CSE344 – Final Projesi Raporu

Abdulhakim Emre ARTIŞ - 141044034

Server.c:

struct clientInfo -> İsim, öncelik, ödev, ödevin yapılış süresi bilgilerini tutar.

struct providerInfo->İsim,sıradaki client bilgileri, client sayısı, providerın aktif olup olmadığı , performans, ücret, online olma süresi , ücret, toplam online olma süresi, yapılan ödev sayısı bilgilerini tutar.

-Provider bilgilerini belirtilen dosyadan okuyarak kişi sayısı kadar yer alır ve bilgilerini doldurur.

-Her provider için thread oluşturulur.

-Bağlantı için socket ve her bağantı için ayrı thread oluşturulur.

-Her bağlantı sağlandığında önceliğine göre en iyi eşleşme yapılan provider'ın(mutex lock yapılarak) ödev sırasına eklenir, provider dan gelen cevap client'a iletilir.

-terminalden SIGINT sinyali gelene kadar devam eder, sinyalin gelmesiyle provider bilgileri print edilir ve ayrılan yerler free edilir.

PROVIDER:

Ödev sırası boşsa gelene kadar bekler . Geldiğinde işlemi yapıp ödev sırasını düzenler . Ardından ödev sonucu bağlantı sağlayan thread aracılığıyla client a gönderilir.

Client.c:

Parametlere göre ödev isteği socket aracılığıyla servera iletilir ve cevap gelene kadar beklenir. Giden ve gelen bilgiler terminale print edilir .

CSE344 – System Programming – Final Projesi PDF Özeti

SERVER

- [x] Admits as command line argument:
  - the connection port for listening to clients
  - file of providers
  - and the log file
- [x] Server reads providers: Provider -> "name , price , quality ,time"
- [x] Server creates a pool of provider threads. THREADLERİN SONLANMASI???
- [x] Every new connection MUST be handled as a new thread.
- [x] This new thread will post the job to provider(if available) and wait for it to finish.
- [x] All server/provider messages must be logged at a log file and printed on stderr.
- [x] The task queue of every provider has size 2. If provider's queue is full, then the server must forward the task to the next best performing provider.
- [x] Once a client connects, your server will forward the homework to the corresponding provider.
- [x] If no provider is available an error message "NO PROVIDER IS AVAILABLE" will be sent to the client.
- [ ] Your server will continue to work until a termination signal is received, in which case it will first close up all client sockets by sending a polite message such as "SERVER SHUTDOWN" and terminate all provider threads.

CLIENT

- [x] Every client commandline arguments:
  - client name as a string
  - priority 'C' for low cost || 'Q' for high quality || 'T' for high speed
  - the homework as an integer denoting degree
  - server address
  - server port address

- [x] Will connect to the server send a request in the form of a simple string:
  "Hileci C 45" the client's name is Hileci, his priority is low cost and homework is cos(45)

Sample output for a client console:

- [x] -Client Hileci is requesting Q 45 from server 127.0.0.1:5555
- [x] -Hileci's task completed by Ayse in 7.53 seconds, cos(45)=0.707, cost is 900TL, total time spent 7.89 seconds.

PROVIDER

- [x] Every provider will wait for tasks in her/his queue, complete the task, sleep for a random duration between 5-15 seconds (to simulate hard work...), and return its result to the client.

- [x] Once the login time is up, the provider will logout and her/his thread will be terminated.

Rules

- [x] Your program must handle eventual errors gracefully according to the POSIX traditions.

- [x] Your program must print usage information in case of incorrect or missing command line arguments.

- [x] "You are expected to submit a report explaining in detail how your project design provides the required behavior and how you have solved any eventual problems you have encountered." Report format will be pdf.

- [x] Provide a makefile to compile your homework. Do not run your program inside makefile. Just compile

REFERANS

Socketlerle iletişimi sağlarken burdaki kodu değiştirerek kullandım :

http://www.linuxhowtos.org/C_C++/socket.htm