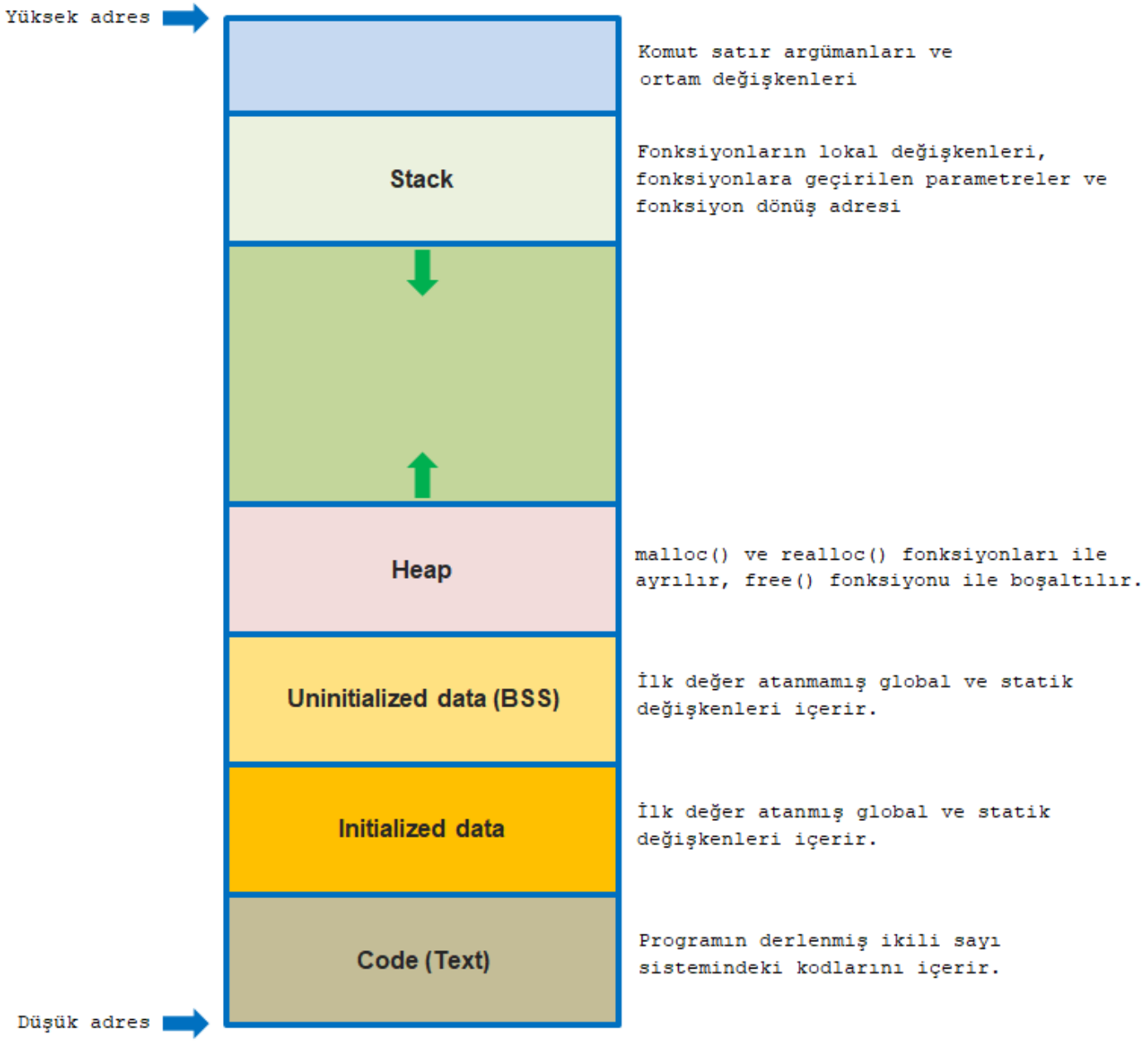
C Bellek Sistemi

Bir C programının çalıştığı bilgisayarın belleğinde aşağıdaki bölümlerden oluşur.

- Kod (Metin) bölümü (Code segment)
- İlk değer atanmış veri bölümü (Initialized data segment)
- İlk değer atanmamış veri bölümü (Uninitialized data segment)
- Stack
- Heap



Kod (Metin) bölümü (Code segment)

Code segment bir C programının çalıştırılabilir talimatlarını içeren bir bellek bölgesidir.

Code segment derlenmiş programın ikili sayı sistemine göre oluşturulmuş değerlerini içerir.

Yanlışlıkla değiştirilmemesi için, code segment sadece okunur olarak düzenlenmiştir.

Code segment, heap ve stack bellek bölümlerinin aşırı kullanımlarının kendi bellek bölgesi üzerine yazma sorununu engellemek için heap ve stack bölgelerinin altında yer alır.

Fonksiyonlar ve karakter dizisi sabitleri bu bölümde yer alır.

İlk değer atanmış veri bölümü (Initialized data segment)

İlk değer atanmış veri bölümü, programcı tarafından ilk değer atanmış olan global ve static değişkenleri içeren, bir programın sanal adres boşluğunu bir parçasıdır.

İlk değer atanmış veri bölümü, değişkenlerin değerleri programın çalışma zamanında değiştirilebileceğinden, sadece okunur nitelikte değildir.

Bu bölüm ileride sadece okunur ve okunur-yazılır olarak iki bölüme ayrılabilir.

İlk değer atanmış veri bölümü, programın kaynak kodunda bir ilk değer atanmış olan global ve statik değişkenleri içerir.

İlk değer atanmamış veri bölümü (Uninitialized data segment)

İlk değer atanmamış veri bölümü, BSS bölümü olarak ta adlandırılır. Bu bölümdeki verilere, program çalışmaya başlamadan önce 0 değeri verilir.

İlk değer atanmamış veri bölümü, veri bölümünün sonundan başlar ve programın kaynak kodunda bir ilk değer verilmemiş ve otomatik olarak 0 değeri atanmış bütün global ve statik değişkenleri içerir.

Stack

Stack alanı geleneksel olarak heap alanına bitişiktir ve ters yöne doğru büyür; stack işaretçisi heap işaretçisi ile karşılaşması boş belleğin tükendiğini gösterir.

Belleğin üst bölümlerinde yer alan stack alanı, LIFO (Last In First Out - Son Giren İlk Çıkar) yapısına sahip program belleğini içerir. Standart PC x86 bilgisayar mimarisinde stack adresin sıfır değerine doğru büyür; diğer bazı mimarilerde ise ters yönde büyür. Bir “stack işaretçisi” kaydı stack bölümünün üstünü izler; stack içine her değer yerleştirilmesinde ayarlanır. Bir fonksiyon çağrısı için stack bölümüne aktarılan değer kümesine “stack çerçevesi” denir. Bir stack çerçevesi en az bir dönüş adresinden oluşur.

Stack bölümünde, bir fonksiyonun her çağrılmasında kaydedilen bilgilerle birlikte otomatik değişkenler saklanır. Bir fonksiyon her çağrıldığında, geri dönülecek yerin adresi ve fonksiyonu çağıranın ortamı ile ilgili bazı bilgisayar register değerleri gibi bilgiler stack bölümüne kaydedilir. Yeni çağrılan fonksiyon daha sonra otomatik ve geçici değişkenleri için stack içinde yer ayırır.

Stack bölümünde, fonksiyonun lokal değişkenleri (işaretçi değişkenler dahil), fonksiyona geçirilen parametreler ve fonksiyon dönüş adresi yer alır.

Heap

Heap dinamik bellek tahsisinin yapıldığı bellek bölümüdür.

Heap alanı BSS bölümünün sonunda başlar ve oradan daha büyük adreslere doğru büyür. Heap alanı, malloc(), realloc() ve free() fonksiyonları tarafından yönetilir.

Aşağıdaki örnekte, farklı bellek bölümlerine yapılan atamaları göstermek üzere, biri ilk değer atanmış diğeri atanmamış olmak üzere 2 adet global değişken, biri ilk değer atanmış diğeri atanmamış olmak üzere 2 adet static değişken, dinamik bellek tahsisi yapılan bir karakter işaretçisi ve bir adet yerel değişken oluşturularak, önce değişken değerleri sonra bellek adres değerleri ekrana yazılmıştır.

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

int g_uninit;    /* İlk değer atanmamış global değişken BSS bölgesinde yer alır. */
int g_init = 7;  /* İlk değer atanmış global değişken DS bölgesinde yer alır. */

int main()
{
```

```
    static int id_uninit;                       /* İlk değer atanmamış static değişken BS
    static int id_init = 21;                    /* İlk değer atanmış static değişken DS b
    char *pdizi = malloc (sizeof(char) * 14);   /* Dinamik olarak tahsis edilmiş Heap böl
    int id=35;                                  /* Stack bölgesine yüklenen yerel değişke

    strcpy(pdizi, "Bellek düzeni");

    printf("g_uninit: %d g_init: %d id_uninit: %d id_init: %d pdizi: %s id: %d\n", g_uni
    printf("g_uninit: %p g_init: %p id_uninit: %p id_init: %p pdizi: %p id: %p", &g_unin

    free(pdizi);

    return 0;
}
```

Yukarıdaki örnekte, program ekrana aşağıdaki satırları yazar:

```
g_uninit: 0 g_init: 7 id_uninit: 0 id_init: 21 pdizi: Bellek düzeni id: 35
g_uninit: 00405054 g_init: 00402000 id_uninit: 00405008 id_init: 00402004 pdizi: 00DE1530 id: 0060FF08
```