



İSTANBUL NİŞANTAŞI ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ
EBLG341.1 – İŞLETİM SİSTEMLERİ
ÖDEV 1: İŞLEMCİ ZAMANLAMA SİMÜLASYONU

ÖĞRETİM GÖREVLİSİ:	Dr. Öğr. Üyesi Ulaş VURAL
DERS KODU/ADI:	EBLG341.1 / İŞLETİM SİSTEMLERİ
BELGE TÜRÜ:	KULLANICI KILAVUZU
HAZIRLAYAN:	
Adı Soyadı:	CEMAL EREN ALDIBAŞ
Öğrenci Numarası:	20232013071

Teslim Tarihi: 14 Aralık 2025

CPU ZAMANLAMA SİMÜLASYONU KULLANICI KİLAVUZU

1. Proje Tanıtımı ve Genel Bakış

1.1. Simülatörün Amacı

Bu uygulama, verilen süreç setleri için 6 farklı CPU zamanlama algoritmasının (FCFS, SJF P/NP, Priority P/NP, Round Robin) performans metriklerini (WT, TAT, Throughput, CPU Verimliliği, Bağlam Değiştirme) eş zamanlı (multi-threaded) olarak hesaplar ve raporlar.

1.2. Uygulanan Algoritmalar

- FCFS (First Come First Served)**
- Non-Preemptive SJF (Shortest Job First)**
- Preemptive SJF (SRTF)**
- Non-Preemptive Priority Scheduling**
- Preemptive Priority Scheduling**
- Round Robin**

1.3. Sabit Sistem Parametreleri

Parametre Adı	Değer	Açıklama
Bağlam Değiştirme Süresi	0.001 birim zaman	Algoritmalar arası geçişte eklenen maliyet.
Round Robin Quantum (Q)	4.0 birim zaman	Bir dilimde çalışılacak maksimum süre.
Öncelik Sırası	high (1) > normal (2) > low (3)	Düşük sayı, yüksek öncelik anlamına gelir.

2. Kurulum ve Çalıştırma

2.1. Sistem Gereksinimleri

Proje, C++11 standardını ve üstünü destekleyen herhangi bir derleyici (GCC, MinGW) gerektirir.

Geliştirme Ortamı Notu: Kodun geliştirme süreci ve ilk testleri Dev-C++ IDE'si kullanılarak yapılmıştır. Programın doğru derlenebilmesi için C++11 veya C++14 standartı seçeneğinin aktif olduğundan emin olunmalıdır.

2.2. Eş Zamanlı Çalıştırma

Kod, std::thread kütüphanesini kullanarak algoritmaları paralel çalıştırmak üzere tasarlanmıştır.

Derleme Notu: Bu yapı, derleme sırasında ek bir komut (-pthread) gerektirebilir: g++ main.cpp -o simulator -std=c++11 -pthread.

2.3. Çalıştırma Adımları

- 1. Derleme sonrasında oluşan çalıştırılabilir dosyayı (simulator.exe) başlatın.**
 - 2. Program, gömülü olan Durum 1 ve Durum 2 için 6 algoritmayı eş zamanlı olarak çalıştıracaktır.**
 - 3. Tüm simülasyonlar bittiğinde konsolda onay mesajı görünecektir.**
-

3. Veri Girişi ve Esneklik

3.1. CSV Dosyası Formatı

Simülatörün doğru çalışması için herhangi bir CSV dosyası aşağıdaki zorunlu sütun yapısına sahip olmalıdır:

Sütun Adı	Açıklama	Örnek Veri
Process_ID	Süreç kimliği	P045
Arrival_Time	Sürecin sisteme varış zamanı	120
CPU_Burst_Time	Sürecin işlem süresi	18
Priority	Öncelik seviyesi	high / normal / low

3.2. Harici Dosya Kullanımı

Proje, harici bir dosyayı parametre olarak alabilme koşulunu karşılar. Yeni bir CSV dosyası kullanmak için main.cpp dosyasında aşağıdaki değişiklik yapılmalıdır:

- 1. Yeni dosyanızı projenin yanına koyun (Örn: test_verisi.csv).**
- 2. main fonksiyonu içindeki veri yükleme satırlarını aşağıdaki gibi düzenleyin:**

C++

```
// Gömülü veriler yorum satırı yapılır. [cite: 32]
```

```
// datasets["Durum 1"] = load_processes_from_string(DATA_CASE_1);
```

```
// Yeni dosya, adı belirtilerek yüklenir. [cite: 34]
```

```
datasets["Yeni_Simülasyon"] = load_processes_from_file("test_verisi.csv");
```

4. Çıktı Raporu Formatı ve Analiz

4.1. Çıktı Dosyaları

Her bir simülasyon (Durum + Algoritma) için ayrı bir .txt dosyası oluşturulur.

Adlandırma standarı: Durum 1_FCFS_results.txt.

4.2. Zaman Tablosu (Gantt Çizelgesi)

Zaman Tablosu (a Maddesi) çıktısı, ondalık hassasiyeti koruyarak virgülden sonra üç basamak gösterir. Tüm çalışma ve boşta kalma (IDLE) segmentleri eksiksiz olarak raporlanır.

Örnek Çıktı Formatı:

[0.000] - - P001 - - [1.000] [cite: 44]

[1.000] - - P002 - - [3.001] [cite: 45] (Bağlam değiştirme süresi bu hassasiyet sayesinde görünür.)

4.3. Metrikler (b, c, d, e, f Maddeleri)

Tüm raporlar, Maksimum ve Ortalama Bekleme Süresi (WT), Tamamlanma Süresi (TAT), İş Tamamlama Sayısı (Throughput), CPU Verimliliği ve Toplam Bağlam Değiştirme Sayısını içerir.