



**Alaa ALMARAWİ**

Proje Konusu:

**C Dili ile Thread Senkronizasyonu**

Ders:

**İşletim Sistemleri**

## İçindekiler

1- Proje Konusu .....	3
2- Proje Süresince Yapılanlar .....	3
3- Ek Açıklamalar .....	5
4- Screenshots: .....	5
5- Kaynakça .....	6



## 1- Proje Konusu

Bu porje thread senkronizasyonu konusunun pratiğinin yapılmasını ön görmektedir.

Sistemde “numbers” isimli bir metin dosyası içinde sayılar yazmaktadır. Bu sayılar “ $x + y + z$ ” işlemine girilecektir.

Sistem 2 programdan oluşmaktadır. Programların ismi “xyztotalizer” ve “numberreader”dir.

## 2- Proje Süresince Yapılanlar

### Proje dokümanı :

Bu porje thread senkronizasyonu konusunun pratiğinin yapılmasını ön görmektedir.

Sistemde “numbers” isimli bir metin dosyası içinde sayılar yazmaktadır. Bu sayılar “ $x + y + z$ ” işlemine sokulacaktır.

Sistem 2 programdan oluşmalıdır. Programların ismi “xyztotalizer” ve “numberreader”dir. Birinci program “xyztotalizer” işlemi yapacak olan ana program ve diğer program “numberreader” servis olarak çalışacak programdır. Bu iki program da iki farklı terminalde çalışacaklardır.

Servis programı “numberreader” kendisine mesaj olarak gelen tek karakter değişken ismi ve satır numarası alacaktır. Satır numarasını dosyadan okumak için kullanacak yani o satırda bulunan sayıyı “numbers” isimli metin dosyasından okuyacak ve mesaj olarak ana programa gönderecektir (pipeden önce bir değişken ismi satır numarası değeri okuyor ve metin dosyasından o satır numarasındaki sayıyı okup pipe yazıyor.). Program dosyadan okuduğu her sayıyı satır numarası değişken ismi ve değer olarak ekrana basmalıdır

Örn:

1.  $x = 7$
2.  $z = 2$
3.  $y = 3$
4.  $y = 6$
5.  $x = 83$
6.  $z = 34$

Ana program “xyztotalizer” program içerisinde “xthread”, “ythread”, “zthread” 3 adet isimli thread ve integer tipi global bir “linenum” ve “readcount” değişkeni bulundurmaktadır. Program “readcount” kadar sayıyı okuyabilir ve “readcount” sadece 3’ün katları olabilir (“readcount” ana programa parametre olarak geçilecektir.). X,Y,Z threadleri kendileri her seferinde sadece biri “numberreader” programına o anki “linenum” değerini göndermekte ve geri o satırdaki sayıyı almakta ve ekrana hangi thread okuma yapmışsa “ $x = \dots$ ”, “ $y = \dots$ ”, “ $z = \dots$ ” şeklinde aldıkları değerleri yazmalıdır. Ardından “linenum” değerini artırmalıdır (X,Y,Z threadlerinin ayrı fonksiyonları olmalı ve “numberreader” programı ile iletişim için bir fonksiyon olmalı, bu fonksiyon threadlerden çağırılmalı).

Ana program threadlerden okunan değerleri toplayacaktır.

Örn:

```
./xyztotalizer 6
x = 7
z = 2
y = 3
7 + 3 + 2 = 12
```

z = 34

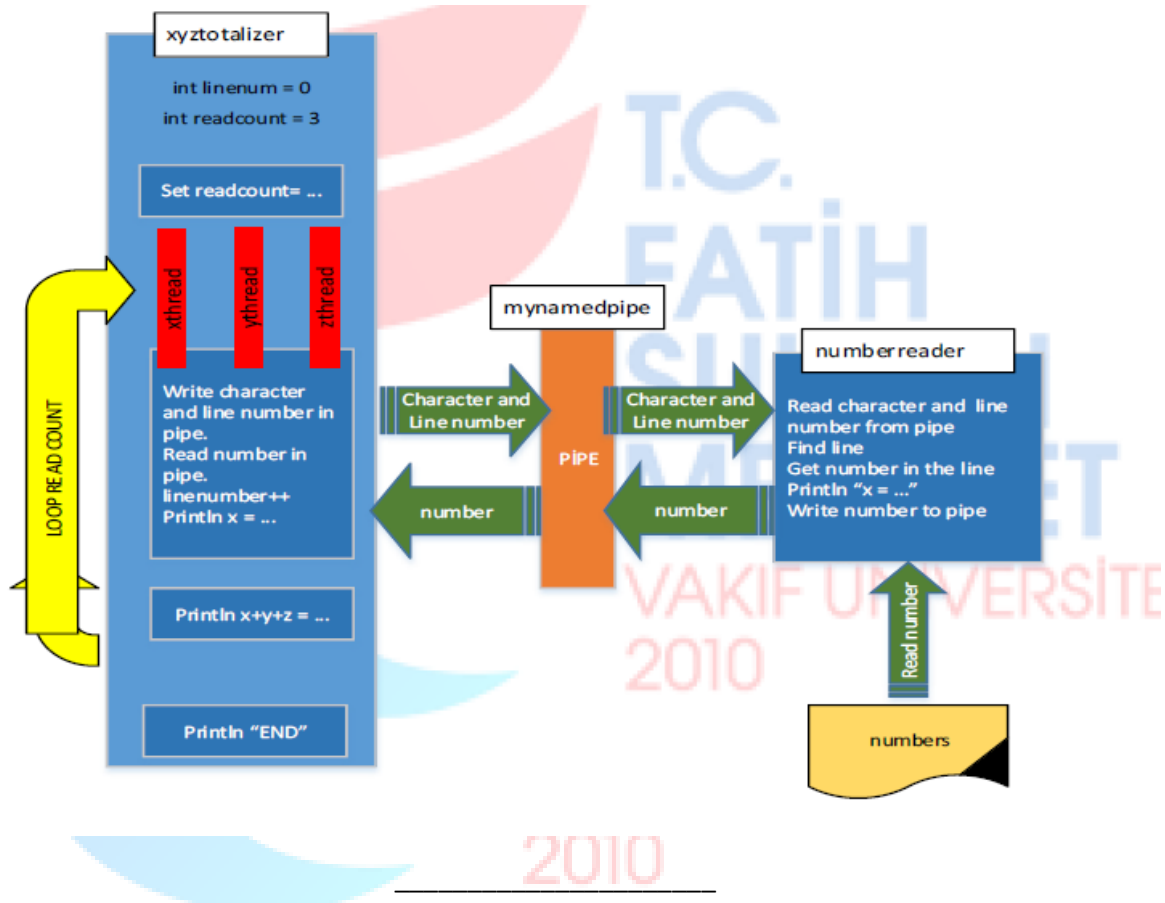
y = 6

x = 83

83 + 6 + 34 = 123

#### Program Akışı:

- İki program farklı terminallerde çalışır.
- Namedpipe yapısı kullanılmış olması gerekir.
- Programların senkronizasyon gerektiren yerlerini yapmalısınız.
- Sleep ve benzeri yapılan senkronizasyonlar pek makbul değildir.
- Size gönderilen dosya içindeki rakam ve sıra ile çalışınız(numbers.txt).



- Numberreader:

Servis olarak çalışır. Pipe'ten xyztotlizer'ın gönderdiği stringi okur, karakter ve sayıya ayırır.

Sayı satır numarası olarak alır ve o satırdaki değer okuyup pipe'e geri yazar.

Eğer dosya sonuna geldi ise pipe END yazar. Xyztotlizer programı da dosya sonuna geldiğini algılar ve toplam son hali yazıp programı bitirir.

- Xyztotlizer :

Kullanıcıdan alınan read count kadar numbers.txt dosyasından satır okuyacaktır.

Named pipe kullanarak sırayla x,y,z karakterlerinden biri ve okunacak satır numarasını (1string olarak) pipe'e gönderir.

Sonrasından pipe ile geri gönderilen değer (o satırdaki rakam) 'ı okuyup o anki karaktere atar.

X,y,z değerleri belli olunca toplamı ekrana yazdırır. Ve daha okunacak satırlar varsa yeniden

x,y,z leri kullanır.

- Thread senkronizasyonu için mutex lock mekanizması kullanıldı. Sırayla x,y,z threadleri kendi fonksiyonlarını çalıştırabilmek için ilk önce y,z threadleri kilitlendi. Xthread işleme girince xthreadi kilitlenir ve bitirince ythread kilidini açar... sürekli dönebilmek için zthread işini bitirince yine zthreadin kilidi açar.

- Program read count kadar döner. Ve her döngüde 3 tane thread oluşturur ve bir sonraki döngüye girmeden pthread\_join() ile bu threadleri bekler.

- Program bitince tanımlanan mutex anahtarları yok edilir.

### 3- Ek Açıklamalar

karşılaşılan zorluklar :

- named pipe ile peşpeşe iki bilgi gönderemedim, karakter ve satır numarasını strcat() ile bir string'te birleştirip gönderdim, diğer programda ayırıp satır numarası bilgisi kullanır.
- numberread programında stringi pipten okuyup sstring[1] elemanını (line number) integer'e çevirmek için atoi gibi bir araç kullanmam gerekir ancak atoi, strint ve diğer çevirme metodları char\* tipinden bir parametre bekler (sstring[0] char tipi). Çözüm olarak pipe göndermeden önce arada ayraç "-" koydum , numberreader'da ayrıca böler ve iki string döndürür.
- strcat() kullanınca bazen doğru çalışmadı. String'i ilk tanımlayınca doğru yer ayırmam gerektiğini buldum ve karakter,ayraç,line numarası uzunlukların toplamı kadar calloc ile yer ayırdım.
- numberreader programı servis olarak çalışmalı, dolayısıyla ana program kapanıp birdaha çağırıldığında servise yeni değer okumaya hazır olmalı ve bu şamada hata veriyordu. Bunu sebebi line değişkeni (satır değeri) son okuma işleminden dolu olmasındandır. Dolu bir string'e değer atamaya c dilinde strcpy kullanmad izin vermez. Çözüm olarak her okuma işleminden önce o değişkeni boşalttım(null).

### 4- Screenshots:

```
alaa@AlaaDesktop-KO4HM62: /mnt/g/universite/BLM/5.Donem/İşletimSistemleri/LAB
project3$ make
gcc numberreader.c -o numberreader
gcc xyztotalizer.c -o xyztotalizer -l pthread
alaa@AlaaDesktop-KO4HM62: /mnt/g/universite/BLM/5.Donem/İşletimSistemleri/LAB
project3$ ./xyztotalizer
Enter read counter (must be a multiple of 3): 8
4
3
Send x-1
Recieved Number: 7
Send y-2
Recieved Number: 2
Send z-3
Recieved Number: 3
x + y + z = 12
alaa@AlaaDesktop-KO4HM62: /mnt/g/universite/BLM/5.Donem/İşletimSistemleri/LAB
project3$ ./xyztotalizer
Enter read counter (must be a multiple of 3): 30

alaa@AlaaDesktop-KO4HM62: /mnt/g/universite/BLM/5.Donem/İşletimSistemleri/LAB
project3$ ./numberreader
** numberreader.c
recieved: |x-1|
recieved character:x
recieved line: 1
File opened
1. x = 7
-----
recieved: |y-2|
recieved character:y
recieved line: 2
File opened
2. y = 2
-----
recieved: |z-3|
recieved character:z
recieved line: 3
File opened
3. z = 3
-----
```

```
alaa@AlaaDesktop-KO4HM62: /mnt/g/universite/BLM/5.Donem/IşletimSistemleri/LA...  
Received Number: 3  
x + y + z = 12  
Send x-4  
Received Number: 6  
Send y-5  
Received Number: 83  
Send z-6  
Received Number: 34  
x + y + z = 123  
Send x-7  
Received Number: 1  
Send y-8  
Received Number: 4  
Send z-9  
Received Number: 6  
x + y + z = 11  
Send x-10  
Received Number: 78  
Send y-11  
Received Number: 5  
Send z-12  
Received Number: 7  
x + y + z = 90  
Send x-13  
Received Number: 2  
Send y-14  
Received Number: 4  
Send z-15  
Received Number: 6  
x + y + z = 12  
Send x-16  
Received Number: 3  
Send y-17  
Received Number: 4  
Send z-18  
Received Number: 5  
x + y + z = 12  
Send x-19  
File end has reached!  
Last sum= 0  
END  
alaa@AlaaDesktop-KO4HM62: /mnt/g/universite/BLM/5.Donem/IşletimSistemleri/LA...  
project:$  
-----  
received: |y-14|  
received character:y  
received line: 14  
File opened  
14. y = 4  
-----  
received: |z-15|  
received character:z  
received line: 15  
File opened  
15. z = 6  
-----  
received: |x-16|  
received character:x  
received line: 16  
File opened  
16. x = 3  
-----  
received: |y-17|  
received character:y  
received line: 17  
File opened  
17. y = 4  
-----  
received: |z-18|  
received character:z  
received line: 18  
File opened  
18. z = 5  
-----  
received: |x-19|  
received character:x  
received line: 19  
File opened  
File End Reached!  
Return value E  
19. x = END  
-----
```

## 5- Kaynakça

<https://stackoverflow.com/>

<https://www.tutorialspoint.com/index.htm>

T.C.  
FATİH  
SULTAN  
MEHMET  
VAKIF ÜNİVERSİTESİ  
2010