4.Ders : C Bellek Ayırma

C Dinamik Bellek Ayırma

Bu eğitici yazıda, standart kütüphane işlevlerini kullanarak C programınıza bellek ayırmayı öğreneceksiniz: **malloc()**, **calloc()**, **free()** ve **realloc()**.

Bildiğiniz gibi, bir dizi sabit sayıda değerden oluşan bir koleksiyondur. Bir dizinin boyutu bildirildiğinde, değiştiremezsiniz.

Bazen bildirdiğiniz dizinin boyutu yetersiz olabilir. Bu sorunu çözmek için çalışma zamanı sırasında belleği manuel olarak ayırabilirsiniz. Bu, C programlamada dinamik bellek tahsisi (Dinamik Bellek Ayırma) olarak bilinir.

Belleği dinamik olarak tahsis etmek için kütüphane fonksiyonları malloc (), calloc (), realloc () ve free () kullanılır. Bu işlevler başlık dosyasında tanımlanmıştır.

C malloc()

“**Malloc**” ismi **hafıza tahsisini** gösterir.

Malloc () işlevi, belirtilen bayt sayısında bir bellek bloğu ayırır. Ve herhangi bir formdaki işaretçilere atılabilecek bir

boşluk göstergesi döndürür.

ptr = (cast-type*) malloc(byte-size)

Örneğin :

ptr = (int*) malloc(100 * sizeof(float));

Bu ifade 400 bayt hafıza ayırır. Çünkü int boyutu 4 bayttır. Ve, ptr işaretçisi, ayrılan bellekteki ilk baytın adresini tutar.

Bellek tahsis edilemezse, ifade bir NULL işaretçisiyle sonuçlanır.

C calloc()

“**Calloc**” adı **bitişik tahsisi** ifade eder.

Malloc () işlevi, tek bir bellek bloğunu ayırır. Oysa calloc () birden fazla bellek bloğunu ayırır ve sıfırla başlatır.

ptr = (cast-type*)calloc(n, element-size);

Örneğin :

```
ptr = (float*) calloc(25, sizeof(float));
```

C free()

Calloc () veya malloc () ile oluşturulan dinamik olarak ayrılmış bellek kendi başına serbest bırakılmaz. Boşluk bırakmak için açıkça free () işlevini kullanmalısınız.

```
free(ptr);
```

Bu ifade, ptr ile gösterilen bellekte ayrılan alanı boşaltır.

Örnek 1: malloc () ve free()

```
int main()
{
    int n, i, *ptr, sum = 0;
    printf("Öğe sayısını girin: ");
    scanf("%d", &n);
    ptr = (int*) malloc(n * sizeof(int));

    // if memory cannot be allocated
    if(ptr == NULL)
    {
        printf("Hata! hafıza tahsis edilmedi.");
        exit(0);
    }
    printf("Öğeleri gir: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
```

```
    printf("Toplam = %d", sum);
    // deallocating the memory
    free(ptr);
    return 0;
}
```

Burada, n sayısını int için dinamik olarak ayırdık.

Örnek 2: calloc() ve free()

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int n, i, *ptr, sum = 0;
    printf("Öğe sayısını girin: ");
    scanf("%d", &n);
    ptr = (int*) calloc(n, sizeof(int));
    if(ptr == NULL)
    {
        printf("Hata! hafıza tahsis edilmedi.");
        exit(0);
    }
    printf("Öğeleri gir: ");
    for(i = 0; i < n; ++i)
    {
        scanf("%d", ptr + i);
        sum += *(ptr + i);
    }
    printf("Toplam = %d", sum);
    free(ptr);
    return 0;
}
```

C realloc()

Dinamik olarak atanmış hafıza gerektiğinde yetersiz veya daha fazla ise, realloc () işlevini kullanarak önceden ayrılan hafızanın boyutunu değiştirebilirsiniz.

```
ptr = realloc(ptr, x);
```

Örnek 3: realloc ()

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int *ptr, i , n1, n2;
    printf("Boyut giriniz: ");
    scanf("%d", &n1);
    ptr = (int*) malloc(n1 * sizeof(int));
    printf("Önceden ayrılmış hafızanın adresleri: ");
    for(i = 0; i < n1; ++i)
         printf("%u\n",ptr + i);
    printf("\nYeni Boyut giriniz: ");
    scanf("%d", &n2);
    // rellocating the memory
    ptr = realloc(ptr, n2 * sizeof(int));
    printf("Yeni ayrılan hafızanın adresleri: ");
    for(i = 0; i < n2; ++i)
         printf("%u\n", ptr + i);
    return 0;
}
```

Programı çalıştırdığınızda, çıktı:

```
Boyut giriniz: 2
Önceden ayrılmış hafızanın adresleri:26855472
26855476

Yeni Boyut giriniz: 4
Yeni ayrılan hafızanın adresleri:26855472
26855476
26855480
26855484
```