

BIL 344

SYSTEM PROGRAMMING

FINAL PROJECT

Gözde DOĞAN

131044019

Course Teachers:

Erkan ZERGEROĞLU

Erchan APTOULA

CLIENT:

Aşağıdaki şekildeki gibi çalıştırılıyor;

`./HomeworkClient "Name" Priority IntegerDenotingDegree ServerAddr ServerPortAddr`

- Priority, Q | q , C | c ve T | t için çalışmaktadır. Kullanıcının yanlış girme ihtimali göz önüne alınarak bu şekilde tasarlanmıştır.
- ServerAddr, 127.0.0.1 gibi bir adres olmalıdır. Socket haberleşmesinde girilen bu değer kullanılmaktadır.
- ServerPortAddr, server çalıştırılırken girilen port adresi ile aynı olmalıdır. Yoksa client bir bağlantı kuramaz ve bağlantı kuramadığına dair bir mesaj yazıp çıkar.

Client sadece verilen bilgiler ile socket'e bağlanıp verileri yolluyor. Bunun içinde bir struct yapısı kullandım.

```
typedef struct{  
  
    char ClientName[MAX_STRING_SIZE];  
  
    char Priority;  
  
    int Degree;  
  
    pid_t pid;  
  
} Client;
```

Kullanıcının girdiği argümanlara ek olarak bir de kendi pid'sini server'a yolluyor.

Server tarafından işleminin yapılıp sonucunun kendisine gönderilmesini bekliyor. Gönderilen veriyi almak için de bir struct yapısı oluşturdum.

```
typedef struct {  
  
    char providerName[MAX_STRING_SIZE];  
  
    double homeworkResult;
```

```
double spendTime;  
  
int cost;  
  
} ReceiveInformation;
```

Server tarafından gönderilen, ödevi kimin çözdüğü, ödevin sonucu, ne kadar sürede yapıldığı ve maliyeti bilgilerini alıp terminal'e yazdırarak işlemini bitiriyor.

NOT: Client valgrind ile çalıştırıldığı zaman bir sürü error olduğu görülmektedir. Bu errorlar çözülememiştir. Kaynağı bulunamamıştır. Farklı ger input değeri için error değeri de değişmektedir.

SERVER:

Aşağıdaki şekildeki gibi çalıştırılıyor;

```
./HomeworkServer PortAddress "FileOfProviders" "LogFileName"
```

Girilen argümanları alarak socket yapısını oluşturur ve socket'e bağlanılmasını bekler.

Socket'e bir client bağlanıp verilerini yollayana kadar;

- FileOfProviders tarafından tutulan provider bilgilerini alarak bir struct yapısında tutar.

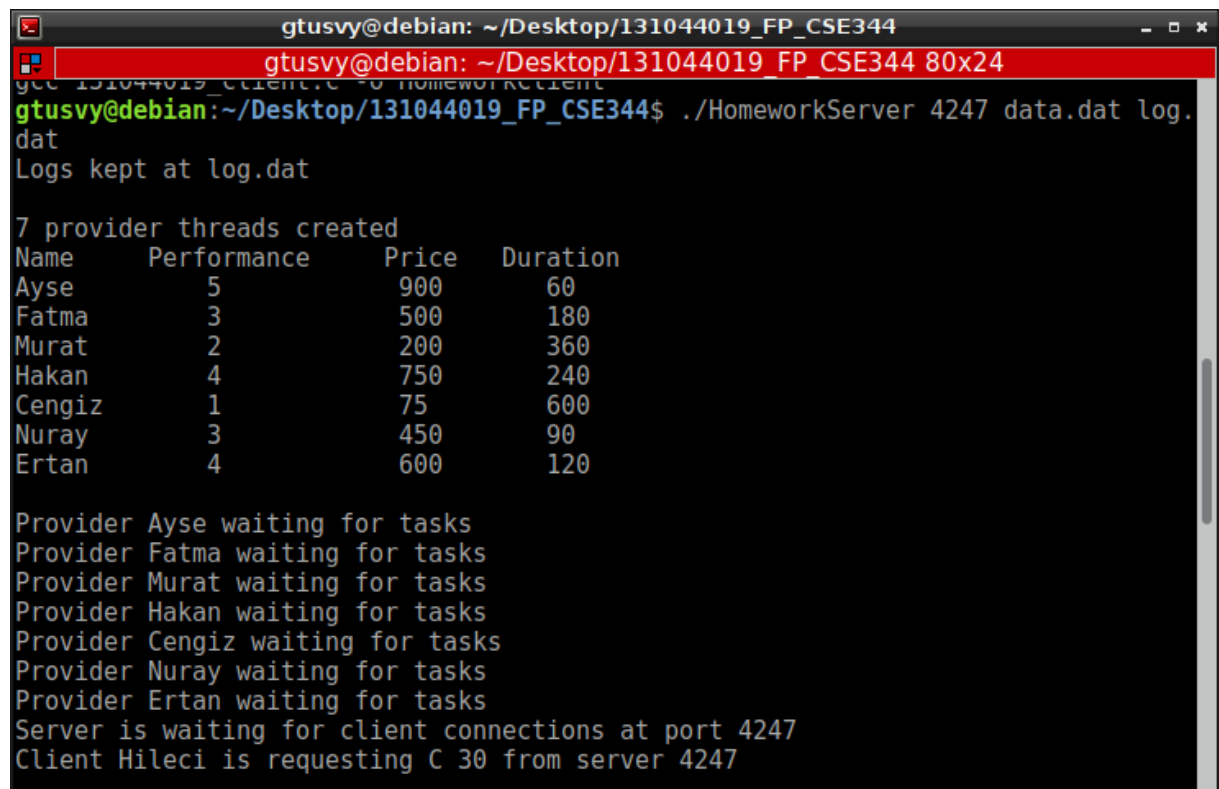
```
typedef struct {  
  
    char ProviderName[MAX_STRING_SIZE];  
  
    int performance;  
  
    int price;  
  
    int duration;  
  
    int numOfClients;  
  
    char queue[MAX_QUEUE_SIZE][MAX_STRING_SIZE];  
  
    int indexQueue;
```

} Providers;

- İlk 4 değişken dosyada yer alan veriler ile doldurulur.
- Son 3 değişken client bağlandıkça değişen değişkenlerdir.
- numOfClients; provider tarafından ödevi yapılan client sayısını tutar.
- queue; İşlem yapacağı clientları tutar. Clientlar önce bu queue'ya eklenir. Ardından provider'in kendisini alıp işini yapmasını bekler.
- indexQueue; queue için tutulmuş bir index.
- (Queue'da maksimum 2 eleman olacağı için indexQueue kontrol edilerek queue'ya eleman eklenir. Kuyruktan eleman alınıp işleme tabi tutulduğu zaman kuyruğa tekrar eleman eklenebilmektedir.)

NOT: Bir provider'in toplam da yapacağı işlem sayısının maksimum 2 olacağı düşünüülerek yapılmıştır.

- Provider bilgilerini “cost”, “performance”, “duration” özelliklerine göre sıralar. Bunları gelen client'in priority bilgisine göre hangi işlemi yapmasının uygun olacağını belirlemek için yapar.



```
gtusvy@debian: ~/Desktop/131044019_FP_CSE344
gtusvy@debian: ~/Desktop/131044019_FP_CSE344 80x24
gtusvy@debian:~/Desktop/131044019_FP_CSE344$ ./HomeworkServer 4247 data.dat log.
dat
Logs kept at log.dat

7 provider threads created
Name      Performance    Price    Duration
Ayse      5              900      60
Fatma     3              500      180
Murat     2              200      360
Hakan     4              750      240
Cengiz    1              75       600
Nuray     3              450      90
Ertan     4              600      120

Provider Ayse waiting for tasks
Provider Fatma waiting for tasks
Provider Murat waiting for tasks
Provider Hakan waiting for tasks
Provider Cengiz waiting for tasks
Provider Nuray waiting for tasks
Provider Ertan waiting for tasks
Server is waiting for client connections at port 4247
Client Hileci is requesting C 30 from server 4247
```

Figure 1: Server socket'e client bağlanmasını bekliyor

- Bir “while(1)” döngüsü (sonsuz döngü) içinde kendisine client bağlanmasını bekler.
- Bağlanan clientların işlemlerini halledip yolladığında yeni bir client bağlantısı bekler.
- “CTRL-C” sinyali gelene kadar beklemeye devam eder. (“CTRL-C” sinyali burada Termination Signal olarak kullanıldı.)
- Termination Signal – “CTRL-C” Sinyali geldiğinde bütün provider’ları ve kaç client için işlem yaptıklarını (Providers struct’ındaki “int numOfClients” değişkeni) ekrana yazdırır, açık her şeyi kapatır ve çıkar.

```

gtusvy@debian: ~/Desktop/131044019_FP_CSE344
gtusvy@debian: ~/Desktop/131044019_FP_CSE344 80x24
Client Hileci7 (Q 0) connected, forwarded to provider Murat
Provider Murat is processing task number 0: 0
Provider Murat completed task number 0: cos(0)=1.00 in 0 seconds.
Ending connection
^C
SERVER SHUTDOWN

Termination signal received
Terminating all clients
Terminating all providers

Statistics
Name          Number of clients served
-----
Ayse          1
Fatma         0
Murat         1
Hakan         0
Cengiz        2
Nuray         0
Ertan         0

Goodbye.
gtusvy@debian:~/Desktop/131044019_FP_CSE344$

```

Figure 2: CTRL-C sinyali geldiğinde server'da son durum

- Client bağlandığında;
 - Client bilgilerini okur.
 - Bilgilerini okuduğu client’i threadpool’a ekler.
 - Threadpool bu işlemi çözmek için aldığı (void *) fonksiyonu ve (void *) argümanı olarak işlem yapar.
 - (void *) fonksiyon içinde ödevi yapabilmek için;
 - Priority bilgisine göre işlemi yapacak provider seçilir.

- Provider seçerken “queue” ‘nun dolu olup olmadığı da kontrol edilir.
- Seçilen provider’ın kuyruğuna client eklenir.
- Client ödevi çözülür.
- Ödev çözümü için taylor serisi mantığı kullanıldı.
- Çözülen ödevi bir fifo’ya yazar.
- Fifo ile thread’i oluşturan process (süreç) ‘e yollar.
- Thread fonksiyonu içinde socket’e yazarak geri yollanmaya çalışıldığında sorunlar çıktı, bazen doğru gitti, bazen yanlış gitti, bazen eksik gitti, bu nedenle thread’i oluşturan sürece gönderildi ve orada yollandı. Böyle olduğunda verilerin doğru aktarıldığı gözlemlendi.
- Thread’i oluşturan process, fifo’ya yazılan verileri okur ve socket ile client’a yollar.

*NOT: void *ThreadServer(void *arguman); fonksiyonu, (void *) olarak tanımlandığı zaman bir uyarı veriyor ama sorunsuz çalışıyor. (void) olarak kullanıldığında ise uyarı vermiyor ve sorunsuz çalışıyor.*

```

gtusvy@debian: ~/Desktop/131044019_FP_CSE344
gtusvy@debian: ~/Desktop/131044019_FP_CSE344 80x38
Cengiz      1      75      600
Nuray       3      450     90
Ertan       4      600     120

Provider Ayse waiting for tasks
Provider Fatma waiting for tasks
Provider Murat waiting for tasks
Provider Hakan waiting for tasks
Provider Cengiz waiting for tasks
Provider Nuray waiting for tasks
Provider Ertan waiting for tasks
Server is waiting for client connections at port 4247
Client Hileci is requesting T 60 from server 4247

Client Hileci (T 60) connected, forwarded to provider Ayse
Provider Ayse is processing task number 0: 60
Provider Ayse completed task number 0: cos(60)=0.50 in 0 seconds.
Client Hileci3 is requesting C 27 from server 4247

Client Hileci3 (C 27) connected, forwarded to provider Cengiz
Provider Cengiz is processing task number 0: 27
Provider Cengiz completed task number 0: cos(27)=0.89 in 0 seconds.
Client Hileci2 is requesting T 90 from server 4247

Client Hileci2 (T 90) connected, forwarded to provider Ayse
Provider Ayse is processing task number 1: 90
Provider Ayse completed task number 1: cos(90)=0.00 in 0 seconds.
Client Hileci7 is requesting Q 0 from server 4247

Client Hileci7 (Q 0) connected, forwarded to provider Cengiz
Provider Cengiz is processing task number 1: 0
Provider Cengiz completed task number 1: cos(0)=1.00 in 0 seconds.
Client Hileci5 is requesting C 97 from server 4247

Client Hileci5 (C 97) connected, forwarded to provider Murat
Provider Murat is processing task number 0: 97
Provider Murat completed task number 0: cos(97)=-0.12 in 0 seconds.

Hileci's task completed by Ayse in 0.00 seconds, cos(60)=0.50, cost is 900TL, total time spent 0.00 seconds.
gtusvy@debian:~/Desktop/131044019_FP_CSE344$ ./HomeworkClient Hileci3 C 27 127.0.0.1 4247

Client Hileci3 is requesting C 27 from server 127.0.0.1:4247

Hileci3's task completed by Cengiz in 0.00 seconds, cos(27)=0.89, cost is 75TL, total time spent 0.00 seconds.
gtusvy@debian:~/Desktop/131044019_FP_CSE344$ ./HomeworkClient Hileci2 T 90 127.0.0.1 4247

Client Hileci2 is requesting T 90 from server 127.0.0.1:4247

Hileci2's task completed by Ayse in 0.00 seconds, cos(90)=0.00, cost is 900TL, total time spent 0.00 seconds.
gtusvy@debian:~/Desktop/131044019_FP_CSE344$ ./HomeworkClient Hileci7 Q 0 127.0.0.1 4247

Client Hileci7 is requesting Q 0 from server 127.0.0.1:4247

Hileci7's task completed by Cengiz in 0.00 seconds, cos(0)=1.00, cost is 75TL, total time spent 0.00 seconds.
gtusvy@debian:~/Desktop/131044019_FP_CSE344$ ./HomeworkClient Hileci5 C 97 127.0.0.1 4247

Client Hileci5 is requesting C 97 from server 127.0.0.1:4247

Hileci5's task completed by Murat in 0.00 seconds, cos(97)=-0.12, cost is 200TL, total time spent 0.00 seconds.
gtusvy@debian:~/Desktop/131044019_FP_CSE344$

```

Figure 3: Client'lar servera bağlanıyor (sleep yok)

Threadpool mantığı oracle sitesinden incelenerek implement edilmiştir. Kodların büyük çoğunluğu oradan alınmıştır.

Oracle Sitesi: <https://docs.oracle.com/cd/E19253-01/816-5137/ggedn/index.html>

Threadpool mantığını kullanırken,

- Öncelikle threadpool verilen sayı ile initialize edilmiştir. (Burada verilen sayı provider sayısıdır.) (while döngüsünden önce initialize edilmiştir. Çünkü provider thread lerinden oluşan havuzda gelen client için işlem yapılacaktır.)
- Her bağlanan client, threadpool'a bir görev olarak eklenmiştir.
- Ardından threadpool'un bunu çözmesi beklenmiştir.
- ThreadPool bunu çözebilmek için bir (void *) fonksiyon adı ve bir (void *) argüman almaktadır.

Valgrind ile çalıştırıldığında birçok hataya (ERROR) rastlanmıştır. Bir başka hata ise server'a gelen 5 ya da 6 client isteğinden sonra server donmaktadır, bu sorun çözülememiştir. Ve bu soruna bazen rastlanılmaktadır.

Threadpool kaldırılarak normal thread kullanılarak yapıldığında ise server'a gelen sonsuz client isteği sorun çıkarmamıştır. (50 isteğe kadar denendiğinde bir sorun yaşanmamıştır.)

NOT: Ödev threadpool açık şekilde yollanmıştır. Yani threadpool kullanılmaktadır. Ama Threadpool olmadan denemek isterseniz, 105-114 satır aralarını, 157-169 satır aralarını ve 280. satırı yoruma almalısınız ve ardından 171-181 satır aralarını ise yorumdan çıkarmalısınız.

Log file'da server terminalinde çıktı olarak verilen her şey yazılmıştır. Bunun dışında ek olarak her client bağlandığında "client connected to server, clientpid:..." şeklinde bir yazı eklenmiştir.


```
File Edit Selection Find View Goto Tools Project Preferences Help
131044019_server.h x 131044019_server.c x log.dat x 1310440
1
2 7 provider threads created
3 Provider Ayse waiting for tasks
4 Provider Fatma waiting for tasks
5 Provider Murat waiting for tasks
6 Provider Hakan waiting for tasks
7 Provider Cengiz waiting for tasks
8 Provider Nuray waiting for tasks
9 Provider Ertan waiting for tasks
10 Server is waiting for client connections at port 4247
11 Client Hileci1 connected server. ClientPid: 11716
12 Client Hileci1 is requesting Q 45 from server 4247
13
14 Client Hileci1 (Q 45) connected, forwarded to provider Cengiz
15 Provider Cengiz is processing task number 0: 45
16 Provider Cengiz completed task number 0: cos(45)=0.71 in 0 seconds.
17 Ending connection
18 Client Hileci5 connected server. ClientPid: 11719
19 Client Hileci5 is requesting T 0 from server 4247
20
```

Figure 6: Log dosyası _ part1

```
Client Hileci8 (C 270) connected, forwarded to provider Murat
Provider Murat is processing task number 0: 270
Provider Murat completed task number 0: cos(270)=-0.00 in 0 seconds.
Ending connection
Client Hileci3 connected server. ClientPid: 11726
Client Hileci3 is requesting T 172 from server 4247
|
Client Hileci3 (T 172) connected, forwarded to provider Ayse
Provider Ayse is processing task number 1: 172
Provider Ayse completed task number 1: cos(172)=-0.99 in 0 seconds.
Ending connection

SERVER SHUTDOWN

Termination signal received
Terminating all clients
Terminating all providers

Statistics
Name          Number of clients served
-----
Ayse          2
Fatma         0
Murat         1
Hakan         0
Cengiz        2
Nuray         0
Ertan         0

Goodbye.
```

Figure 7: Log dosyası _ part2