YILDIZ TEKNİK ÜNİVERSİTESİ

ELEKTRİK-ELEKTRONİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

BLM2512 – VERİ YAPILARI VE ALGORİTMALAR

1.ÖDEV RAPORU

Konu : Cache Buffer (Önbellek Tamponu) Tasarımı

Ad Soyad : Eren TUTUŞ

Okul No : 17011702

E-posta : l1117702@std.yildiz.edu.tr

Programın Çalışma Süreci

1-) node *olustur() fonksiyonu :

Her düğüm node *olustur fonksiyonu icinde oluşturulup BasaEkle() fonksiyonuna yollanmakta. Ayrıca oluşturulan ilk düğümlerin *next ve *prev komutlarıda burada oluşturulmakta.

2-) void BasaEkle() Fonksiyonu :

Her farklı adres en başa yüklenecektir eğer aynı adres gelirse sayacı arttırılacak ve program akışı bu şekilde devam edecek. Ancak, ne zaman herhangi bir adresin sayacı belirlenen eşik değerini aşması durumunda sayacını arttırıp adresi listenin en başına taşıyacaktır.

Burada dikkat edilecek diğer husus ise belirlenen kapasite değerine karşılık kaç farklı adres bilgisi çekilmekte. Eğer girilen farklı adres sayısı belirlenen kapasite değerini aşarsa listenin sonundaki düğüm silinecek ve yeni gelen adres listenin en başına atanacaktır.

3-) main () Fonksiyonu :

Program başlarken kullanıcıdan Cache Buffer için eşik ve kapasite değerleri istenir. Daha sonra kullanılacak adreslerin dosyadan mı çekilmesi veya kullanıcının klavyeden girmesi için seçim yapması istenir. Eğer klavyeden giriş yapılması istenirse kaç adet adres girişi yapacağı da kullanıcıdan giriş yapılması istenmektedir.

Son aşamada ise Cache Buffer'ın tamamen temizlenmesine olanak sağlamak için kullanıcıya 2 seçenek sunulmuştur. Eğer konsol ekranından ' 1 ' tuşuna basılırsa Cache Buffer tamamen temizlenecek ' 2 ' tuşuna basılırsa temizleme işlemi yapmadan listenin son halini ekranda yazdıracaktır.

C Kodu :

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct link{
	char adres[30];
	int sayac;
	struct link *next; // Doubly Linked List yapısına uyarlamak için ->next ve ->prev oluşturuyorum.
	struct link *prev;
}node;

node *head = NULL;

node *olustur(char adres[30]){  // Her düğümü node *olustur fonksiyonu icinde oluşturup BasaEkle() fonksiyonuna yolluyorum..
	node *yenidugum = (node *) malloc(sizeof(node));  // yollanan char icin bellekte yer acıyorum..
	yenidugum->sayac = 1;  // ilk defa olusturuldugu icin sayac=1 olarak başlatıyorum..
	strcpy(yenidugum->adres,adres);
	yenidugum->next = NULL; // Her deger aldıgımda ilk deger gibi aldıgımdan dolayı ->next = NULL ve ->prev = NULL ataması yapıyorum.
	yenidugum->prev = NULL;
	return yenidugum;
}

void BasaEkle(char adres[30],int kapasite,int esik){
	node *basaekle = olustur(adres);
	if(head==NULL) // head nodu NULL ise boş demektir o zaman node *olustur() fonksiyonundan gelen bilgiyi head noduna ekliyorum.
	{
		head = basaekle;
		return;
	}

	int kontrol=0; // kontrol değişkeni koyma nedenim dosyadan gelen string bilginin linked list icinde olup olmadığını sorgulamak icin kullanacağım.
	node *current = head;
	while(current!=NULL)
	{
		if (strcmp(current->adres,adres) == 0) // Dosyadan gelen deger linked list icinde varmı ? eğer var ise if bloğuna giriş yapacağız.
		{
			if (current->sayac <esik || current->sayac >esik) // Gelen string'in sahip oldugu sayac eşik değerden düşük veya eşik değerden büyükse sadece sayacında arttırma olacak.
			{
				current->sayac++;
				kontrol++;
				return;
			}
			else // else durumunda ise geriye tek koşul kalıyo oda string'in gelen adres'e eşit olması.
			{
```

```
                        if (current->prev == NULL){ // Önceki değer NULL
ise o en baştaki elemandayız bu nedenle sayacını sadece arttırmamız
yeterlidir.
                                current->sayac++;
                                return;
                        }
                        if (current->next == NULL)  // Sondaki degerin
next'i NULL ise o zaman sondaki elemandayız ve bunu başa alacağız...
                        {
                                current->sayac++;
                                node *onceki_kisim = current->prev;
                                onceki_kisim->next = NULL;
                                node *item2 = (node*)malloc(sizeof(node));
                                strcpy(item2->adres,current->adres);
                                        item2->sayac = current->sayac;
                                        item2->prev = NULL;
                                        free(current);
                                        node *eski = head;
                                        head = item2;
                                        head->next = eski;
                                        eski->prev = head;
                                        return;
                                }


                                        // Önceki iki if komutunda ilk baştaki
node yada son durumdaki node'un olup olmadıgını sorguladık ve başta ise
zaten başta kalacak sonda ise başa alacaktık.
                                        //Eğer node başta ve sonda değilse tek
seçenek geriye kalıyor oda node'un ortada olması.
                                        current->sayac++;
                                        node *onceki = current->prev;
                                        node *sonraki = current->next;
                                        onceki->next = sonraki;
                                        sonraki->prev = onceki;
                                        node *item =
(node*)malloc(sizeof(node));
                                        strcpy(item->adres,current->adres);
                                        item->sayac = current->sayac;
                                        item->prev = NULL;
                                        free(current);
                                        node *eski = head;
                                        head = item;
                                        head->next = eski;
                                        eski->prev = head;
                                        return;
                                }

                        }
                                current = current->next;
                }


        if (kontrol == 0) // kontrol degiskeninin hala sıfır olması demek
yani linked list icinde dosyadan gelen string ile eşleşen bir veri
bulamadığı anlamına geliyor.
        {
                // KAPASİTE KONTROLÜ
                node *say = head;
                int sayici=0;
```

```
            while(say->next != NULL) // Listemizde ne kadar kutu var onu
öğrenmek icin bir sayici degiskeni kullandık bunu kapasite kontrolünde
kullanacağız.
            {
                  say = say->next;
                  sayici++;
            }
            if (sayici == kapasite){  // Listemizdeki kutular sayıldıktan
sonra artık kullanıcının girdiği kapasite ile eşit olup olmadığını
sorgulayabiliriz.
                  node *tempp = head;
                  while(tempp->next != NULL)
                  {
                        tempp = tempp->next;
                  }
                  node *oncekii = tempp->prev;
                  oncekii->next = NULL;
                  free(tempp);

                  node *eski = head;
                  head = basaekle;
                  head->next = eski;
                  eski->prev = head;
            }
            else {
                  node *eski = head;
                  head = basaekle;
                  head->next = eski;
                  eski->prev = head;
            }
      }
      else
      {
            head = current;
      }

}

void yazdir(node *baslangic) // Ekrana yazma fonksiyonumuz..
{
      node *temp = baslangic;
      printf("\n");
      while(temp!=NULL)
      {
            printf("%s,%d\t",temp->adres,temp->sayac);
            temp = temp->next;
      }
}

int main(){
      char adres[30];
      int secim,secim2,n,i=0;
      int kapasite,esik;
      printf("Kapasite kac olsun : ");
      scanf("%d",&kapasite);
      printf("Esik deger kac olsun : ");
      scanf("%d",&esik);
      printf("Klavyeden Okumak icin ' 1 ' tusuna basiniz.. \nDosyadan
Okumak icin ' 2 ' tusuna basiniz.  ");
      scanf("%d",&secim2);
      int sayac=0;
```

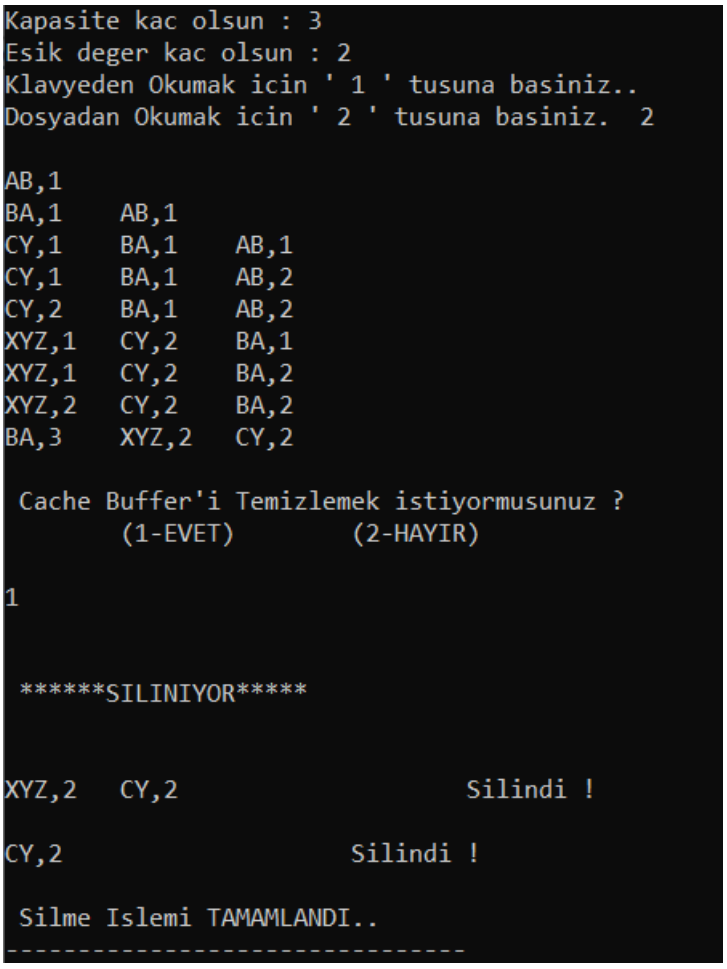
```
	FILE *fp;
	char buff[255];
	switch(secim2){   // Burada switch kullanmamın sebebi; dosyadan mı adresleri çekeceğiz yoksa klavyeden adres mi gireceğiz tercihini kullanıcıya yaptırmak için.
		case 1: // CASE 1: Klavyeden girilecek.
			printf("Kac string gireceksiniz : ");
			scanf("%d",&n);
			while(i<n){
			printf("\n Eklemek istediginiz metni girin .. ");
			scanf("%s",&adres);
			BasaEkle(adres,kapasite-1,esik);
			yazdir(head);
			printf("\n");
			i++;
	}
		break;

		case 2: // CASE 2: Dosyadan okunacak.
				fp = fopen("input.txt", "r");
				while(fscanf(fp, "%s", buff)!=EOF){
				strcpy(adres,buff);
				BasaEkle(adres,kapasite-1,esik);
				yazdir(head);
				sayac++;
				}
				fclose(fp);
				break;
		default: break;

	}
	printf("\n\n Cache Buffer'i Temizlemek istiyormusunuz ?\n \t(1-EVET)\t(2-HAYIR) \n\n");
	scanf("%d",&secim);
	switch (secim) // Kullanıcının Cache Buffer'i silebilmesine olanak tanımak için yeniden bir seçim yaptırdım.
	{
		case 1: printf("\n\n ******SILINIYOR*****\n\n");
				int j;
				for (j=0;j<kapasite;j++)
				{
					if (head==NULL){
						return;
					}
					if (head->next == NULL)
					{
						head = NULL;
						printf("\n Silme Islemi TAMAMLANDI..");
						return;
					}
					node *tut = head->next;
					tut->prev = NULL;
					free(head);
					head = tut;
					yazdir(head);
					printf("\t\tSilindi !\n");
				}
		break;
		case 2:
			printf("Silmemeyi tercih ettiniz.Listenin son hali\n");
			yazdir(head);
```

```
                    break;
                default: break;
        }
        return 0;
}
```

ÖRNEK EKRAN ÇIKTILARI



```
Esik deger kac olsun : 3
Klavyeden Okumak icin ' 1 ' tusuna basiniz..
Dosyadan Okumak icin ' 2 ' tusuna basiniz.  2

A,1
B,1     A,1
B,1     A,2
AA,1    B,1     A,2
BBB,1   AA,1    B,1     A,2
BBB,1   AA,1    B,2     A,2
BBB,1   AA,1    B,2     A,3
AB,1    BBB,1   AA,1    B,2
A,1     AB,1    BBB,1   AA,1
B,1     A,1     AB,1    BBB,1
B,1     A,2     AB,1    BBB,1
BB,1    B,1     A,2     AB,1

 Cache Buffer'i Temizlemek istiyormusunuz ?
        (1-EVET)        (2-HAYIR)

1


 ******SILINIYOR*****


B,1     A,2     AB,1                    Silindi !

A,2     AB,1                    Silindi !

AB,1                    Silindi !

 Silme Islemi TAMAMLANDI..
--------------------------------
Process exited after 60.08 seconds with return value 27
```