Základy programovania 25ZS
Štruktúry

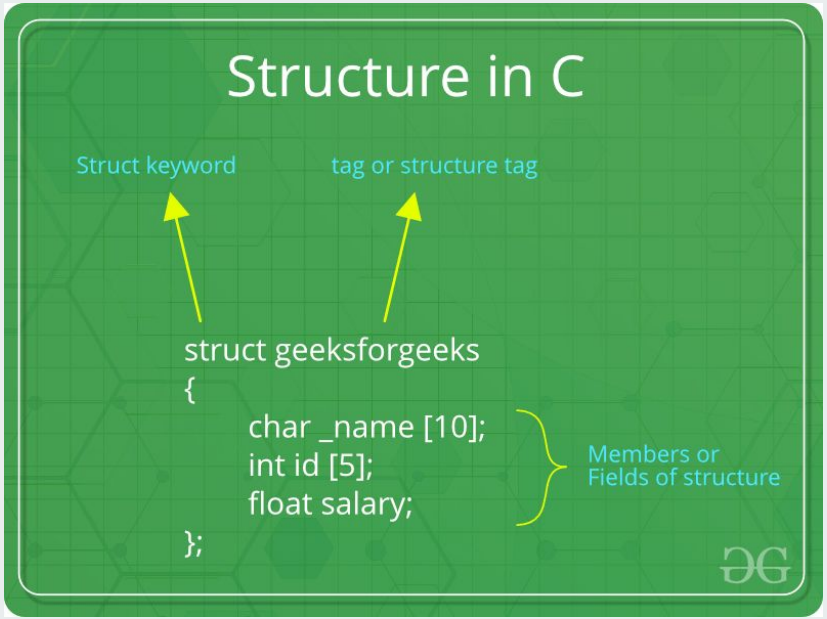
Prednášajúci: prof. Gabriel Juhás
Cvičiaci: Milan Mladoniczky

Štruktúra

- Programátorom definovaný zložený dátový typ.
- Zložená z viacerých atribútov - členov, ktorý každý má definované meno a dátový typ.
- Každý člen môže byť rôzneho dátového typu.
- Hodnoty sú uložené v pamäti sekvencií za sebou podobne ako pri poli v poradí v akom sú definované v štruktúre.

```
struct Osoba {

    char meno[15];

    char *priezvisko;

    int vek;

};
```

Štruktúra




```
struct Osoba {
        char meno[15];
        char *priezvisko;
        int vek;
};
```

- Definícia štruktúry sa nazýva aj šablóna štruktúry (angl. structure template). Nezaberá žiadne miesto v pamäti.
- Premennú dátového typu štruktúry je možné definovať za definíciou alebo za označením štruktúry.

```
struct Osoba {
    char meno[15];
    char *priezvisko;
    int vek;
} milan, martin, …;
```

- Definícia štruktúry sa nazýva aj šablóna štruktúry (angl. structure template). Nezaberá žiadne miesto v pamäti.
- Premennú dátového typu štruktúry je možné definovať za definíciou alebo za označením štruktúry.

```
struct Osoba milan, martin, … {
    char meno[15];
    char *priezvisko;
    int vek;
};
```

- Definícia štruktúry sa nazýva aj šablóna štruktúry (angl. structure template). Nezaberá žiadne miesto v pamäti.
- Premennú dátového typu štruktúry je možné definovať za definíciou alebo za označením štruktúry.
- Východzie hodnoty členov štruktúry nie je možné inicializovať priamo v definícií. Program nie je možné skompilovať.

```
struct Osoba {

      char meno[15] = "Meno?";

      char *priezvisko;

      int vek = 0;

};
```

- Pristupovať k členom štruktúry je možné cez operátor ‘ . ’
- Členy nemajú **nie sú** automaticky inicializované na NULL hodnoty.

```
struct Osoba {
     char meno[15];
     char *priezvisko;
     int vek;
} milan, martin, …;

milan.meno;
milan.vek = 30;
```

- Pristupovať k členom štruktúry je možné cez operátor ‘ . ’
- Členy nemajú **nie sú** automaticky inicializované na NULL hodnoty.
- Hodnoty členov štruktúry je možné inicializovať cez tzv. inicializačný zoznam (angl. initializer list)
    - hodnoty sú definované v poradí ako v štruktúre
    - pomenované členy majú priradené hodnoty

```
struct Osoba {

	char meno[15];

	char *priezvisko;

	int vek;

};

struct Osoba milan = { “Milan”, “Prvý”, 30 };
struct Osoba martin = { .meno = “Martin”, .vek = 20 };
```

typedef

- Kľúčový slovom typedef je možné definovať alias pre existujúci dátový typ.
- Často sa používa pri použití štruktúr pre kratšiu definíciu štruktúry a následne jej použitie.

```
typedef struct {

	char meno[15];

	char *priezvisko;

	int vek;

} Osoba;

Osoba milan = { “Milan”, “Prvý”, 30 };


printf(“meno: %s\n”, milan.meno);
printf(“vek: %d\n“, milan.vek);
```

Pointer štruktúry

- Pointer štruktúry je definovaný ako adresa pamäte začiatku rozsahu alokovaného pre celú štruktúru.
- K členom štruktúry je možné pristupovať cez pointer pomocou operátor ‘ -> ‘ (tzv. indirect access)

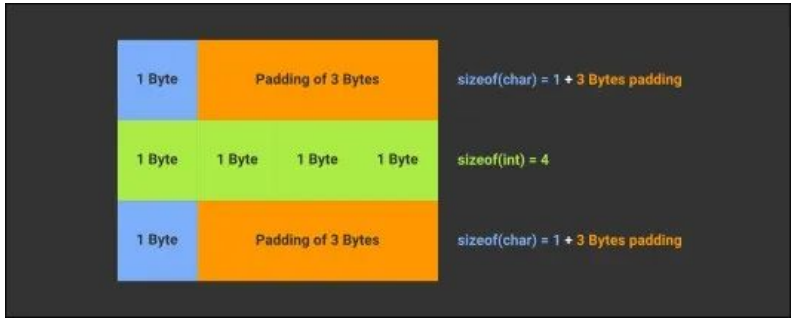
```
typedef struct {

      char meno[15];

      char *priezvisko;

      int vek;

} Osoba;

Osoba milan = { “Milan”, “Prvý”, 30 };
Osoba *ptr = &milan;


printf(“meno: %s\n”, ptr->meno);
printf(“vek: %d\n“, ptr->vek);
```

Praktická ukážka –>

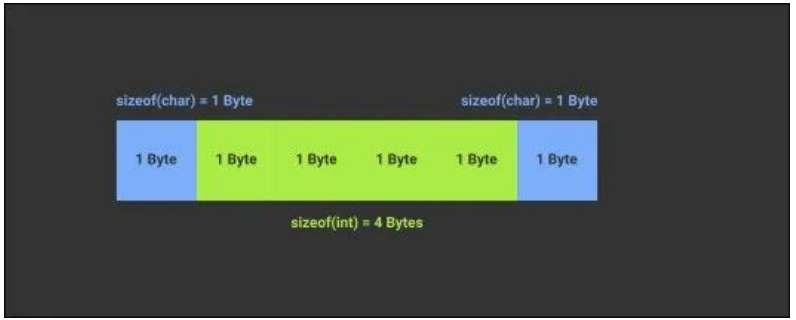
Veľkosť štruktúry

- Na prvý pohľad sa môže zdať, že veľkosť štruktúry je súčet veľkostí jej členov. V skutočnosti to tak nemusí byť nakoľko je pri alokácii použitá technika tzv. Structure padding.
- Structure padding pridá viacero prázdnych bytov pre lepšie usporiadanie členov v blokoch pamäte tak aby štruktúra bola načítaná za čo najmenej procesorových cyklov (t.j. prístupov procesoru do pamäte).

Veľkosť štruktúry



Rozbalená (unpacked) štruktúra



Zabalená (packed) štruktúra

Limitácie štruktúr

- **Vyššia náročnosť programu na pamäť** - spôsobená najmä technikou structure padding
- **Neumožňuje skrývanie dát (privátny člen)** - Všetky dáta/členy v štruktúre sú prístupným všetkým a všade.
- **Neumožňuje funkcie v štruktúre** - V štruktúre nie je možné definovať funkcie ako členy, napr. pre lepšiu enkapsuláciu logiky programu.
- **Žiaden konštruktor a deštruktor** - Nie je možné definovať funkciu, resp. spôsob, akým sa má štruktúra inicializovať.

https://zapr.interes.group/exercises/exercise-8

Doplňujúci materiál

- https://medium.com/@hatronix/mastering-structs-in-c-an-in-depth-guide-4a524af06fd4
- https://www.programiz.com/c-programming/c-structures
- https://hatchjs.com/how-to-free-a-struct-in-c/