

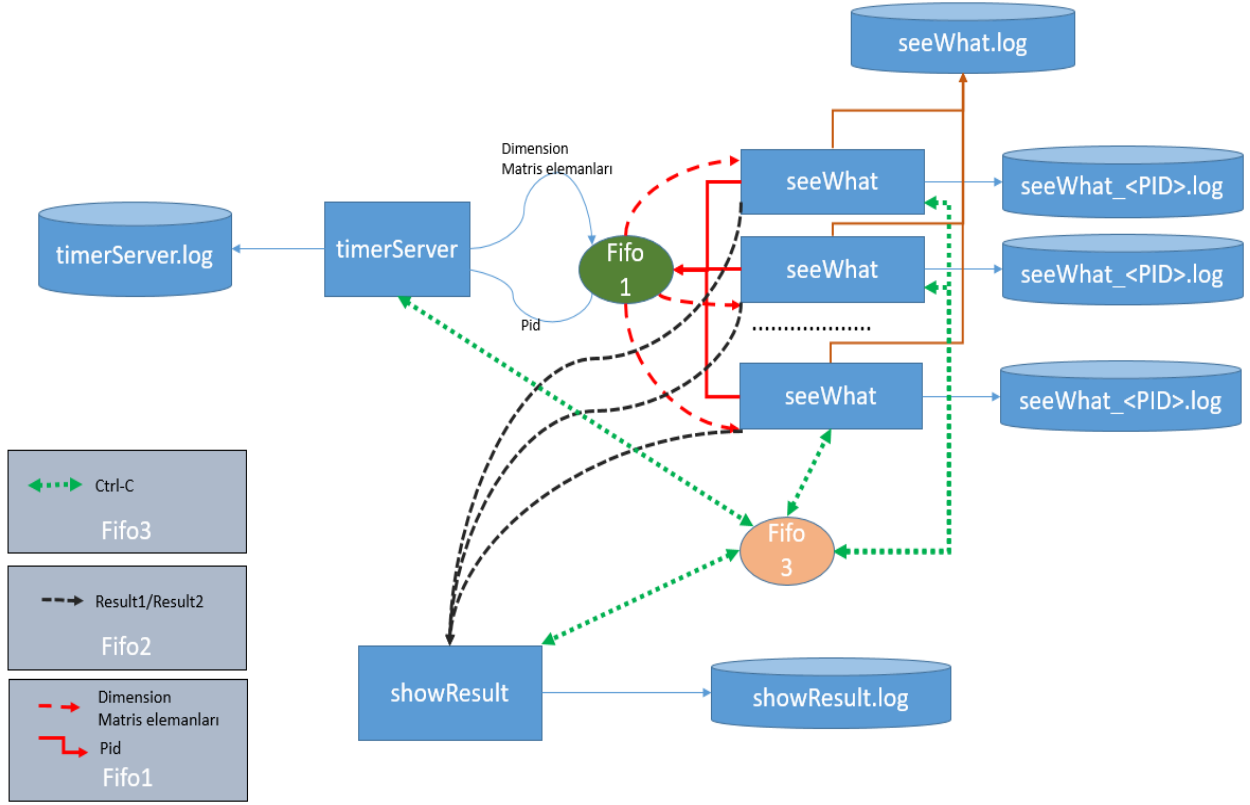
Gebze Technical University
Computer Engineering

CSE 244
2017 Spring

MIDTERM PROJECT REPORT

Elif Şeyma ARMAĞAN
151044042

Lecturer:
Prof. Dr. Erkan ZERGEROĞLU



Projeyi tasarlarken temelde 3 fifo yapısı ile processler arasındaki haberleşmeyi gerçekleştirdim.

Oluşturduğum ilk fifonun adı fifo1 ve bu fifonun ismi kullanıcıdan alınıyor. Bu fifo timerServer ve seeWhat arasındaki haberleşmeyi sağlıyor.

İkinci fifonun adı fifo2 ve ismi resFIFO. Bu fifo seeWhat ve showResult arasında haberleşmeyi sağlıyor.

Üçüncü fifo ise fifo3 ve ismi ctrlFIFO. Bu fifo ise 3 program arasındaki haberleşmeyi sağlıyor.

Fifo1 yani ismi kullanıcıdan alınan fifo iki yönlü çalışmaktadır. seeWhat processleri ile timerServer processleri arasındaki iletişimin koordinasyonunu fifo1 sağlamaktadır. Her seeWhat processi timerServer ile iletişime geçebilmek için fifo1 fifosunu oluşturmak zorundadır. Eğer oluşturuyorsa iletişime geçer. Oluşturamazsa bekler. Bekliyor olması başka bir seeWhat processinin çalıştığını gösterir. seeWhat processleri matris değerlerini timerServer dan aldıktan sonra fifoyu silerek bekleyen seeWhat processlerine fırsat vermektedir.

seeWhat procesi fifo1 i oluşturduktan sonra öncelikle PID değerini timerServer ile paylaşmakta ve paylaştığı PID değerinin timerServer tarafından okunduğundan emin olmak için fifo1 i okumakta ve eğer okuduğu değer gönderdiği PID değeri ise PID değerini tekrar fifoya yazarak tekrar beklemektedir. timerServer process i PID değerini okuduktan sonra öncelikle matris boyutunu göndermekte sonra da girilen matris boyutuna göre matris elemanlarını teker teker göndermektedir. Sonra fifo1 i kapatarak başka bir process in PID değeri göndermesi için beklemeye geçmektedir. seeWhat process i ise PID değerini gönderdikten sonra önce matris boyutunu okumakta daha sonra girilen boyuta göre 1 döngü yardımıyla sürekli fifo1 i okuyarak girilen matrisi tamamlamaktadır. Matrisin değerlerinin okunması tamamlandıktan sonra fifo1 kapatılmakta ve silinmektedir. Daha sonra processler oluşturularak res1 ve res2 değerlerinin hesaplanması yapılmaktadır.

Fifo2 yani resFIFO ise see What processleri ile showResult processleri arasındaki iletişim için kullanılmaktadır. Oluşturulan her bir seeWhat process i PID değeri, res1, res 2 ve bu w result için elapsed time değerlerini sırasıyla fifo2 üzerinden showResult a göndermektedir. showResult process i de belirtilen sırada bu değerleri okuyarak kendi log dosyasına ve ekrana yazar.

Fifo3 yani ctrlFIFO ise tüm processler arasında ctrl-c durumunun paylaşıldığı bir fifodur. Öncelikle her programda INThandler ve checkForCtrlC fonksiyonları bulunmaktadır. Her programın başında SIGINT fonksiyonu ile ctrl-c kesintisinin tetikleyeceği fonsiyon INThandler olarak belirtilmektedir. Böylelikle programa girilen ctrl-c tuşu INThandler fonksiyonu tetikler hale gelmektedir. Böylelikle bütün programlarda ctrl-c kesintisi bu fonksiyon ile yakalanmakta ve ctrl-c kesintisinin yakalandığı program diğer programları bu durumdan haberdar etmek için fifo3 fifosuna benim belirlemiş olduğum 999 değerini yazmaktadır. checkForCtrlC fonksiyonu ise her 3 programın içerisinde düzenli olarak çağrılarak fifo3 ü okuyarak diğer processlerden ctrl-c sinyali alınıp alınmadığını kontrol etmektedir. 999 değerinin okunması durumunda diğer processlerden birinin ctrl-c sinyali aldığını farketmekte ve fifo3 e 999 değerini tekrar yazarak hala sonlanmamış olan processlerin de bu olayı farketmesini sağlayıp gerekli logları yazarak ilgili programı sonlandırmaktadır.

Örnek çalıştırmalar

n 3 için ve ticks 500 için

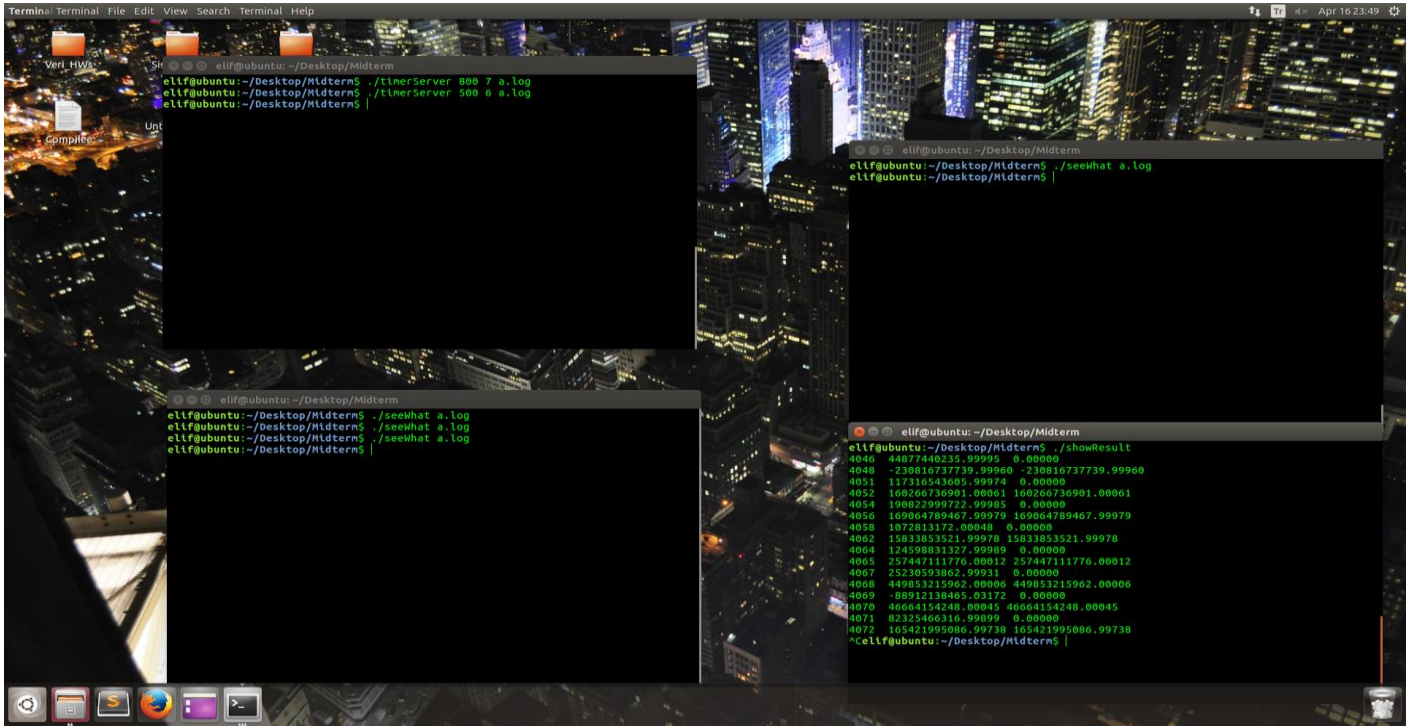
```
Terminal Terminal File Edit View Search Terminal Help
elif@ubuntu: ~/Desktop/Midterm
elif@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
3270 -34576.05707 -214266.00000
3275 1737.00026 -62545.00000
3276 73418.00003 -335066.00000
3277 -4374.00110 29650.00000
3279 -9834.00010 395496.00000
3283 31123.99999 47738.00000
3284 172901.00003 173419.00000
3286 132128.99762 -698253.00000
3289 -126339.99999 -376294.00000
3291 60189.99985 639762.00000
3292 134940.00002 -52624.00000
3296 -86582.99699 105682.00000

elif@ubuntu:~/Desktop/Midterm$ ./timerServer 500 3 a.log

elif@ubuntu:~/Desktop/Midterm$ ./showResult
3270 -34576.05707 -214266.00000
3275 1737.00026 -62545.00000
3276 73418.00003 -335066.00000
3277 -4374.00110 29650.00000
3279 -9834.00010 395496.00000
3283 31123.99999 47738.00000
3284 172901.00003 173419.00000
3286 132128.99762 -698253.00000
3289 -126339.99999 -376294.00000
3291 60189.99985 639762.00000
3292 134940.00002 -52624.00000
3296 -86582.99699 105682.00000

elif@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
```

n 7 için ve ticks 500 için

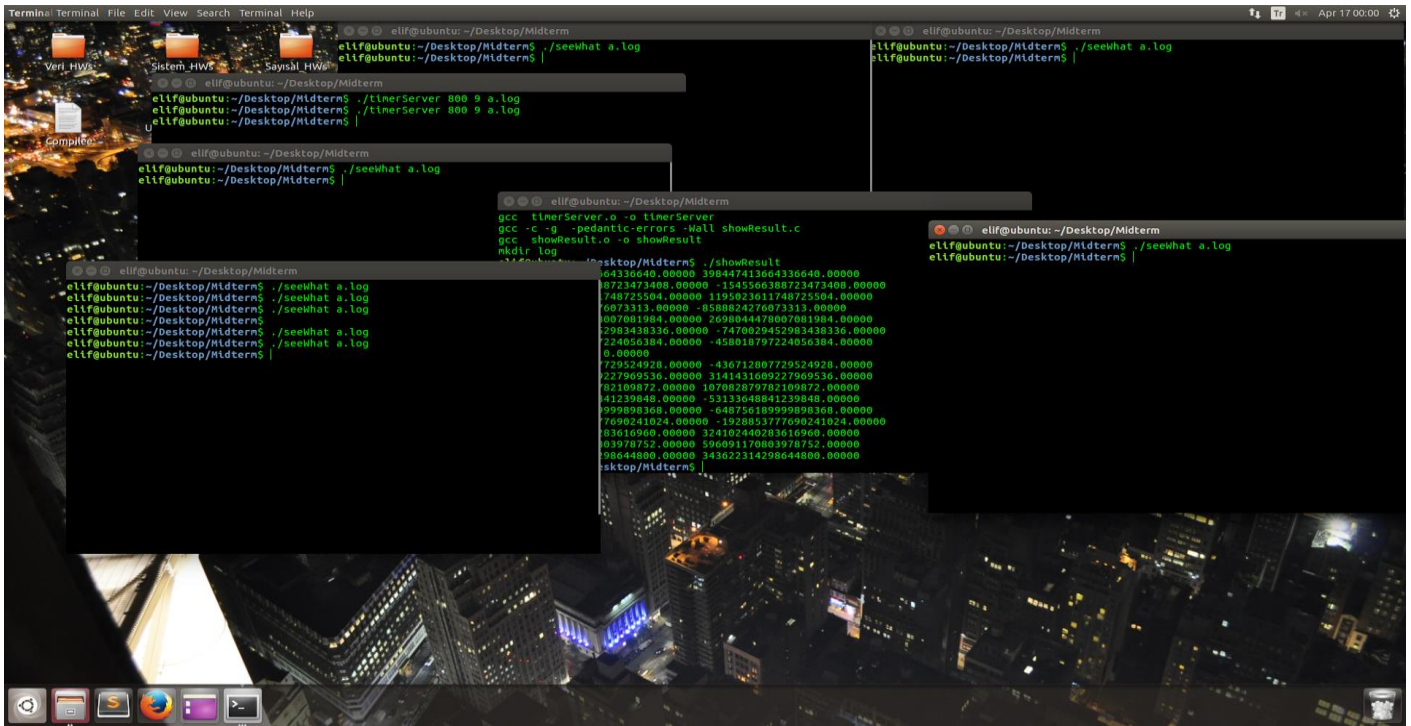


```
Terminal Terminal File Edit View Search Terminal Help
Veri HWs
Uyg
Compu...
elif@ubuntu: ~/Desktop/Midterm
elif@ubuntu:~/Desktop/Midterm$ ./timerServer 800 7 a.log
elif@ubuntu:~/Desktop/Midterm$ ./timerServer 500 6 a.log
elif@ubuntu:~/Desktop/Midterm$ |

elif@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
elif@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
elif@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
elif@ubuntu:~/Desktop/Midterm$ |

elif@ubuntu:~/Desktop/Midterm$ ./showResult
4046 44877440235.99995 0.00000
4048 -230816737739.99960 -230816737739.99960
4051 117316543605.99974 0.00000
4052 160266736901.00061 160266736901.00061
4054 190822999722.99985 0.00000
4056 169064789467.99979 169064789467.99979
4058 1072813172.00048 0.00000
4062 15833853521.99978 15833853521.99978
4064 124508831327.99989 0.00000
4065 25744711776.80012 25744711776.80012
4067 25230593862.99931 0.00000
4068 449853215962.00006 449853215962.00006
4069 -38912139465.03172 0.00000
4070 46664154248.00045 46664154248.00045
4071 82325466316.99899 0.00000
4072 165421995086.99738 165421995086.99738
^Celif@ubuntu:~/Desktop/Midterm$ |
```

n 9 için ve ticks 800 için




```
Terminal Terminal File Edit View Search Terminal Help
Veri HWs
Sistem HWs
Sayısal HWs
Compu...
elif@ubuntu: ~/Desktop/Midterm
elif@ubuntu:~/Desktop/Midterm$ ./timerServer 800 9 a.log
elif@ubuntu:~/Desktop/Midterm$ ./timerServer 800 9 a.log
elif@ubuntu:~/Desktop/Midterm$ |

elif@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
elif@ubuntu:~/Desktop/Midterm$ |

gcc timerServer.o -o timerServer
gcc -c -g -pedantic-errors -Wall showResult.c
gcc showResult.o -o showResult
mkdir log
elif@ubuntu:~/Desktop/Midterm$ ./showResult
64336640.00000 398447413664336640.00000
8723473408.00000 -1545566388723473408.00000
748725504.00000 1195023611748725504.00000
60873313.00000 -8508824276073313.00000
8087081984.00000 2698044478007081984.00000
2983438336.00000 -7470029452083438336.00000
224056384.00000 -458018797224056384.00000
0.00000
729524928.00000 -436712807729524928.00000
227909536.00000 314431609227909536.00000
82109872.00000 107082879782109872.00000
41239848.00000 -5313648841239848.00000
99980368.00000 -6487561899980368.00000
7690241024.00000 -192885377690241024.00000
83616960.00000 324102440283616960.00000
83978752.00000 596891170883978752.00000
9864800.00000 34362231429864800.00000
sktop/Midterm$ |
```


n 10 için ve ticks 500 için



```
Termin: Terminal File Edit View Search: Terminal Help
elf@ubuntu:~/Desktop/Midterm$ ./timerServer 500 3 a.log
[1]+ Stopped ./timerServer 500 3 a.log
elf@ubuntu:~/Desktop/Midterm$ ./showResult
Unit test - pass: 94080, 94016, 718573, 00000
elf@ubuntu:~/Desktop/Midterm$ ./timerServer 500 3 a.log
[[B]] [A*Z
[2]+ Stopped ./timerServer 500 3 a.log
elf@ubuntu:~/Desktop/Midterm$ ./timerServer 500 3 a.log
elf@ubuntu:~/Desktop/Midterm$ ./timerServer 500 3 a.log
elf@ubuntu:~/Desktop/Midterm$ ./timerServer 500 10 a.log
elf@ubuntu:~/Desktop/Midterm$

elf@ubuntu:~/Desktop/Midterm$ make
gcc -c -g -pedantic-errors -Wall seeWhat.c
gcc seeWhat.o -o seeWhat
gcc -c -g -pedantic-errors -Wall timerServer.c
gcc timerServer.o -o timerServer
gcc -c -g -pedantic-errors -Wall showResult.c
gcc showResult.o -o showResult
mkdir log
elf@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
C segmentation fault (core dumped)
elf@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
elf@ubuntu:~/Desktop/Midterm$ ./seeWhat a.log
elf@ubuntu:~/Desktop/Midterm$

elf@ubuntu:~/Desktop/Midterm$ ./showResult
1887 -158544701923369910272.00000 -158544701923369910272.00000
1888 449232979372388057800.00000 449232979372388057800.00000
1889 -55585728420215799800.00000 -55585728420215799800.00000
1891 -108725677531793293312.00000 -108725677531793293312.00000
1893 80215735279510339584.00000 80215735279510339584.00000
1895 -393888202366906513664.00000 -393888202366906513664.00000
1896 451343911457992605696.00000 451343911457992605696.00000
^Celf@ubuntu:~/Desktop/Midterm$
```

Eksik ve hatalı kısımlar

- Convolution hesabı yapılırken bazı durumlarda hatalı sonuçlar elde edilebilmektedir. Bunun nedeni convolution kodunun tamamen düzgün çalışmamasıdır.
- Convolution kodu tamamen doğru çalışmadığı için bazı durumlarda nan ya da inf değerleri gelebilmektedir.
- Oluşturduğum fifo dosyalarının hangilerinin ne zaman biteceğini bilmediğim için onları program içinde silemedim. Program her çalıştırılışında make clean komutu ile onların silinip tekrar make yaparak çalıştırılması gerekmektedir.
- 5 – 6 client tan sonrası için senkronizasyon probleminden dolayı hatalar olabiliyor.

2d Convolution

2d convolution matris ler görüntü işlemede sıklıkla kullanılır. Belirlenen kernel ve görüntü arasında gerçekleştirilir.

Kullanılan Kaynaklar

inverse için

<http://www.programming-techniques.com/2011/09/numerical-methods-inverse-of-nxn-matrix.html>

determinant için

<http://www.programming-techniques.com/2011/09/numerical-methods-determinant-of-nxn.html>

convolution için

[https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))