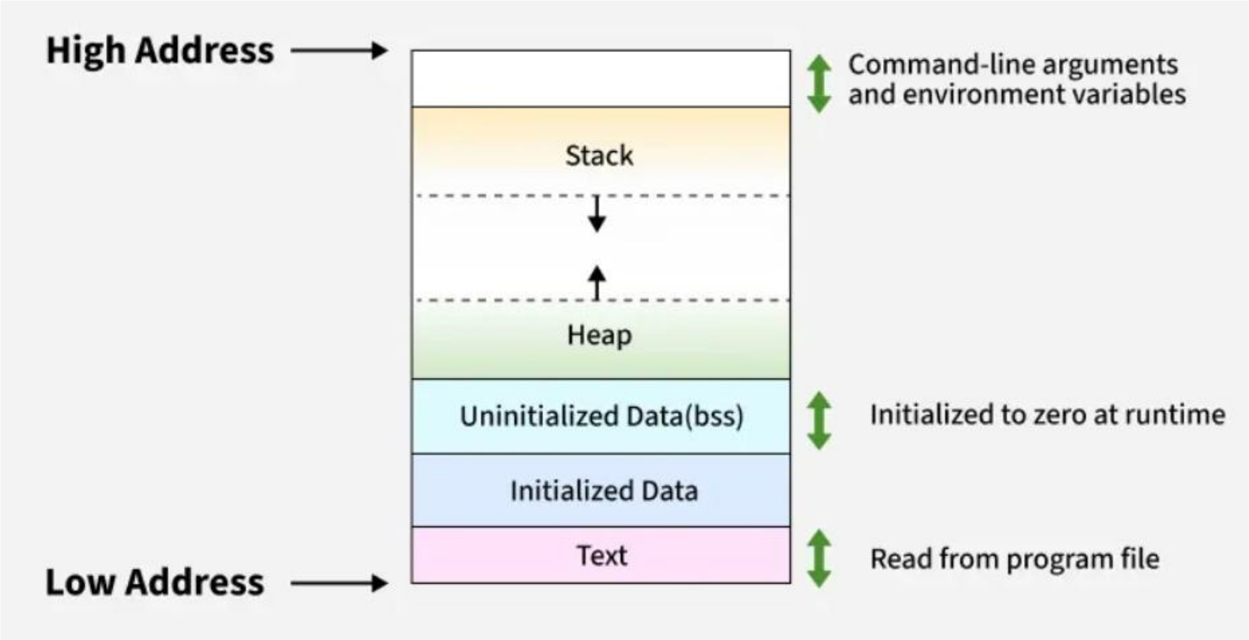

Text/Code segment (instructions)

Data segment

- Initialized global/static variables (.data)
- Uninitialized global/static variables (.bss)

Heap (dynamic memory allocation)

Stack (local variables, function calls)

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

int g_init = 10;   // .data
int g_uninit;      // .bss

int main() {
    int local = 20;                // stack
    int *heap = malloc(sizeof(int)); // heap
    *heap = 30;

    printf("main() function      : %p (text)\n", (void*)&main);
    printf("g_init (data)        : %p\n", (void*)&g_init);
    printf("g_uninit (bss)       : %p\n", (void*)&g_uninit);
    printf("local (stack)        : %p\n", (void*)&local);
    printf("heap (malloc)        : %p\n", (void*)heap);

    free(heap);
    return 0;
}
```

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

// ===== Globals =====
int g_init = 10;   // .data segment
int g_uninit;      // .bss segment

int main() {
    int local = 20;                  // stack
    int *heap = malloc(sizeof(int));  // heap
    *heap = 30;

    printf("Address of main()   : %p (Text/Code)\n", (void*)&main);
    printf("Address of g_init   : %p (.data)\n", (void*)&g_init);
    printf("Address of g_uninit : %p (.bss)\n", (void*)&g_uninit);
    printf("Address of local    : %p (Stack)\n", (void*)&local);
    printf("Address of heap var : %p (Heap)\n", (void*)heap);

    free(heap);
    return 0;
}
```

Text/Code Segment

- Stores compiled **instructions** of program.
- Example: main() function address.

Data Segment (.data)

- Stores **initialized global/static variables**.
- Example: g_init = 10;

BSS Segment (.bss)

- Stores **uninitialized global/static variables** (default = 0).
- Example: g_uninit;

Heap

- Used for **dynamic memory allocation** (malloc, calloc, realloc).
- Grows **upward**.
- Example: heap = malloc(sizeof(int));

Stack

- Stores **local variables, function call frames**.
- Grows **downward**.
- Example: local.