FATİH SULTAN MEHMET VAKIF ÜNİVERSİTESİ

Öğrencinin;

ADI: Sümeyye Zülal

SOYADI: Dik

NO: 1721221014

BÖLÜM: Bilgisayar Mühendisliği

Projenin;

KONUSU: Dosya Yöneticisi

Dersin;

ADI: Operating Systems

EĞİTMEN: Ali Yılmaz ÇAMURCU

Samet KAYA

Tuğçe GÖKSU

# İçindekiler

## 1- Proje Konusu

Projenin konusu named_pipe, multi_thread ve senkranizasyon mekanizmaları kullanarak bir dosya yöneticisi oluşturmaktır. Projede iki adet program yazılmıştır: file_manager, file_client. File_manager, ana program olup servis gibi çalışmaktadır. File_client, File_manager'a komutlar göndermektedir. File_manager aldığı bu komutları yerine getirerek file_client'a bir cevap verir. Bu işlemler File_client'tan exit komutu gelene kadar yapılır. Create komutu girilirse File_List'te dosya ismi yoksa boş olan sıraya dosya ismini ekler ve sistemde dosyayı oluşturur. Delete komutu girilirse File_List'te dosya ismi varsa sistemden ve File_List'ten(indexten) dosyayı siler. Read komutu girilirse File_List'te dosya ismi varsa sistemden ilgili indexteki lineni dosyadan okuyarak ilgili File_Client'a gönderir. Write komutu girilirse File_List'te dosya ismi varsa File_client'tan gelen datayı dosyaya yazar.

## 2- Proje Süresince Yapılanlar

file_client.c isimli bir dosya oluşturdum. Bu dosyada biri genel 6 farklı named pipe yolu tanımladım. Dosya ilk çalıştırıldığında ortak named pipe üzerinden file_manager.c dosyasının okuması için "client" komutu yazılır. Ve yine bu ortak named pipe üzerinden cevap olarak bir id alır. Aldığı id'ye göre bir while döngüsü içinde exit komutu girilene kadar id'sine özel açılan named pipe'a yazma, yazdıktan sonra da okuma işlemi yapılır. Exit komutu girilirse managerdan cevabı da aldıktan sonra döngüden çıkılır ve program sona erer.

file_manager.c isimli bir dosya oluşturdum. Bu dosyada listen_pipe isimli bir fonksiyon yazdım bu fonksiyonda 5 id bilgisi içeren char array(string) ve 4 thread tanımladım ve file_client'taki ortak named pipe adresini verdim. While döngüsü içinde önce bir locklama yapılır ve named pipe'tan okumak için client'tan komut beklenir. Komut okunduktan sonra pipe tekrar kapatılır. Okunan komut "client" ise global olan client sayısını tutan sayaç bir arttırılır. Client numarasına göre genel named pipe bu sefer yazmak için açılır ve id'si client'a gönderilir. "client" komutu okunmadan önce döngünün başında konulan lock kaldırılır. Bir thread create edilir. Bu thread id numarasına göre yazılan 5 farklı listen pipe fonksiyonundan birini çalıştırır.

Yazılan 5 farklı listen pipe fonksiyonunun da çalışma mantığı aynıdır. Yalnızca pipe'ların yolları farklıdır. Listenpipe1 fonksiyonunda önce bir lock konulur. While döngüsü içinde id 1 e özel named pipe ile iletişim kurulur. Önce bu named pipe'tan okumak için client'tan komut beklenir. Okunan komutlar boşluğa göre ayrılarak bir string array'e (char*[]) atılır. Bu string array execute_commands isimli fonksiyona gönderilir. Bu fonksiyondan yapılan işlemin ne olduğuna, yapılıp yapılamayışına göre bir int döner. Dönen bu int degree göre client'a cevap vermek için tutulan response değişkenine ilgili cevap yazılır. Id1'e özel olan named pipe bu sefer yazmak için açılır, cevap client'a gönderilir ve pipe tekrar kapatılır. Execute_commands fonksiyonundan dönen değer -4 ise bu açılan bütün client'ların çıkış isteği yaptığı anlamına gelir. Fonksiyonun başında yapılan lock kaldırılır ve exit ile çıkış yaplır -4 değilse döngü devam eder. Fonksiyonun sonuda yine lock kaldırılır. Diğer listen pipe'larda da sadece named pipe'ın yolu değiştirilmiştir. Temel yapı aynıdır.

Execute_commands fonksiyonu parameter olarak aldığı komutları control ederek gerekli fonksiyonları çağırır. Daha sonra bu fonksiyonların işleyişine göre bir int değer döndürür. . Gelen komut "exit" ise global olan client sayısını tutan sayaç bir azaltılır client kalmamışsa fonksiyondan geri -4 döndürülür hala client varsa -3 döndürülür.

Gelen komutlar en az iki kelimeden oluşuyorsa ikinci kelime dosya ismi olacağı için dosya ismini almak üzere getFileName fonksiyonuna gönderilir. Komutlar '\n' ile beraber geldikleri için dosya isimlerinin yalın hallerini elde etmek amacıyla yazılan bu fonksiyonda dosya isminin son elemanı '\n' ise '\0' ile değiştirilir.

Gelen komutların tutulduğu arrayin ilk elemanı create ise create_file fonksiyonu çağırılır. Create_file fonksiyonunda parameter olarak alınan file name eğer fileList'te yoksa ve fileList boyutu aşılmamışsa NULL olan ilk indexe atılır. Sistemde dosyayı oluşturmak için fopen fonksiyonu ile a modunda dosya açılıp kapatılır ve geriye 1 değeri döndürülür. Eğer dosya zaten fileList'te ise geriye 0 değeri, eğer dosya sayısı 10'u aşmışsa geriye -1 değeri döndürülür.

Gelen komutların tutulduğu arrayin ilk elemanı write ise write_to_file fonksiyonu çağırılır. write_to_file fonksiyonunda parameter olarak alınan file name fileList'te ise dosya fopen ile "a+" modunda açılır ve komutlardan alınan veri fprintf fonksiyonu ile dosyaya yazılır, dosya kapatılır ve geriye 1 değeri döndürülür. File fileList'te yoksa -1 değeri döndürülür.

Gelen komutların tutulduğu arrayin ilk elemanı read ise read_file fonksiyonu çağırılır. read_file fonksiyonunda parameter olarak alınan file name fileList'te ise dosya fopen ile "r" modunda açılır ve EOF olana kadar yani dosyanın sonuna kadar karakter karaketer dosya okunarak print file ekrana basılır ve dosya kapatılıp geriye 1 değeri döndürülür. File fileList'te yoksa -1 değeri döndürülür.

Gelen komutların tutulduğu arrayin ilk elemanı delete ise delete_file fonksiyonu çağırılır. delete _file fonksiyonunda parameter olarak alınan file name fileList'te ise ilgili index NULL ile boşaltılır ve remove fonksiyonu ile dosya sistemden silinir. Geriye 1 değeri döndürülür File fileList'te yoksa -1 değeri döndürülür.

Main fonksiyonda önce bütün tanımlanan global locklar initialize edilir ve bir thrad oluşturularak listen_pipe fonksiyonu çalıştırılır ve join ile bu thread beklenir. En son bütün locklar destroy edilir.

## 3- Zorlandığım Kısımlar

Başlangıçta dosya isimlerindeki \n'i dikkate almadığımda create metodunu çağırdığımda listedeki bütün elemanlar değişiyordu. Problemin neyden kaynaklandığını farketmem zamanımı aldı.

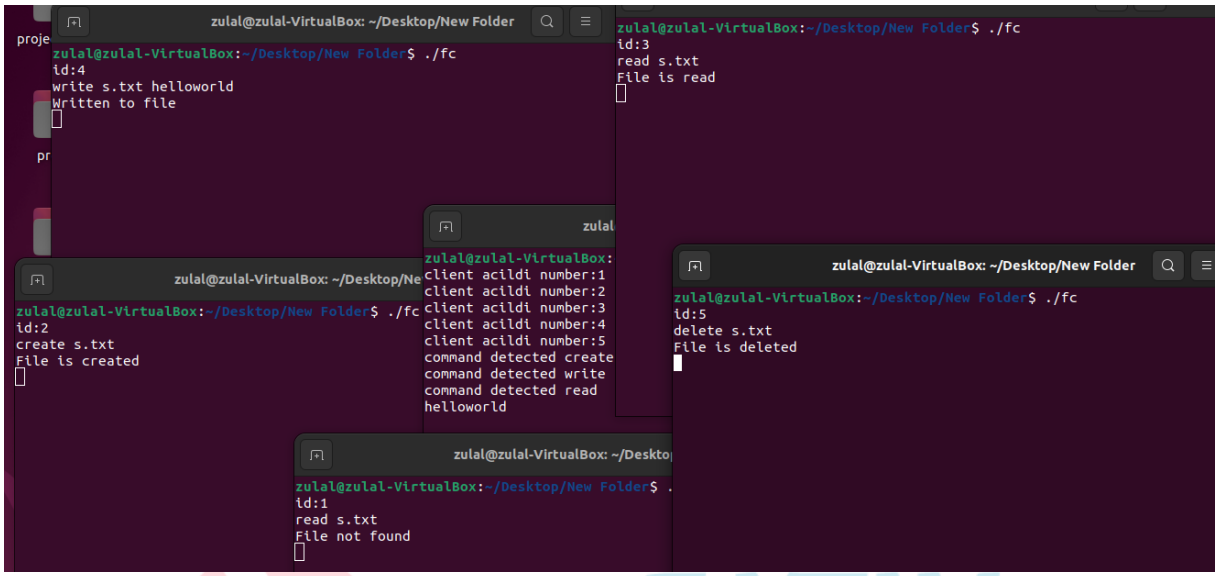
Çoklu named pipe yapısını kurmakta zorlandığım için önce tek named pipe ile yapmayı denedim. Daha sonra ortak bir named pipe ile id vererek yapma fikri aklıma gelince bütün projeyi baştan düzenledim. İlk başta kurduğum yapı şu şekildeydi (Daha sonra iki dosyayı da tamamen değiştirdim);

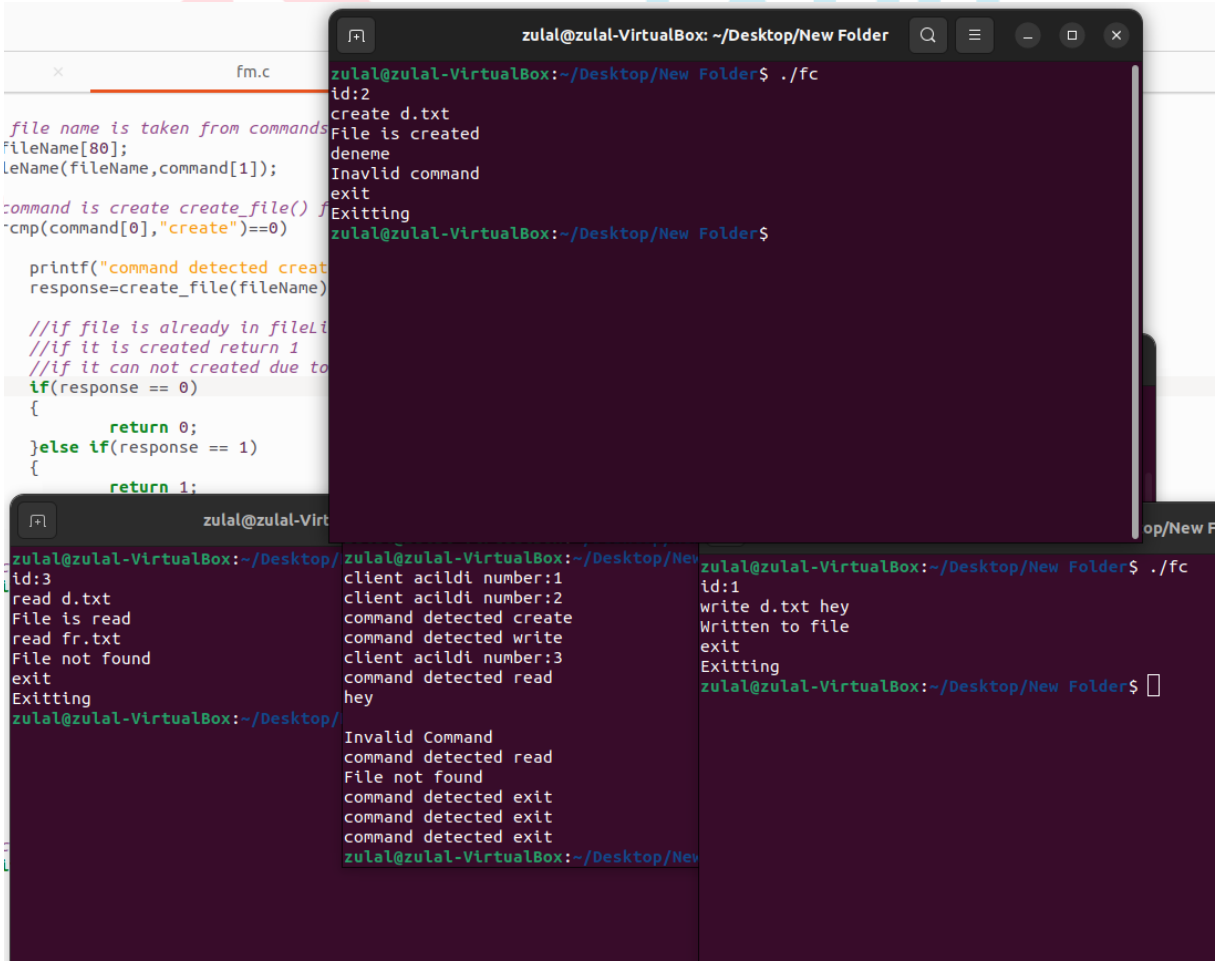
file_client.c isimli dosya her çalıştırıldığında file_manager.c dosyasının okuması için aralarında haberleşmeleri için açılan named pipe'a "client" komutu yazılır. Daha sonra bir while döngüsü içinde exit komutu girilene kadar açılan bu named pipe'a yazma, yazdıktan sonra da okuma işlemi yapılır. Exit komutu girilirse managerdan cevabı da aldıktan sonra döngüden çıkılır ve program sona erer.

file_manager.c isimli dosyada listen_pipe isimli bir fonksiyon yazdım bu fonksiyonda 4 thread tanımladım ve bir named pipe adresi verdim. While döngüsü içinde bu named pipe'tan okumak için client'tan komut beklenir. Okunan komutlar boşluğa göre ayrılarak bir string array'e (char*[]) atılır. Gelen komut "client" ise global olan client sayısını tutan sayaç bir arttırılır. komutu yazılır. Gelen komut "exit" ise global olan client sayısını tutan sayaç bir azaltılır ve client'a "exitting" response'u dönülür. Client sayacı sıfır ise exit ile program sona erer. Client sayacı sıfır olana kadar ise thread'te dinlemede kalınır ve gelen komutlara göre bir thread oluşturulur ve bu thread gerekli fonksiyonu (create_file, read_file vs.) çalıştırır. Pthread_join ile bu threadin bitmesi beklenir bittikten sonra pthread_exit ile thread sonlandırılır.

Bu fonksiyonlar global olarak tanımlanan response değişkenine gerkli cevabı atarlar. Response alındıktan sonra named pipe yazma modunda açılır ve client'a response döndürülür. Pipe tekrar kapanır. Main fonksiyonda bir thrad oluşturularak listen_pipe fonksiyonu çalıştırılır ve join ile bu thread beklenir. En son bütün locklar destroy edilir.

## 4- Proje Çıktıları ve Başarı Ölçütleri




## 5- Kaynakça

Lab çalışmalarındaki kodlar.