

C a C++

V předchozím ročníku jste probírali PHP, což je skriptovací jazyk. Skriptovací jazyk je *interpretovaný*, tedy ke spuštění programu v něm napsaném je zapotřebí jiný program (*interpret*), který jej vykonává příkaz po příkazu. Skriptovací jazyk je dále *volně typovaný*, tj. typ proměnných je určen *dynamicky* (za běhu) a může se měnit (v PHP jsou všechny proměnné prostě `$`).

C není ani volně typovaný, ani interpretovaný. Při práci s pamětí musí být znám její typ. Program je *kompilovaný* do *objektového kódu* konkrétní *architektury* (HW+OS). Objektový kód připomíná strojový kód, ale adresy jsou v něm uvedeny jako relativní¹. Objektový kód je binární kód zabalený do kontejneru (PE, ELF), který spolu s ním tvoří knihovnu nebo spustitelný kód. Spustitelný kód není přenositelný (program pro Windows nelze přímo spustit v Linuxu a naopak, program pro 64bitovou architekturu nelze spustit na 32bitovém stroji, některé programy jsou optimalizovány přímo pro konkrétní typ procesoru).

Pojmy

Ujistěte se, že dokážete odpovědět na následující otázky:

- Jak se liší interpretovaný kód od kompilovaného?
- Co je to skriptovací jazyk?
- Co se z hlediska kompilace myslí slovem *architektura*?

Tvorba spustitelného kódu jazyka C

Tvorba spustitelného kódu v C probíhá několika fázemi:

1. Zdrojový kód upraví preprocesor, který z něj odstraní komentáře a vykoná všechny příkazy začínající na `#`.
2. Takto upravený kód *zkompiluje* kompilátor do objektového kódu.
3. *Linker* sestaví spustitelný kód z několika objektových kódů.

Těto posloupnost nástrojů (*toolchain*) obsahuje ještě některé další (např. pro tvorbu knihoven), které nemusí být vždy použity.

Pojmy

Ujistěte se, že rozumíte:

- Co je to linker a kompilátor?
- Co je to *toolchain*?

¹ V době spuštění programu OS přidělí HW prostředky, mj. segmenty paměti určené offsetem (začátkem) a velikostí (počtem stránek). Relativní adresa se přičte k offsetu, čímž se získá fyzická adresa. OS může její načítání optimalizovat použitím cache paměti, tuto optimalizaci lze však v C u konkrétní proměnné zakázat klíčovým slovem `volatile`.

První *Hello World* program by v C mohl vypadat nějak takto:

```
#include <stdio.h>

void main() {
    puts("Hello world");
}
```

Preprocesor zde vykoná příkaz `include`, který připojí metodou Ctrl+C, Ctrl+V obsah hlavičkového souboru uvedený ve špičatých závorkách, `stdio.h` obsahuje mimo jiné deklaraci funkce `puts`, která vytiskne text na konzoli.

Kompilátor pak vytvoří objektový kód, který obsahuje funkci `main`: tato funkce je vstupním bodem spustitelného kódu (OS ji při spuštění spustí).

Linker pak program propojí s objektovým kódem knihovny obsahující definici funkce `puts` (je to systémová knihovna, kterou linker připojuje automaticky) a výsledkem pod Windows je `exe` soubor, který můžeme spustit jako samostatný program.

Pojmy

Všimněte si, že v jedné části kódu je *deklarace* (soubor `stdio.h`) a v druhé části *definice* (systémová knihovna) funkce `puts`. Deklarace umožňují linkeru správně napojit kód.

- Jaký je rozdíl mezi deklarací a definicí?

Deklarace je uvedení typu (zde hlavička funkce, tj. její návratová hodnota a typ jejích argumentů). Definice je obsah (zde tělo funkce).