

Gebze Technical University

Computer Engineering

CSE 244
2017 Spring

MIDTERM REPORT

GULZADA IISAEVA
131044085

İstenilen programlar:

- TimerServer
- SeeWhat
- ShowResults

1. **TimerServer**: Bizim ana serverimiz

Çalıştırılma şekli:

`./TimerServer <ticks in miliseconds> <n> <mainpipename>`

`./TimerServer` : program ismi

`<ticks in miliseconds>` : Serverın kaç milisaniyede bir clientdan gelen requestleri kontrol edeceği değer.

`<n>` : İnteger bir değer. Server her client için $2n \times 2n$ invertible matris oluşturacak.

`<mainpipename>` : Server ile clientin haberleşeceği ana fifonun adı.

Algoritma:

- İlk çalıştırıldığı zaman arguman olarak verilen isimde bir tane mainfifo oluşturacak (client server arası request için). Sonra while(1) döngüsü içinde sürekli clienta kendi pidsini gönderecek, client dan request (pid) bekleyecek, matris sayısını n i gönderecek. Sonra process üretilecek. Parent process pid beklerken child pocess while(1) döngüsü içinde pid isimdeki oluşturulan fifoya sürekli random tersi alınabilen matrix (det!=0 ana kadar üretilecek kontrolü sağlandı) üretilip gönderecek. "serverLog.log" dosyasına matrisin oluşturulduğu zaman (milisaniye olarak), pid , determinantın yazacak.

- Log: Time Pid Determinant

...

- Eğer CTRL +C sinaylı gelirse:
 - Her pid geldiğinde onları bir arrayOfClients[1000] arrayde tutuyorum, aynı anda sizeofClients ı arttırıyorum.
 - Eğer sinyal yakalandıysa signalHandler(int signo) açılan tüm fifoları unlink ediyor, sizeofClients kadar bütün pidlere kill sinyali

gönderiliyor.En son log dosyasına hangi zamanda CTRL+C geldiğini basıp ölüyor.

2. **SeeWhat** : client. Birden fazla çalıştırabilir

Çalıştırılma şekli:

./seeWhat <mainpipename>

Algoritma:

- İlk çalıştırıldığında kendi pid isminde fifo oluşturuyor ve mainFifo üzerinden serverin pidsini bekleyecek onu ShowResults a gönderecek ve ShowResultın pidsini alacak, request olarak kendi pidsini(kendi pid sini)gönderecek. Sonra while(1) döngüsü içerisinde serverden sürekli matrix okuyacak. Fork oluşturulacak. Oluşturacağı proses n x n lik shifted inverse matrisi bulacak ve result i hesaplayacak.

Result = $\det(\text{orjinal matris}) - \det(\text{shifted inverse matris})$

Diğer proses 2d convolution matris oluşturarak ve result2 yi hesaplayacak.

Program result nin ne kadar zamanda(time elapsed) hesaplandığını

bulacak.Sonra bunu ShowResulta gönderiyor

- Eğer CTRL+C sinyali gelirse kendisini ve çocukları , server ve ShowResults pidlerini öldürecek

3. **ShowResults** : Her cliendan result ve clientın pidsini alacak ve onları dosyasına ve ekrana aşağıdaki gibi yazdıracak.

Ekrana yazılacak bilgiler

pid Result1 Result2

...

Log dosyasına yazılacak bilgiler

m1 pid of client

result1, time elapsed

..

Not:

Ekrana takip edebilmek için her seferinde pid leri , size ı, matrisi yazıyorum.

Yararlanan kaynaklar: UNIX System Programming ders kitabı

<http://www.geeksforgeeks.org/adjoint-inverse-matrix>