

# İşletim Sistemleri Proje 2 Raporu

*Hazırlayan: 1821221005 Furkan Gündoğan*

## 1. Projenin Adı: Database Yönetimi

**2. Proje Konusu:** Terminal üzerinden çalışıp pipe ve dosya okuma teknolojilerini kullanan ve veritabanı sorgusu gerçekleştirebilen bir program yazmak.

## 3. Tamamlanan Talimatlar:

- 3.1 Sürekli çalışarak kullanıcıdan sorgu girmesini bekleyen program.c'yi oluşturdum. Bu program kullanıcı sorgusunu named pipe ile daha sonra oluşturacağımız database.c'ye ulaştırmaktadır.
- 3.2 Database.c programı ilk olarak veri1.txt ve veri2.txt dosyalarından bilgiler okumaktadır. Ardından named pipe ile gelen sorgu metnini parse etmektedir.
- 3.3 Database.c'nin sorguyu "ad, numara, \*" seçenek tipleri ve "ad, number" kıyasları ile gerçeklemektedir. Daha sonra sonuç named pipe'a yazılarak ana programın sonucu kullanabilmesi sağlanmıştır.
- 3.4 Ana program kullanıcıya "dönen verileri kayıt etmek istiyor musunuz?" sorusunu yöneltmektedir. Eğer klavyeden cevap olarak "e" girildiyse ana program fork => exec ile kaydet.c programını çalıştırıyor ve isimsiz pipe ile verileri o programa iletebiliyor. "h" girildiyse program başa dönüyor.
- 3.5 Kaydet.c programı sonuc.txt'yi "a+" ile açıyor ve verileri bu dosyaya yazıyor.
- 3.6 Kayıt programının sonlanmasıyla ana programın 'exit' girilene kadar sorgu istemeye devam etmektedir.

## 4. Pipe Teknolojisi

Bu projede normal pipe ve named-pipe teknolojilerini kullandık. Pipe ana ve çocuk prosesler arasında iletişim kurulabilmesini sağlıyor. Bunlar fd[0] (okuma) ve fd[1] (yazma) dosya tanımlayıcıları. Named pipe fifo olarak da bilinir bir tarafın yazdığını diğer tarafın okuyabilmesini ve cevap gönderebilmesini sağlar.

Projede named pipe kullanımını labdaki örnek programımızdaki gibi message buffer ile yaptık. İlk olarak program.c'den "sbuf.text" içerisine sorguyu yazıp database.c programına gönderdik. Database.c içerisinden "r.buf.text" ile gönderdiğimiz veriyi okuduk. Sorgu işlemlerini gerçekleştirdik ve sorgu sonuçlarını "s.buf.text" içine yazıp ana programa gönderdik. Ardından ana programdan bir pipe oluşturduk, kaydedilecek verileri pipe'a yazdık ve exec ile kaydet programını çağırdık. Bu program da verileri sonuc.txt'ye verileri kaydetti.

## 5. Zorlandığım Kısımlar:

- Sorgudaki sayı ile dosyadan okuduğum sayıyı karşılaştırırken dosyadan okurken sayının yanında iki fazla karakter olduğunu fark ettim. Bu sebeple karşılaştırma yapılamıyordu. Karakter işlemleriyle bu sorunu çözdüm.
- İsimli pipe oluşturduktan sonra verileri fd[1]'e write edip kaydet.c'den 3 ile read yapıyordum. Program ikinci sorguyu çalıştırdığında okunan değerde problem çıkmaya başlamıştı. Daha sonra write işleminde fd[1] yerine 4 yazarak bu sorunu çözdüm.
- Daha önce çift terminalli program çalıştırmadığım için bazen database.c'yi başlatmayı unutuyordum. Program.c'de hata meydana geliyordu. Bu duruma alışmak projenin başlarında bana biraz vakit kaybettirmişti.

## 6. Gerçekleştirilemeyenler ve Sınırlandırmalar:

- Bu ödevde kullanıcının gireceği sorgu sonucu eşleşme olmazsa uyarı vermekte ve program çalışmaya devam etmektedir. Ancak sorgu metninin hatalı girilmesi, eksik girilmesi ya da boş bırakılması durumlarını kontrol etmedim. Bu sebeple bazı durumlarda çekirdek hataları alabiliyoruz.
- Yukarıdaki belirttiğim kontrolleri yapmamış olmamın sebebi çalışmaktan kaçmaktan ziyade karışıklılığı azaltmak. Ödevde istenen programların içeriği çok uzun değil ve lablardan dolayı alışık olduğumuz kodlar. Elimizdeki parametre sayısı fazla ve bunlara string kontrolleri yapmak kod sayısını ve karmaşıklığını epey arttıracaktı.

## 7. Yardım Aldığım Kaynaklar

- Laboratuvar derslerindeki örnek programlar.