

SİSTEM PROGRAMLAMA DERSİ

VİZE RAPORU

BENGİ YÖRÜKOĞLU
131044046

timerServer.c

Bu programdan beklenen her client için bir process oluşturup $2n \times 2n$ lik içi random sayılarla dolu bir matris oluşturup bu matrisi clienta yollamasıdır. Bu matrisi de client'a özel oluşturduğu matris üzerinde yapmasıdır.

#define FIFO_WRITE (O_CREAT | O_WRONLY) fifo'ya write işlemi yapılmak istendiğinde fifo'yu bu modda açarız. O_CREAT ekleme sebebi program akışında fifo'yu oluşturmama rağmen açılmıyor hatalarını almaktır.

#define MAX_SIZE 400 clienta, serverda determinantı alınabilen bir matrix yollanmak istendiğinde çift boyutlu yollayamayacağım için tek boyutlu bir arraya çekip fifo ile clienta yolladım. MAX_SIZE olarak 400 aldım.

int getClient[MAX_SIZE]; Client pidlerini bu arrayde saklamamız için tanımlanmış bir global arraydir. Bu arraye ihtiyacımız olma sebebi servera sinyal geldiğinde tüm clientları bilgilendirebiliyor olmak isteğimizdir.

int countGetClient=0; Client sayılarını tutan global değişken.

char commonMainFifo[255]; Ana fifo adını saklamak için oluşturulmuş global array.

Fonksiyonlar :

void createFifo(char *fifoName); Bu fonksiyonda parametre olarak girilen isimde bir fifo oluşturuluyor. timerServer'da ana fifo'yu create ettim.

float determinant(float a[25][25], float k); Bu fonksiyon sonsuz döngüde oluşturulan matrislerin determinantını bulur. Determinant sıfıra eşit olduğu takdirde server tekrar tekrar matris üretir. Bunun sebebi oluşturulan matrisimizin clientta tersini bulma gibi bazı işlemlerden geçecek olması. Ancak determinant sıfır ise tersini bulamayız.

void getClientFunction(); Bu fonksiyonun amacı clientların pidlerini getClient arrayinde tutabilmemiz. Tutmak istememizin sebebi servera SIGINT sinyali geldiğinde her clienta serverin ben ölüyorum mesajı yollamak istemesidir.

Sinyaller :

SIGINT sinyali timerServer'a ilk geldiğinde timerServer ilk olarak tüm child

processleri öldürür. Daha sonra bağlı olan tüm client processlere kill ile SIGUSR1 sinyalini gönderir. Bu sinyalin amacı timerServer'ın SIGINT sinyali ile öldüğünü bildirmektir ve tüm client processlerin sonlanması sağlamaktır. Server sona erer.

seeWhat.c

Bu programdan beklenen serverdan matrisi alalmak ve üzerinde bazı işlemler yapabilmek.

```
#define TWO 2 Argüman sayısının eksik girildiğinin if statement'taki kontrolü için 2 yi  
define ettim  
#define FIFO_WRITE (O_CREAT | O_WRONLY) fifo'ya write işlemi yapılmak  
istendiğinde fifo'yu bu modda açarız. O_CREAT ekleme sebebi program akışında  
fifo'yu oluşturmama rağmen açılmıyor hatalarını almaktır.  
#define ZERO 0 exit kontrolleri için 0'ı define ettim  
#define FIFO_READ (O_CREAT | O_RDONLY) fifo'ya read işlemi yapılmak  
istendiğinde fifo'yu bu modda açarız. O_CREAT ekleme sebebi program akışında  
fifo'yu oluşturmama rağmen açılmıyor hatalarını almaktır.  
#define MAX_SIZE 400 serverdan matrisi alırken n en fazla 10 girilebileceği için 20'ye 20'lik  
matrisi 400'lük sızlık bir tek boyutlu array ile alabilirim. serverdan matrisi alırken bu size'ı  
kullanıyorum  
#define DEBUG 0
```

char commonMainFifo[255]; Ana fifo adını saklamak için oluşturulmuş global array

int openFifo; file descriptor fifo açmak için kullandığımız belirteç

int openClientFifo; file descriptor her client'ın özel fifosunu açmak için kullandığımız
belirteç

int clientPid; getpid ile aldığımız client'ın pid'ini bu değişkende saklarız.

int getClient[MAX_SIZE]; client pid'lerinin saklandığı array

int countGetClient=0; getClient arrayine pid'ler doldukça countGetClient dolan pid sayısını
verir

char arrayForCreateClientFifo[255]; client ile server arasındaki iletişimi sağlayabilmek için
her client'a özel fifo oluşturmamızı sağlayan sprintf ile içinde client'ın pid'ini saklayabildiğimiz
array

static int logDegiskeni=1; her client'ın ayrı bir log dosyası oluşmasını istediğim için o log
dosyalarının adını belirlemede logDegiskenini kullandım

char arrayForCreateClientFifo[255]; Her client'ın özel fifosunu oluşturmak için ismini
tuttugumuz array

float kopyaArray1[25][25];

float tersBas[25][25]; Inverse Array'i bastırmak için gerekli olan array

float yedek1[25][25]; girilen 2n 2n matrisin ilk n n lik kısmının saklandığı array

float yedek2[25][25]; girilen 2n 2n matrisin ikinci n n lik kısmının saklandığı array

float yedek3[25][25]; girilen 2n 2n matrisin üçüncü n n lik kısmının saklandığı array

float yedek4[25][25]; girilen 2n 2n matrisin dördüncü n n lik kısmının saklandığı array

float global1[25][25]; girilen $2n \times 2n$ matrisin ilk $n \times n$ lik kısmının saklandığı array
float copyForDortluTers[25][25]; girilen $2n \times 2n$ matrisin her dört parçasının dörde alınmış şekli saklanır
int kernel[3][3]={1,1,-1},{-1,1,-1},{1,1,1}}; Convolution için gerekli 3'e 3'lük matrisimizden biri
float sonucArray[25][25]; Convolution işleminin döndürdüğü array bu global arrayde saklanır ki istedigimiz yerde kullanabilelim
int static flag=1; Bu flagi her clientın loguna tersi,convolutionu ve arrayin kendisini yazdırma işlemlerinde fork oluştuğunda yazma tekrar olmasın diye kontrol amaçlı tutuyorum
float tersiAlinmisArray[25][25];Tersi alınmış arrayi tutar

PROGRAMIN MANTIK AKIŞI

Clienttan servara programı başlatma amaçlı kill komutu ile sinyal gelir.Server ana fifo'yu oluşturur.Time Server'da determinantı sıfır olmayan bir $2n \times 2n$ 'lik 2 boyutlu array oluşturur.Bu matrisin oluşturulma zamanı ve client pid'si ve oluşturulan matrisin determinantı loga yazıldı.Daha sonra bu array client adıyla oluşmuş özel fifodan clienta aktarılır.Aktarılırken 2 boyutlu array tek boyutlu arraye düşürülerek clienta aktarımı kolaylaştırılır.Clientta yani seeWhat programında bu matrix Convolution ve Shifted Inverse Matrix işlemlerinden geçer.Burada Orijinal matrisin determinantından shifted inverse matris determinantı çıkartılır.Ve Orijinal matrisin determinantında convolution işlemi yapılan matrisin determinantı çıkartılır.seeWhat programında 2 fork yaptım ve her bir forkun child'ında bu farkları hesapladım.Convolutionda elemanları $\{1,1,-1\},\{-1,1,-1\},\{1,1,1\}$ olan skew-symetric matrix'i kullandım.seeWhat programında her işlem için ayrı bir log dosyası oluşturdum ve her bir log dosyasının içinde Orijinal,shifted inverse ve convolution matrisleri matlab formunda yazdık.

Burada forkların devreye girmesinden ötürü log dosyalarında tekrarlı yazım sıkıntısı ile karşılaştım.Bu sıkıntıyı global bir değişken tutarak sadece fork yapmadığı zamanlarda dosyaya yazdıracak bir algoritma kurdum.seeWhat programında belirtmek istediğim önemli noktalardan biri shifted inverse yaparken $n \times n$ lik parçanın determinantının alınmadığı oldu.Biliyoruz ki determinantı alınamayan bir matrisin tersi bulunamaz bu sebeple o parçayı aynı şekilde hiç değiştirmeden bıraktım ki farklı hatalar ile karşı karşıya kalmayayım.Matrisin determinant ve tersi işlemlerinde

<http://www.sanfoundry.com/c-program-find-inverse-matrix/> sitesinden yardım aldım. 2D Convolution işlemi için ise <http://www.songho.ca/dsp/convolution/convolution.html> sitesinden yararlandım.

ÖRNEK ÇIKTILAR

Oluşan matrix :

1	2	5	9	9	9	5	8
4	5	2	3	4	6	5	8
3	1	5	0	8	9	4	5
6	0	7	1	8	1	1	1
6	9	0	7	0	5	6	7
3	0	0	7	6	7	5	0
9	0	2	9	1	8	4	7
8	1	9	7	3	2	8	9

MY ORIGINAL MATRIX=

```
[1.000000 2.000000 5.000000 9.000000 9.000000 9.000000 5.000000 8.000000
4.000000 5.000000 2.000000 3.000000 4.000000 6.000000 5.000000 8.000000
3.000000 1.000000 5.000000 0.000000 8.000000 9.000000 4.000000 5.000000
6.000000 0.000000 7.000000 1.000000 8.000000 1.000000 1.000000 1.000000
6.000000 9.000000 0.000000 7.000000 0.000000 5.000000 6.000000 7.000000
3.000000 0.000000 0.000000 7.000000 6.000000 7.000000 5.000000 0.000000
9.000000 0.000000 2.000000 9.000000 1.000000 8.000000 4.000000 7.000000
8.000000 1.000000 9.000000 7.000000 3.000000 2.000000 8.000000 9.000000
]
```

MY CONVOLUTION MATRIX=

```
[8.000000 -1.000000 -6.000000 0.000000 -2.000000 2.000000 -5.000000 6.000000
6.000000 10.000000 14.000000 23.000000 39.000000 25.000000 13.000000 17.000000
17.000000 5.000000 22.000000 -2.000000 20.000000 6.000000 10.000000 14.000000
25.000000 -1.000000 10.000000 6.000000 21.000000 24.000000 25.000000 10.000000
6.000000 13.000000 -1.000000 36.000000 4.000000 15.000000 -5.000000 -2.000000
27.000000 5.000000 20.000000 16.000000 4.000000 18.000000 19.000000 11.000000
21.000000 -6.000000 15.000000 20.000000 2.000000 28.000000 16.000000 9.000000
16.000000 -5.000000 12.000000 7.000000 12.000000 4.000000 16.000000 12.000000
]
```

MY INVERSE MATRIX=

```
[-0.092647 0.099106 -0.043947 0.078610 -0.027078 0.026494 0.068902 -0.022591
0.031307 -0.140417 0.059499 0.039898 0.179802 -0.057733 -0.054129 -0.038247
0.067565 -0.392957 0.201254 -0.009291 0.174087 -0.122160 -0.023861 0.061620
]
```

0.089403 -0.096494 -0.070751 0.016415 0.042346 0.002263 0.022564 -0.006700
0.006702 0.264329 -0.145776 0.092891 -0.126962 0.079548 -0.032000 -0.046614
0.014452 -0.229246 0.198760 -0.059848 0.105548 -0.046185 0.046839 -0.031367
-0.097852 0.079379 0.022087 -0.097828 -0.038451 0.150137 -0.100105 0.122786
0.023307 0.287791 -0.128981 -0.008491 -0.149887 -0.046443 0.040320 -0.007601
]

Oluşan matrix :

**3 1 6 6 8 4 3 1
4 5 8 1 2 5 3 3
7 7 3 9 6 7 6 6
9 7 5 4 2 6 5 5
7 3 3 7 8 6 1 4
1 9 7 6 7 0 9 4
0 2 3 6 2 2 2 3
1 8 7 3 4 4 1 3**

MY ORIGINAL MATRIX=

**[3.000000 1.000000 6.000000 6.000000 8.000000 4.000000 3.000000 1.000000
4.000000 5.000000 8.000000 1.000000 2.000000 5.000000 3.000000 3.000000
7.000000 7.000000 3.000000 9.000000 6.000000 7.000000 6.000000 6.000000
9.000000 7.000000 5.000000 4.000000 2.000000 6.000000 5.000000 5.000000
7.000000 3.000000 3.000000 7.000000 8.000000 6.000000 1.000000 4.000000
1.000000 9.000000 7.000000 6.000000 7.000000 0.000000 9.000000 4.000000
0.000000 2.000000 3.000000 6.000000 2.000000 2.000000 2.000000 3.000000
1.000000 8.000000 7.000000 3.000000 4.000000 4.000000 1.000000 3.000000
]**

MY CONVOLUTION MATRIX=

**[19.000000 0.000000 -3.000000 -13.000000 2.000000 1.000000 -6.000000 4.000000
23.000000 16.000000 34.000000 46.000000 57.000000 47.000000 21.000000 21.000000
42.000000 22.000000 25.000000 10.000000 22.000000 20.000000 18.000000 20.000000
51.000000 8.000000 30.000000 33.000000 42.000000 41.000000 37.000000 25.000000
36.000000 42.000000 12.000000 49.000000 12.000000 27.000000 15.000000 6.000000
31.000000 24.000000 32.000000 31.000000 24.000000 19.000000 38.000000 12.000000
38.000000 24.000000 34.000000 41.000000 14.000000 43.000000 26.000000 25.000000
11.000000 0.000000 19.000000 10.000000 19.000000 9.000000 17.000000 19.000000
]**

MY INVERSE MATRIX=

[0.068748 -0.185376 -0.198592 0.287776 0.046561 0.004038 0.007099 0.005452

0.052961 -0.229490 0.008555 0.139280 -0.107188 -0.017181 -0.084749 0.213166
0.058828 0.021428 -0.188675 0.109422 0.008744 0.008643 0.126882 0.003877
0.183717 -0.330738 -0.102419 0.266421 -0.137735 -0.069764 0.182116 0.124851
-0.069278 0.128753 0.052514 -0.207607 0.166750 0.078851 -0.139734 -0.052409
0.051690 0.179205 0.337931 -0.226123 -0.156438 -0.131833 -0.135660 0.024594
0.035499 0.131527 0.186187 -0.105580 -0.127762 0.038854 -0.091002 -0.130223
-0.473511 0.500051 0.003388 -0.375577 0.416483 0.151758 0.142983 -0.290332
]