

Kocaeli Üniversitesi

Bilgisayar Mühendisliği Bölümü

Programlama Laboratuvarı 1-3

Mahmut Emre Terzi-180201087

Mahmut Bilgi-200201037

Özet

Bu doküman Programlama Laboratuvarı 1 dersi 3. Projesi için çözümümü açıklamaya yönelik oluşturulmuştur. Dokümanda 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 ve proje derlenirken dikkat edilmesi gereken hususlar bulunmaktadır.

1-Proje Tanıtımı

PROJENİN ADI: Havalimanı Uçuş Yönetim Sistemi
PROJENİN AMACI: Öncelikli kuyruk (priority queue) kullanarak bir havalimanı uçuş yönetim sistemi oluşturmak. Kuyruk (Queue): Kuyruk, ilk giren eleman ilk çıkar (First In First Out – FIFO) mantığında çalışan bir veri yapısıdır. Örneğin, kuyruk veri yapısı bankada işlem yaptırmak için sıraya girmiş insanlara benzetilebilir. Sıraya ilk giren kişi ilk işlem yaptıracaktır. Kuyruk tasarımı için dizi ya da bağlı liste kullanılabilir. Dizi kullanılan sabit boyutlu, bağlı liste kullanılarak değişken boyutlu kuyruk oluşturulabilir. Kuyrukta işlemler iki uçtan yapılır. Kuyruk veri yapısında yapılabilecek işlemlerden bazıları aşağıdaki gibidir:

- enqueue(): Kuyruğun önüne eleman ekler.
- dequeue(): Kuyruğun sonundan eleman çıkarır.
- peek(): Silme işlemi uygulamadan sıradaki elemanı (front işaretçisinin gösterdiği düğüm) döndürür.

Öncelikli Kuyruk (Priority Queue): Bazı problemlerin çözümünde doğrudan kuyruk oluşturulamaz. Örneğin; bir hastanede muayene sırasına girmiş insanlar arasında durumu acil olan birisi bulunabilir ve bu kişi muayene için öncelikli hale gelebilir. Bu gibi durumlarda öncelikli kuyruk kullanılır. Öncelikli kuyrukta ilk giren ilk çıkar mantığı geçerli değildir, önemli olan önceliktir. Öncelikli kuyruk veri yapısında yapılabilecek işlemlerden bazıları aşağıdaki gibidir:

- add(): Kuyruğa eleman eklemek için kullanılır.
- poll(): Kuyruktaki son elemanı döndürür ve elemanı kuyruktan siler.
- peek(): Kuyruktaki son elemanı silmeden döndürür.
- clear(): Kuyruktaki bütün elemanları siler.
- remove(): Kuyruktaki belirtilen elemanı siler.

2-Proje yöntemi Deney ve Araştırma

1 iniş 1 kalkış olmak üzere 2 pisti bulunan İstanbul Havalimanı'nda gün içerisinde (1-24 saat dilimi boyunca) yapılan uçuşların yönetimi için bir sistem geliştirilecektir. 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 sırasının belirlenmesi hedeflenmektedir.

- Havalimanına iniş yapacak uçaklar öncelikle kuleden iniş yapabilmek için izin talep etmelidir.
- İniş izni talep eden her bir uçak için havalimanında yeterli kapasite olup olmadığı kontrol edilmelidir (inis_pisti_kullanim_sirasi öncelikli kuyruğunda yeni uçak eklemek için boş alan var mı?).
- Kuleden iniş izni talep eden uçaklar için öncelikle, iniş talep edilen saatte pistin dolu mu boş mu olduğu kontrol edilmelidir. Pist boş ise iniş yapılmak istenen saate izin verilmeli ve inis_pisti_kullanim_sirasi'nda uygun yere eklenmelidir. Aksi halde uçakların iniş sıralaması önceliğe göre belirlenmelidir.
- İniş izni talep eden her uçak için “İniş izni talebiniz onaylanmıştır” veya “... nedeniyle iniş izni verilememektedir.” şeklinde ekranda yazdırılmalıdır.
- Uçakların iniş ve kalkış saatleri önceliğe göre belirlenecektir. Uçakların öncelik (oncelik_id) sıralaması şu şekildedir (yüksekten düşüğe):
 1. Ambulans uçağı
 2. Savaş uçağı
 3. Yolcu uçağı
 4. Kargo uçağı

- Havalimanına iniş talep eden uçakların önceliğı (oncelik_id), uçak numarası (ucak_id) ve talep ettiğı iniş saati (talep_edilen_inis_saati) input.txt dosyasından okunacaktır.
- Tüm uçakların iniş ve kalkış süreleri eşittir ve hesaplamalara dâhil edilmeyecektir.
- Havalimanına iniş yapan her uçağın, kalkış için bekleme süresi 1 saattir. Uçakların kalkış saatine, ötelenmeden dolayı oluşan gecikme süreleri dâhil edilmelidir. Kalkış saati bu bilgiler göz önünde bulundurularak hesaplanmalıdır.
- Aynı önceliğe sahip iki uçak, aynı saatte kalkış yapacaksa öncelik ilk iniş yapan uçağa verilmelidir.
- Önceliğı yüksek olan uçaklar nedeniyle önceliğı düşük olan herhangi bir uçağın uçuşu, maksimum 3 kez ertelenebilir. Eğer 3'ten fazla ertelenme durumu söz konusuysa, öncelik gözetilmeksizin beklemede olan uçağın kalkışı gerçekleştirilmelidir.
- Kuleden bir günde maksimum 24 uçak iniş için izin talep edebilir. Eğer bu kapasite dolmuşsa;
 - İniş için onay alan uçaklardan en az birinin önceliğı (X uçağı olsun), iniş izni onayı bekleyen uçağın (Y uçağı olsun) önceliğinden düşükse; yüksek öncelikli yeni uçağa (Y) iniş onayı verilir. Daha önce onay almış ve önceliğı düşük olan uçak (X) başka bir havalimanına yönlendirilmelidir.
 - İniş izni daha önceden onaylanan uçağın (X) izni iptal edilmişse; “Acil iniş yapması gereken ...(Y) uçağı nedeniyle iniş izniniz iptal edilmiştir, iniş için Sabiha Gökçen Havalimanı'na yönlendiriliyorsunuz.” şeklinde ekranda yazdırılmalıdır.
- İniş izni talep eden uçakların her biri satır satır input.txt'den okunmalıdır. Okunan her bir satır ekranda gösterilmelidir.

- Her yeni input satırı okunduğunda, kalkış yapacak olan uçakların bulunduğu output.txt dosyası güncellenmeli ve güncel kalkis_pisti_kullanım_sirasi öncelikli kuyruğu ekranda gösterilmelidir.
- Projede kullanılacak input.txt dosyası ekte verilmiştir. output.txt dosyası ise proje içerisinde öğrenci tarafından oluşturulacaktır.
- input.txt dosyası aşağıdaki bilgileri içermektedir:
 - Havalimanına 1 günde iniş izni talep eden uçakların listesi
 - Dosyadan sırasıyla (satır satır) boşluk ile ayrılmış 3 bilgi (oncelik_id, ucak_id, talep_edilen_inis_saati) okunacaktır.
- output.txt dosyası uçakların kalkış saati bilgisine göre oluşturulmalıdır. Dosyanın içeriği şu şekilde olmalıdır: o Her satırda oncelik_id, ucak_id, talep_edilen_inis_saati, inis_saati, gecikme_suresi, kalkis_saati olmak üzere toplamda 6 bilgi içermelidir. o inis_saati bilgisi; eğer uçak talep ettiği saatte iniş yapmışsa talep_edilen_inis_saati ile aynı olmalıdır. Ancak talep edilen saatte önceliği yüksek başka bir uçak iniş yapacaksa belirlenen yeni iniş saati, kalkış saatinin hesaplanmasında kullanılmalıdır.
- Bilgiler input.txt’de olduğu gibi boşluk ile ayrılmalıdır.
- Projede kullanılacak veri yapıları aşağıdaki gibidir: o inis_pisti_kullanım_sirasi: Havalimanına iniş yapacak uçakların iniş pistini kullanım sırasını içeren öncelikli kuyruktur. Uçak öncelik ve iniş saati bilgisi dikkate alınarak hazırlanacaktır.
- kalkis_pisti_kullanım_sirasi: Havalimanından kalkış yapacak uçakların kalkış pistini kullanım sırasını içeren öncelikli kuyruktur. Uçak öncelik ve kalkış saati bilgisine göre hazırlanacaktır.

3-Yalancı Kod

```

Struct plane tanımlaması yapıyoruz;
Struct plane içinde
oncelikli_id,uçak_id,talep_edilen_inis_saati.ve
erteleme adlı global değişkenlerimizi tanımlıyoruz;
İnt dizi sıfırlama adlı int fonksiyonumuzu
tanımlıyoruz;
swap_inis_saati = (plane_info+d)-
>talep_edilen_inis_saati;//plane_info[counter].tale
p_edilen_inis_saati      (plane_info+d)-
>talep_edilen_inis_saati = (plane_info+d+1)-
>talep_edilen_inis_saati;      (plane_info+d+1)-
>talep_edilen_inis_saati = swap_inis_saati;
swap_oncelik_id = (plane_info+d)-
>oncelik_id;//plane_info[counter].oncelik_id
(plane_info+d)->oncelik_id = (plane_info+d+1)-
>oncelik_id; (plane_info+d+1)->oncelik_id =
swap_oncelik_id; swap_ucak_id = (plane_info+d)-
>ucak_id;//plane_info[counter].ucak_id
(plane_info+d)->ucak_id = (plane_info+d+1)-
>ucak_id;(plane_info+d+1)->ucak_id =
swap_ucak_id;
for(int i=0; i<counter; i++)
printf("%d %d %d\n", (plane_info+i)-
>oncelik_id, (plane_info+i)-
>ucak_id, (plane_info+i)->talep_edilen_inis_saati);
int directedDizi[SIZE];
int *inisDizi;
inisDizi=(int *)malloc(SIZE*sizeof(int));
for(int i=0; i<SIZE; i++)
inisDizi[i]=0;
int sayac=1;
int sayac2=0;
int sayac3=0;
adlı işlemleri yapıyoruz;
printf("\n");
for(int i=0; i<28; i++){if((plane_info+i)-
>oncelik_id==sayac)or(sayac2=0; sayac2<4;
sayac2++)
if(inisDizi[((plane_info+i)-
>talep_edilen_inis_saati)-1+sayac2]!=0)
continue;
else if(inisDizi[((plane_info+i)-
>talep_edilen_inis_saati)-
1+sayac2]=inisDizi[((plane_info+i)-
>talep_edilen_inis_saati)-1+sayac2] =
(plane_info+i)->ucak_id;
break;

```

```
if(sayac2==4)
directedDizi[sayac3] = (plane_info+i)-
>ucak_id;
sayac3++;
else if((plane_info+i)->oncelik_id!=sayac)
i--;
sayac++;
int *inisDizi2;
inisDizi2=(int *)malloc(SIZE*sizeof(int));
for(int i=0;i<SIZE;i++)
inisDizi2[i]=0;
int size ile bellekte yerimizi açıyoruz;
printf uçak talebi alınıyor;
printf uçak bilgisi;
printf uçak no;
printf istediği iniş saati;
printf talep edilen saatte başka bir uçak
yerleştirildi;
printf tarif edilen başka bir uçak vardır.
printf talep edilen saatte başka bir uçak
yoktur yerleştirildi;
printf talep edilen saatte başka bir uçak
vardır;
printf yeni talepte bulunan uçağın önceliği
daha yüksek;
printf öncelik numaralarını karşılaştırılıyor;
printf Yeni uçak ve önce talepte bulunan
ucagin oncelik id'si aynidir
printf yeni talepte bulunan uçağın. Numarası
kaydırılmıştır;
printf yeni talepte bulunan uçağın saati
kaydırılmıştır;
printf yeni uçağın önceliği düşüktür;
printf yeni talepte bulunan uçağın saati
kaydırılmıştır;

int selection uçakların iniş kalkış saatini
göster;
case 1 uçak listesi;
case 2 exit;
fclose(input);
free(plane_info_2);
free(plane_info);
free(inisDizi);
free(inisDizi2);
```

4-Kaynakça

https://cdn-acikogretim.istanbul.edu.tr/auzefcontent/20_21_Guz/veri_yapilari/7/index.html

<https://ckaynak.com/c-programlama-dili-tek-yonlu-kuyruk-yapisi-1407>

<http://www.yazilimtuneli.com/2019/11/c-programlama-ygn-ve-kuyruk-ornek.html>

<https://www.youtube.com/watch?v=1hKLn12poa8>

[https://tr.wikipedia.org/wiki/Kuyruk_\(veri_yap%C4%B1s%C4%B1\)](https://tr.wikipedia.org/wiki/Kuyruk_(veri_yap%C4%B1s%C4%B1))

<https://stackoverflow.com/questions/21282647/c-tailing-a-file-and-intercepting-output>

<https://stackoverflow.com/questions/38047194/how-to-code-my-own-version-of-tail-unix-command-in-c-language>