

# UNIX Shell Uygulaması

**Proje Grubu: Sistem Mimarları (#21)**

**Grup Üyeleri:**

1. **B221210588 Younes Rahebi** - 1-C
2. **B221210090 Umut Direk** - 1-C
3. **B221210577 Nurcan Ahmet** - 1-A
4. **B221210070 Şevval Nur Kalaycı** - 1-A
5. **B221210055 Şeyma Orhan** - 1-A

**GitHub Bağlantısı:** <https://github.com/Nurcanahmet/SistemMimarlari.git>

## 1. Proje Amacı

Bu proje, bir UNIX tabanlı sistem üzerinde çalışan, temel shell komutlarını destekleyen ve çeşitli fonksiyonlar sağlayan bir komut satırı uygulaması geliştirmeyi amaçlamaktadır. Shell, kullanıcıların komutları çalıştırabilmesi, yönlendirme işlemleri yapabilmesi ve arka plan işlemleri yönetebilmesi gibi temel yeteneklere sahiptir.

## 2. Projenin Özellikleri

**Shell Uygulamasının Temel Özellikleri:**

### 1. Komut Yürütme:

- Basit komutları çalıştırır (örn: ls, pwd, echo).
- Arka planda çalışan işlemleri destekler (örn: command &).

### 2. Dosya Yönlendirme:

- Giriş yönlendirme (örn: command < input.txt).
- Çıkış yönlendirme (örn: command > output.txt).

### 3. Pipe İşlemleri:

- Komutlar arasında pipe kullanarak veri aktarımı (örn: command1 | command2).

### 4. Birden Fazla Komut Çalıştırma:

- ; karakteri ile birden fazla komutu sıralı şekilde çalıştırma (örn: command1; command2).

### 5. İnkrement Fonksiyonu:

- Verilen bir dosyadaki sayıyı veya standart girdiden alınan bir sayıyı 1 artırır.

### 6. Arka Plan İşlemleri Yönetimi:

- Shell üzerinden arka planda başlatılan işlemleri takip eder ve işlem durumlarını günceller.

### 7. Signal Yönetimi:

- SIGCHLD sinyalini ele alarak arka plan işlemlerinin durumunu takip eder ve kullanıcıyı bilgilendirir.

### 3. Kod Yapısı

#### Ana Modüller ve Fonksiyonlar:

- **Yönlendirme İşlemleri:**
  - `giris_yonlendir`: < operatörü ile giriş dosyası yönlendirme.
  - `cikis_yonlendir`: > operatörü ile çıkış dosyası yönlendirme.
- **Pipe İşlemleri:**
  - `command`: Pipe işlemi için tek bir komut çalıştırır.
  - `with_pipe_execute`: Pipe içeren birden fazla komutu yürütür.
- **Arka Plan İşlemleri Yönetimi:**
  - `add_background_process`: Yeni bir arka plan işlemi ekler.
  - `wait_for_all_bg_processes`: Shell kapanmadan önce tüm arka plan işlemlerini bekler.
- **Signal Yönetimi:**
  - `sigchld_handler`: SIGCHLD sinyallerini ele alarak tamamlanan işlemler hakkında bilgi verir.

### 4. Çalışma Ortamı ve Geliştirme Araçları

- **Geliştirme Ortamı**: Linux tabanlı işletim sistemi.
- **Dil**: C Programlama Dili.
- **Kütüphaneler**:
  - `stdio.h`, `stdlib.h`, `string.h`
  - `unistd.h`, `sys/wait.h`, `fcntl.h`, `signal.h`

### 5. Sonuç ve Değerlendirme

Bu proje, UNIX sistemlerinin temel özelliklerini anlamak ve bir shell uygulaması geliştirmek için kullanıcılara önemli deneyimler kazandırmıştır. Projede gerçekleştirilen hedefler başarıyla tamamlanmış, sistem çağrıları ve process yönetimi gibi temel kavramlar etkin bir şekilde uygulanmıştır.