

BLM102

PROGRAMLAMA DİLLERİ II

Yrd. Doç. Dr. Baha ŞEN

baha.sen@karabuk.edu.tr

KBUZEM

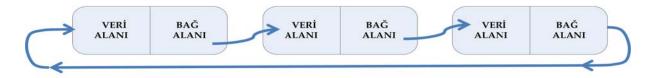
Karabük Üniversitesi Uzaktan Eğitim Araştırma ve Uygulama Merkezi

1. DAİRESEL LİSTELER

Dairesel tek bağlı listeleri doğrusal listelerden ayıran özellik; son bağ alanında NULL yerine listebaşının adresinin saklanmasıdır. Buradan hareketle çift bağlı dairesel listelerde bu duruma ek olarak ilk düğümün ilk bağ alanında da son düğümün adresi saklanmaktadır.

1.1. Tek Bağlı Dairesel Listeler

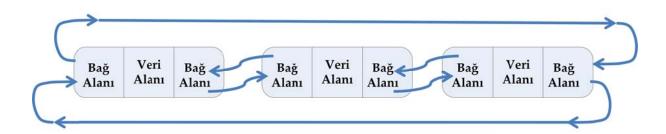
Bu veri yapısı iki ayrı alandan oluşur. Veri alanı ve bağ alanı. Şekilden de anlaşılacağı üzere listenin son düğümü, listenin ilk düğümünü işaret etmektedir.



Şekil 1: Tek Bağlı Dairesel Listelerin Genel Yapısı

1.2. Çift Bağlı Dairesel Listeler

Bu veri yapısının tek bağlı dairesel listeden farkı, önceki düğümün de adresini saklayan bir işaretçi bulunmasıdır. Bu sayede listede iki yönlü ilerleme yeteneği sağlanmış olur.



Şekil 2: İki bağlı dairesel listenin yapısı

Örnek:

```
#include <stdlib.h>
#include <string.h>
#include <stdio.h>
#include <conio.h>
struct listyapi
  struct listyapi *onceki;
  char adi[21];//veri alani
  struct listyapi *sonraki;
};
typedef struct listyapi listnode; //artık listyapi
yerine listnode kullanılacak
listnode *listebasi; /* Herzaman listenin basini
gosteren bir işaretçi */
void basaekle(char *s)
  listnode *yeni dugum;
  yeni dugum = (listnode*) malloc(sizeof(listnode));
/* yeni kayida yer al */
  strcpy(yeni dugum->adi, s);
                                         /*bilgiyi
yeni kayida yaz */
  if (listebasi==NULL)
    {
      listebasi=yeni dugum;
      yeni dugum->onceki=yeni dugum;
      yeni dugum->sonraki=yeni dugum;
    }
  else
      listnode *son node=listebasi->onceki;
      yeni dugum->sonraki = listebasi;
      yeni dugum->onceki = son node;
                         = yeni dugum;
      listebasi->onceki
      son node->sonraki = yeni dugum;
      listebasi=yeni dugum;
}
void sonaekle(char *s)
  listnode *yeni dugum;
  yeni dugum = (listnode*) malloc(sizeof(listnode));
/* yeni kayida yer al */
```

```
strcpy(yeni dugum->adi, s);
                                          /*bilgiyi
yeni kayida yaz */
  if (listebasi==NULL)
      listebasi=yeni dugum;
      yeni dugum->onceki=yeni dugum;
      yeni dugum->sonraki=yeni dugum;
    }
  else
    {
      listnode *son node=listebasi->onceki;
      yeni dugum->sonraki = listebasi;
      yeni dugum->onceki = son node;
      listebasi->onceki = yeni_dugum;
      son node->sonraki = yeni dugum;
    }
void listlist(void)
  listnode *aktif node = listebasi;
  while (aktif node != NULL)
    {
      printf("%s ",aktif node->adi);
      aktif node = aktif node->sonraki;
      if(aktif node->onceki == listebasi->onceki)
break;
  printf("\n");
int main()
  char
           sec;
  char
          *s;
  do
      system("cls");
      listlist();
      printf("\n\n1 - Basa Ekle\n2 - Sona Ekle\n3 -
Son\n\nSec :");
      sec = getche();
      switch (sec)
        case '1':
          printf("\nAdi :");
```

```
gets(s);
basaekle(s);
break;
case '2':
    printf("\nAdi :");
    gets(s);
    sonaekle(s);
    break;
case '3':
    return(0);
    break;
}
while (1);
}
```