BIL 244 Linux Sistem Programlama

Multiprocess-Based Matrix Multiplication Competition Application

Burak Orçun Özkablan
06104403

Bilgisayar Mühendisliği Bölümü
Gebze Yüksek Teknoloji Enstitüsü
Gebze, Kocaeli

Giriş

Bu projede, birden fazla client'ın bir server'a bağlanarak, server'dan gelen matrislerin çarpımını yaparak yarıştıkları bir uygulama geliştirilmiştir.

Uygulama en az 3 etaptan ve en az 3 tane yarışmacı ile çalışır. Yarışmanın galibi, en çok etap kazanan client'tir. Yarışma sonlandığı zaman server ve client'ler kendilerine has bir log dosyası hazırlar. Burada tüm etapların ve genel yarışmanın kazananı ve etap süreleri yer alır. Bu proje ile amaçlanan tema, farklı programların birbirleri ile haberleşmesidir. Bu haberleşme Processler Arası Haberleşme olarak bilinir. Her bir program bir tane process'ten oluşur ve bu process'ler server ile FIFO'lar sayesinde haberleşirler.

Sonuç olarak, uygulama farklı programların iş yapabilme süreleri arasında bir mukayese yaparak ve FIFO ile haberleşerek çalışan bir içeriğe sahiptir.

Multiprocess-Based Nedir ?

Multiprocess-based, birden fazla process tabanlı demektir. Projede birden fazla client olduğu için ve her bir program içerisinde bir tane process barındırdığı için, bu uygulama multiprocess tabanlı bir uygulamadır.

Matrix Multiplication

Server tarafından client'lara her etapta onluk bir matris havuzundan seçilen 10x10'luk ve her elemanı tek basamaklı olan iki tane matris gönderilir. Client'lar bu iki matrisi çarpma sürelerini ölçerek server'a bildirirler. Böylece her etabın birincisi server tarafından belirlenir.

Uygulamanın Çalışma Süreci

Uygulamada öncelikle server çalıştırılır. Server çalışmadan client'ların hiçbiri çalışamaz çünkü bağlanacak server bulamazlar. Server'a çalıştırmak için parametrik olarak yarışmanın etap sayısı ve yarışmacı sayısı girilir. Server parametre olarak aldığı etap sayısı kadar yarışma yapar. En çok 3 tane client, server'a bağlandığı zaman yarışma başlar. Bu 3 yarışmacı yarışa devam ederken en az bir tanesi yarışmayı terk ederse, yarışma durur ve dışarıdan yarışmaya katılabilecek başka bir yarışmacı bekler. 3 yarışmacı yarışırken ve yarışma devam ederken dışarıdan başka yarışmacılar yarışmaya katılabilir. Bir yarışma en fazla 10 etaptan ve 10 tane yarışmacıdan oluşabilir. Yarışma esnasında, client'lar veya server kendisini sonlandırabilir. Client'ın kendini sonlandırmasında sonunda server'a haber vererek kendisini yarışmadan egale etmesini sağlar. Server kendini sonlandırdığı zaman client'lara yarışmanın bittiğini bildirir.

Yarışma devam ederken her client etabı bitirme süresini server'a bildirir. Server, client'lerden gelen süreleri karşılaştırarak etabın galibini belirler. Yarışma sonunda ise en çok etap kazanan client'ı yarışmanın galibi olarak belirler. Yarışmanın süresi server'ın log dosyasında tutulur.

Fonksiyonel Olmayan Gereksinim Analizi

Uygulama kodlaması Ubuntu 10.10 adlı işletim sisteminde standart C kullanılarak yapılmıştır. Geliştirme araçları olarak gedit editörü ve gcc compiler'ı kullanılmıştır.

Fonksiyonel Gereksinim Analizi

1) Server

Server tarafında standart C kütüphaneleri ile birlikte Unix işletim sistemine has birkaç tane kütüphane daha kullanılmıştır.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <fcntl.h>
#include <errno.h>
#include <dirent.h>
#include <signal.h>
#include <setjmp.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/time.h>
```

Yukarıda server tarafında kullanılan kütüphaneler gösterilmektedir.

Server'ın oluşturduğu log dosyası ve FIFO dosyalarının yazma, okuma ve değiştirilme izinleri kullanıcıya, gruba ve diğerlerine şu şekilde verilmiştir.

```
#define REQ_PERMS (S_IRUSR | S_IWUSR | S_IWGRP | S_IWOTH)
#define RES_PERMS (S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH)
```

Server tarafından birden fazla FIFO kullanılmıştır.

```
#define ENTRY_FIFO_NAME "entry.dat"
#define QUIT_FIFO_NAME "quit.dat"
#define QUESTION_FIFO_NAME "question.dat"
#define ABORT_FIFO_NAME "abort.dat"
#define FEEDBACK_FIFO_NAME "feedback.dat"
#define SERVER_LOG_FILE "serverLog.log"
```

enrty.dat : Client'ların server'a yarışmaya katıldığını bildirdikleri FIFO'dur.
quit.dat : Server tarafından yarışmanın bittiğinin client'lara bildirildiği FIFO'dur.
question.dat : Server'ın matrisleri client'lere yolladığı FIFO'dur.
abort.dat : Yarışmayı yarıda bırakan client'lerin server'a yarışmadan çıktığını bildirdiği FIFO'dur.
feedback.dat : Matris çarpım sürelerinin client'ler tarafından server'a bildirildiği FIFO'dur.
serverLog.log : Server'ın yarışma detaylarını yazdığı dosyadır.

Server, 3 tane yarışmacı gelir gelmez yarışmayı başlatır. Her etap başında yarışmadan çıkan veya yarışmaya giren birisi var mı diye kontrol eder, değişen bir durum varsa server bu

değişikliklere göre elindeki dataları günceller. Yarışma bitince de server log dosyasına yarışma detaylarını yazar. Server'a ctrl+c ile bir kesme gelirse bunu client'lara bildirerek yarışmayı sonlandırır. Server'ın fonksiyonları aşağıdaki resimde gösterilmiştir.

```
void usage();
void help();
int quitCompetitorFromCompetition(long *, long, int);
int isDirectory(char *);
void freeMemory(char **);
int fillMatrixArray(char *, char *, char *);
long findWinner(long *, long *, int, int);
void closeMatrixFile(int *);
static void chandler(int);
```

usage : Programın kullanımı gösterir.
help : Programın içeriği hakkında kullanıcıya bilgi verir.
quitCompetitorFromCompetition : Client'tan çıkma isteği gelince client listesinden çıkmak isteyen client id'si çıkarılır, kayıtlı verileri silinir.
isDirectory : Parametre olarak aldığı char pointer'ın bir dosyaya mı yoksa başka bir şeye mi ait olduğunu kontrol eder.
freeMemory : Parametre olarak aldığı char pointer pointer'ı alanını hafızaya geri verir.
fillMatrixArray : Server'dan gelen matrisleri arraylere ayıran fonksiyondur.
findWinner : Yarışmayı kazanan process'in id'sini bulup, dönderir.
closeMatrixFile : Açılan matris dosyalarını kapatır.
chandler : ctrl+c sinyalini yakalayan fonksiyondur.

2) Client

Client'lerin header bilgileri aşağıdaki gibidir.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <sys/time.h>
#include <fcntl.h>
#include <errno.h>
#include <signal.h>
#include <setjmp.h>
```

Client kodları içerisinde herhangi bir FIFO yaratma işlemi yoktur. Client sadece server'ın yarattığı FIFO'ları kullanır. entry.dat FIFO'suna kendi process id'sini yazdırarak

yarışmaya katılma isteğini belirtir. Her etap başlangıcında yarışmanın bitip bitmediğini kontrol eder. quit.dat FIFO'sunda herhangi bir şey yoksa yarışmaya devam eder. Kendisine gelebilecek bir ctrl+c sinyali yakalayarak server'a abort.dat FIFO'sundan yarışmadan çıkma isteğini belirtir. Server, yarışmadan çıkan yarışmacıyı diskalifiye eder, her bu yarışmacıya ait bilgileri dataları arasından siler. Client yarışmayı question.dat FIFO'sundan kendisine gelen matrisleri çarparak ve çarpma süresini ölçerek yapar. Yarışma bitiminde server ile birlikte kendisini sonlandırır. Client fonksiyonları aşağıda verilmiştir.

```
void usage();
void help();
void multiplicateTwoMatrix(char *, char *, FILE *);
char *pidToFileName(long, char *);
static void chandler(int);
```

usage : Programın kullanımı gösterir.
help : Programın içeriği hakkında kullanıcıya bilgi verir.
multiplicationTwoMatrix : Server'dan gelen 2 matrisi çarparak, çarpma süresini ölçer.
pidToFileName : Her process'in kendisine has bir dosyası olacağından, dosya isimleri process id'leri olarak atanır. Bu fonksiyon process id'sini dosya adı için char pointer'ına çevirir.
chandler : ctrl+c sinyalini yakalayan metottur.

Sonuç

Bu projede amaçlanan hedeflere tamamen ulaşılmıştır. Proje sonucunda FIFO haberleşmesi, sinyal yakalama, mikro saniye değerinde süre ölçme, bloklanmış ve bloklanmamış dosya işlemleri başarı ile uygulanmıştır. Uygulama tüm Unix sistemlerinde sorunsuz bir şekilde çalışmaktadır.