

PROGRAMLAMA LABORATUVARI 1

3. PROJE

Bilgisayar Mühendisliği Bölümü Kocaeli Üniversitesi

Ali Murat Ekmekçi *
200202114

Melih Turaneri †
200202075

Aralık 2021

1 Özet

Bu doküman Programlama Laboratuvarı 1 dersi 3. Projesi için çözümümüzü açıklamaya yönelik oluşturulmuştur. Dökümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklara yer verilmiştir. Doküman sonunda projemizi hazırlarken kullandığımız kaynaklar bulunmaktadır.

2 Proje Tanımı

2.1 Proje Tanımı

Bilginin geliş sırasına göre, ilk önce gelen elemana ilk erişilen liste yapısına kuyruk (queue) denir. Bu erişimde First-In-First-Out (FIFO) prensibi vardır. Yani ilk giren eleman, ilk çıkar. Örneğin sinema bileti

almak için sıraya girmiş kişileri düşünebiliriz. İlk önce gelen kişi bileti daha önce alacaktır. Queue veri yapısında, verilere iki uçtan erişim vardır. Bir uçtan eleman ekleme (enqueue), diğer uçtan eleman çıkarma (dequeue) işlemleri yapılır. Queue tasarımı dizi veya bağlı liste ile yapılabilir. Bağlı liste kullanarak boyutu sabit olmayan bir queue oluşturabiliriz. Dizi kullanmak için ise sabit bir boyut belirlemeliyiz. Bazı problemlerin çözümünde doğrudan kuyruk oluşturamayız. Örneğin uçakların inişi sırasında, acil inmesi gereken uçaklar bulunabilir. Veya muayene sırasında bekleyen hastalar için farklı bir öncelik belirlenebilir. Bu gibi senaryolarda öncelikli kuyruk ile çözüm üretilir. Öncelik sırası belirlenir ve program sırasında uygulanır.

Projede bizden istenen uçakların öncelik durumları, uçak tanım numaraları ve talep ettikleri iniş saatine göre öncelikli kuyruk (priority queue) kullanarak bir havalimanı uçuş yönetim sistemi oluşturmaktır.

*alimuratekmekci@gmail.com

†turaneri45@gmail.com

Projemizin temel amacı öncelikli kuyruk veri yapısı kullanarak sıralama işlemleri yapmaktır.

Projede C dili kullanılmıştır.

2.2 İsterler

1 iniş 1 kalkış olmak üzere 2 pisti bulunan İstanbul Havalimanı'nda gün içerisinde yapılan uçuşların yönetimi için bir sistem geliştirmemiz beklenmektedir. Havalimanında aynı anda sadece 1 uçak kalkış yapabiliyorken sadece 1 uçak iniş yapabilmektedir. Uçakların her biri iniş ve kalkışta farklı önceliklere sahiptir ve 1 günde maksimum 24 uçak iniş için izin isteyebilmektedir. Havalimanındaki uçakların öncelik sırası, iniş saati, gecikme süresi ve kalkış saati bilgileri kullanılarak; iniş pistini ve kalkış pistini kullanım sırasını belirlememiz beklenmektedir. Buna göre aşağıdaki durumlar göz önünde bulundurulmalıdır:

- Uçakların iniş ve kalkış saatleri önceliklerine göre belirlenecektir ve öncelik sıralamaları şu şekildedir.
 1. Ambulans Uçağı
 2. Savaş Uçağı
 3. Yolcu Uçağı
 4. Kargo Uçağı
- İniş yapmak isteyen uçak öncelikle iniş izni talep etmeli.
- Her talep sonrası kuyruğa eleman eklemek için havalimanında yeterli kapasite olup olmadığı kontrol edilmeli.
- İniş talep edilen saatte pistin dolu mu boş mu olduğu kontrol edilmelidir. Pist

boş ise iniş yapılmak istenen saate izin verilmeli ve kuyruğa eklenmelidir. Aksi halde uçakların iniş sıralaması önceliğe göre belirlenmelidir.

- İniş izni alan uçaklar için "İniş izniniz onaylanmıştır.", alamayan her uçak için ise "Talebiniz öncelikli uçak inişi sebebiyle ertelenmiştir." yazılmalıdır.
- İniş talep eden uçakların önceliği (oncelik_id), uçak numarası (ucak_id) ve talep ettiği iniş saati (talep_edilen_inis_saati) input.txt dosyasından okunacaktır.
- Önceliği yüksek olan uçaklar nedeniyle önceliği düşük olan herhangi bir uçağın inişi, maksimum 3 kez ertelenebilir. Eğer iki uçak için 3'ten fazla ertelenme durumu söz konusuysa önceliği düşük olan uçak iniş yapmalı diğer uçak Sabiha Gökçen Havalimanına gönderilmelidir.
- Her yeni input satırı okunduğunda, kalkış yapacak olan uçakların bulunduğu output.txt dosyası güncellenmeli ve güncel öncelikli kuyruk ekranda gösterilmelidir.
- Output.txt dosyasında oncelik_id, ucak_id, talep_edilen_inis_saati, iniş_saati, gecikme_suresi, kalkış_saati olmak üzere toplamda 6 bilgi içermelidir.

3 Araştırmalar

Projemize başlamadan önce, C dilinde öncelikli kuyukları ayrıntılı bir şekilde araştırdık. Ne işe yaradıkları, nerede ve nasıl

kullanıldıklarına dair çeşitli bilgiler topladık. Bu araştırmalar ışığında öncelikli kuyrukları oluştururken dizi veya bağlı liste kullanarak oluşturabileceğimizi gördük. Bunun yanı sıra öncelikli kuyruk oluştururken kullanabilecek çeşitli işlemler var olduğunu gördük. Öncelikli kuyruk veri yapısını kullanmanın avantaj, dezavantaj ve kullanım alanları şu şekildedir. Avantajları;

1. Geliş sırasına göre hizmet verilmesi gereken senaryolarda avantajlıdır.
2. Üretici-tüketici problemlerinde fayda sağlar.
3. Hastanelerde, uçakların inişinde, araç geçişlerinde öncelikli kuyruk kullanılabilir.

Dejavantajları;

1. Üzerinde arama yapmak zahmetlidir. En baştan başlanıp ilerlemek gerekir.
2. Kuyruğun aralarına eleman eklemek karmaşıktır.

Kullanım alanları;

1. İşletim sistemlerinde çalışma önceliği kuyruk ile yapılır.
2. Ağ yazıcılarında, belgeler öncelikli kuyruk ile çalışır.

Bu bilgileri edindikten sonra projemize başlamaya hazır hale geldik ve başladık.

4 Yöntem

Bu kısımda projemizi oluştururken izlediğimiz yöntemleri açıklayacağız.

Biz projemizde öncelikli kuyruk oluştururken dizi kullanmayı tercih ettik. Projemizi ilk olarak çalıştırdığımızda içerisinde `öncelik_id`, `uçak_id` ve `talep_edilen_iniş_saati` bilgileri .txt uzantılı input dosyasından okunuyor. Ardından okunan bilgiler bir struct dizisine aktarılıyor. Bu struct dizisi üzerinde bir takım işlemlerin ardından öncelik şartlarına göre elemanlar teker teker öncelikli kuyruğa alınıyor. Ardından talep edilen iniş saatlerine göre bir karşılaştırma daha yapılıyor. İlk uçağın inişi ardından iniş talebi onaylanmayan önceliği düşük uçakların iniş talep saatleri bir saat erteleniyor. Bu erteleme sonrası, sıradaki iniş için talep edilen inişler kendi arasında da öncelik durumlarına göre tekrar sıralanıyor ve öncelikli kuyruğa ekleniyor. Erteleme durumlarında bir uçak için maksimum erteleme sayısını üç olarak ayarladık ve üç erteleme sonrası önceliği ne olursa olsun üç kez ertelenmiş olan uçağı indirmeyi başardık. Ancak aynı anda iki adet uçak eğer üç kez ertelenmiş ise tek bir uçağın inişi gerçekleştirilebileceğinden önce uçakların önceliklerine baktık ve önceliği düşük olan uçağı indirip diğer uçağı Sabiha Gökçen Havalimanına yönlendirdik. Ancak uçakların `öncelik_id`'lerinin aynı olması durumunda da bu kez uçakların `uçak_id`'lerine baktık ve `uçak_id`'si küçük olan uçağı indirip diğer uçağı Sabiha Gökçen Havalimanına yönlendirmeyi başardık. Bu işlemler kuyruksa maksimum 24 uçak olana kadar tekrar ediyor. Eğer kuyruksa uçakların iniş saatleri ertesi güne kalmış ise bu uçakları da Sabiha Gökçen Havalimanına yönlendirdik. Tüm bunlar olurken inen tüm uçaklar iniş saatlerinden bir saat sonra kalkış kuyruğına giriyor ve kalkışı gerçekleşiyor. Bu verileri

oluşturduğumuz bir Output.txt dosyası içinde sırasıyla oncelik_id, ucak_id, talep_edilen_inis_saati, inis_saati, gecikme_suresi ve kalkis_saati şeklinde saklanıyor ve yazdırılıyor. Ayrıca ekranda da kuyruktan çekilen uçakların oncelik_id, ucak_id, talep_edilen_inis_saati, inis_saati ve gecikme_suresi yazdırılarak kuleden iniş talebinde bulunuluyor. Eğer iniş izin talebi onaylanmış ise ekranda "İniş İzin Talebiniz Onaylanmıştır" yazdırılıyor. Eğer iniş izin talebi onaylanmamışsa ekranda "İniş İzin Talebiniz Öncelikli Uçak İnişi Sebebiyle Ertelenmiştir" yazdırılıyor. Eğer 3 erteleme sonrası uçak başka bir havalimanına yönlendiriliyorsa ekranda "İniş Saatiniz Kotasından Fazla Ertelendiği İçin Sabiha Gökçen Havaalanına Yönlendiriliyorsunuz" yazdırılıyor. Eğer uçağın inişi ertesi güne kaldıysa ekranda "İnişiniz Ertesi Güne Kaldığı İçin Sabiha Gökçen Havaalanına Yönlendiriliyorsunuz" yazdırılıyor.

4.1 Kazanımlar

5 Geliştirme Ortamı

Projemizi Linux işletim sisteminde , CodeBlocks programı üzerinde gerçekleştirdik. Proje raporunu oluşturmak için LaTeX programı kullandık.

6 Kod Bilgisi

6.1 İstatistik

Program kodu toplam 526 satırdan oluşmaktadır.

6.2 Akış Diyagramı

Kısım ektedir. [1](Akış Diyagramı)

7 Sonuç

Projenin gerektirdiği tüm isterleri sağladık.

8 Kaynakça

1. <https://www.texdizgi.com/category/genel/>
2. <https://app.diagrams.net/> -Akış diyagramı oluşturmak için.
3. <https://medium.com/@tolgahan.cepel/dogrusal-veri-yapilari-4-kuyruk-queue-dcbd07e8ba77>
4. https://en.wikipedia.org/wiki/Linked_list#Language
5. <http://www.baskent.edu.tr/~tkaracay/etudio/ders/prg/dataStructures/Collections/Class>
6. <https://www.programiz.com/dsa/priority-queue>
7. [https://tr.wikipedia.org/wiki/Yığın_\(veri_tipi\)](https://tr.wikipedia.org/wiki/Yığın_(veri_tipi))