasyon hatası, veri yolu hatası, dou-ble free, vb.) çıkmamalıdır. Bu gerçekleşirse, projeniz işlevsel değil olarak kabul edilecek ve değerlendirme sırasında 0 alacaktır.
- Tüm heap-ayrılmış bellek gerektiğinde uygun şekilde serbest bırakılmalıdır. Bellek sızıntılarına tolerans gösterilmeyecektir.
- Konu gerektiriyorsa, kaynak dosyalarınızı cc kullanarak -Wall, -Wextra ve -Werror bayraklarıyla gerekli çıktıya derleyen bir Makefile göndermelisiniz. Ayrıca, Makefile dosyanız gereksiz yeniden bağlantılar gerçekleştirmemelidir.
- Makefile dosyanız en azından \$(NAME), all, clean, fclean ve yeniden.
- Projenize bonuslar göndermek için, Makefile dosyanıza, projenin ana bölümünde izin verilmeyen tüm çeşitli başlıkları, kütüphaneleri veya işlevleri ekleyecek bir bonus kuralı eklemelisiniz. Bonuslar, konu aksini belirtmediği sürece \_bonus.{c/h} dosyalarına yerleştirilmelidir. Zorunlu ve bonus kısımların değerlendirilmesi ayrı ayrı yapılır.
- Projeniz libft kullanmanıza izin veriyorsa, kaynaklarını ve ilişkili Makefile'ını bir libft klasörüne kopyalamanız gerekir. Projenizin Makefile'ı kütüphaneyi Makefile'ını kullanarak derlemeli, ardından projeyi derlemelidir.
- Bu çalışmanın **teslim edilmesi gerekme ve not verilmeyecek** olsa da, projeniz için test programları oluşturmanızı teşvik ediyoruz. Bu size kendi çalışmanızı ve akranlarınızın çalışmalarını kolayca test etme fırsatı verecektir. Bu testleri özellikle savunmanız sırasında faydalı bulacaksınız. Nitekim, savunma sırasında kendi testlerinizi ve/veya değerlendirdiğiniz akranınızın testlerini kullanmakta özgürsünüz.
- Çalışmanızı atanan Git deposuna gönderin. Yalnızca Git repos-itory'sindeki çalışmalara not verilecektir. Çalışmanıza not vermek için Deepthought görevlendirilirse, bu gerçekleşecektir.

akran değerlendirmelerinizden sonra. Deepthought'un notlandırması sırasında çalışmanızın herhangi bir bölümünde bir hata meydana gelirse, değerlendirme duracaktır.

# Bölüm III Genel

## Bakış

İşte bu ödevde başarılı olmak için bilmeniz gereken temel şeyler:

- Bir veya daha fazla filozof yuvarlak bir masada oturuyor. Masanın ortasında büyük bir kase spagetti vardır.
- Filozoflar sırayla yemek yer, düşünür ve uyurlar. Yemek yerken ne düşünüyorlar ne de uyuyorlar; düşünürken ne yemek yiyorlar ne de uyuyorlar; Ve tabii ki uyurken de ne yemek yiyorlar ne de düşünüyorlar.
- Masanın üzerinde çatal da var. **Filozoflar kadar çok çatal** vardır.
- Spagettiye tek bir çatalla yemek pratik olmadığından, bir filozof yemek yemeden önce hem sağındaki hem de solundaki çatalı almalıdır.
- Bir filozof yemeğini bitirdiğinde çatallarını masaya geri koyar ve uyumaya başlar. Uyandıklarında tekrar düşünmeye başlarlar. Bir filozof açlıktan öldüğünde simülasyon durur.
- Her filozofun yemek yemesi gerekir ve asla açlıktan ölmemelidir.
- Filozoflar birbirleriyle iletişim kurmazlar.
- Filozoflar başka bir filozofun ölmek üzere olup olmadığını bilmezler.
- Söylemeye gerek yok, filozoflar ölmekten kaçınmalıdır!

# Bölüm IV

## Küresel kurallar

Zorunlu bölüm için bir program ve bonus bölümü için başka bir program yazmalısınız (eğer bonus bölümünü yapmaya karar verirsiniz). Her ikisi de aşağıdaki kurallara uymak zorundadır:

- Global değişkenler yasaktır!
- Program(lar)ınız aşağıdaki argümanları almalıdır:  
number\_of\_philosophers time\_to\_die time\_to\_eat time\_to\_sleep  
[number\_of\_times\_each\_philosopher\_must\_eat]
  - number\_of\_philosophers: Filozof sayısı ve ayrıca çatal sayısı.
  - time\_to\_die (milisaniye cinsinden): Bir filozof son yemeğinin başlangıcından veya simülasyonun başlangıcından itibaren time\_to\_die milisaniye içinde yemeye başlamazsa ölür.
  - time\_to\_eat (milisaniye cinsinden): Bir filozofun yemek yemesi için gereken süre. Bu süre zarfında iki çatal tutmaları gerekecektir.
  - time\_to\_sleep (milisaniye cinsinden): Bir filozofun uyumak için harcayacağı zaman.
  - number\_of\_times\_each\_philosopher\_must\_eat (isteğe bağlı bağımsız değişken): Tüm filozoflar en az number\_of\_times\_each\_philosopher\_must\_eat kez yemek yemişse simülasyon durur. Belirtilmezse, bir filozof öldüğünde simülasyon durur.
- Her filozofun 1 ile filozof\_sayısı arasında değişen bir numarası vardır.
- Filozof sayısı 1, filozof sayısı\_of\_filozof'un yanında oturur. N numaralı diğer herhangi bir filozof, N - 1 ve N + 1 filozofları arasında yer alır.

Programınızın günlükleri hakkında:

- Bir filozofun herhangi bir durum değişikliği aşağıdaki gibi biçimlendirilmelidir:
  - timestamp\_in\_ms X bir çatal aldı
  - timestamp\_in\_ms X yemek yiyor
  - timestamp\_in\_ms X uyuyor
  - timestamp\_in\_ms X düşünüyor
  - timestamp\_in\_ms X öldü

*timestamp\_in\_ms yerine milisaniye cinsinden geçerli zaman damgasını ve X yerine de filozof numarasını yazın.*

- Görüntülenen bir durum mesajı başka bir mesajla örtüşmemelidir.
- Bir filozofun öldüğünü bildiren bir mesaj, gerçek ölümünden sonraki 10 ms içinde görüntülenmelidir.
- Tekrar ediyorum, filozoflar ölmekten kaçınmalıdır!



Programınızda herhangi bir **veri yansı** olmamalıdır.



# Bölüm V Zorunlu

## kısım

<b>Program adı</b>	philo
<b>Dosyaları teslim edin</b>	Makefile, *.h, *.c, philo/ dizini içinde
<b>Makefile</b>	NAME, all, clean, fclean, re
<b>Argümanlar</b>	filozof_sayısı ölmek_için_zaman yemek_için_zaman time_to_sleep [number_of_times_each_philosopher_must_eat]
<b>Harici fonksiyonlar.</b>	memset, printf, malloc, free, write, usleep, gettimeofday, pthread_create, pthread_detach, pthread_join, pthread_mutex_init, pthread_mutex_destroy, pthread_mutex_lock, pthread_mutex_unlock
<b>Libft yetkili</b>	Hayır
<b>Açıklama</b>	İş parçacıkları ve muteksler ile filozoflar

Zorunlu bölüm için özel kurallar şunlardır:

- Her filozof ayrı bir iş parçacığı olarak temsil edilmelidir.
- Her filozof çifti arasında bir çatal vardır. Bu nedenle, birden fazla filozof varsa, her filozofun sol tarafında bir çatal ve sağ tarafında bir çatal vardır. Sadece bir filozof varsa, sadece bir çatala erişimleri olacaktır.
- Filozofların çatalları çoğaltmasını önlemek için, her çatalın durumunu bir muteks ile korumalısınız.

# Bölüm VI Bonus

## kısımı

<b>Program adı</b>	philo_bonus
<b>Dosyaları teslim edin</b>	Makefile, *.h, *.c, philo_bonus/ dizini içinde
<b>Makefile</b>	NAME, all, clean, fclean, re
<b>Argümanlar</b>	filozof_sayısı ölmek için zaman yemek için zaman time_to_sleep [number_of_times_each_philosopher_must_eat]
<b>Harici fonksiyonlar.</b>	memset, printf, malloc, free, write, fork, kill, exit, pthread_create, pthread_detach, pthread_join, usleep, gettimeofday, waitpid, sem_open, sem_close, sem_post, sem_wait, sem_unlink
<b>Libft yetkili</b>	Hayır
<b>Açıklama</b>	Süreçleri ve semaforları olan filozoflar

Bonus bölümünün programı zorunlu programla aynı argümanları alır. *Global kurallar* bölümündeki gerekliliklere uymak zorundadır.

Bonus bölümü için özel kurallar şunlardır:

- Tüm çatalar tablonun ortasına yerleştirilir.
- Bellekte durumları yoktur, ancak mevcut çataların sayısı bir semafor ile temsil edilir.
- Her bir filozof ayrı bir süreç olarak temsil edilmelidir. Ancak, ana süreç bir filozof olarak hareket etmemelidir.



Bonus kısmı sadece zorunlu kısım MÜKEMMEL ise değerlendirilecektir. Mükemmel, zorunlu kısmın bütünsel olarak yapıldığı ve hatasız çalıştığı anlamına gelir. TÜM zorunlu gereklilikleri geçmediyseniz, bonus bölümünüz hiç değerlendirilmeyecektir.

## Bölüm VII

### Sunum ve akran değerlendirmesi

Ödevinizi her zamanki gibi Git deponuza gönderin. Savunma sırasında yalnızca depolarınızdaki çalışmalar değerlendirilecektir. Doğru olduklarından emin olmak için dosyalarınızın adlarını iki kez kontrol etmekten çekinmeyin.

Zorunlu parça dizini: philo/

Bonus bölüm dizini: philo\_bonus/

