Úvod

Karel Richta a kol.

katedra počítačů FEL ČVUT v Praze

Programování v C++, B6B36PJC
2017, Lekce 1

https://cw.fel.cvut.cz/wiki/courses/b6b36pjc/start

Proč se učit C++?

- Protože...
  - ... se ve vašem oboru používá
  - ... chcete psát rychlý kód
  - ... chcete, aby vás váš počítač zcela poslouchal
  - ... znáte C a nestačí vám
  - ... znáte Python a chcete poznat svět z jiné strany
  - ...
  - ... vám bylo dáno jako povinný předmět ☺

Kde se tedy C++ používá?

Počítačová grafika

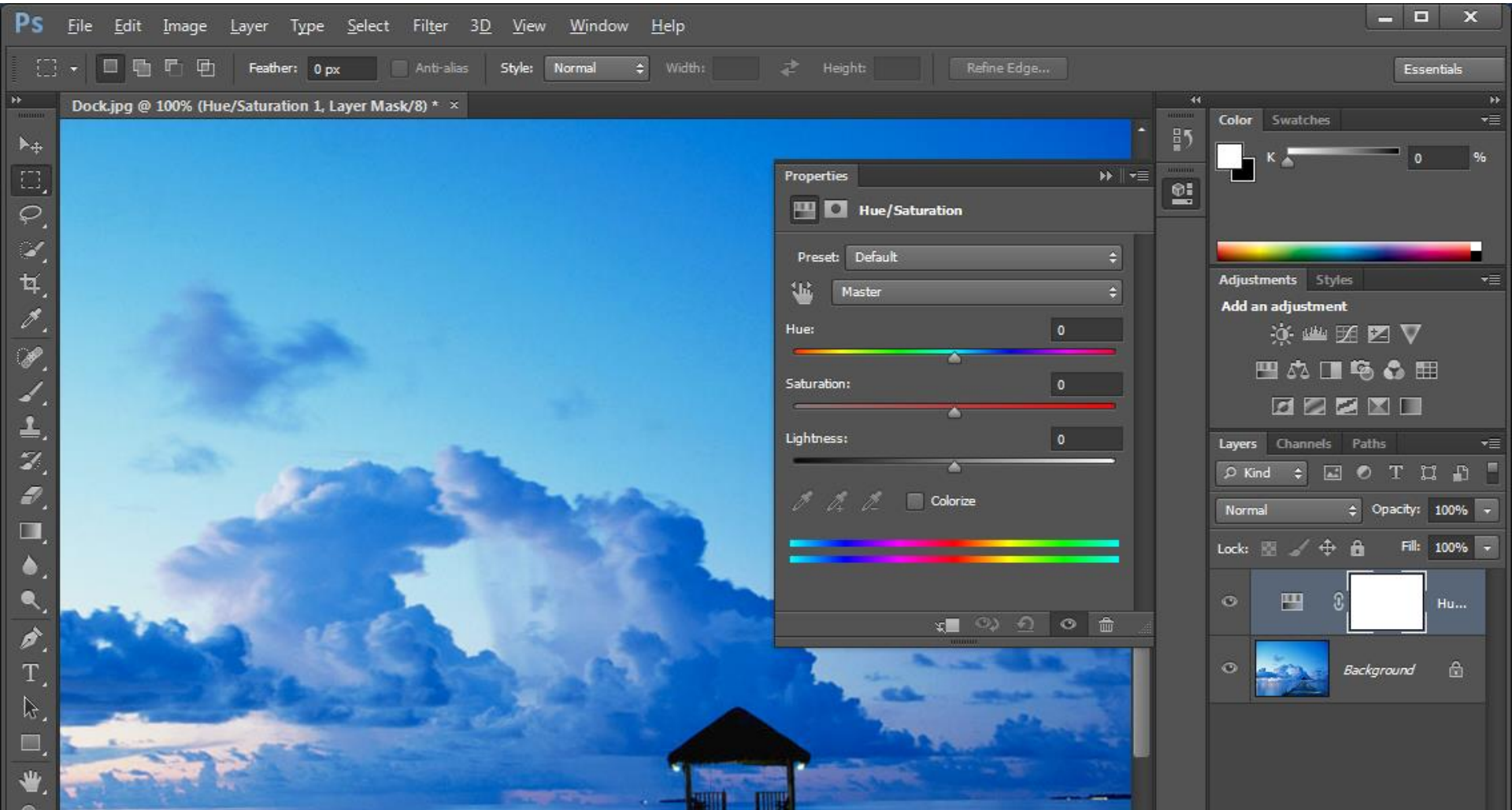
- Unreal Engine, idTech, CryEngine, Unity, ...

Analýza dat

- CERN Root

Produkční software

- CAD/CAM, editory, modeláře (Autodesk, Adobe…)

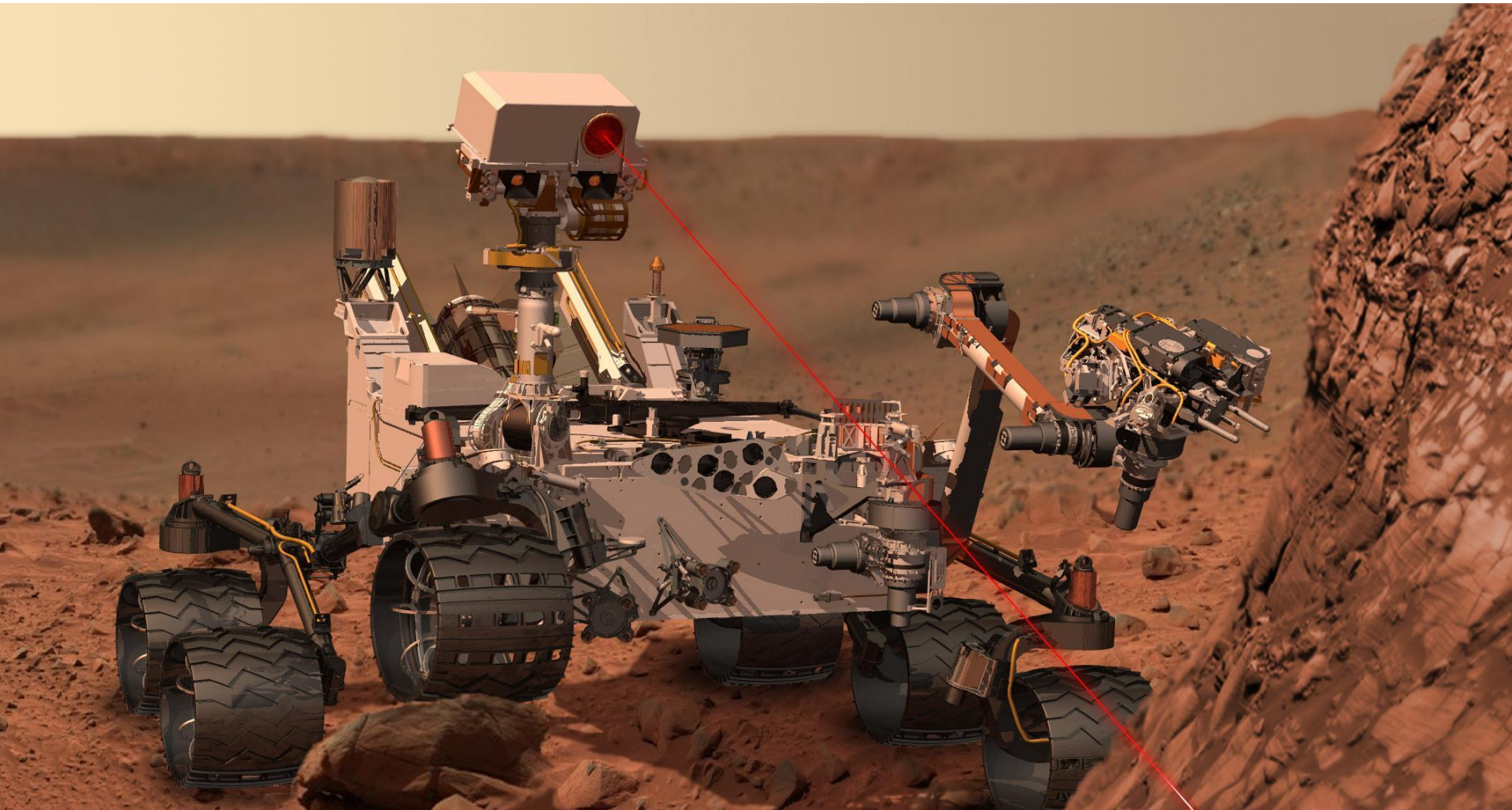


Burza a finance

- High Frequency Trading, Bloomberg

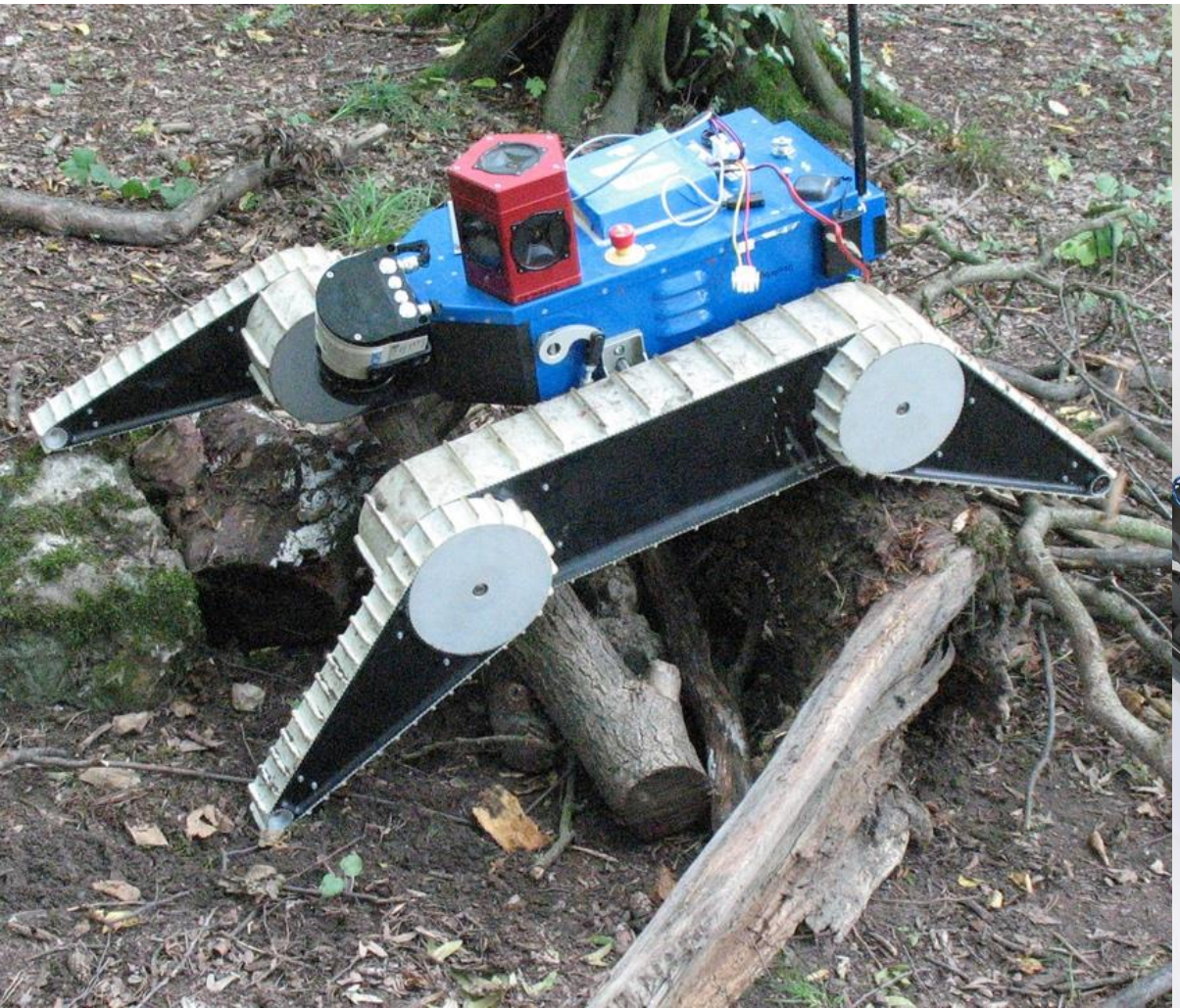
Real-time systémy

- Průmyslové technologie, na obrázku: Mars Rover



Robotika

- Robot Operating System



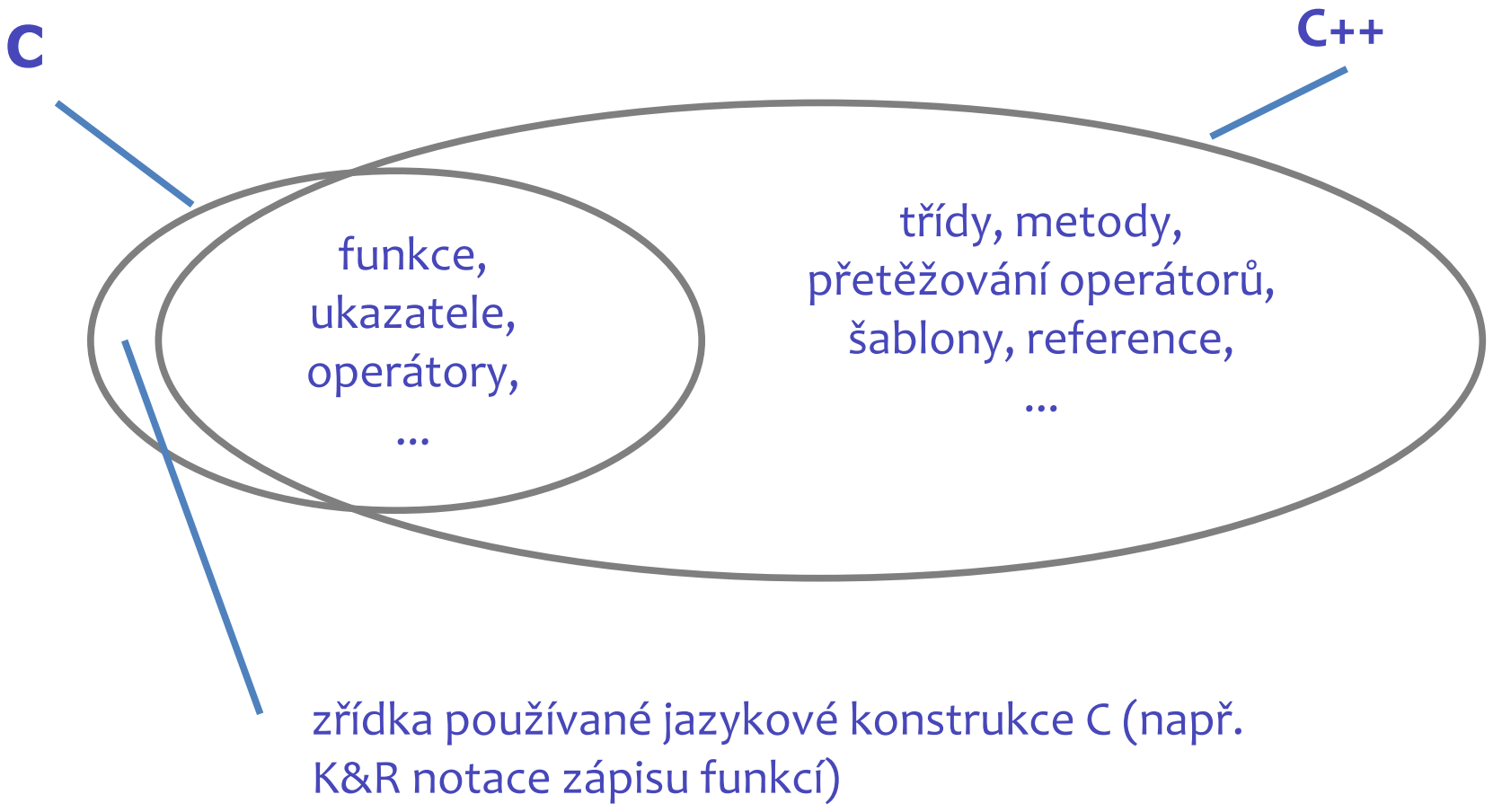
Zdravotnická zařízení

- Diagnostika, MRI, kardiostimulátory aj.

Vlastnosti jazyka C++

- Je převážně zpětně kompatibilní s jazykem C
- Umožňuje vysokou úroveň abstrakce
  - S nulovou, nebo minimální cenou
- Dává poslední slovo programátorovi
- Podporuje řadu programovacích stylů
- Umožňuje přístup až k hardwaru počítače

Kompatibilita C a C++



Vysoká úroveň abstrakce

- C++ obsahuje nejenom funkce, třídy, objekty
  - Které jazyky dneska ne?
- Ale i šablony
  - Parametrizace kódu dle typů
- Anonymní funkce, funkce vyššího řádu
  - funkce jako samostatný typ
- A velmi silný typový systém
  - V tomto předmětu se vám kouzla s ním vyhnou

Programátor má poslední slovo

- Filozofie C++ říká, že je důležitější umožnit nějaké použití, než zabránit všem chybám
  - Není vždy výhoda
  - C++ takto skončilo s turing-complete šablonami
- Programátor má možnost „přehlasovat“ typový systém.
  - Ale pokud nemá pravdu, tak pak bude mít problém

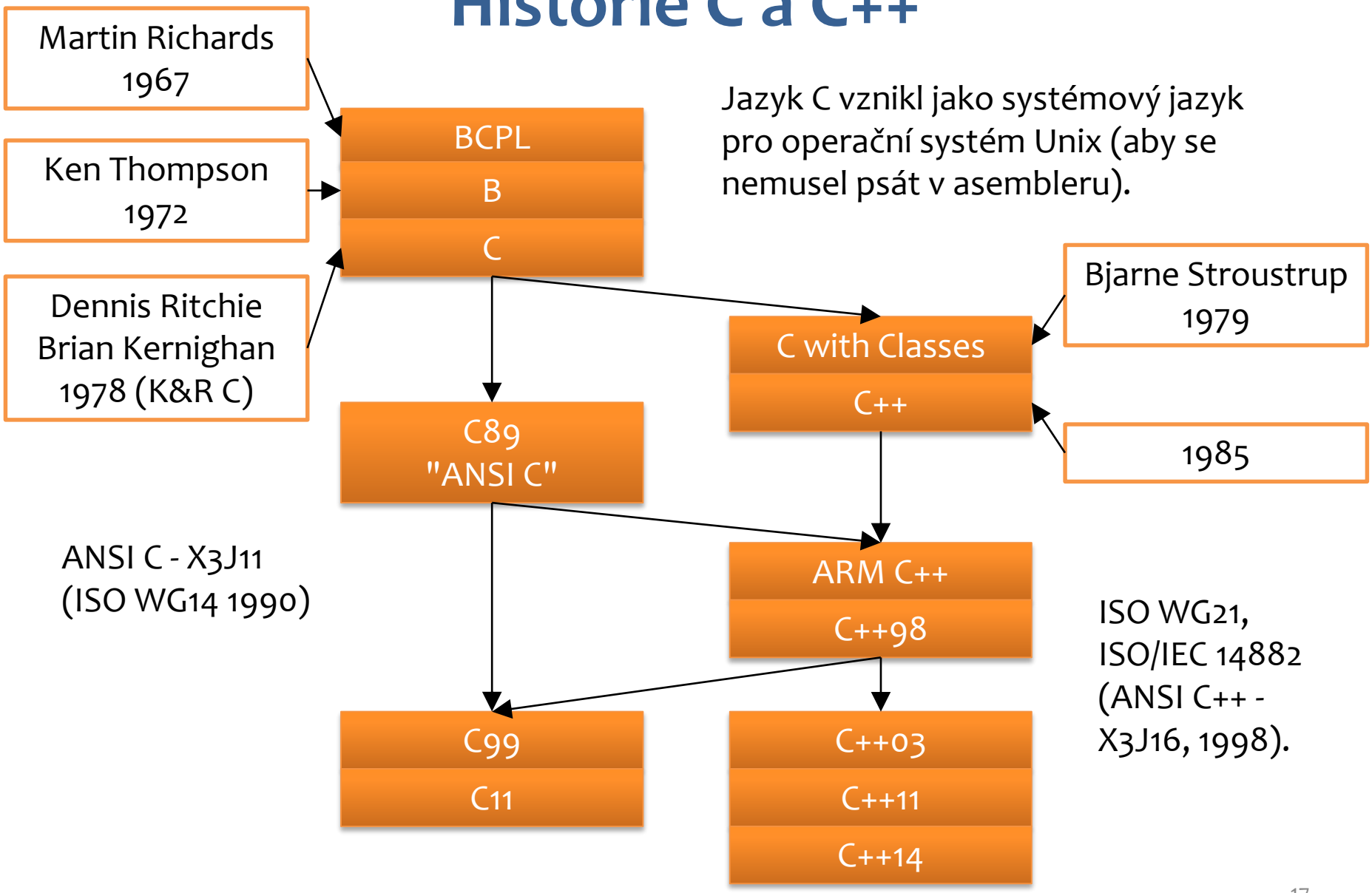
Programovací styly v C++

- K dispozici je několik programovacích stylů:
  - procedurální programování
  - objektově orientované programování
  - funkcionální programování
  - metaprogramování se šablonami
- C++ je *imperativní* programovací jazyk, tzn. podstatou programování je tvorba *příkazů*.

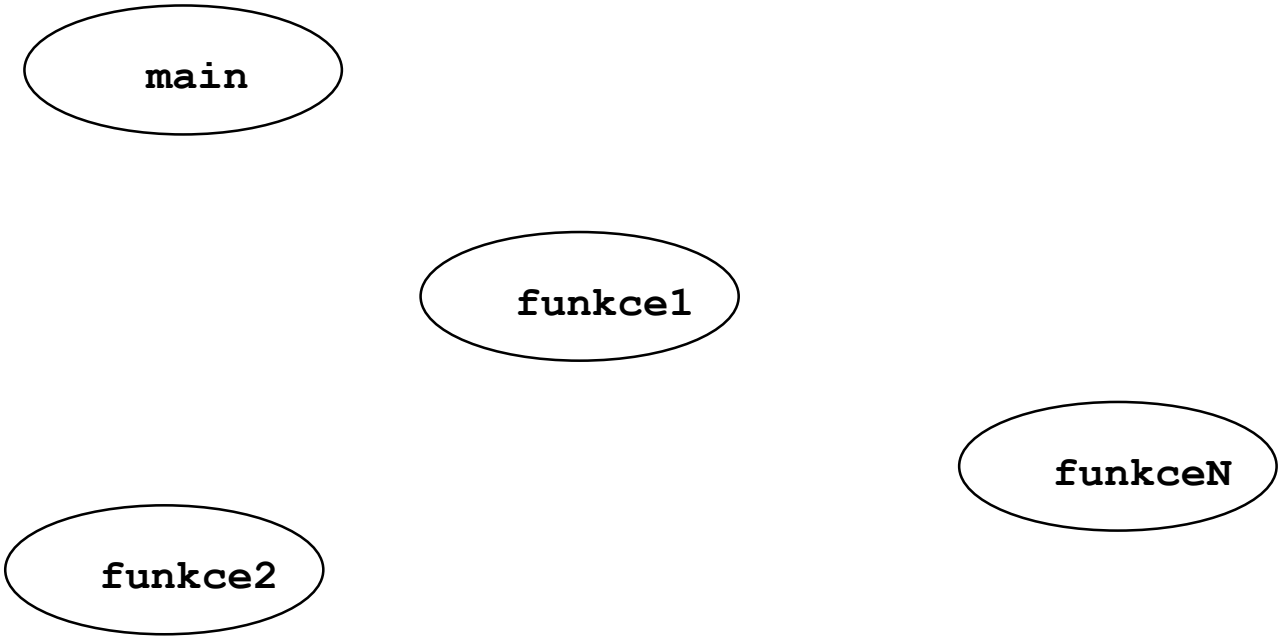
Spolupráce s hardwarem

- Přímý přístup k paměti pomocí ukazatelů
- *Knihovny* a *ovladače* umožňují práci
  - se službami operačního systému, např. se sítěmi, souborovým systémem, správou paměti, standardním vstupem a výstupem
  - se zařízeními, např. s grafickou kartou, tiskárnou, čtečkou SD karet

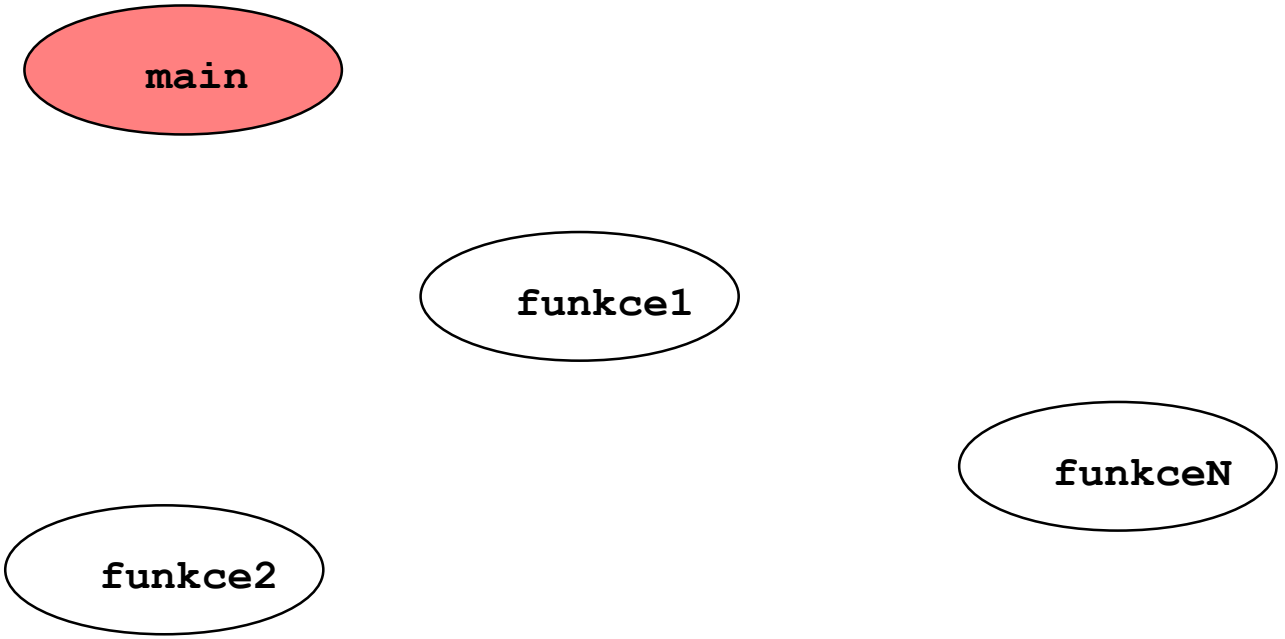
Historie C a C++



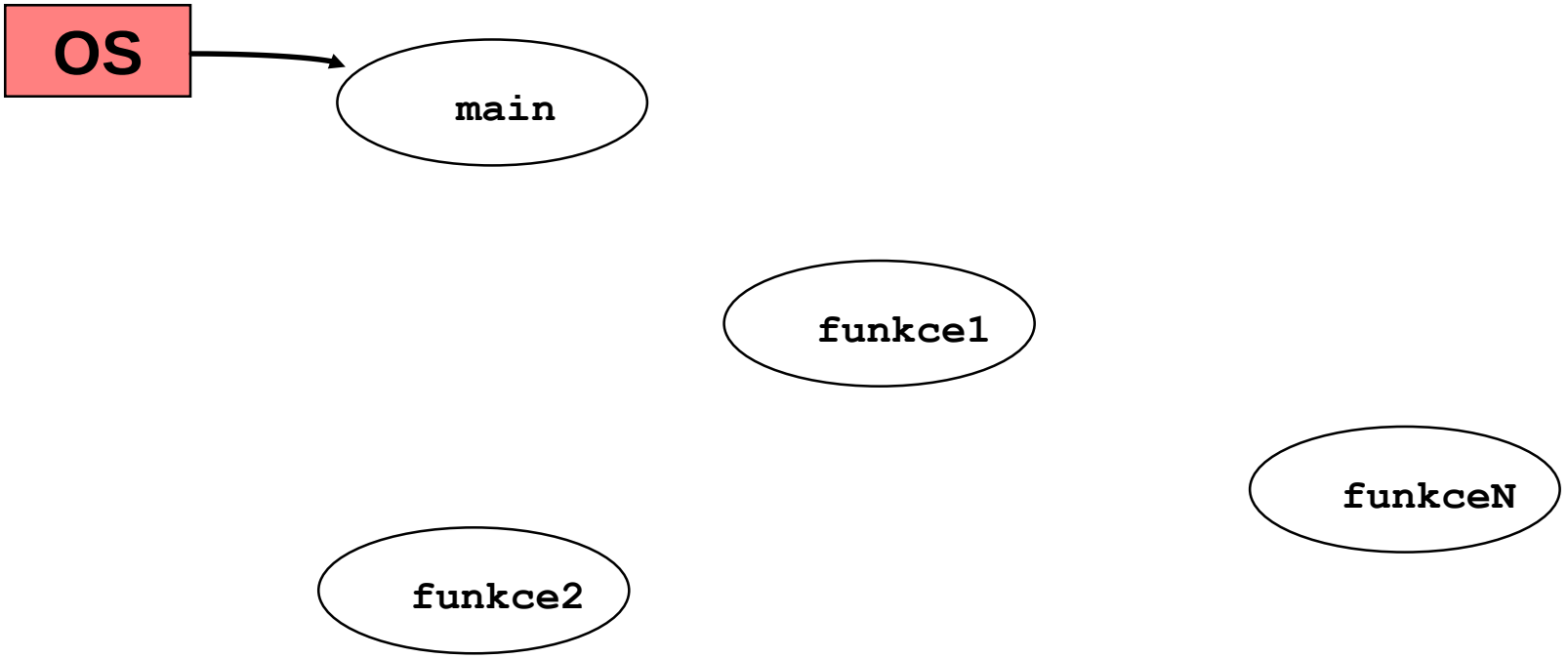
Program v C++ je sada funkcí



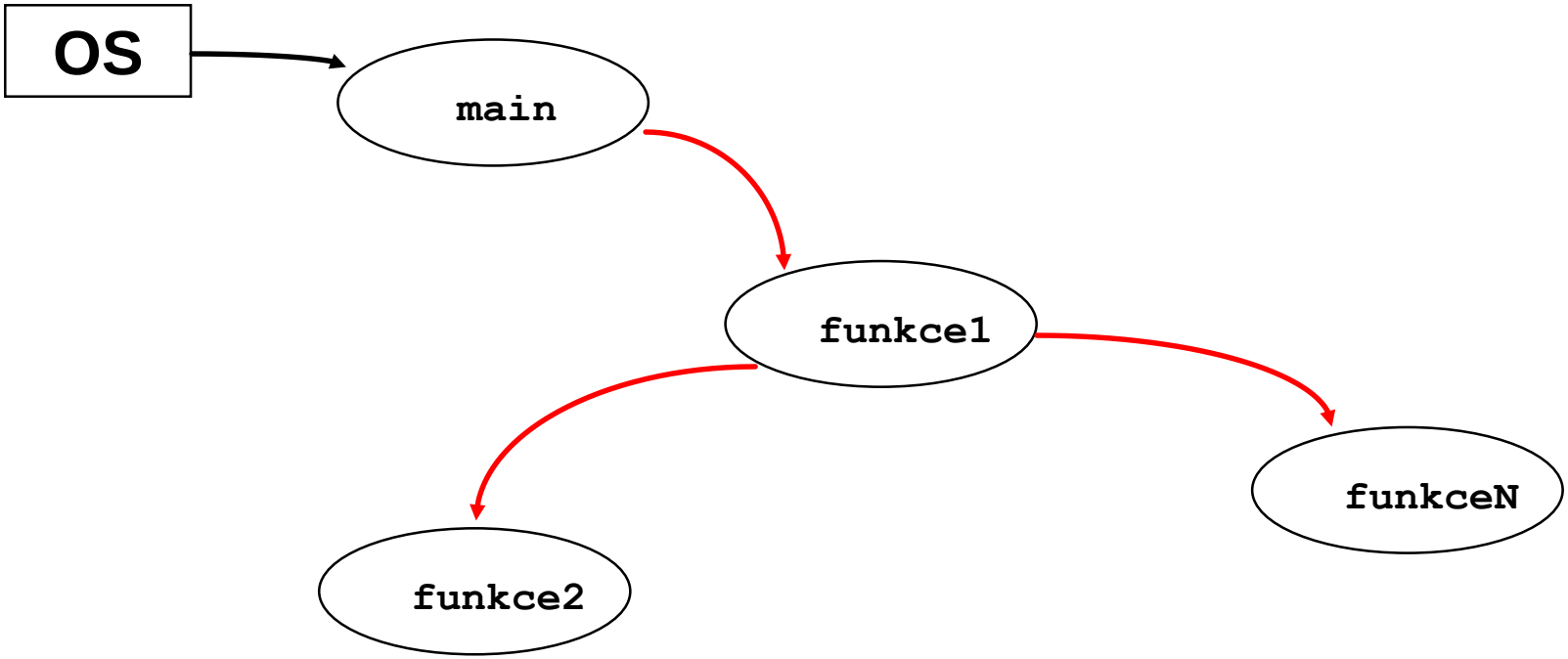
Jedna z nich se jmenuje `main`



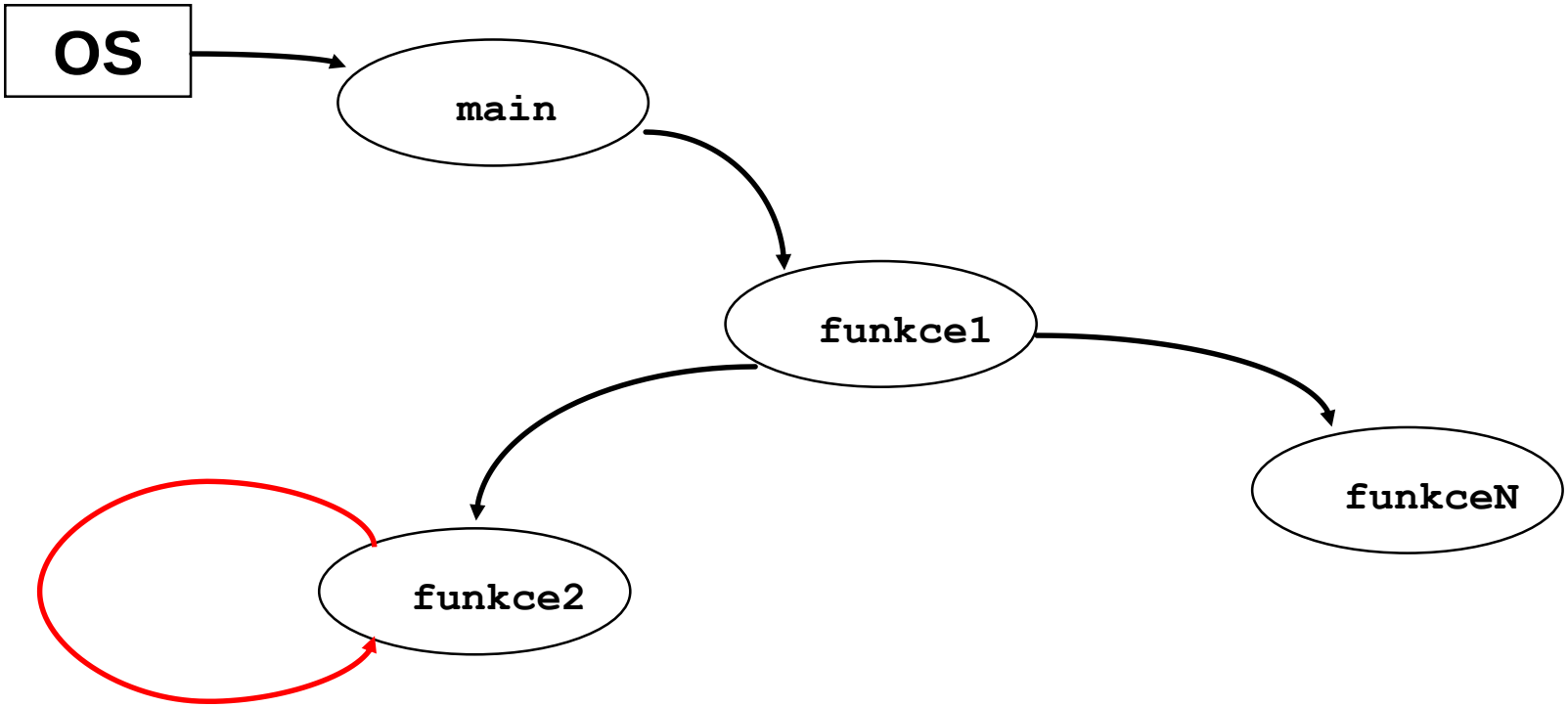
Funkci `main` spustí OS



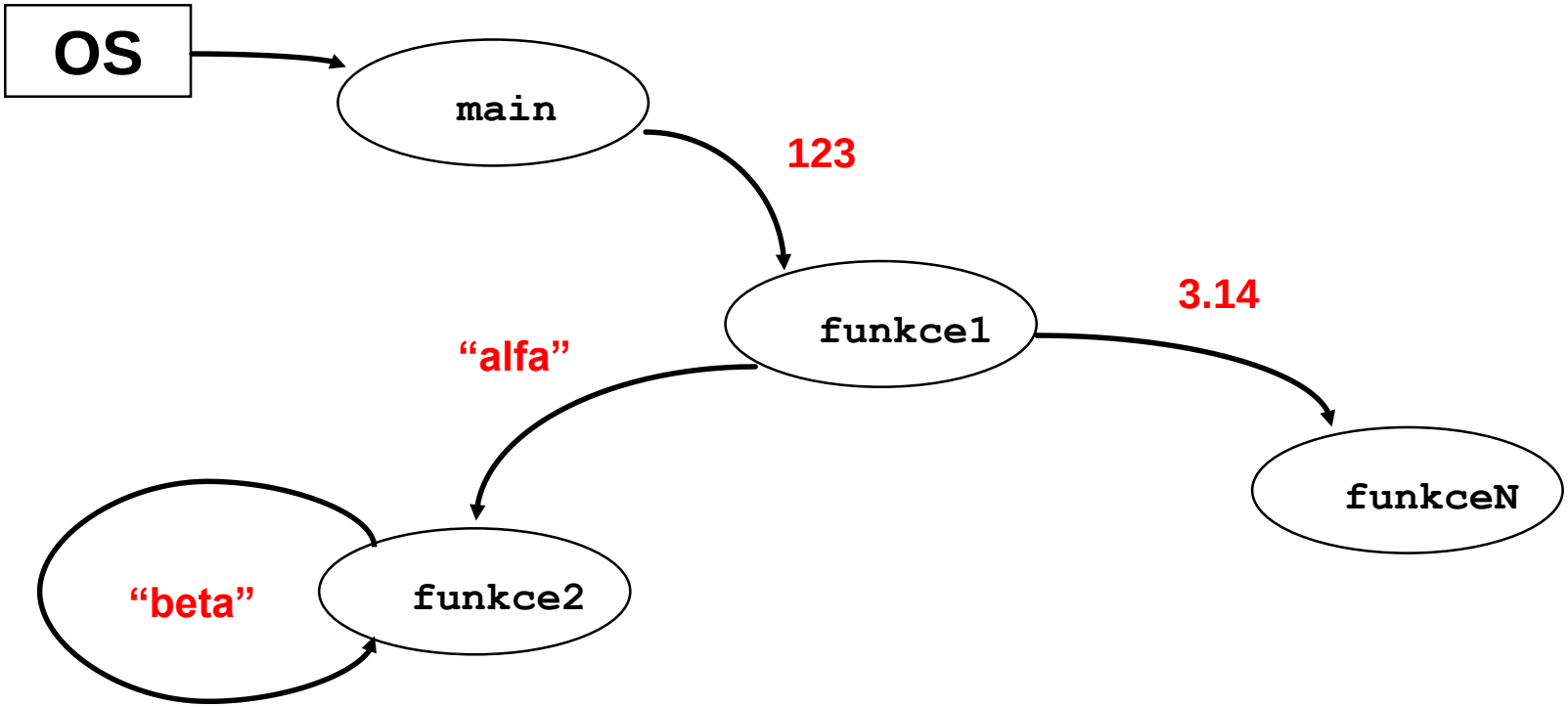
Funkce spolu komunikují (volají se)



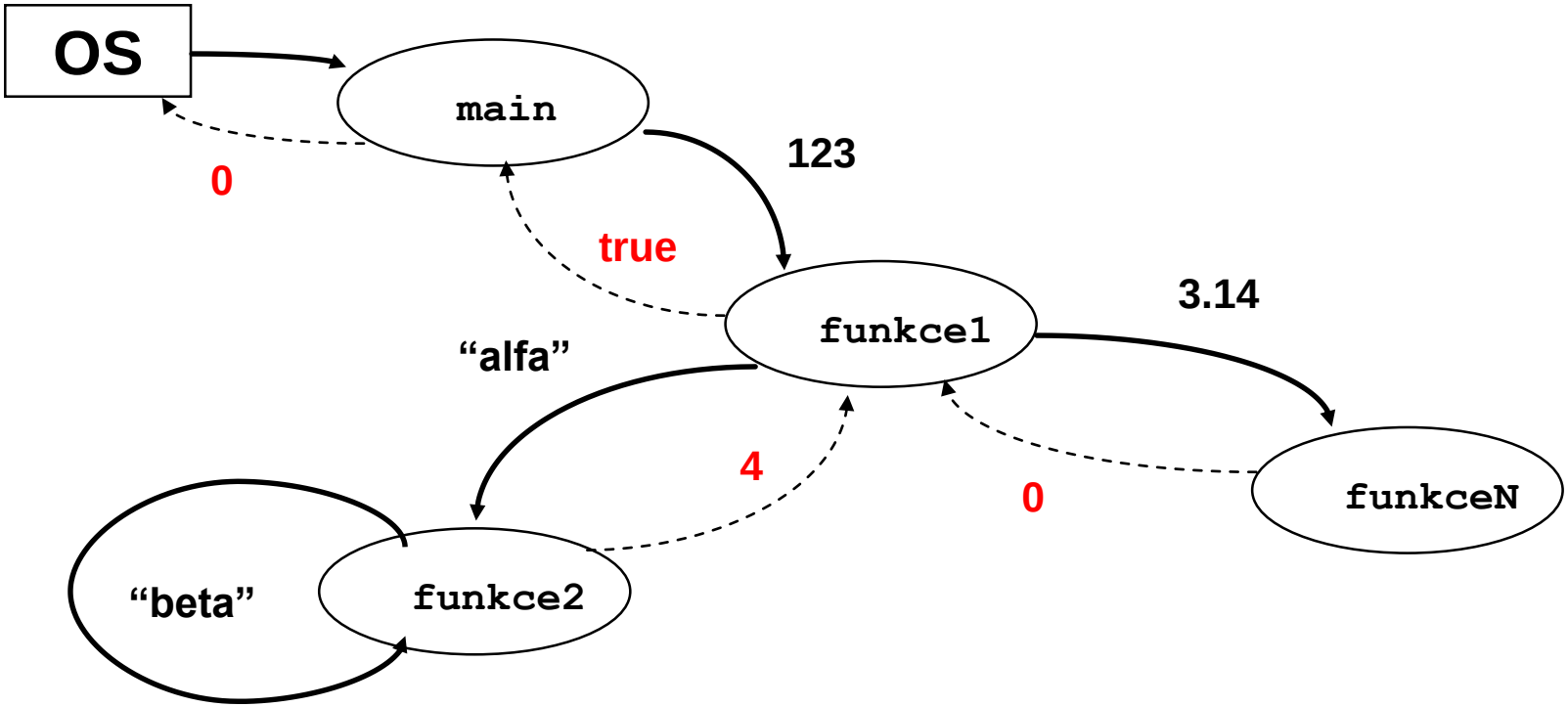
Případně i rekurzivně



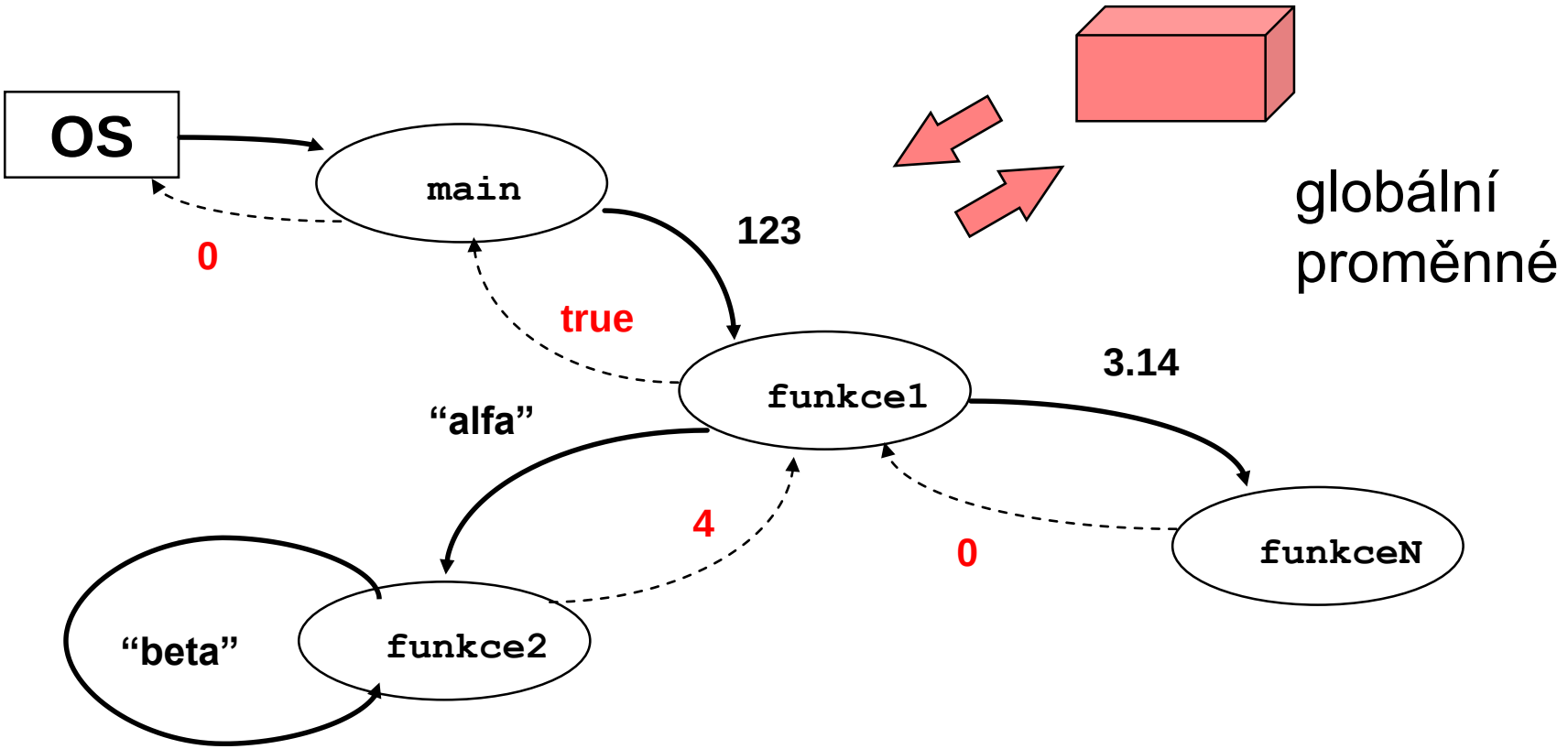
Předávají si parametry



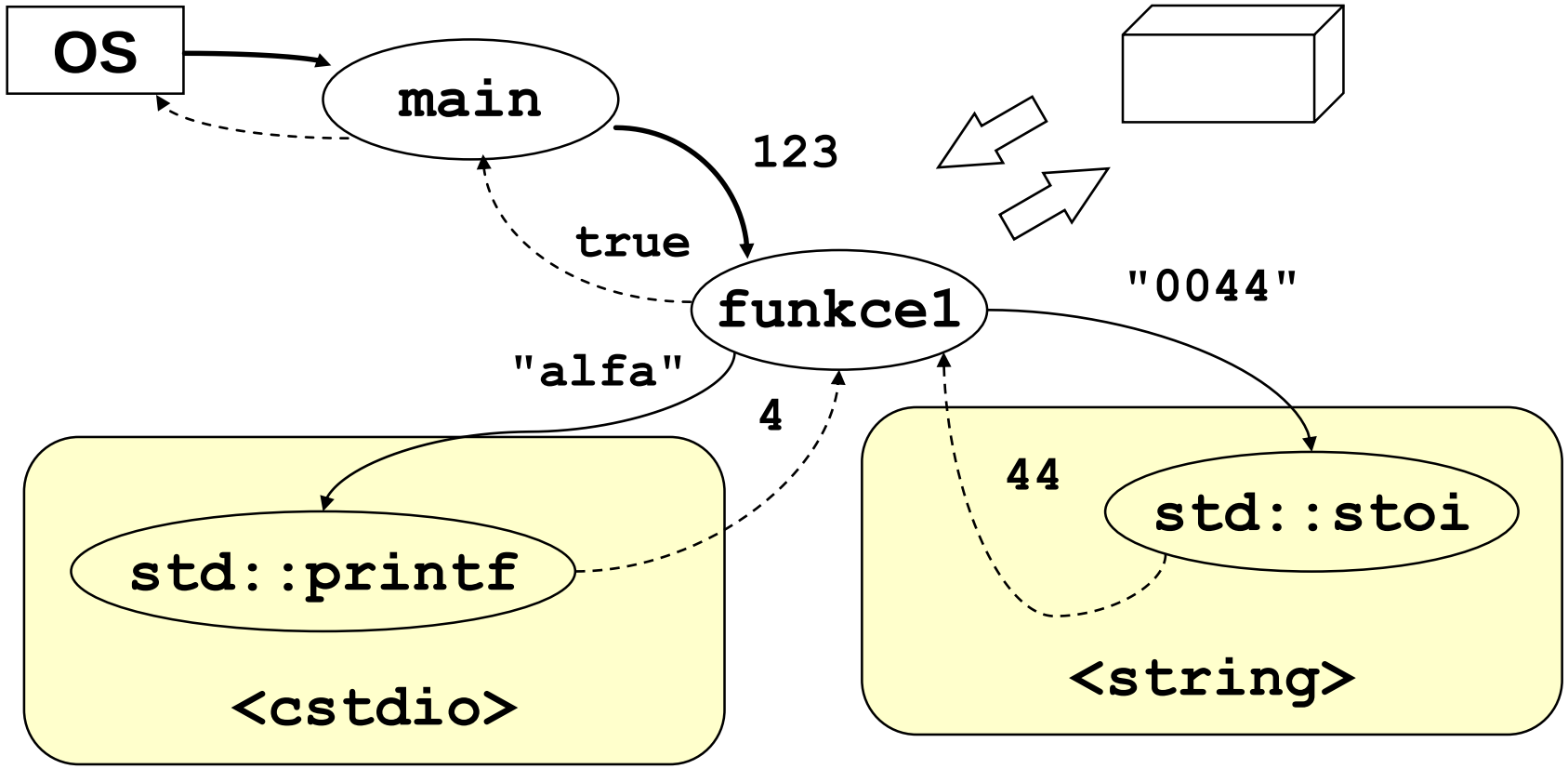
A vracejí výsledky



Navíc existuje globální paměť



Některé funkce jsou připraveny v knihovnách funkcí



Vstup a výstup v C++

- Každý program se chápe jako filtr – filtruje vstupní data na výstupní.
- Každý program proto má standardní vstup (`cin` – console input), standardní výstup (`cout` – console output), výstup pro hlášení chyb (`cerr` – console errors, `clog` – console log).
- Vstup pomocí >>

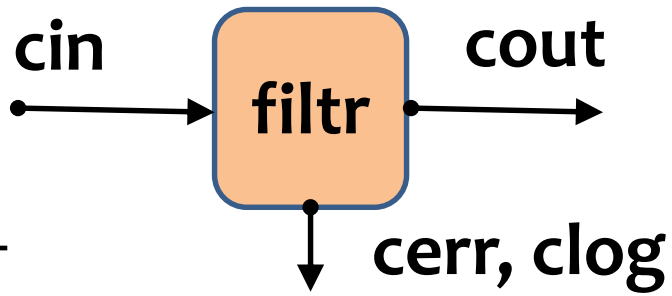
```
std::cin >> x;
```

- Výstup pomocí <<

```
std::cout << x;
```

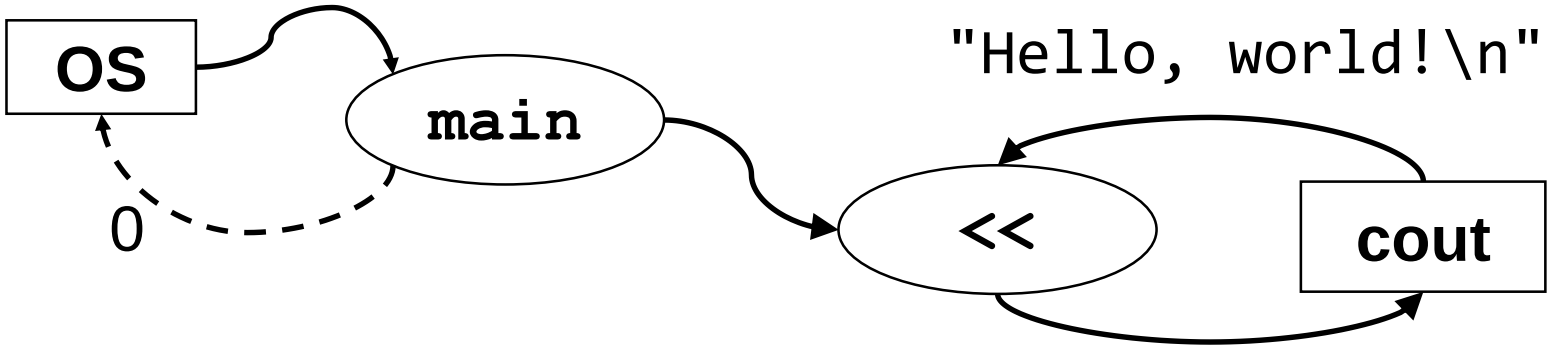
- Konec řádku: `"\n"` nebo `endl`

```
std::cout << "\n";
std::cout << std::endl;
```



Příklad “Hello, world!” v C++

<< znamená vypiš na výstup




```
#include <iostream>

int main() {
    std::cout << "Hello, world!\n";
}
```

Struktura programu v C++

- Program v C++ realizující řešení problému je sada definic tříd a funkcí, jedna z funkcí se jmenuje **main** a ta představuje hlavní program, který se spustí. Funkce **main** pak případně volá jiné funkce, či metody realizované ve třídách.
- Definice funkce **main** má např. tvar:

```
#include <iostream>

int main() {
    std::cout << "Priklad – tisk vstupu do zadani nuly\n";
    int hodn;
    do {
        std::cin >> hodn;
        std::cout << "Hodnota = " << hodn << "\n";
    } while (hodn != 0);
    std::cout << "Konec\n";
}
```

Formát zápisu programu v C není předepsán

```
#include <stdio.h>
int main(int t, int _, char *a){return!0<t?t<3?main(-79,-13,a+main(-87,1-_,
main(-86,0,a+1)+a)):1,t<_?main(t+1,_,a):3,main(-94,-27+t,a)&&t==2?_<13?
main(2,_+1,"%s %d %d\n"):9:16:t<0?t<-72?main(_,t,
"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#\
;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K w'K:'+}e#';dq#'l \
q#'+d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n')(}#}w'){){nl]'/+#n';d}rw' i;# \
){nl]!/n{n#'; r{#w'r nc{nl]'/#{l,+'K {rw' iK{;[{nl]'/w#q#n'wk nw' \
iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/*{q#'ld;r'}{nlwb!/*de}'c \
;;{nl'-{}rw]'/+,}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/ ') }+}{rl#'{n' ')# \
}'+}##(!!/")
:t<-50?_==*a?putchar(31[a]):main(-65,_,a+1):main((*a=='/')+t,_,a+1)
:0<t?main(2,2,"%s"):*a=='/'||main(0,main(-61,*a,
"!ek;dc i@bK'(q)-[w]*%n+r3#l,{}:\nuwloca-O;m .vpbks,fxntdCeghiry"),a+1);}
```

Zdroj: Wikipedia, International Obfuscated C Code Contest

Výstup tohoto programu

On the first day of Christmas my true love gave to me a partridge in a pear tree.

On the second day of Christmas my true love gave to me two turtle doves and a partridge in a pear tree.

On the third day of Christmas my true love gave to me three French hens, two turtle doves and a partridge in a pear tree.

On the fourth day of Christmas my true love gave to me four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the fifth day of Christmas my true love gave to me five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the sixth day of Christmas my true love gave to me six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the seventh day of Christmas my true love gave to me seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the eighth day of Christmas my true love gave to me eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the ninth day of Christmas my true love gave to me nine ladies dancing, eight maids a-milking, seven swans a -swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the tenth day of Christmas my true love gave to me ten lords a-lea ping, nine ladies dancing, eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the eleventh day of Christmas my true love gave to me eleven pipers piping, ten lords a-lea ping, nine ladies dancing, eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the twelfth day of Christmas my true love gave to me twelve drummers drumming, eleven pipers piping, ten lords a-lea ping, nine ladies dancing, eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings;

four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

Výstup tohoto programu (česky)

Na první den Vánoc mi moje pravá láska dala koroptev v hrušce (Ježíš Kristus).

Na druhý den Vánoc mi moje pravá láska dala dvě hrdličky (Starý a Nový zákon) a koroptev v hrušce.

Na třetí den Vánoc mi moje pravá láska dala tři francouzské slepice (víra, naděje, láska), dvě hrdličky a koroptev v hrušce.

Na čtvrtý den Vánoc mi moje pravá láska dala čtyři volající ptáky (evangelisté), tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Na pátý den Vánoc mi moje pravá láska dala pět zlatých prstenů (5 knih Mojžíšových), čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Šestý den Vánoc mi má pravá láska dala šest snášejících hus (šest dní stvoření světa), pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Sedmý den Vánoc mi moje pravá láska dala sedm plovoucích labutí (sedm darů Ducha svatého), šest snášejících hus, pět zlatých prstenů, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Osmého dne Vánoc mi moje pravá láska dala osm dojících služek (osm blahoslavenství), sedm plovoucích labutí, šest snášejících hus, pět zlatých prstenů, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

V devátý den Vánoc mi moje pravá láska dala devět tančících dam (ovoce Ducha svatého), osm dojících služek, sedm plovoucích labutí, šest snášejících hus, pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Desátý den Vánoc mi má pravá láska dala deset skákajících pánů (Desatero přikázání), devět tančících dam, osm dojících služek, sedm plovoucích labutí, šest snášejících hus, pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Jedenáctého dne Vánoc mi moje pravá láska dala jedenáct dudajících dudáků (jedenáct věrných apoštolů), deset skákajících pánů, devět tančících dam, osm dojících služek, sedm plovoucích labutí, šest snášejících hus, pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Na dvanáctý den Vánoc mi moje pravá láska dala dvanáct bubnujících bubeníků (dvanáct bodů víry apoštolů), jedenáct dudajících dudáků, deset skákajících pánů, devět tančících dam, osm dojících služek, sedm plovoucích labutí, šest snášejících hus, pět zlaté prsteny, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Hlavičkové soubory

hello.h

```
#ifndef SAY_HELLO_H
#define SAY_HELLO_H
void sayHello();
#endif
```

hello.cpp

```
#include "hello.h"
#include <iostream>
void sayHello() {
    std::cout << "Hello\n";
}
```

world.h

```
#pragma once
void sayWorld();
```

world.cpp

```
#include "world.h"
#include <iostream>
void sayWorld() {
    std::cout << "World\n";
}
```

```
#include "hello.h"
#include "world.h"
int main() {
    sayHello();
    sayWorld();
}
```

Preprocesor

- Povinná součást prostředí C a C++
- Provádí textové úpravy před vlastním překladem
- Řídí se příkazy pro preprocesor – **direktivami**
- Formát direktivy:

```
#direktiva parametry
```

- Příklady direktiv:
  - Vložení textu

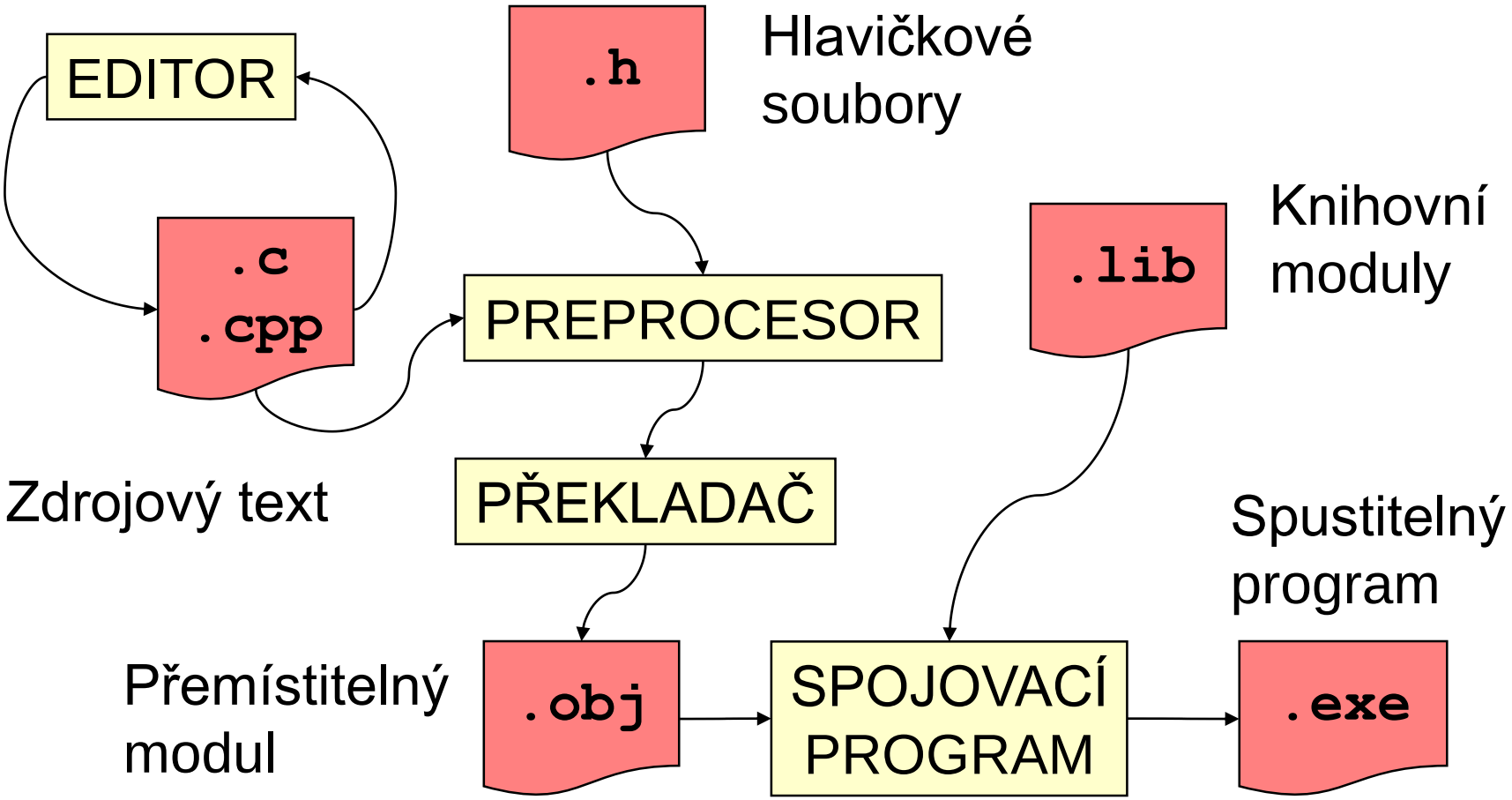
```
#include <soubor>
#include "soubor"
```

  - Definice makra

```
#define jméno text
#define jméno (parametry) text
```

  - Podmíněný překlad, parametry překladu (později)

Fáze překladu



Výpočetní problém

- Specifikace výpočetního problému má dvě části:
  - definici vstupu (vstupních dat) a
  - definici výstupu (výstupních dat).
- Instance výpočetního problému → nějaká konkrétní přípustná vstupní data. K jedné instanci problému může existovat několik správných výstupů (obecně se jedná o výpočet představující relaci).
- Příklad: problém řazení čísel (Sorting Problem for Numbers).
  - Vstup: Posloupnost *n* čísel *In* = $\langle a_1,\dots,a_n\rangle$.
  - Výstup: Taková permutace *Out* = $\langle a'_1,\dots,a'_n\rangle$ stejných čísel z *In*, pro kterou platí, že: $a'_1 \leq \dots \leq a'_n$.
- Instance problému řazení je např. zadání: seřaďte vzestupně posloupnost *In* = ⟨75, 11, 34, 176, 59, 6, 54⟩. Správným výstupem řazení je pak posloupnost *Out* = ⟨6, 11, 34, 54, 59, 75, 176⟩.
- Pozn.: Pro tento vstup je výstup určen jednoznačně, ale neplatí to pro každý

Výpočetní problém – příklad

- **Úloha:** najděte největšího společného dělitele (NSD) dvou přirozených čísel (víme, co je největší společný dělitel dvou přirozených čísel?)
- Řešení:
  - Popišme postup tak, aby byl použitelný pro dvě libovolná přirozená čísla:
    - označme zadaná čísla **x** a **y** a menší z nich **d**
    - není-li d společným dělitelem **x** a **y**, pak **d** zmenšíme o 1, test opakujeme a skončíme, až **d** bude společným dělitelem **x** a **y**
- Poznámka:
  - Význam symbolů **x**, **y** a **d** použitých v algoritmu:
    - jsou to proměnné (paměťová místa), ve kterých je uložena nějaká hodnota, která se může v průběhu výpočtu měnit

Algoritmus pro největšího spol. dělitele

Úloha: najděte největšího společného dělitele dvou přirozených čísel

Přesnější popis:

- **Vstup**: přirozená čísla **x** a **y**
- **Výstup**: NSD(**x**,**y**)
- **Postup**:
  1. Je-li **x**<**y**, pak d má hodnotu **x**, jinak **d** má hodnotu **y**
  2. Pokud je **d** dělitelem **x**, a zároveň je **d** dělitelem **y**, jdi na krok 5
  3. Zmenši **d** o 1
  4. Jdi na krok 2
  5. Výsledkem je hodnota **d**

- Sestavili jsme algoritmus pro výpočet největšího společného dělitele dvou přirozených čísel.

Zápis algoritmu NSD v C++

```
#include <iostream>  // pro std::cin a std::cout
#include <algorithm> // pro std::min

int deli(int d, int x) {
    return x % d == 0;
}

int main() {
    int x, y;
    std::cout << "Zadej dve cisla: ";
    std::cin >> x >> y;
    int d = std::min(x, y);
    while (!(deli(d, x) && deli(d, y))) {
        d -= 1;
    };
    std::cout << "NSD(" << x << "," << y << ") = " << d << "\n";
}
```

Různé styly řešení problémů

- Zápis programu realizuje určitou funkci či algoritmus, které představují řešení určitého problému.
- K řešení problému můžeme dospět různými způsoby – styly.
- Dva základní styly, kterými se budeme zabývat, jsou strukturovaný styl a objektově-orientovaný styl.
- Každý z nich se hodí na řešení trochu jiných problémů.
- **Strukturovaný styl** se hodí pro řešení procesních problémů, kdy řešení skládáme z posloupností jednotlivých kroků.
- **Objektově-orientovaný styl** se hodí spíše pro řešení různých informačních, či návrhových systémů, které uvažujeme jako síť komunikujících objektů.

Rozdíly mezi Javou a C++

Ačkoliv syntaxe obou jazyků je velmi podobná, chování se často liší. Některé z rozdílů jsou:

- Typový systém
- Správa prostředků
- Chování při inicializaci
- Nedefinované chování
- Generické typy

Typový systém

- C++ nerozlišuje mezi primitivními typy a objekty. Například, při předávání argumentů do funkce:

Java

```
public void foo(int count, Bar b) {
    count = 3;
    b.set(4);
}
```

C++

```
void foo(int count, Bar b) {
    count = 3;
    b.set(4);
}
```

Správa prostředků

- V obou jazycích jde použít operátor new. C++ ale očekává, že je poté zavolán operátor delete.

Java

```
public void foo() {
    MyObject obj = new MyObject();
    obj.doStuff();
}
```

C++

```
void foo() {
    MyObject* obj = new MyObject();
    obj->doStuff();
}
```

Správa prostředků

- V obou jazycích jde použít operátor new. C++ ale očekává, že je poté zavolán operátor delete.

Java

```
public void foo() {
    MyObject obj = new MyObject();
    obj.doStuff();
}
```

C++

```
void foo() {
    MyObject* obj = new MyObject();
    obj->doStuff();
    // nebyl zavolán delete: ztráta paměti
}
```

Správa prostředků

- V obou jazycích jde použít operátor new. C++ ale očekává, že je poté zavolán operátor delete.

Java

```
public void foo() {
    MyObject obj = new MyObject();
    obj.doStuff();
}
```

C++

```
void foo() {
    MyObject* obj = new MyObject();
    obj->doStuff();
    delete obj;
}
```

Správa prostředků

- V obou jazycích jde použít operátor new. C++ ale očekává, že je poté zavolán operátor delete.

Java

```
public void foo() {
    MyObject obj = new MyObject();
    obj.doStuff();
}
```

C++

```
void foo() {
    MyObject obj;
    obj.doStuff();
}
```

Správa prostředků

- V obou jazycích jde použít operátor new. C++ ale očekává, že je poté zavolán operátor delete.

Java

```
public void foo() {
    MyObject obj = new MyObject();
    obj.doStuff();
}
```

C++

```
void foo() {
    MyObject obj;
    obj.doStuff();
}
```

Lepší

Správa prostředků

- Narozdíl od Javy, C++ nemá *garbage collector*, tj. neposkytuje automatickou správu prostředků.
- Výhodou pro C++ je, že prostředky jsou spravovány deterministicky:
  - buďto je paměť uvolněna v daný okamžik v kódu,
  - nebo nikdy.
- To umožňuje používat C++ v hard real-time aplikacích.

Chování při inicializaci

- Java zaručuje, že pokud vytvoříme proměnnou bez rovnítka =, proměnná bude mít danou hodnotu.

```
int i;              // hodnota i je 0
Bar b;              // reference b je null
```

- To v C++ neplatí.

```
int i;              // hodnota i je nedefinovaná
Bar* b;             // ukazatel b je nedefinovaný
```

- Proto je dobré vždy určit hodnotu proměnné:

```
int i = 0;          // hodnota i je 0
Bar* b = nullptr; // ukazatel b je nullptr
```

Nedefinované chování

- C++ standard říká, že některé situace způsobují tzv. *nedefinované chování* programu.
- Například pokud čteme z neinicializované proměnné, náš program má nedefinované chování:

```
int i;          // i je neinicializovaná proměnná
int j = i + 1; // způsobí nedefinované chování
```

- Nedefinované chování je zakázáno.
  - Kompilátor předpokládá, že program nedefinované chování nemá. To občas vede ke zvláštním situacím...
- Pokud má program nedefinované chování, může se stát *cokoliv*.

Generické typy

- V Javě je možné tvořit generické typy, ale dosadit lze pouze objekty.

```
ArrayList<int> list = new ArrayList<>();      // Chyba
ArrayList<Integer> list = new ArrayList<>(); // OK
```

- V C++ lze dosadit libovolný typ.

```
std::vector<int> list; // OK
```

Použité obrázky

- Mars Rover
  http://marsprogram.jpl.nasa.gov/msl/multimedia/images/?ImageID=3710
- Unreal Engine 4
  http://cdn.wccftech.com/wp-content/uploads/2015/03/Unreal-Engine-4-Quixel%E2%80%99s-Jungle-Environment-2.jpg
- CERN
  http://home.cern/about/experiments/cms
- Burza v New Yorku
  https://en.wikipedia.org/wiki/Stock_exchange
- Robot ROS
  http://wiki.ros.org/Robots/Erle-Rover
- Photoshop
  http://www.fullypcgames.net/2013/10/adobe-photoshop-cs6.html
- Robot NIFTI
  https://cw.felk.cvut.cz/w/_media/misc/projects/nifti/sw/adaptive_traversability/robot_obstacle_intro.jpg
- Zdravotnická zařízení
  http://getingegroup.episerverhotell.net/sv/Press/Bildgalleri/Koncern-och-affarsomraden/

Konec