

## SYSTEM PROGRAMMING FINAL REPORT

Serhat Güzel 131044065

Final projesine başlarken midtermden bi şeyler öğrenerek geldim . Öncelikle client server yapısının nasıl olduğu hakkında bilgim vardı . Bu projede socket programming üzerinde durarak çalışmalarına başladım . İlk başta eğlenceli gibiydi . Socket programlama gayet zevk alarak başladığım kısımdı . Ancak gerek thread olsun gerek semaphore , mutex , shared memory bu konular hakkında yeterli deneyime sahip değildim. Onlar hakkında biraz daha bilgi sahibi olduktan sonra tasarımı yapmaya başladım . Proje anlaşılması zor bir konu üzerineydi . Açıkçası anlamak çok zamanımı aldı ve bir çok şeyi hala anlayamayarak yapmaya çalıştım . Midterme göre daha çok zorlandığımı söyleyebilirim . Öncelikle işe socket ile client server bağlantısı yaparak başladım . Clientta kullanıcıdan aldığım değerler olan matris sütun , satır , oluşması gereken client sayısı ve port numarası üzerinde karşı tarafa bilgi yollamaya çalıştım . Socketle veri göndermek için send() fonksiyonunu kullandım . Diğer taraftan serverda ise recv() fonksiyonu ile aldım . Client içinde threadler ile kullanıcıdan girilen sayı kadar mini client oluşturdum. Her biri semaphore sayesinde sırayla istek yolayacak şekilde ayarlandı. Serverda ise accept() her bir isteği kabul ederek bir thread oluşturuyor . Bu her bir thread ise paralel olarak 3 processe ayrılıyor . Bu processlerden 1. si ile A ve B matrislerini oluşturdum . Shared memory ile P2 processine paylaştım . Bu sırada processler paralel olmasına rağmen birbirlerine senkronize olacak şekilde ayarlamaya çalıştım ancak çok da randımanlı olmadı . P2 processi 3 paralel threade ayrıldı bunlar 3 farklı yöntemle  $Ax=B$  işleminde x matrisinin bulunması gerekiyordu. Matematiksel ifadeler kullanmadım . Kendim x değerleri belirledim . Bu değerler Shared Memory ile P3 processise paylaşılabilecek şekilde ayarlandı . Serverda bir tane server.log dosyası basarak A B ve X matrislerini yazdırdım . Programın multithread ve processlerinin paralel olarak çalışmasına özen göstermeye çalıştım . Server da sadece bir tane implementasyona zamanım yetti . Program clientta CTRL-C yaparsa sadece client eğer serverda CTRL-C yaparsa tüm programlar kapanacak şekilde ayarlandı . Programı 1 - 2- 3 - 5 - 10 - 20 - 100 mini client kullanılacak şekilde test ettiğimde aynı anda matrislerin oluştuğunu gözlemledim. Performans zaman kontrolü pek sağlıklı olmadı . Projede ne kadar çok zorlansam da bir çok yeni şeyi deneyimlemeyebildim.