# BLGM 343 – DENEY 5*
# İSİMSİZ TÜNELLER

## *Amaçlar*

1. İşlemler arası iletişim konusunda bilgi sahibi olma
2. POSIX tabanlı işletim sistemlerinde isimsiz tünellerin kullanımı

## *İşlemler arası iletişim*

İşlemlerin bazı durumlarda veri alışverişi yapması gerekmektedir. Bu durum daha çok ana işlem alt işlem arasında gerçekleşir. Örneğin, hesaplama işlemini yapan alt işlem, ana işleme sonucu göndermek isteyebilir. Bu tarz durumlarda, işlemler arası iletişim yöntemleri kullanılmaktadır. Bu deneyde bu yöntemlerden en sık kullanılan, ancak yalnızca ilişkili işlemler arasında (ana işlemle alt işlem yada bir ana işlemin alt işlemleri arasında) kullanılabilen isimsiz tüneller hakkında pratik yapacağız.

## *İsimsiz tüneller*

## *Ön Bilgi*

Tünel (pipe), iki işlem arasında tek yönlü veri akışını sağlayan bir yapıdır. Her tünelin iki ucu vardır, yazma ve okuma. Bu iki uç için iki ayrı tam sayı dosya tanımlayıcısı vardır. Bu tanımlayıcılar başarılı bir pipe sistem çağırısının ardından oluşturulurlar. Tünel kullanarak bir işlem tünele yazarken, diğeri de tünelden veri okuyabilir.

Tüneller genel olarak ilişkili işlemlerin aralarında veri transferini sağlamak için kullanılırlar. Açılan tünel tanımlayıcıları, dosya tanımlayıcıları gibi alt işlemler tarafından erişilebilir.

**pipe()** sistem çağırısı aşağıdaki gibi kullanılabilir:

```
int tanimlayicidizisi[2];
sonuc = pipe(tanimlayicidizisi);
```

* BLGM 343 dersi için Güz 2013/2014 döneminde Gürcü Öz ve Cem Kalyoncu tarafından hazırlanmıştır

Burada, sonuc değeri eğer çağrı başarılıysa 0, hata oluştuysa -1 değerini alır. tanimlayicidizisi, iki elemanlı bir dizi olmalıdır. Eğer çağrı başarılıysa, 0. eleman okuma, 1. eleman yazma dosya tanımlayıcısının değerine atanır.

## *Deney 1*

Bir ana işlem ve iki alt işlemden oluşan bir program yazınız. Bu programda, ilk alt işlem komut satırından verilen dosyayı okuyarak, tünel aracılığıyla diğer alt işleme göndersin. Bu işlem 100'er bytelık bloklar halinde yapılsın. Diğer alt işlem ise bu okuduğu bilgiyi ekrana yazdırsın.

**Not:** açılan tünelin alt işlemler tarafından erişilebilir olması için **fork()** sistem çağrılarından önce olması gerekmektedir.

## *Dosya tanımlayıcı değişikliği*

Gerekli olması durumunda, bir dosya tanımlayıcısı başka bir tanımlayıcı numarayla değiştirilebilir. Bunun için **dup2()** sistem çağrısı kullanılmaktadır. **dup2()** sistem çağrısı ilk parametre olarak değiştirilecek olan dosya tanımlayıcısını alır. İkinci parametresi ise hedef tanımlayıcıdır. Eğer bu tanımlayıcı kullanılmaktaysa, önce kapatılır. Aşağıdaki kod parçası açılan bir dosyanın tanımlayıcısını 1 numaralı dosya tanımlayıcısıyla (standart çıktı) değiştirmektedir. Bunun sonucunda, printf() fonksiyonu yazdırma işlemini ekrana değil dosyaya yapmaktadır.

```
int dosya;
dosya=creat("dosya.txt", 0644);
dup2(dosya, 1);
printf("merhaba!\n");
```

## *Deney 2*

**execlp()** sistem çağrısıyla **ls** komutunu çalıştıran bir program yazınız (benzer bir programı önceki deneylerde hazırlamıştık). Bu programda **dup2()** sistem çağrısını kullanarak **ls** programının sonucunun ekrana değilde bir dosyaya yazılmasını sağlayınız.

## *Ek çalışmalar*

1. Genelde birden fazla işlem aynı tünele yazabilir veya birden fazla işlem aynı tünelden okuyabilir. Üç alt işlem yaratıp, iki alt işlemin tünele veri yazmasını, diğerinin de bu tünelden veri okuyarak, ekrana yazmasını sağlayınız

2. Yeni bir program yazarak oluşturulan iki işlem arasında çift yönlü iletişimi isimsiz tüneller kullanarak sağlayınız.

3. Bir program yazarak ls komutunun çıktısını sort komutunun girdisine yönlendiriniz.

## *Sorular*

1. Bu deneyin amacı nedir?

2. Tünellerin amacı nedir?

3. Linux/UNIX İşletim sistemlerinde kullanılan tüneller nelerdir?

4. pipe() sistem çağrısının parametrelerini açıklayınız.

5. pipe() çağrısına gönderilen parametrenin, bu çağrıdan önce anlamlı bir değere sahip olması gerekir mi? pipe() sistem çağrısı çalıştıktan sonra bu parametrenin değeri ne olur?

6. Bir programda, oluşturulan tünelin uçlarını nasıl standart girdi ve çıktıya bağlarsınız?

7. İsimsiz tüneller, bir biriyle bağlantısı olmayan işlemler tarafından kullanılabilirler mi?

8. Bir tünel, onu kullanan işlem bittiği zaman varolmaya devam eder mi?

9. Tüneller, işlemler arasında iki yönlü iletişimi sağlamak için nasıl kullanılabilirler?

10. Eğer iki alt işlem aralarında iletişim kurmak istiyorsa, nasıl bir tünel açma yöntemi kullanılmalıdır? Adımlarını belirtiniz. Bir alt işlemin yarattığı tüneli, bir diğer alt işlem