SAKARYA ÜNİVERSİTESİ

PROGRAMLAMA DİLLERİNİN PRENSİPLERİ DERSİ

2024-2025 Bahar dönemi
Dr.Öğr.Üyesi MUHAMMED FATİH ADAK

SELİM ALTIN	**G231210558 2.A**
selim.altin@ogr.sakarya.edu.tr	

Ödev Raporu

Uzay Aracı Simülasyonu

2025

BİLGİSAYAR MÜHENDİSLİĞİ

1. Rapor Özeti

Bu proje “Uzayda Yaşam Simülasyonu” adıyla başladığım, C dilinde yazdığım bir derinleşme çalışması. Amaç; farklı gezegenlere giden uzay gemilerini, içlerindeki yolcuları ve gezegen takvimlerini saat saat ilerleterek gözlemlemekti. Çalışmayı yaparken nesne-yönelimli düşünce tarzını (constructor, destructor, fonksiyon işaretçileri) C’ye uyarladım; aynı zamanda dosyalardan veri okuyup simülasyon döngüsünü canlı bir konsol çıktısına dönüştürdüm.

2. Projeden Beklenenler

Hocamız bizden, gerçek takvim kurallarına uyan, modüler yazılmış ve DOS/Unix farkı gözetmeden derlenebilen bir program istedi. Klasik C fonksiyonlarının yanında “struct içinde metot” düzenini kurmamız da ayrıca vurgulandı. Sonuçta program; dosyalardaki gezegen, araç ve kişi bilgilerini okuyacak, her geminin kalkış-varış süresini takip edecek, yolcular yaşlandıkça ölebilecek, Gezegenlerin günü farklı uzunlukta olabilecek ve döngü sonunda ekranda tablolardan oluşan net bir rapor verecekti.

3. Projenin Başlangıcı ve Planlama

İlk gün klasör iskeletini kurdum: src/, include/, data/, build/ ve kökte Makefile. Daha sonra her modül için boş .h/.c dosyalarını açtım. Zaman modülünü önce yazdım, çünkü takvim işlerse gerisi rahat gidiyor. Ardından dosya okuma fonksiyonlarını bitirip elimde ham veri olduğuna emin oldum. Üçüncü aşamada UzayAraci, Gezegen ve Kisi struct’larını pointer tabanlı “nesnelere” çevirdim. Son olarak da Simulasyon döngüsünü bağlayıp ekrana basılan çıktıyı düzenledim.

4. Dosya Okuma

Dosya okuma kısmında beni en çok “bilinmeyen satır sayısı” ve “Windows satır sonu” zorluyordu. Çözüm olarak her satırı fgets ile çektikten sonra \r ve \n karakterlerini temizleyen minik bir temizleSatir yazdım. Satırı parçalarken strtok_r kullandım; eksik sütun varsa o satırı es geçiyorum, böylece yanlış format programı göçürmüyor. Ayrıca dizi boyutunu önden tahmin etmediğim için realloc ile iki katına çıkarma modeline gittim.

5. Simülasyon Döngüsü ve Durum Yönetimi

_baslatSimulasyon adında tek bir ana fonksiyonum var; içindeki işi beşe böldüm:

1. Saat başlığı basmak
2. Kalkış kontrolü (handleDepartures)
3. Gemileri ve yolcuları hareket ettirmek (moveShipsAndPassengers)
4. Gezegen zamanlarını ilerletmek
5. Döngü bitince raporları basmak

Her gemi “hasDeparted” bayrağı ile takip ediliyor. Kalkış tarihinde gezegen saati geminin çıkış tarihine denk gelince uzayAraciDepart çağrılıyor; geminin saat sayacı geriye doğru inmeye başlıyor. Yolculuk sırasında yolcuların ömrü çıkış gezegeninin katsayısıyla her tur azalıyor. Kalan yolcu sıfıra düşünce gemi imha oluyor. Kalan saat sıfıra inince varış tarihi geminin içine yazılıyor.

6. Karşılaşılan Zorluklar

- Tarih Sapması: Gezegenlerin gün uzunlukları farklı olduğu için varış tarihini hesaplarken önce birkaç kez hatalı gün atladım. Çözüm; varış tarihini hesaplarken çıkış anındaki gün/saat bilgisini kopyalayıp travelHours kadar ilerle() demek oldu.
- Bellek Sızıntıları: Her toString çağrısından sonra free yapmayı unutunca Valgrind uyarıları aldım. Yapacak başka yol yok, satır satır temizledim.
- Başlık Satırı: Txt dosyalarının ilk satırı insan okuyabilir başlıktı. İlk denemede başlık da “veri” gibi parse edilince program çöktü. Artık ilk token “isim” ise satırı atlıyorum.

7. Eksikler / İyileştirmeler

- Varış tarihini hâlâ bazı rotalarda bir gün şaşırabiliyorum; gün uzunluğu 17-18 saat olan gezegenlerde ekstra test lazım.
- Gezegenler ve gemiler için ayrı thread açıp gerçek zamanlı çizelge yapmak güzel olurdu, ama proje scope’unu aşar diye bırakıldı.

8. Programda Kullandığım Veri Yapıları ve Teknolojiler

- C11 + MinGW GCC
- Dinamik diziler (malloc, realloc)
- Fonksiyon pointer’ları ile sözde-nesne modeli
- strtok_r, sscanf, snprintf ile güvenli string işlemleri
- ANSI escape kodu ile ekran temizleme

9. Sonuç ve Öğrendiklerim

Bu proje bana şunu gösterdi: Düz C’de bile temiz mimari kurmak mümkün, yeter ki sorumlulukları ayırmayı bilin. Fonksiyon pointer’ları sayesinde constructor-destructor, hatta basi+t kalıtım benzetimi yapabildim. Ayrıca gerçek takvim hesabına girip artık yıl, ay gün sayısı gibi ayrıntıları kodlamak; “küçük görünen tarih problemi”nin aslında ne kadar kafa patlatan bir iş olduğunu öğretti. Dosya okurken başlık atlaması, hatalı satırı yumuşakça silmek, bellek taşmasını önlemek gibi pratikler de projelerimde tekrarlayacağım disiplinler oldu. En önemlisi, hata mesajlarını sakin kafayla okuyup sorunu adım adım izole etmenin ne kadar zaman kazandırdığını gördüm.

Kısacası, bu ödev bana hem C dilinde ileri seviye bellek ve pointer kullanımı, hem de yazılım mimarisi açısından büyük katkı sağladı.

10.Çıktı

```
===== iterasyon  319 =====
Gezegen A  : 14.05.2025 07:00
Gezegen B  : 16.05.2025 19:00
Gezegen C  : 11.05.2025 19:00
Gezegen D  : 18.05.2025 13:00
Gezegen E  : 15.05.2025 11:00

  [VARIS]   Y hedefe ulasti: 11.05.2025
  [VARIS]   Z hedefe ulasti: 12.05.2025
  [VARIS]   L hedefe ulasti: 06.05.2025
------------------------

Gezegenler:

                        -- A --            -- B --            -- C --            -- D --            -- E --
Tarih:                  14.05.2025         17.05.2025         11.05.2025         18.05.2025         15.05.2025
NÃ¾fus:                     4                  0                  1                  0                  0

Uzay Araclari:
Arac Adi   Durum      cikis      varis      Hedefe Kalan Saat    Hedef Tarih
X          IMHA       E          D          --                   --
Y          Vardi      B          C          0                    11.05.2025
Z          Vardi      C          A          0                    12.05.2025
Q          IMHA       E          A          --                   --
L          Vardi      E          A          0                    06.05.2025
M          IMHA       A          E          --                   --

Tum araclar hedefe ulasti. Simulasyon tamamlandi.
```