



SAKARYA
ÜNİVERSİTESİ

SimpleFS

1. Proje / Tasarım

2024-2025

Younes Rahebi | B221210588

SİSTEM PROGRAMLAMA

Dr.Öğr.Üyesi ABDULLAH SEVİN

BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Özet:

Proje, bir dosya sisteminin temel işlevlerini simüle eden bir C++ uygulamasıdır. Kullanıcılar, menü tabanlı bir arayüz üzerinden dosya oluşturma, silme, veri yazma, okuma ve diğer işlemleri gerçekleştirebilir. Simülasyon tabanlı bu sistem, eğitim amaçlı tasarlanmıştır ve gerçek dünya uygulamaları için ek geliştirmeler gerektirir.

1. Proje Bileşenlerinin Açıklanması

Bu proje, bir dosya sisteminin temel işlevlerini simüle eden bir C++ uygulamasıdır ve üç ana bileşenden oluşur: fs.h, fs.cpp ve main.cpp. Aşağıda her bir bileşen detaylı bir şekilde açıklanmıştır.

1.1. fs.h (Başlık Dosyası)

Bu dosya, dosya sistemi için gerekli sabitleri, veri yapılarını ve fonksiyon prototiplerini tanımlar:

- **Sabitler:**
 - METADATA_SIZE: 65536 bayt (meta veri alanı).
 - DISK_SIZE: 10 MB (toplam disk boyutu).
 - BLOCK_SIZE: 512 bayt (veri bloğu boyutu).
 - MAX_FILES: 100 (maksimum dosya sayısı).
- **Veri Yapısı:**
 - FileMetadata yapısı, her dosya için şu bilgileri saklar:
 - valid: Dosyanın geçerli olup olmadığını belirtir (0: boş, 1: dolu).
 - name: 100 karakterlik dosya adı.
 - size: Dosya boyutu (bayt cinsinden).
 - start: Dosyanın veri başlangıç konumu.
 - creationTime: Dosyanın oluşturulma zamanı.
- **Fonksiyon Prototipleri:** 19 farklı işlem için fonksiyonlar tanımlanmıştır, örneğin:
 - fs_create: Yeni dosya oluşturur.
 - fs_delete: Dosya siler.
 - fs_read: Dosyadan veri okur.
 - fs_write: Dosyaya veri yazar.
 - fs_ls: Dosyaları listeler.
 - fs_format: Diski biçimlendirir.

1.2. fs.cpp (Uygulama Dosyası)

Bu dosya, fs.h'de tanımlanan fonksiyonların uygulamalarını içerir:

- **Dosya Sistemi Yapısı:**
 - disk.sim adlı bir simülasyon disk dosyası kullanılır.
 - Meta veriler, dosyanın başında saklanır ve dosya sayısı ile FileMetadata dizisini içerir.
 - Dosya verileri, meta veri alanından sonra (METADATA_SIZE sonrası) saklanır.
- **Yardımcı Fonksiyonlar:**
 - load_metadata: Meta verileri diskten belleğe yükler.
 - write_metadata: Güncellenmiş meta verileri diske yazar.
 - get_new_file_start: Yeni dosya için başlangıç konumu hesaplar.
 - find_file_index: Dosya adıyla meta veri dizisinde dosya indeksini bulur.
- **Temel İşlemler:**
 - **fs_create:** Boş bir meta veri yuvası bulur, dosya için yeni bir başlangıç konumu hesaplar, meta veriyi günceller ve işlemi fs.log'a kaydeder.
 - **fs_delete:** Dosyanın meta verisini geçersiz olarak işaretler ve dosya sayısını azaltır.
 - **fs_read:** Dosyanın belirtilen konumundan veri okur, dosya boyutunu kontrol eder ve veriyi bir tampona yazar.
 - **fs_write:** Veriyi yeni bir konuma yazar, meta veriyi günceller ve işlemi loglar.
- **Loglama:** Tüm işlemler, zaman damgalı olarak fs.log dosyasına kaydedilir.

1.3. main.cpp (Kullanıcı Arayüzü)

Bu dosya, kullanıcıların dosya sistemiyle etkileşime geçmesini sağlayan bir menü tabanlı program içerir:

- **Menü Yapısı**
- **Kullanıcı Etkileşimi:** Kullanıcı, bir seçenek numarası girer ve gerekli girdileri (dosya adı, veri, konum, boyut vb.) sağlar. Program, ilgili fonksiyonu çağırır ve sonucu ekrana yazdırır.
- **Hata Yönetimi:** İşlem başarısız olduğunda hata mesajları gösterilir (örneğin, "Dosya bulunamadı").

2. Projenin Çalışması ve Ekran Çıktıları

Program, kullanıcıya bir menü sunar ve seçilen işleme göre gerekli girdileri alarak işlemleri gerçekleştirir.

```
Ubuntu x + v
younes@Younes:~/SP$ make
g++ -Wall -g -Iinclude -c src/fs.cpp -o lib/fs.o
g++ -Wall -g -Iinclude -c src/main.cpp -o lib/main.o
g++ -Wall -g -Iinclude -o simplefs lib/fs.o lib/main.o
younes@Younes:~/SP$ ./simplefs

--- SimpleFS Menu ---
1. Disk formatla (fs_format)
2. Dosya olustur (fs_create)
3. Dosya sil (fs_delete)
4. Dosyaya veri yaz (fs_write)
5. Dosyadan veri oku (fs_read)
6. Dosyalari listele (fs_ls)
7. Dosya yeniden adlandir (fs_rename)
8. Dosyanin varligini kontrol et (fs_exists)
9. Dosya boyutunu ogren (fs_size)
10. Dosyaya veri ekle (fs_append)
11. Dosya icerigini kisalt (fs_truncate)
12. Dosya kopyala (fs_copy)
13. Dosya tasi (fs_mv)
14. Disk defragmente et (fs_defragment)
15. Integrity kontrolu (fs_check_integrity)
16. Disk yedegi al (fs_backup)
17. Disk yedegini geri yukle (fs_restore)
18. Dosyayi görüntüle (fs_cat)
19. Dosyalari karsilastir (fs_diff)
20. Cikis
Seciminiz: █
```

3. Sonuç

Bu proje, dosya sistemi kavramlarını öğrenmek ve uygulamak için etkili bir eğitim aracıdır. Kullanıcı dostu arayüzü ve kapsamlı işlevselliği, temel dosya sistemi işlemlerini anlamak için iyi bir başlangıç noktası sunar. Ancak, gerçek dünya uygulamaları için daha fazla özellik, güvenlik ve performans iyileştirmeleri gereklidir. Proje, dosya sistemi simülasyonu olarak eğitim amaçlı kullanımlar için uygundur ve geliştirilmeye açıktır.