

Jazyk C++

Objektově orientované programování

Den první

ENGETO s.r.o.

Prezentované materiály jsou dostupné na adrese
https://github.com/ChaoticRoman/kurz_cpp

Cíle kurzu

- Základy programování v C++
- Porozumění modernímu **objektově orientovanému programování** v C++
- **Praktické zkušenosti** s OOP v C++
- **Best practices** současného profesionální softwarového inženýrství

Předpoklady

- Chuť se učit
- Počítač při ruce
- Základní znalosti IT a programování

Osnova kurzu

Den první

Historie C a C++

C vs. C++

Praktická část: prostředí

Praktická část: git, GitHub

Základy: proměnné, flow control, streams

Templates, STL: kontejnery, iterátory

Objekty, dědičnost a polymorfismus

Praktická část: zpracování paketů

Den druhý

Verze C++

C++ Core guidelines

Best practices: dokumentace, testování, issue tracking, CI/CD

Přetěžování operátorů

Exceptions

Moderní C++: smart pointers, lambdy, STL algoritmy a funkce

Praktická část: Issue tracker

Den třetí

Standardy

Paralelizace

Přehled užitečných knihoven

Boost

Qt: GUI

Qt: Síťování

Praktická část: chat

Počítání vs. programování

Výpočet: Transformace nějakých **vstupních dat** do nějakého **výstupu**

Vstupní data → **Transformace** → **Výstupní data**

Příklady:

Vstupní data	Transformace	Výstupní data
1, 2, 3	Suma	6
Seznam nákupů a prodejů	Postup pro vyplnění daňového přiznání	Vyplněné daňové přiznání
"Siri, najdi nejbližší pizzerii!"	Rozpoznání hlasu, vyhledání v mapových podkladech	Pizzerie vyznačená na mapě

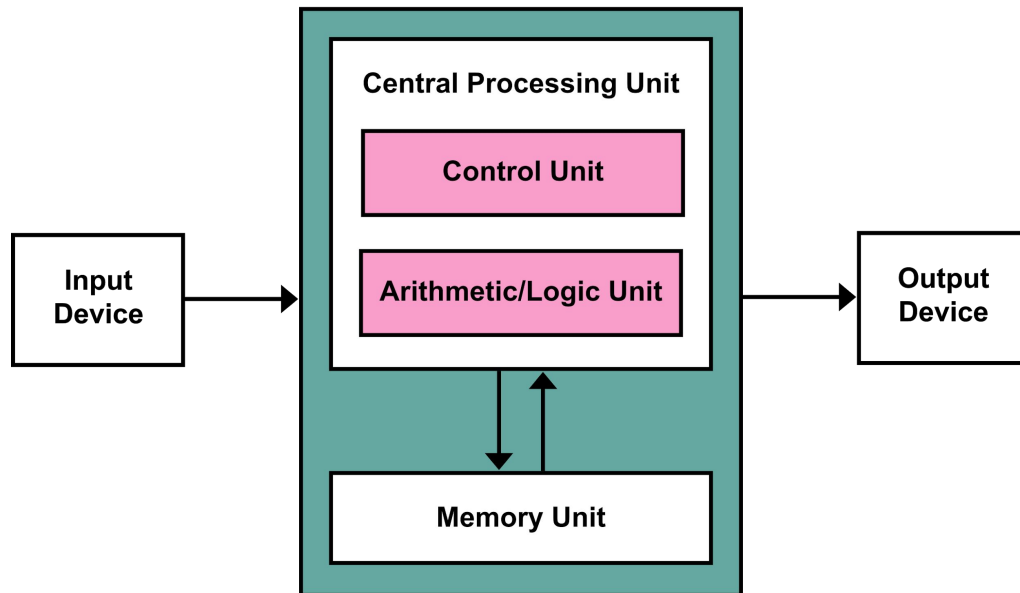
Počítání vs. programování

Programování: Popsání transformace dat pomocí programu

Program: Popis transformace v jazyce, kterému nějaký stroj rozumí

Vstupní data + Program → Programovatelný výpočetní stroj → Výstupní data

Jak funguje počítač



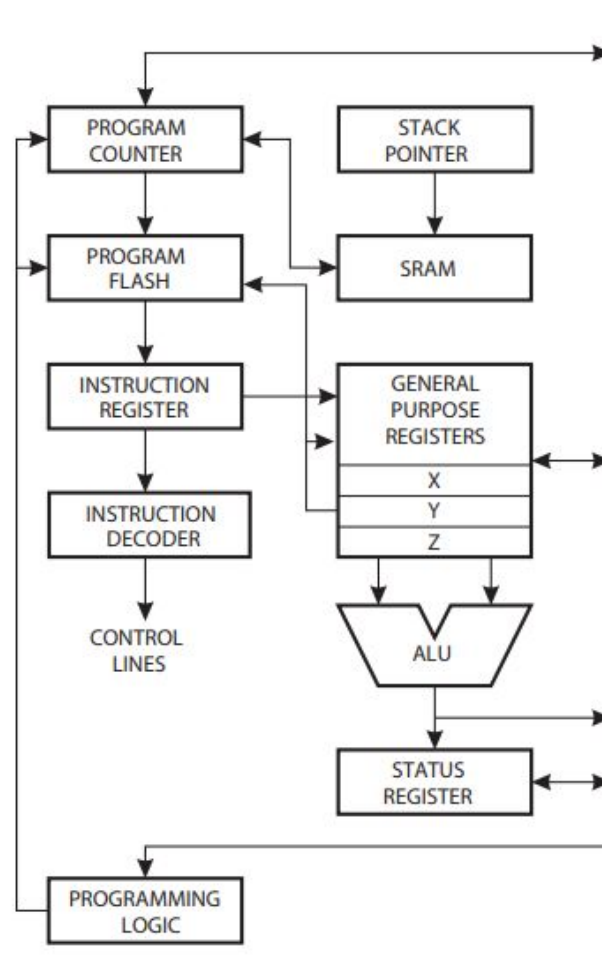
Von Neumannova Architektura: Program je spolu s daty ve stejné paměti

Zdroj: https://en.wikipedia.org/wiki/Von_Neumann_architecture#/media/File:Von_Neumann_Architecture.svg

Jak funguje počítač

Jednoduchý, ovšem reálný moderní 8-bitový procesor
Harvardské architektury (program a data mají každý svou paměť')

Zdroj: Atmel AVR ATtiny 25/V datasheet

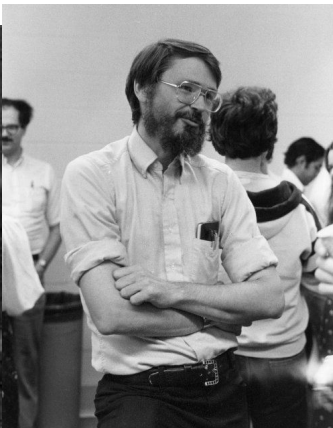
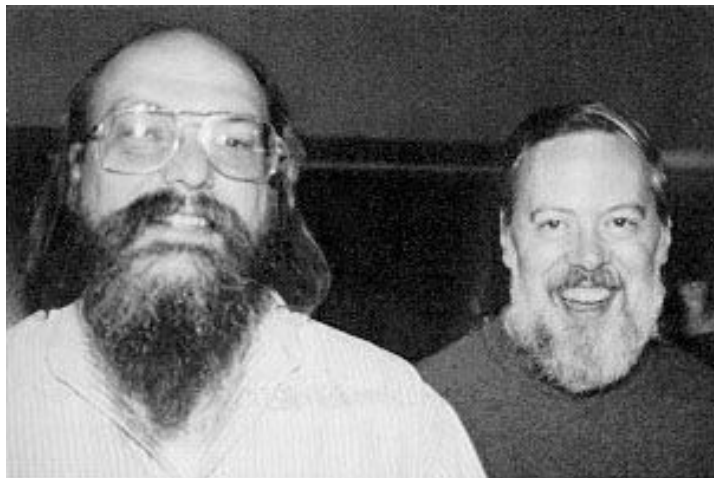


Historie programování a programovací jazyky

- [1801 Jacquard loom](#)
- 1837 Analytical Engine (Charles Babbage, Ada Lovelace)
- 1936 První kód tak, jak jej známe dnes (Alan Turing)
- 1948 ENIAC coding system (Adele Goldstine, John von Neumann)
- 1949 Assembler (David Wheeler)
- 1952 První kompilátor (Grace Hopper)
- 1953 Fortran (John W. Backus)
 - První specializovaný jazyk s optimalizovaným kompilátorem
- 1959 COBOL
 - První jazyk cílící na přenositelnost mezi počítači

Historie programování a programovací jazyky

- 1973 The C Programming Language
 - UNIX, systémový programovací jazyk, První verze "Hello, world!":
"portabilní assembler"



Brian Kernighan

Zdroj: https://blog.hackerrank.com/wp-content/uploads/2015/04/brian_500.jpg

```
#include <stdio.h>

main()
{
    printf("hello, world\n");
}
```



PDP-11

<https://www.chessprogramming.org/PDP-11>



Brian W. Kernighan • Dennis M. Ritchie

PRENTICE HALL SOFTWARE SERIES

Dennis Ritchie, Ken Thompson (B, unix), 1973

Zdroj: [https://en.wikipedia.org/wiki/C_\(programming_language\)#/media/File:Ken_Thompson_and_Dennis_Ritchie--1973.jpg](https://en.wikipedia.org/wiki/C_(programming_language)#/media/File:Ken_Thompson_and_Dennis_Ritchie--1973.jpg)

The C Programming Language, first edition aka K&R, 1978

Zdroj: [https://en.wikipedia.org/wiki/File:The_C_Programming_Language_First_Edition_Cover_\(2\).svg](https://en.wikipedia.org/wiki/File:The_C_Programming_Language_First_Edition_Cover_(2).svg)

Historie programování a programovací jazyky

- 1985 The C++ Programming Language



Bjarne Stroustrup

<https://upload.wikimedia.org/wikipedia/commons/d/da/BjarneStroustrup.jpg>

"C with classes":
objektově orientované
rozšíření C

STL: Standard
Template Library

První verze "Hello, world!" programu:

```
#include <iostream>

int main()
{
    std::cout << "Hello World!";
}
```



The C++ Programming Language, 1985, 1. vydání

<https://upload.wikimedia.org/wikipedia/commons/d/da/BjarneStroustrup.jpg>



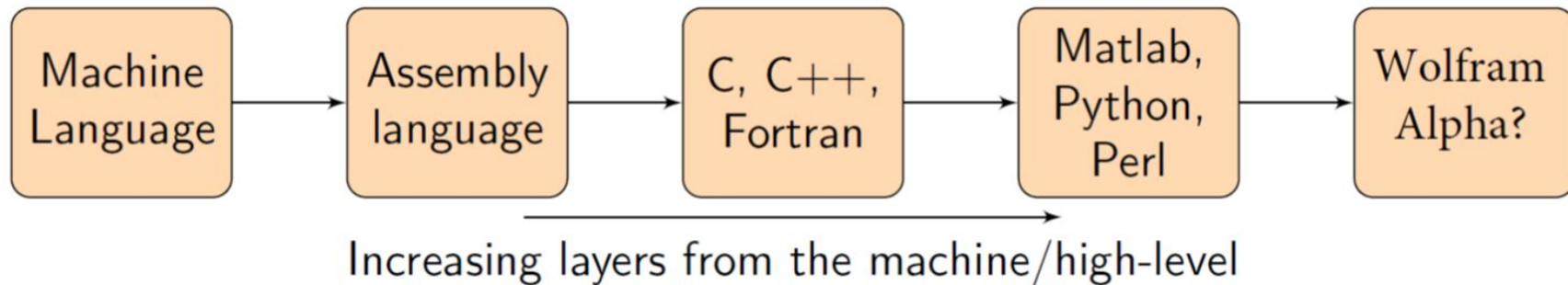
C makes it easy to shoot yourself in
the foot; C++ makes it harder, but
when you do it blows your whole leg
off.

— *Bjarne Stroustrup* —

AZ QUOTES

Historie programování a programovací jazyky

- Vývoj se samozřejmě nezastavil v roce 1985...



Stále přibývají vrstvy abstrakce...

zdroj: cdslab.org

Historie programování a programovací jazyky

- Vyvíjí se ovšem i C a C++

K&R (1978)

ANSI C (1990)

MISRA C (1998, 1.ed)

C99 (1999)

MISRA C (2004, 2.ed)

C11 (2011)

MISRA C (2012, 3. ed)

C18 (2018)

C2x (?)

TC++PL (1985)

C++98 (1990)

C++03 (2003)

MISRA C++ (2008)

C++11 (2011)

C++14 (2014)

C++17 (2017)

C++20 (?)

(MISRA = Motor Industry Software Reliability Association)

C vs. C++

- C
 - Defakto standardní "portable assembler"
 - C je jednoduché: ANSI C má 274 stran
 - poslední verze [ISO/IEC 9899:2018\(en\) Information technology — Programming languages — C](#) má 520 stran
 - procedurální strukturovaný jazyk
 - bez funkcionality skrývání dat a omezení přístupu

C vs. C++

- C++
 - podpora většiny moderních paradigmat programování
 - strukturální, procedurální, OOP, generické, funkcionální
 - plná kontrola nad přístupem k datům
 - přístupové modifikátory, jmenné prostory, "friends" třídy a funkce
 - Komplexní jazyk: C++11 standard má 1376 stran, poslední C++20 standard [ISO/IEC 14882:2020\(en\) Programming languages — C++](#) má 1853 stran

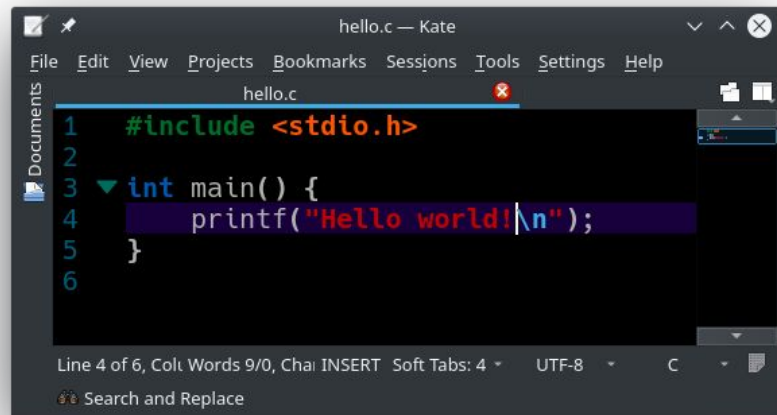
Minimální program, Program "Ahoj světe!"

[Minimální program v C++](#)

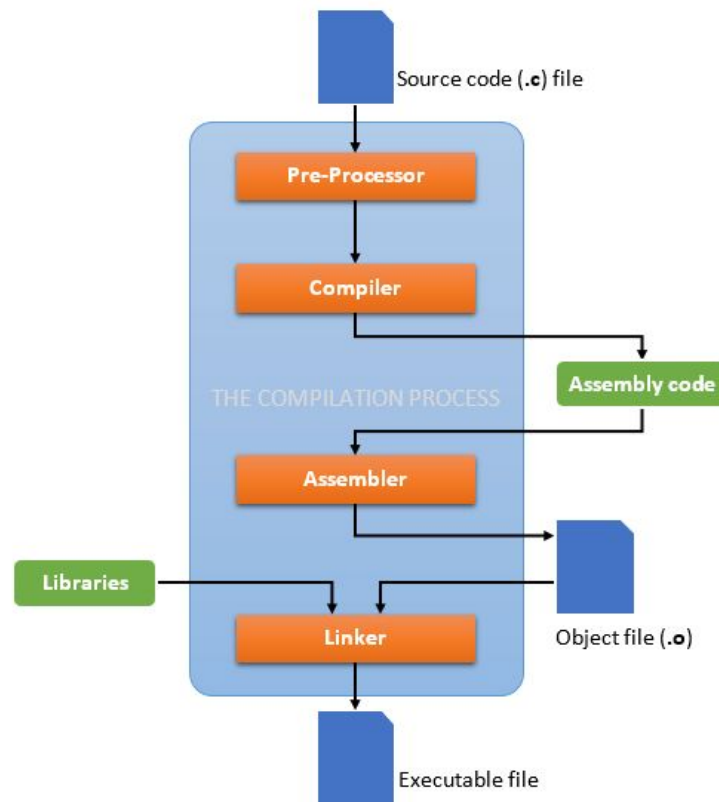
[Hello world program v C++](#)

Textový editor

- Kód je obyčejný textový soubor
- Pro úpravu nám stačí libovolný textový editor
 - notepad.exe, VSCode, PSPad, Atom, Kate...
- Online nástroje
 - <https://www.onlinegdb.com/>
- Existují i propracovanější nástroje...



Jak vznikne program



Proces překladu programu, zdroj: Laura Roudge,
<https://medium.com/@laura.derohan/compiling-c-files-with-gcc-step-by-step-8e78318052>

Spuštění programu

- Ve Windows je vyprodukován normální EXE soubor
- Je vhodné spouštět konzolové aplikace v konzoli

IDE

- Program obsahující textový editor, kompilátor a další nástroje
- Microsoft Visual Studio, Eclipse, CodeLite, ...
- Instalace...

Git a GitHub

- Decentralizovaný verzovací systém
- Slouží pro záznam změn v souborech
- Značně napomáhá spolupráci více programátorů
- Existuje mnoho online služeb spolupracujících s Gitem, nejznámější je github.com
- Tuto prezentaci naleznete celou zde, včetně příkladů: github.com/ChaoticRoman/kurz_cpp



Linus Torvalds, autor linuxového jádra a gitu

Zdroj: https://upload.wikimedia.org/wikipedia/commons/6/69/Linus_Torvalds.jpeg

Terminálový vstup a výstup: Komunikace s uživatelem

- Knihovna iostream

[Příklad v C++](#)

[Další příklad v C++](#)

Namespaces aneb jmenné prostory

- Separace deklarací

Příklad v C++

- Standardní jmenný prostor std
- Šetřeme s nimi z estetických důvodů :-)

Data, jejich typy a proměnné

- int
- float, double, long double
- char
- bool
- C++ řetězce

Příklad v C++

Operátory a aritmetické výrazy

- Standardní aritmetika

Příklad v C++

Funkce

- Funkce slouží pro pojmenování částí programu a usnadňují jejich opakované používání
- název funkce
- návratový typ, klíčové slovo return
- argumenty

Příklad v C++

Funkce v C++

- Přetěžování funkcí (function overloading)

Příklad v C++

- Pozor na "Name mangling"! [Příklad v C/C++](#)

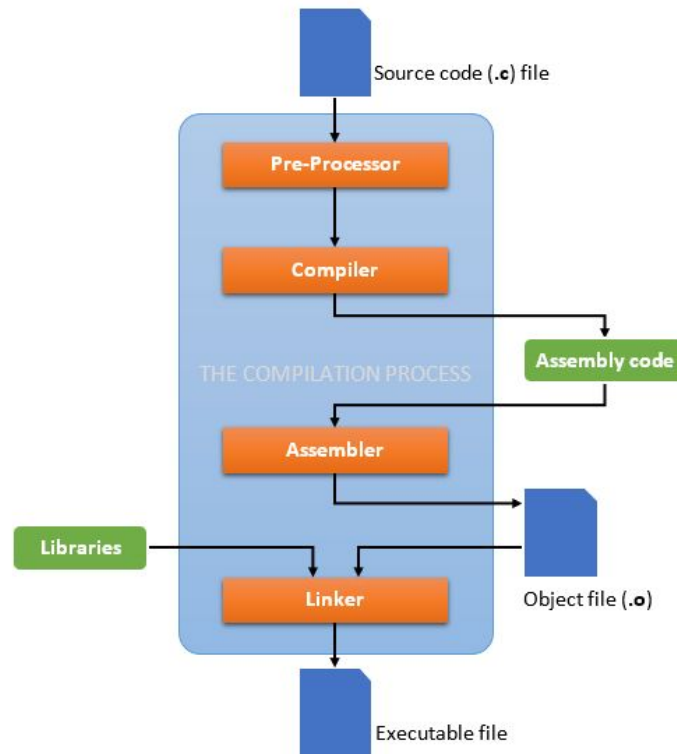
Lze používat najednou C++ a C, pokud nechceme po C schopnosti, které má jen C++, ale pak používáme "extern "C" { /* ... */ }" pro zabránění name manglingu

Detailní vysvětlení

(TLDR: vyhněte se tomu, jestli to jde, např. přeložte vše pomocí C++)

Preprocessor

- V podstatě se jedná o nahrazování textu a možnosti podmíněného překladu
- Nepřehánět jeho využívání!
- [Příklad v C++](#)



Proces překladu programu, zdroj: Laura Roudge,
<https://medium.com/@laura.derohan/compiling-c-files-with-gcc-step-by-step-8e78318052>

Organizace kódu: Zdrojové soubory, hlavičkové soubory a preprocesor

- Deklarace a definice
- Hlavičkové soubory, zdrojové soubory, translační jednotky
- Hierarchie projektu
- Protip: v jednoduchosti je síla, "pravidlo jedné obrazovky"

Příklad v C++

Podmínky

- Pravda a nepravda
- Porovnávací operátory
- if
- if ... else
- switch ... case

Příklad v C++

Cykly

- Cykly provádějí daný úsek kódu opakovaně
- for: předem známý počet operací
- while: počet operací nemusí být předem známý, nemusí se provést ani jednou
- do ... while: počet operací nemusí být předem známý, vždy se aspoň jednou provede
- break, continue

Příklad v C++

Řetězce

Příklad v C++

Viz <http://www.cplusplus.com/reference/string/string/>

Pozor na kódování (cp1250 vs. UTF-8) ve Windows

Práce se soubory: Čtení, psaní, počítání...

- Otevření souboru pro čtení a zápis
- Binární vs. textový mód
- Zavírání souborů
- Čtení a zápis

Příklad v C++

C pole, C++ pole a vektory

Příklad C pole v C++

Příklad C++ pole

Viz <https://en.cppreference.com/w/cpp/container/array>

Příklad C++ vektoru

Viz <https://en.cppreference.com/w/cpp/container/vector>

Templates aneb Šablony

Templates nám umožní vygenerovat kód použitelný pro více datových typů

Příklad v C++

Pozor! Šablony vedou k komplikovaným chybovým hláškám v případě chyby a často se hůře ladí.

Třídy, objekty

- C struktury (data) vs. C++ struktury, třídy a objekty (data a metody)
- class, object, instance, member, member variable, method
- řízení přístupu (access modifiers): public, private

Příklad v C++: třídy a struktury

Příklad v C++: metody

Statické proměnné a metody třídy

statické proměnné a třídy nejsou vázané na konkrétní instanci

Příklad v C++

Dědičnost a polymorfismus

- Dědičnost (inheritance)
 - vytváření potomků (child) odvozených od rodičů (parent)
- Polymorfismus
 - zacházení s různými potomky stejné třídy stejným způsobem
- Virtuální metody, dynamic dispatch, vtable

Příklad v C++

Praktická úloha: překladiště paketů

Váš program akceptuje "vstupní paket" ze standardního vstupu potvrzený Enterem a zapisuje "výstupní paket" do souboru output.txt, pokaždé na nový řádek.

Vstupní paket	Výstupní paket	Poznámky
0	t4ping	
1	t4pong	
2<délka><náklad>	t<délka + 6><Ahoj, <náklad>>	Délka je maximálně 3 a náklad musí být tak dlouhý, kolik říká délka, jinak je paket neplatný Příklad: 23Ivo -> t9Ahoj, Ivo
cokoliv jiného nebo neplatný paket	t7invalid	