

System-Level Programming

34 Organization of Memory

Peter Wägemann

Lehrstuhl für Informatik 4
Systemsoftware

Friedrich-Alexander-Universität
Erlangen-Nürnberg (FAU)

Summer Term 2025

<http://sys.cs.fau.de/lehre/ss25>



Organization of Memory

```
int a;           // a: global, uninitialized
int b = 1;       // b: global, initialized
const int c = 2; // c: global, const

void main(void) {
    static int s = 3; // s: local, static, initialized
    int x, y;         // x: local, auto; y: local, auto
    char *p = malloc(100); // p: local, auto; *p: heap (100 byte)
}
```

Where does the memory for these variables come from?



Organization of Memory

```
int a;           // a: global, uninitialized
int b = 1;       // b: global, initialized
const int c = 2; // c: global, const

void main(void) {
    static int s = 3; // s: local, static, initialized
    int x, y;         // x: local, auto; y: local, auto
    char *p = malloc(100); // p: local, auto; *p: heap (100 byte)
}
```

Where does the memory for these variables come from?

■ Static allocation – allocation during compilation / linking

- Concerns all global/static variables and the code itself
- Allocation by getting placed into a **section**

→ 12-5

.text – contains program code
.bss – contains all variables initialized with 0
.data – contains all variables initialized with other values
.rodata – contains all constant variables

main()
a
b,s
c



Organization of Memory

```
int a;           // a: global, uninitialized
int b = 1;       // b: global, initialized
const int c = 2; // c: global, const

void main(void) {
    static int s = 3; // s: local, static, initialized
    int x, y;         // x: local, auto; y: local, auto
    char *p = malloc(100); // p: local, auto; *p: heap (100 byte)
}
```

Where does the memory for these variables come from?

■ Static allocation – allocation during compilation / linking

- Concerns all global/static variables and the code itself
- Allocation by getting placed into a [section](#)

→ 12-5

[.text](#) – contains program code
[.bss](#) – contains all variables initialized with 0
[.data](#) – contains all variables initialized with other values
[.rodata](#) – contains all constant variables

main()
a
b,s
c

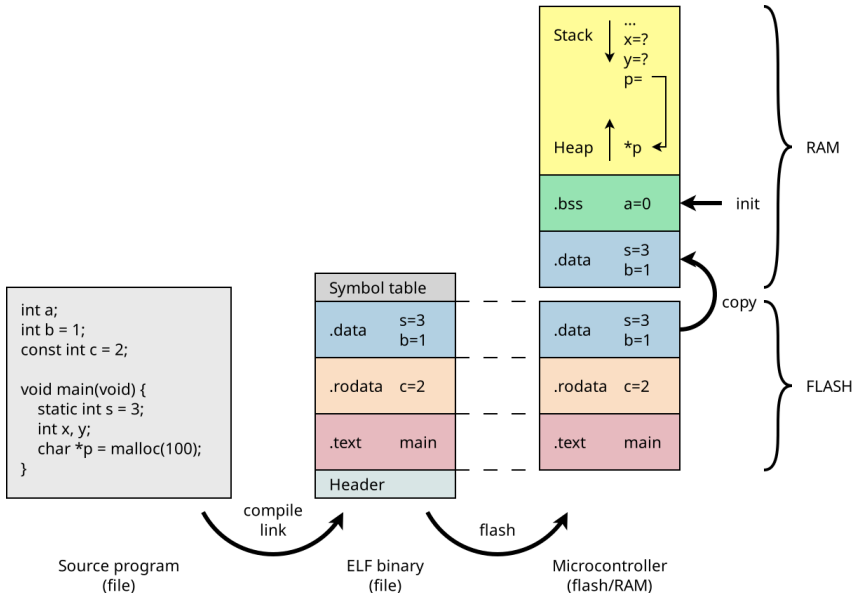
■ Dynamic allocation – allocated during runtime

- Concerns all local automatic variables and explicitly allocated memory

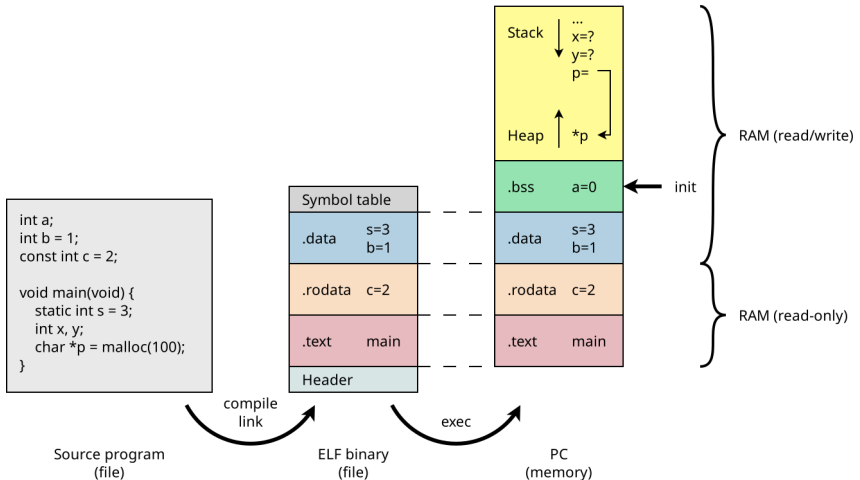
[Stack](#) – contains all auto variables that are [currently alive](#) x,y,p
[Heap](#) – contains with `malloc()` explicitly allocated memory areas *p



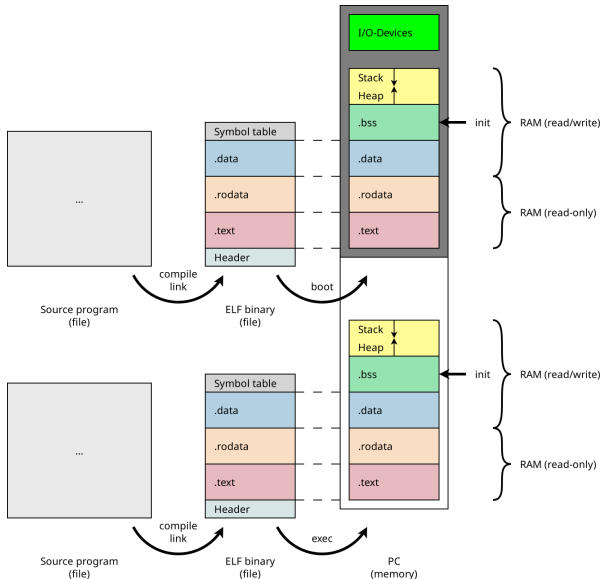
Organization of Memory on a μ C



Organization of Memory with an OS



Organization of Memory with an OS (continued)



Organization of Memory with an OS (continued)

