



Course Name: VERİ YAPILARI VE
ALGORİTMALAR

Course Group: Group 1

Instructor Name: Doç. Dr. M.Elif KARSLIGİL

ÖDEV 1

Student Id: 16011706

Student Name and Surname: Duygu Erduran

Konu : Cache Buffer (Önbellek Tamponu) Tasarımı

Açıklama :

Girişte verilen ifadeyi string olarak adım adım veya metin dosyasından bir seferde alarak, yapılan her istek sonunda cache buffer'ın mevcut linkli liste yapısını, düğümlerin içerikleri ile ekrana yazdırıldı. Bu çözüm yapılırken;

- File Dosya fonksiyonları (fopen, feof, fclose, fscanf) :
Dosyanın içinden karakter alıp kopyalama işlemleri yapıldı.
 - String fonksiyonları (strcpy ,strcmp)
 - Double Linkli Liste yapısı
 - Struct : Node yapısı için oluşturuldu.
 - Typedef tanımlaması
 - While döngüsü
 - If/Else/Else If Statements
 - puts();
 - free();
- kullanıldı.

C Kodu:

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

```
struct Node
{
    char adres[20];
    int sayac;
    struct Node *sonraki;
    struct Node *onceki;
};
```

```
//Buffer adinda bir düğüm yapisi tanimliyoruz
typedef struct Node node;
node *Buffer = NULL;
```

```
// fonksiyonlarimizi tanitiyoruz
node* yeniDugumOlustur(char[] );
void basaDugumEkle(char[] );
void tasanDugum(struct Node* );
void listele();
void temizle();
void dugumSil(struct linked_list* );
void sonDugumuSil();
int dugumBul(char []);
```

```
// degiskenlerimizi tanitiyoruz
int say = 0, T, L;
```

```
//istenilen degeri tutan bir düğüm olusturuyoruz
node* yeniDugumOlustur(char gelenAdres[20])
```

```
{
    node *geciciDugum;
    geciciDugum = (node *) malloc(sizeof(node)); //düğüm için bellekte
    yer aciyoruz
```

```
    //degerleri atiyoruz
    strcpy(geciciDugum->adres,gelenAdres);
    geciciDugum->sonraki=NULL;
    geciciDugum->onceki=NULL;
```

```
    return geciciDugum;
}
```

```
//dugumu listenin en onune ekliyoruz
```

```

void basaDugumEkle(char gelenAdres[])
{
    node *yeniDugum = yeniDugumOlustur(gelenAdres);

    if(Buffer == NULL) //ilk dugum icin
    {
        Buffer=yeniDugum;
        Buffer->sayac = 1;
        say = 1;

        return;
    }

    if(dugumBul(gelenAdres)==1) // eger düğüm varsa sadece sayaci
arttiriyoruz
        return;

    //ilk düğümden sonraki düğümlerin eklenmesi islemleri
    yeniDugum->sonraki = Buffer;
    Buffer->onceki = yeniDugum;

    Buffer = yeniDugum; // düğümümüzü güncelliyoruz
    Buffer->sayac=1;
    say++;

    if(say>L) // eger sayacimiz istenilen L degerinden büyükse
sonDugumuSil fonksiyonunu cagiriyoruz.
        sonDugumuSil();
}

//arama islemleri ve belli bir esik degerinden silme islemleri
int dugumBul(char gelenAdres[20])
{
    node *aramaDugum = Buffer;

    while(aramaDugum!=NULL) //bütün düğümler taraniyor
    {
        if(strcmp(aramaDugum->adres,gelenAdres)==0) //eger gelen
düğüme esitse;
        {
            aramaDugum->sayac++; //esitse sayac artar

            //esik degerini kontrol eder
            if(aramaDugum->sayac > T){
                tasanDugum(aramaDugum);
            }
        }
    }
}

```

```

        }
        return 1;
    }
    else{
        aramaDugum = aramaDugum->sonraki;
    }
}

}

//silme islemleri
void tasanDugum(struct Node* silinecek)
{
    //buradaki islemler sileceğimiz düğümü tutmak için
    node *yeniDugum = (node*) malloc(sizeof(node));
    strcpy(yeniDugum->adres, silinecek->adres);

    yeniDugum->sayac = silinecek->sayac;
    yeniDugum->onceki = NULL;

    //bossa islem yapmaz
    if (Buffer == NULL || silinecek == NULL)
        return;

    //en bastaysa islem yapmaz
    if(silinecek->onceki == 0)
        return;

    //asadaki kısımlarda aradan cikartılan dugumden öncesi ve sonrası
    birlestiriliyor
    if (Buffer == silinecek)
        Buffer = silinecek->sonraki;

    if (silinecek->sonraki != NULL)
        silinecek->sonraki->onceki = silinecek->onceki;

    if (silinecek->onceki != NULL)
        silinecek->onceki->sonraki = silinecek->sonraki;

    //yukarida tuttugumuz düğümü basa ekliyoruz
    yeniDugum->sonraki = Buffer;
    Buffer->onceki = yeniDugum;
}

```

```

    Buffer = yeniDugum;

    free(silinecek); //düğüm hafızadan siliniyor
}

void temizle(){

    node *geciciDugum = malloc(sizeof(node));
    geciciDugum = Buffer;
    while (Buffer != NULL)
    {
        geciciDugum = Buffer;
        Buffer = Buffer->sonraki;
        free(geciciDugum);
    }

}

void sonDugumuSil(){

    node *p,*q= (node*) malloc(sizeof(node));

    if(Buffer==NULL){
        printf("Linked list is empty\n");
    }
    else{
        p=Buffer;
        while(p->sonraki->sonraki!=NULL){
            p=p->sonraki;
        }
        q=p->sonraki;
        p->sonraki=NULL;
        free(q);
    }

}

//listeleme islemleri
void listele()
{
    node *listem;
    listem = Buffer;

    while(1)
    {
        if(Buffer==NULL) break;

```

```

        printf("[%s --> ", listem->adres);
        printf("%d] ", listem->sayac);

        if(listem->sonraki==NULL) break;

        listem = listem->sonraki;
    }
    puts("\n");
}
int main()
{

    char sil[20];
    char adres[20];

    printf("T:"); // T degerini kullanicidan aliyoruz
    scanf("%d",&T);

    printf("L:"); // L degerini kullanicidan aliyoruz
    scanf("%d",&L);

    printf("\n");

    FILE *dosyam; // dosyamizi tanimliyoruz.
    char *tut[50];
    dosyam = fopen("input.txt", "r"); // dosyamizi r yani read,
    okuma modunda cagirarak aciyoruz.

    while(fscanf(dosyam, "%s", tut) != EOF){ //dosya okuma islemi
    tek tek yapilir, her ekleden sonra listelenir.

        strcpy(adres, tut);

        basaDugumEkle(adres);

        listele();

    }

    fclose(dosyam);

    printf("Silin mi ? (EVET / HAYIR) "); //silme islemini
    kullanicuya sorar
    scanf("%s",&sil);

```

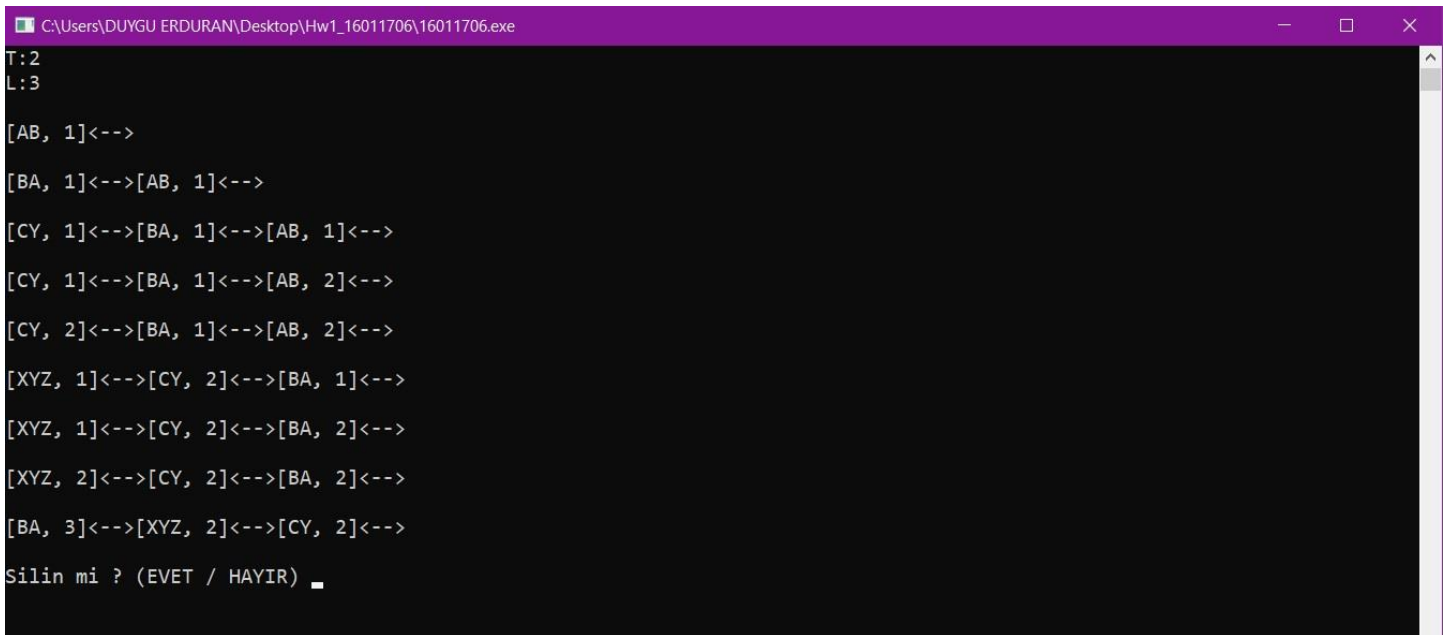
```

        if(strcmp(sil,"EVET") == 0){
            temizle();
            printf("Liste Temizlendi... ");
        }else{
            printf("Listenin Son Hali: ");
            listele();
        }

    return 0;
}

```

Analiz:



```

C:\Users\DUYGU ERDURAN\Desktop\Hw1_16011706\16011706.exe
T:2
L:3

[AB, 1]<-->
[BA, 1]<-->[AB, 1]<-->
[CY, 1]<-->[BA, 1]<-->[AB, 1]<-->
[CY, 1]<-->[BA, 1]<-->[AB, 2]<-->
[CY, 2]<-->[BA, 1]<-->[AB, 2]<-->
[XYZ, 1]<-->[CY, 2]<-->[BA, 1]<-->
[XYZ, 1]<-->[CY, 2]<-->[BA, 2]<-->
[XYZ, 2]<-->[CY, 2]<-->[BA, 2]<-->
[BA, 3]<-->[XYZ, 2]<-->[CY, 2]<-->
Silin mi ? (EVET / HAYIR)

```



```
C:\Users\DUYGU ERDURAN\Desktop\Hw1_16011706\16011706.exe
T:2
L:3

[AB, 1]<-->

[BA, 1]<-->[AB, 1]<-->

[CY, 1]<-->[BA, 1]<-->[AB, 1]<-->

[CY, 1]<-->[BA, 1]<-->[AB, 2]<-->

[CY, 2]<-->[BA, 1]<-->[AB, 2]<-->

[XYZ, 1]<-->[CY, 2]<-->[BA, 1]<-->

[XYZ, 1]<-->[CY, 2]<-->[BA, 2]<-->

[XYZ, 2]<-->[CY, 2]<-->[BA, 2]<-->

[BA, 3]<-->[XYZ, 2]<-->[CY, 2]<-->

Silin mi ? (EVET / HAYIR) EVET
Liste Temizlendi...
-----
Process exited after 101.9 seconds with return value 0
Press any key to continue
```

```
C:\Users\DUYGU ERDURAN\Desktop\Hw1_16011706\16011706.exe
T:2
L:3

[AB, 1]<-->

[BA, 1]<-->[AB, 1]<-->

[CY, 1]<-->[BA, 1]<-->[AB, 1]<-->

[CY, 1]<-->[BA, 1]<-->[AB, 2]<-->

[CY, 2]<-->[BA, 1]<-->[AB, 2]<-->

[XYZ, 1]<-->[CY, 2]<-->[BA, 1]<-->

[XYZ, 1]<-->[CY, 2]<-->[BA, 2]<-->

[XYZ, 2]<-->[CY, 2]<-->[BA, 2]<-->

[BA, 3]<-->[XYZ, 2]<-->[CY, 2]<-->

Silin mi ? (EVET / HAYIR) HAYIR
Listenin Son Hali: [BA, 3]<-->[XYZ, 2]<-->[CY, 2]<-->

-----
Process exited after 4.79 seconds with return value 0
Press any key to continue . . . █
```