



**T.C. MARMARA ÜNİVERSİTESİ TEKNOLOJİ
FAKÜLTESİ**

2025-2026 EĞİTİM ÖĞRETİM YILI

**GÜZ DÖNEMİ BİLGİSAYAR MÜHENDİSLİĞİ
BÖLÜMÜ İŞLETİM SİSTEMLERİ DERSİ PROJE
DOKÜMANI**

171423008 – Hasan Dural

170423039 – Emine Umay Kılıç

171423005 – Doğukan Yılmaz

PROJE RAPORU: KULLANICI BAZLI KOTA VE DOSYA SİSTEMİ SİMÜLASYONU

Ders: İşletim Sistemleri Öğrenci Adı Soyadı: [Adını Buraya Yaz] Öğrenci Numarası: [Numaranı Buraya Yaz] Teslim Tarihi: [Tarih]

1. GİRİŞ VE PROJENİN AMACI

Modern işletim sistemlerinin en temel görevlerinden biri kaynak yönetimi ve güvenliğidir. Çok kullanıcı sistemlerinde, her kullanıcının birbirinin verisine erişememesi (izolasyon) ve disk kaynaklarını adil paylaşması (kota yönetimi) hayati önem taşır.

Bu projenin amacı; **dosya sistemi yönetimi**, **kullanıcı yetkilendirmesi** ve **disk kotası takibi** mekanizmalarını simüle eden, istemci-sunucu mimarisine sahip bir sistem geliştirmektir. Proje, kullanıcıların sanal bir terminal üzerinden dosya oluşturma, okuma, yazma ve silme işlemlerini gerçekleştirmesine olanak tanıırken, arka planda bu işlemleri fiziksel disk üzerinde yönetir ve kısıtlar.

2. KULLANILAN TEKNOLOJİLER VE YÖNTEMLER

Proje, **Model-View-Controller (MVC)** tasarım desenine benzer modüler bir yapıda geliştirilmiştir.

- Programlama Dili (Backend):** Python 3
- Web Çatısı:** Flask (RESTful API yapısı için)
- Arayüz (Frontend):** HTML5, CSS3 ve JavaScript (Fetch API ile asenkron haberleşme)
- Veri Depolama:** JSON (Kullanıcı meta verileri için) ve Fiziksel Dosya Sistemi
- Karakter Kodlaması:** UTF-8 (Türkçe karakter desteği için tüm I/O işlemlerinde zorunlu tutulmuştur)

3. SİSTEM MİMARİSİ VE MODÜLLER

Sistem, sorumlulukların ayrılması ilkesine (Separation of Concerns) göre üç ana backend modülünden oluşmaktadır:

3.1. QuotaManager.py (Veri ve Kota Katmanı)

Sistemin veritabanı yönetimini üstlenir.

- **Kalıcılık (Persistence):** Kullanıcı bilgileri, şifreler ve kota limitleri `users.json` dosyasında tutulur. Sunucu kapansa bile veriler kaybolmaz.
- **Senkronizasyon:** Sistem her başladığında (`load_and_sync_data`), JSON verisi ile diskteki fiziksel klasörleri karşılaştırır. Eğer bir kullanıcının klasörü manuel olarak silinmişse, tutarlılığı sağlamak adına o kullanıcıyı sistemden otomatik olarak temizler.
- **Admin Yönetimi:** Sistem açılışında varsayılan `admin` kullanıcısının varlığını ve yüksek kota yetkisini garanti altına alır.

3.2. FileSystem.py (Mantıksal İşlem Katmanı)

İşletim sistemi çekirdeğinin dosya yönetim birimini simüle eder.

- **Fiziksel-Mantıksal Eşleşme:** Kullanıcılar `/home/user1/dosya.txt` gibi sanal yollar görürken, sistem bunu arka planda proje klasöründeki gerçek yollara (`backend/user1_home/dosya.txt`) yönlendirir.
- **İzin Kontrolü (RWX Simülasyonu):**
 - **Read (Okuma):** `ls` (listeleme) ve `cat` (içerik okuma) komutları ile sağlanır.
 - **Write (Yazma):** `create` (oluşturma), `write` (append/ekleme) ve `delete` (silme) komutları ile sağlanır.
 - **Execute (Çalıştırma):** Güvenlik simülasyonu gereği, `run` komutu engellenmiş ve kullanıcıya yetki hatası döndürülmüştür.
- **Otomatik Onarım:** Dosya oluşturma sırasında, kullanıcının ana dizini kazara silinmişse sistem bunu fark eder ve işlemi iptal etmeden önce dizini otomatik olarak yeniden oluşturur.

3.3. app.py (API ve Sunum Katmanı)

Frontend'den gelen HTTP isteklerini karşılar, kimlik doğrulama (session check) yapar ve ilgili modüllere yönlendirir. Hata yönetimi burada yapılarak istemciye anlamlı JSON cevapları dönülmesi sağlanır.

4. GERÇEKLEŞTİRİLEN İŞLETİM SİSTEMİ KAVRAMLARI

4.1. Kullanıcı İzolasyonu ve Güvenlik

Her kullanıcı sisteme `register` komutu ile kayıt olur ve `login` ile oturum açar. Sistemde, bir kullanıcı (admin değilse) asla başka bir kullanıcının dosyasına erişemez, okuyamaz veya silemez. Bu, Linux/Unix sistemlerindeki dosya sahipliği (ownership) ilkesinin simülasyonudur.

4.2. Kota Yönetimi (Quota Enforcement)

Her yeni kullanıcıya varsayılan olarak **100 MB** kota atanır.

- **Dosya Oluřturma:** Kullanıcı `create 50 dosya.txt` dediğinde, sistem önce kullanıcının mevcut kullanımını ve limitini kontrol eder. Eğer `Mevcut + 50MB > Limit` ise işlem **"Disk Kotası Ařıldı"** hatasıyla reddedilir.
- **Dosya Silme:** Dosya silindiğinde, dosyanın boyutu kullanıcının kullanılan kotasından düşölür.

4.3. Yönetici (Superuser) Yetkileri

`admin` kullanıcısı, sistemdeki "root" kullanıcısı gibi davranır:

- Tüm kullanıcıları listeleyebilir (`list_users`).
- İsteddiği kullanıcıyı silebilir (`delete_user`).
- Kullanıcıların kotalarını değiřtirebilir (`set_quota`).
- Ancak veri gizliliğı simölasyonu gereğı, admin bile olsa kullanıcıların özel dosyalarının içeriğini okuması kısıtlanmıřtır.

5. KOMUT SETİ VE KULLANIM SENARYOLARI

Geliřtirilen web terminali üzerinden ařağıdaki komutlar işlenebilmektedir:

Komut	Açıklama	Gerekli İzin
<code>ls</code>	Dizin içeriğini listeler.	Read
<code>create <MB> <yol></code>	Belirtilen boyutta sanal dosya oluşturur.	Write
<code>write <yol> <metin></code>	Dosyanın sonuna metin ekler (I/O işlemi).	Write
<code>cat <yol></code>	Dosyanın içeriğini terminale basar.	Read
<code>delete <yol></code>	Dosyayı siler ve kotayı günceller.	Write
<code>run <yol></code>	Çalıştırma iznini test eder (Engellidir).	Execute
<code>status</code>	Kullanıcının kota doluluk oranını gösterir.	Read
<code>set_quota <id> <MB></code>	(Admin) Kullanıcı kotasını günceller.	Root

Kodlu Dosya Sistemi (Web Termin: X)

127.0.0.1:5000

54 kullanıcıları içeri aktar

https://www.beitc...

ChatGPT

Schabet et

Diger ok kullanilarlar

Sanal İşletim Sistemi Modülü

```
logout          : Çıkış yap.
create          : Dosya oluştur (Yazma izni).
write           : Dosyaya metin ekle (Yazma izni).
cat             : Dosya oku (Okuma izni).
run             : Dosya çalıştır (Çalıştırma izni - Engelli).
ls              : Listele (Okuma izni).
delete          : Sil (Yazma izni).
Admin Komutları (login admin admin):
list_users      : Kullanıcıları listele.
delete_user     : Kullanıcıyı sil.
set_quota       : Kota güncelle.

CMD> login user1 123
[user1] Başarıyla giriş yapıldı.
CMD> ls
[user1] Dizin içeriği (10.00 MB Kullanım):
-> a.txt (0.00 MB)
CMD> status
[user1] Kota Durumu: 10.00 MB / 200.00 MB
CMD> write /home/user1/a.txt Hello World
BAŞARILI: '/home/user1/a.txt' dosyasına metin eklendi.
CMD> cat /home/user1/a.txt
Bu dosya 10.0 MB (simülasyon).
Hello World

CMD>
```

Kodlu Dosya Sistemi (Web Termin: X)

127.0.0.1:5000

54 kullanıcıları içeri aktar

https://www.beitc...

ChatGPT

Schabet et

Diger ok kullanilarlar

Sanal İşletim Sistemi Modülü

```
-> Kullanıcı ID: admin, Kota: 100000.00 MB, Kullanım: 0.00 MB
CMD> logout
[admin] Oturum kapatıldı.
CMD> register user1 123
BAŞARILI: Kullanıcı 'user1' (100 MB) kaydedildi.
CMD> login user1 123
[user1] Başarıyla giriş yapıldı.
CMD> ls
[user1] Dizin boş: /home/user1/
CMD> status
[user1] Kota Durumu: 0.00 MB / 100.00 MB
CMD> create 1 /home/user1/Hello.txt
BAŞARILI: '/home/user1/Hello.txt' oluşturuldu. -> Kullanım Güncellendi: 0.00 MB -> 1.00 MB
CMD> write /home/user1/Hello.txt Hello World
BAŞARILI: '/home/user1/Hello.txt' dosyasına metin eklendi.
CMD> cat /home/user1/Hello.txt
Bu dosya 1.0 MB (simülasyon).
Hello World
CMD> ls
[user1] Dizin içeriği (1.00 MB Kullanım):
-> Hello.txt (1.00 MB)
CMD> status
[user1] Kota Durumu: 1.00 MB / 100.00 MB

CMD>
```