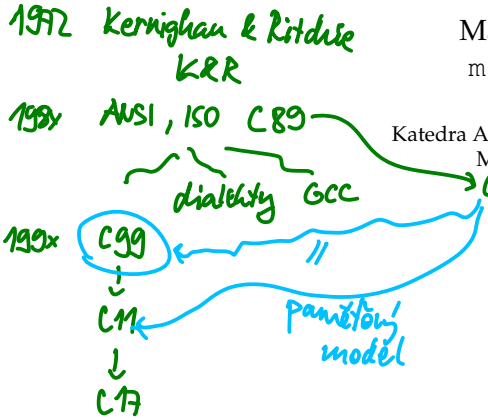
Medvědův průvodce po Céčku



Martin Mareš
mj@ucw.cz

Katedra Aplikované Matematiky
MFF UK Praha

2019

Typový systém: Základní typy

Základní typy: (norma, GCC/amd64)

- Celočíselné typy:

	norma	GCC AMD64
`_Bool (bool)`	1 bit	1 b
`char`, `(un)signed char`	8 bitů, ∀ znak	8 b
`(unsigned) short int`	≥ 16 bitů	16 b
`(unsigned) int`	≥ 16 bitů	32 b
`(unsigned) long int`	≥ 32 bitů	64 b
`(unsigned) long long int`	≥ 64 bitů	64 b

  - bit-fields

- Floating-point typy:

`float`	≥ 6 číslic, ≥ 10^{37}	32 b
`double`	≥ 10 číslic, ≥ 10^{37}	64 b
`long double`	≥ 10 číslic, ≥ 10^{37}	80 b

  - jejich komplexní varianty (`_Complex`)

- Výčty (`enum`), chovají se celočíselně
- Typ `void`

Typový systém: Odvozené typy

Odvozené typy:

- Pole
- Struktury
- Uniony union
- Funkce
- Ukazatele (speciální: `void *`)


```
struct bod {
    double x,y,z;
    int barva;
};
```

```
struct bod A;
int *a;
```

malloc(velikost) -> void *

Částečné typy:

- `struct who_knows_what *`
- `int f()` vs. `int f(void)`
- `int x[]`

Typový systém: Reprezentace hodnot

Reprezentace:

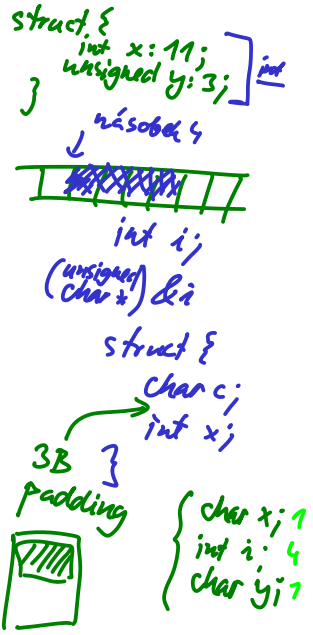
- Hodnoty jsou složeny z bytů (`char`ů), výjimka: bitová pole
- K reprezentaci lze přistupovat pomocí `unsigned char`ů
- Může obsahovat padding a vyžadovat zarovnání
- Některé reprezentace mohou být nekorektní (trap)

Celá čísla:

- `unsigned` čistě binární, může mít padding (třeba paritu)
- `unsigned char` nemá padding
- `signed` má znaménkový bit, bity hodnoty a padding
- Hodnota z `signed` ∩ `unsigned` vypadá v obou stejně

Složené typy:

- Struktury: padding mezi prvky, prefixová vlastnost
- Pole: bez paddingu



Literály

- Integery: `12345`, `0xdeadbeef`, `0177`
- Typované integery: `12345U`, `12345L`, `12345ULL`
- Floaty: `1.5`, `1..1`, `1e20`, `1e-20`
- Typované floaty: `1.5D`, `1.5L`
- Hexadecimální floaty: `0x1.fffep10`
- Znaky (int): `'x'`, `'\n'`, `'\033'`, `'\x1b'`
- Řetězce (const char []): `"brekekex\n"`, `"str" "ing"`
- “Široké” znaky (wchar_t): `L'ž'`, `L"žluťoučký kůň"`
- Unicode: `L'\u2302'`, `L'\U00002302'` (též v identif.)
- Složené literály: `(char []){ 1, 2, 3 }`
  (pozor na to, kde jsou uložené)
- Pojmenovaně: `(struct point){ .x=1, .y=2 }`

Operátory a jejich priority

1. (*podvýraz*) *ident* *literál* `_Generic`
2. [*index*] (*volání*) . *prvek* -> *prvek* `++` `--` (postfixové)
3. `++` `--` `sizeof` `_Alignof` `&` `*` `+` `-` `~` `!` (*typ*) (prefixové)
4. `*` `/` `%`
5. `+` `-`
6. `<<` `>>`
7. `<` `>` `<=` `>=`
8. `==` `!=`
9. `&`
10. `^`
11. `|`
12. `&&`
13. `||`
14. `?:`
15. `=` `+=` `*=` ... (asoc. zprava)
16. `,`