

# C2115

# Praktický úvod do superpočítání

IX. lekce

Petr Kulhánek, Tomáš Bouchal

[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta,  
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno

## ➤ **Architektura počítače**

CPU, paměť, grafická systém, disky, síť, periferie

## ➤ **Reprezentace číselných hodnot v číslicové technice**

celá čísla, reálná čísla

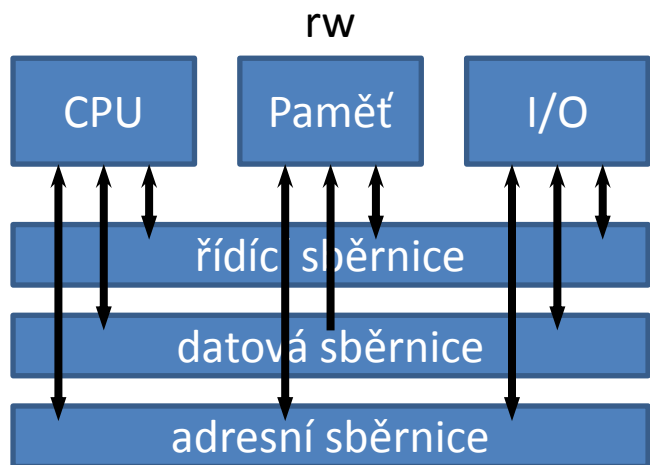
## ➤ **Od problému k výsledku**

algoritmus, zdrojové kódy, překlad, spouštění programu, programovací jazyky

# Architektura počítače

# Přehled

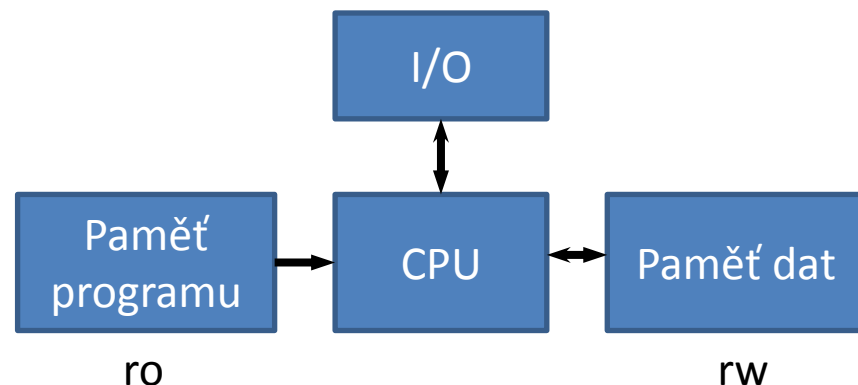
1945 von Neumannova architektura



- program může sebe modifikovat
- program a data nelze načítat současně

**John von Neumann**, původem maďarský matematik, působící ve spojených státech

1944 Hardwariská architektura

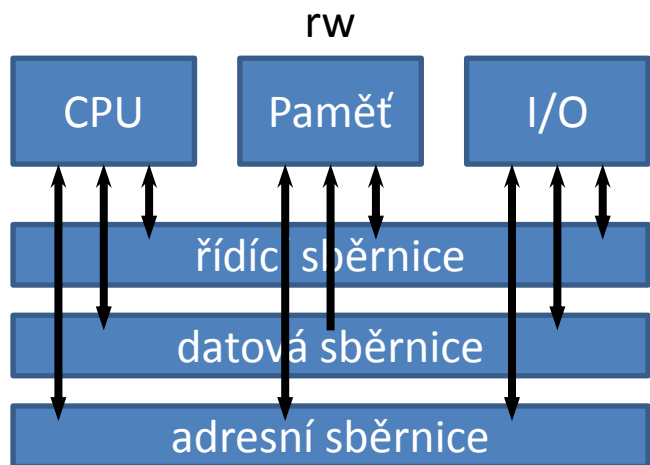


- program se nemůže modifikovat
- program a data se mohou načítat současně

**Harvard Mark I** - počítač složený z relé, 24 bitové instrukce

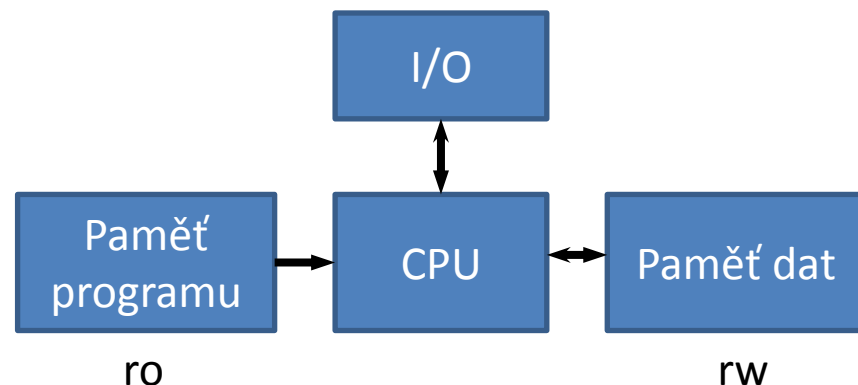
# Přehled

1945 von Neumannova architektura



- program může sebe modifikovat
- program a data nelze načítat současně

1944 Hardwarská architektura



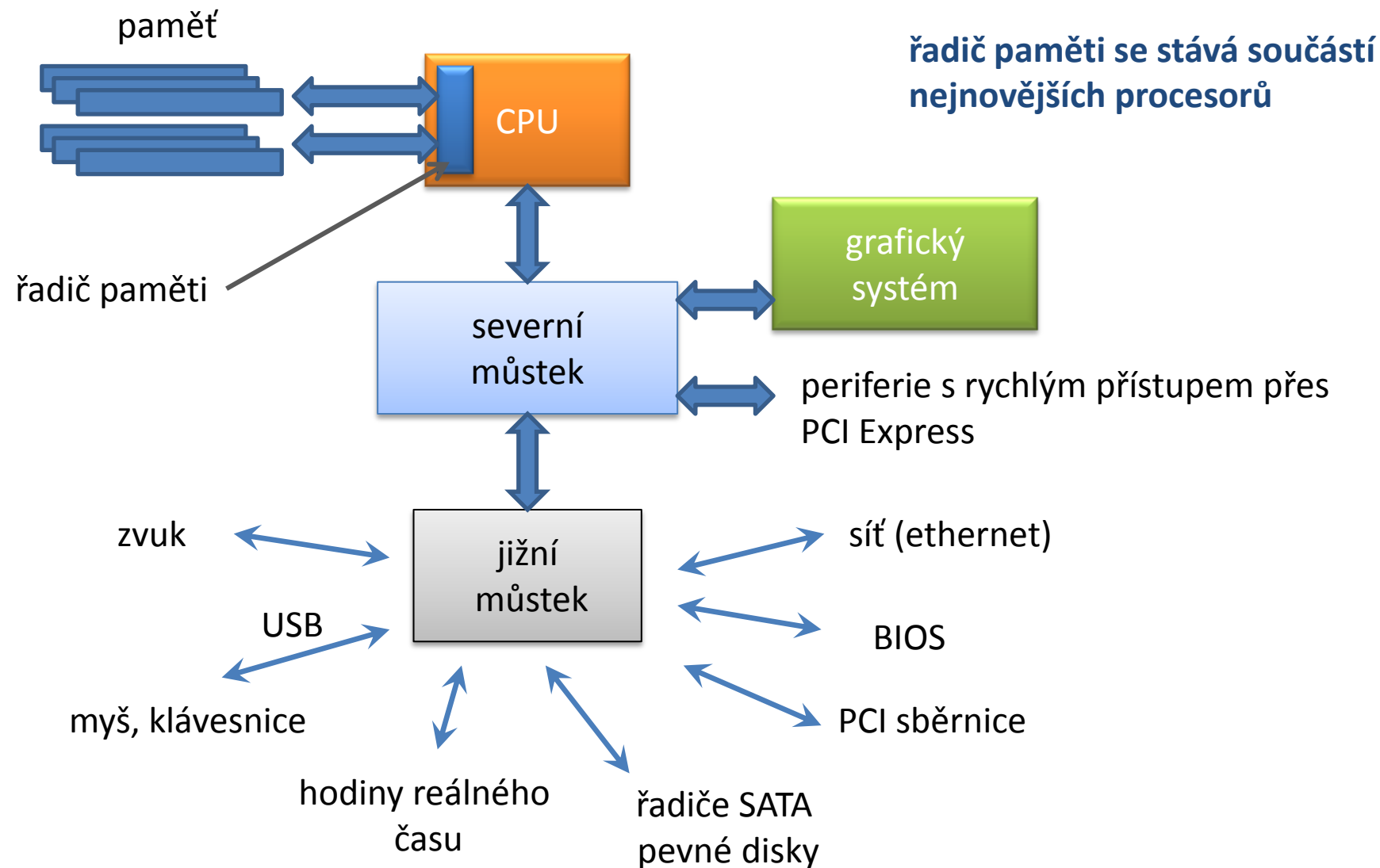
- program se nemůže modifikovat
- program a data se mohou načítat současně

**v dnešních počítačích se kombinují obě architektury**

**John von Neumann**, původem maďarský matematik, působící ve spojených státech

**Harvard Mark I** - počítač složený z relé, 24 bitové instrukce

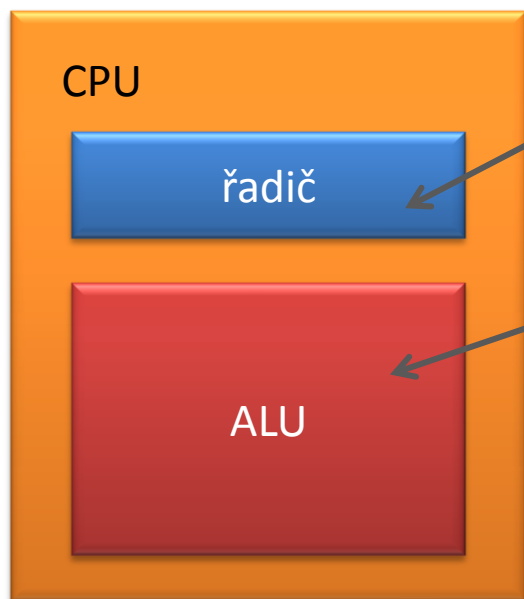
# Typické schéma počítače



# CPU

**Procesor** též **CPU** (anglicky **Central Processing Unit**) je základní součástí počítače; jde o velmi složitý sekvenční obvod, který **vykonává strojový kód** uložený v operační paměti počítače. Strojový kód je složen z jednotlivých strojových instrukcí počítačových programů nahraných do operační paměti.

[www.wikipedia.org](http://www.wikipedia.org)



načítá strojové instrukce a data a připravuje jejich zpracování v ALU

ALU (arithmetic and logic unit), vykonává aritmetické operace, vyhodnocuje podmínky

sekvenční zpracovávání strojových instrukcí je řízeno vnitřním hodinovým taktem

# Reprezentace číselných hodnot



# Celá čísla

Nejmenší jednotkou informace v číslicové technice je jeden **bit**. Z bitů se skládají slova. Nejmenším slovem je **byte**, který obsahuje 8 bitů.

Jeden byte může popsat celá čísla s rozsahem od 0 do 255.

128	64	32	16	8	4	2	1	
0	1	0	1	0	1	1	1	= 87

Lze vyjádřit i celá čísla se znaménkem. V tomto případě je jeden bit vyhrazen pro znaménko, zbývající bity pro číslo. Existuje několik možností implementace. Intel architektura využívá **dvojkový doplněk**, který vede k rozsahu od -128 do 127.

	128	64	32	16	8	4	2	1	
	0	1	1	1	1	1	1	1	= 127
	0	1	0	1	0	1	1	1	= 87
	0	0	0	0	0	0	0	1	= 1
	0	0	0	0	0	0	0	0	= 0
	1	1	1	1	1	1	1	1	= -1
	1	0	1	0	1	0	0	1	= -87
	1	0	0	0	0	0	0	0	= -128

bit vyhrazený pro znaménko

# Celá čísla, II

Celé čísla s větším dynamickým rozsahem lze vyjádřit pomocí větších slov typicky složených ze čtyř bajtů (32 bitové slovo) nebo osmi bajtů (64 bitové slovo).

32 bitové celé číslo bez znaménka:	0 až 4.294.967.295
32 bitové celé číslo se znaménkem:	-2.147.483.648 až 2.147.483.647
64 bitové celé číslo bez znaménka:	0 až 18.446.744.073.709.551.615
64 bitové celé číslo se znaménkem:	-9.223.372.036.854.775.808 až 9.223.372.036.854.775.807

Při práci s celými čísly je nutné brát v potaz, že s nimi **nelze vyjádřit libovolně velké číslo** a je nutné se důsledně vyvarovat možnosti **podtečení** nebo **přetečení** hodnoty.

# Reálná čísla

Reálná čísla se vyjadřují v následujícím formátu (formát s **pohyblivou čárkou**, angl. **floating point**):

$$X = (-1)^s \cdot (1 + Q) \cdot 2^E$$

← exponent

↑  
mantisa

$$Q = m_1 \frac{1}{2^1} + m_2 \frac{1}{2^2} + m_3 \frac{1}{2^3} + m_4 \frac{1}{2^4} \dots$$

$m_1, m_2, m_3$  jsou bity mantisy

V číslicové technice se reálná čísla nejčastěji vyjadřují ve formátu definovaném standardem **IEEE 754**.

typ	šířka	mantisa	exponent
jednoduchá přesnost (single precision)	32	23	8
dvojnásobná přesnost (double precision)	64	52	11

# Reálná čísla, II

typ	rozsah	přesnost
jednoduchá přesnost (single precision)	$\pm 1,18 \times 10^{-38}$ až $\pm 3,4 \times 10^{38}$	přibližně 7 desetinných míst
dvojnásobná přesnost (double precision)	$\pm 2,23 \times 10^{-308}$ až $\pm 1,80 \times 10^{308}$	přibližně 15 desetinných míst

Speciální kombinací hodnoty mantisy a exponentu, lze vyjádřit následující **speciální hodnoty**:

- 0 kladná nula
- 0 záporná nula
- NaN not a number, např. výsledek dělení nulou
- +Inf kladné nekonečno (číslo je příliš velké pro vyjádření)
- Inf záporné nekonečno (číslo je příliš velké pro vyjádření)

Při práci s reálnými čísly je nutné dbát ohled na šíření **zaokrouhlovacích chyb**, v logických porovnáváních **není vhodné** používat operátory **rovná se** a **nerovná se**, kromě situace porovnávání s nulou.

# Cvičení 1

1. Proměnná typu **signed char** (bajt se znaménkem) obsahuje číslo 127. Jakou hodnotu bude mít proměnná pokud ji zvýšíme o jedničku?
2. Proměnná typu **unsigned char** (bajt bez znaménka) obsahuje číslo 88. Jak se číselná hodnota změní, pokud se provede posunutí bitové reprezentace čísla o jednu pozici doprava nebo doleva? Jaký matematický význam má daná operace.
3. Jaký bude výsledek součtu reálných čísel reprezentovaných v dvojnásobné přesnosti a majících hodnotu:  
 $0,1346978 \cdot 10^{-12}$   
 $1,2312657 \cdot 10^6$
4. Co je to big-endian a little-endian? Uvedte architektury, které daný typ endianity používají. Jaký vliv má endianita na přenos binárních dat?

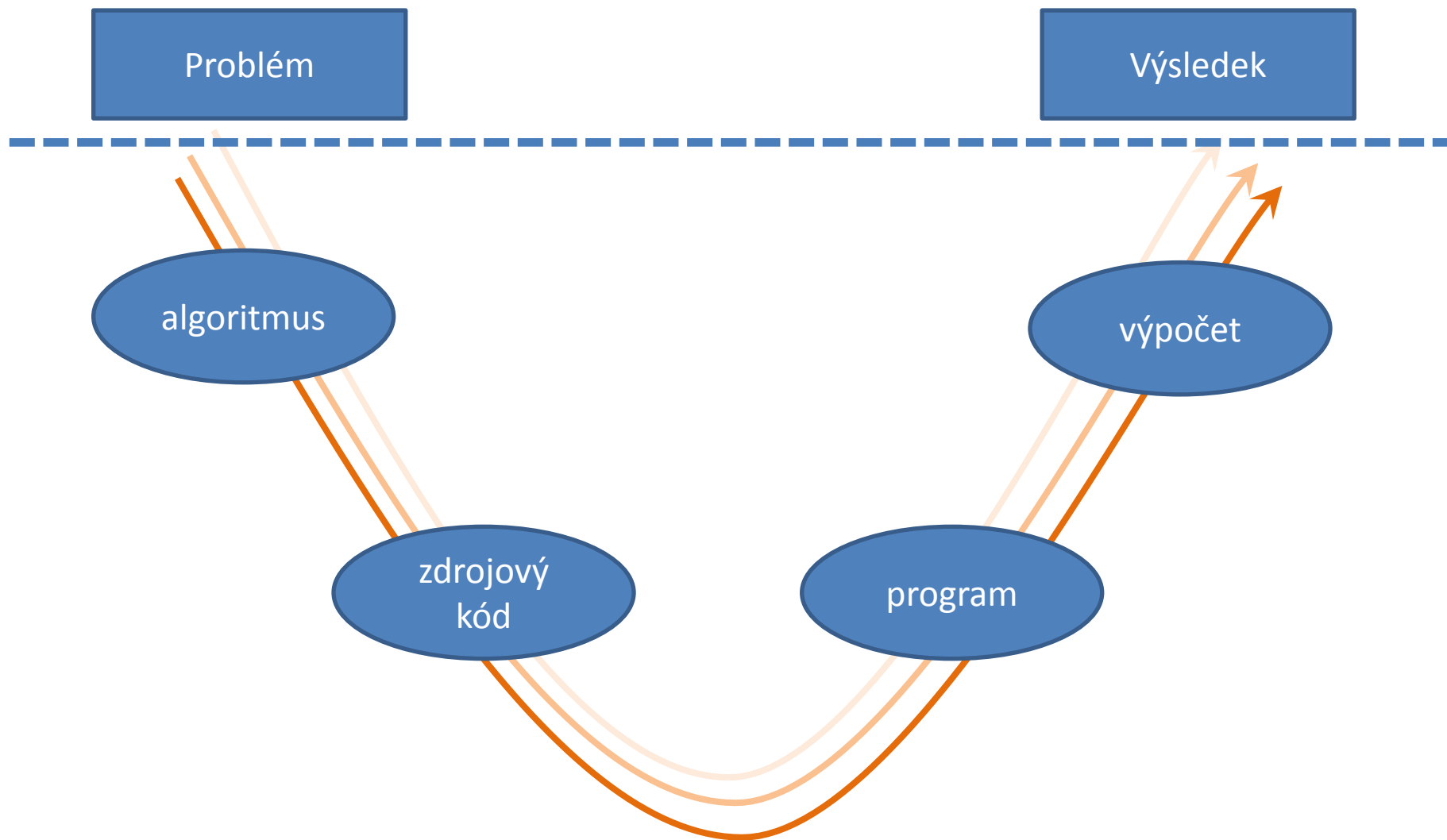
**Společné cvičení:** převod číselných hodnot z dekadické do dvojkové a hexadecimální soustavy

# Od problému k výsledku

# Od problému k výsledku ...

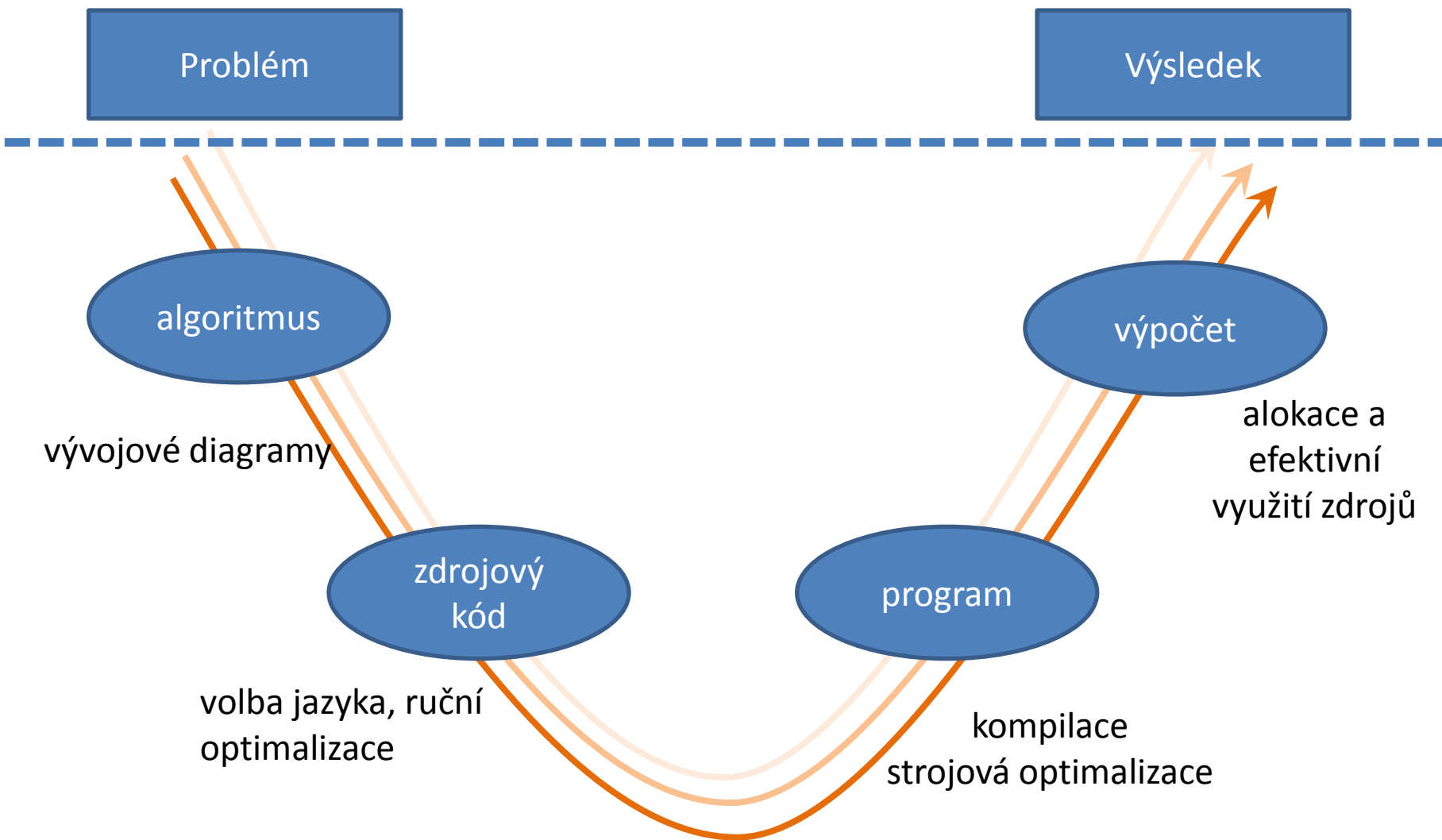


# Od problému k výsledku ...

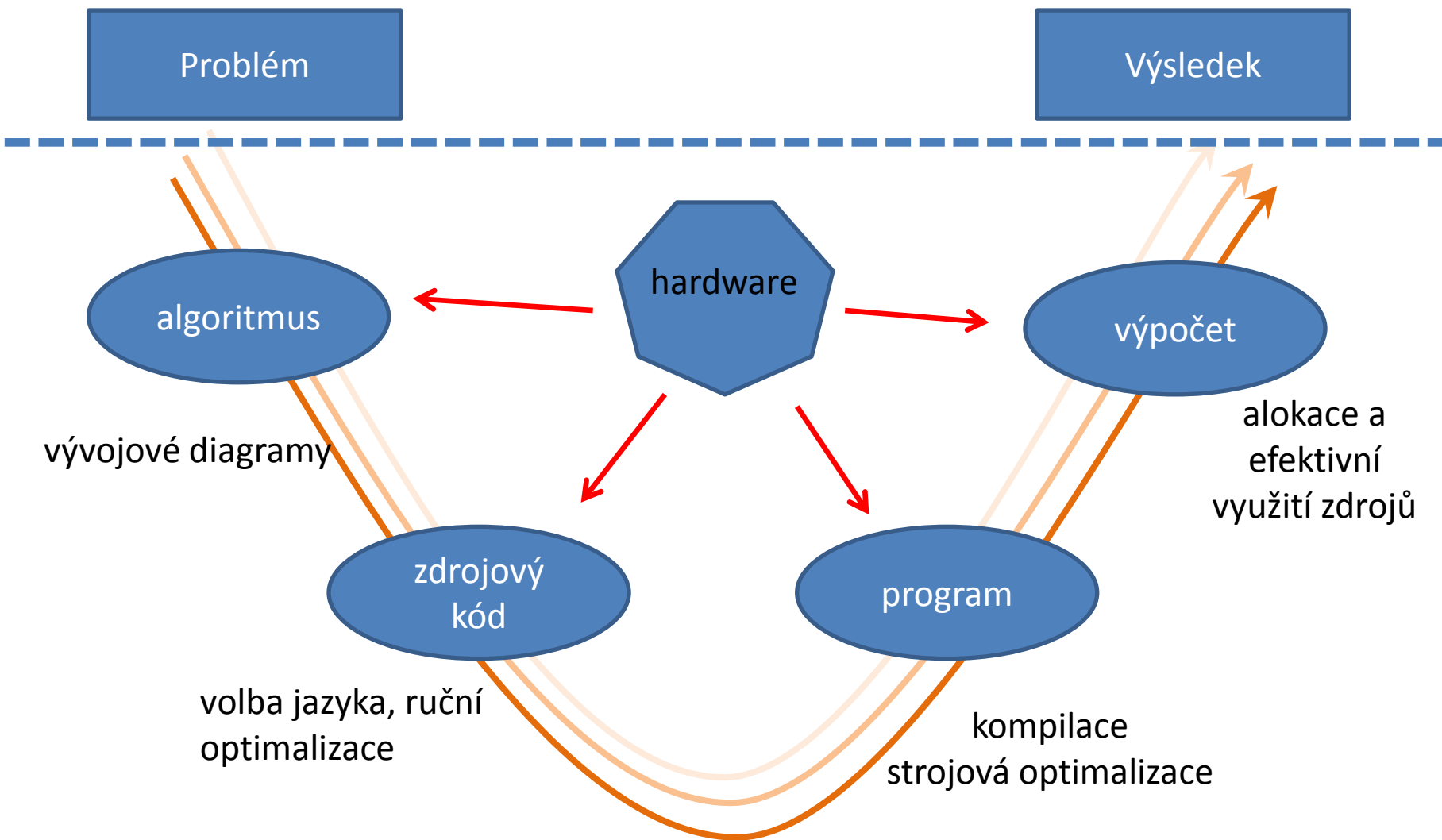




# Od problému k výsledku ...

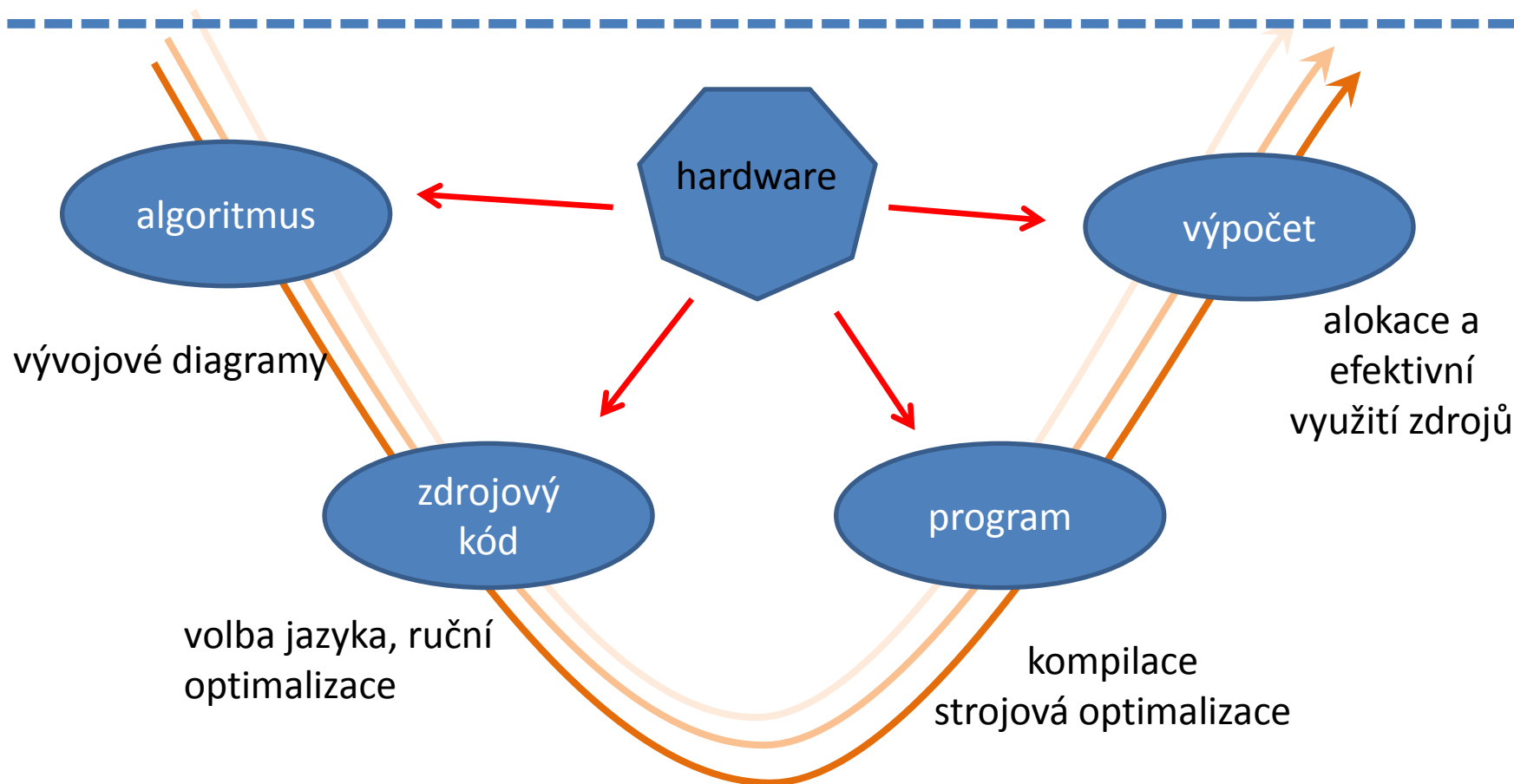


# Od problému k výsledku ...



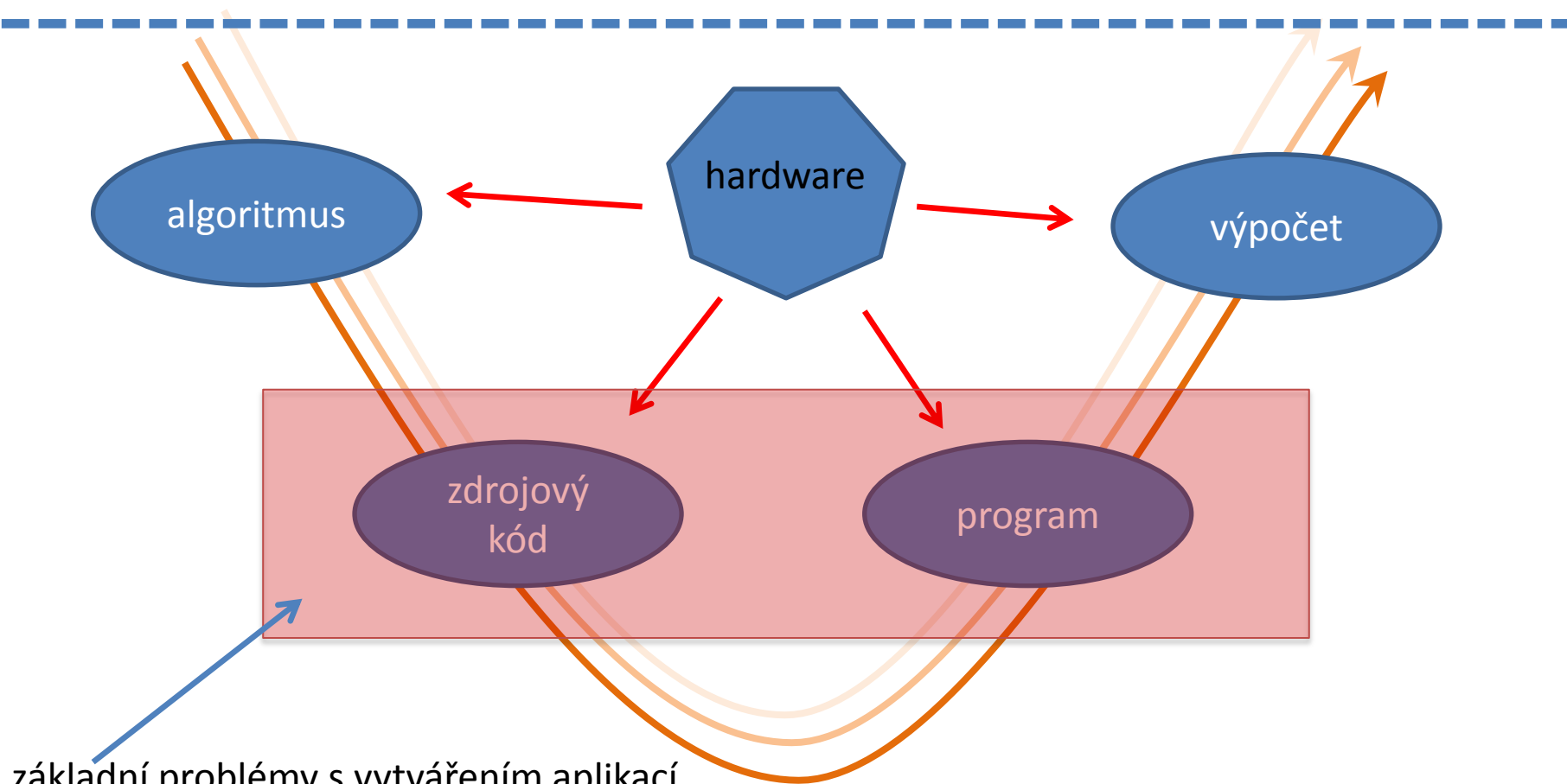
# Od problému k výsledku ...

Při řešení problémů za použití výpočetní techniky (superpočítačů) je nutné **komplexně zhodnotit** celou řadu aspektů, které zahrnují i použitý hardware a jeho architekturu.



# Probírané okruhy ...

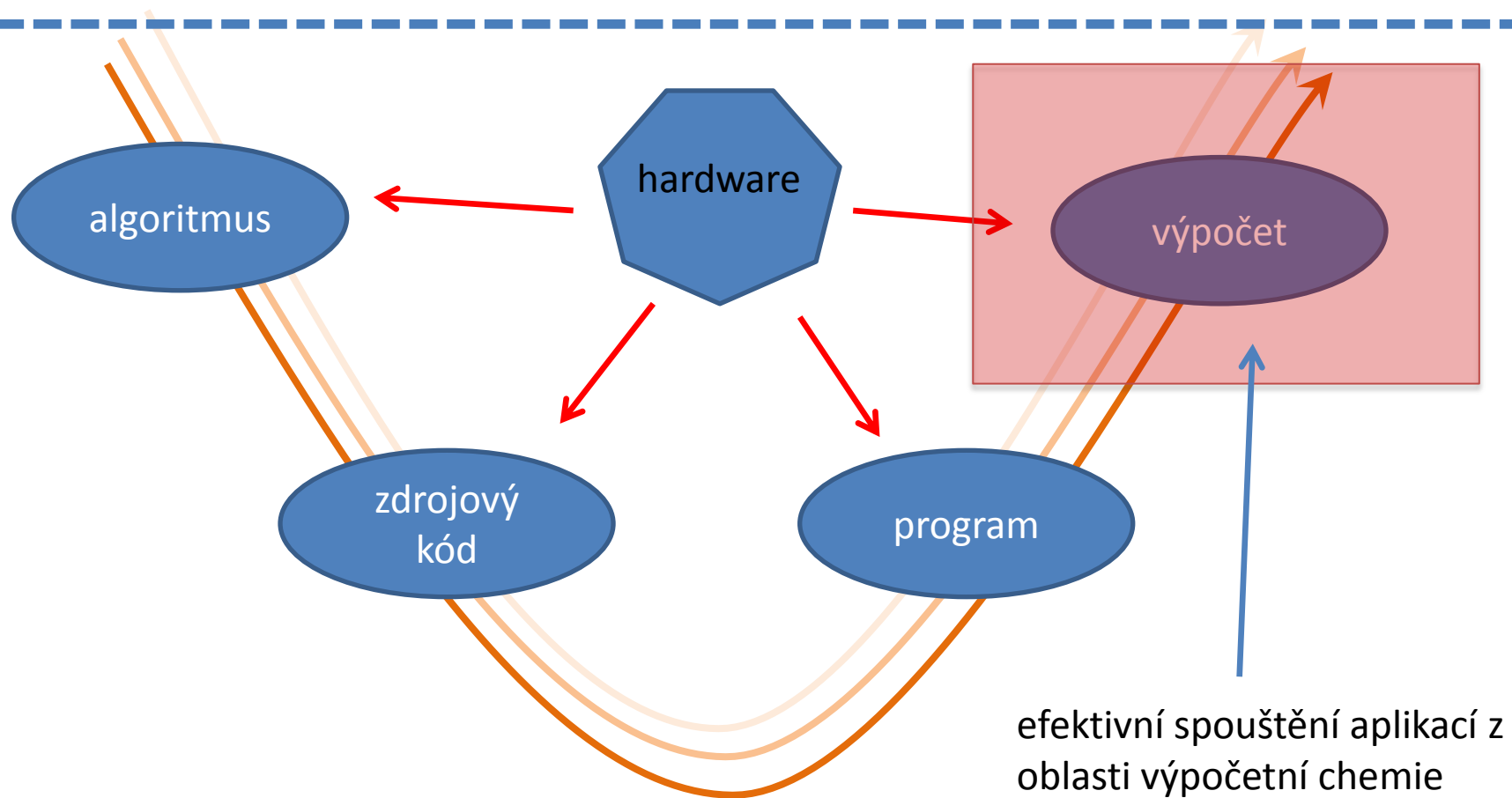
Při řešení problémů za použití výpočetní techniky (superpočítačů) je nutné **komplexně zhodnotit** celou řadu aspektů, které zahrnují i použitý hardware a jeho architekturu.



základní problémy s vytvářením aplikací  
pro náročné výpočty, paralelizace

# Probírané okruhy ...

Při řešení problémů za použití výpočetní techniky (superpočítačů) je nutné **komplexně zhodnotit** celou řadu aspektů, které zahrnují i použitý hardware a jeho architekturu.



efektivní spuštění aplikací z oblasti výpočetní chemie (MetaCentrum, malé klastry)

# Programy vs Skripty

**Program** je soubor strojových instrukcí zpracovávaných přímo procesorem. Program vzniká **překladem** zdrojového kódu programovacího jazyka.

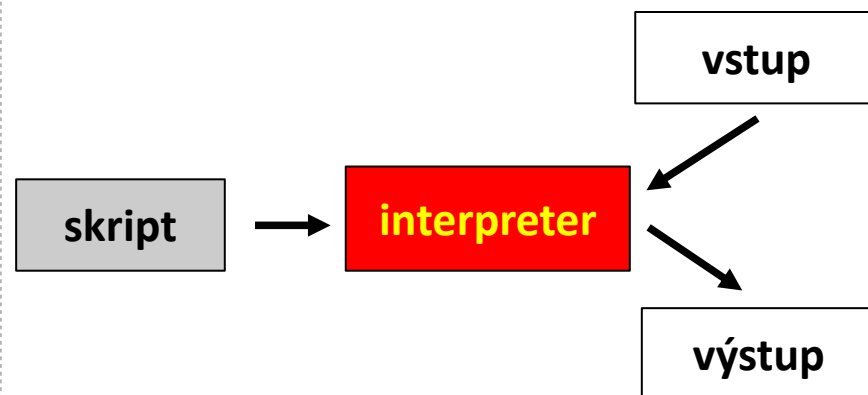
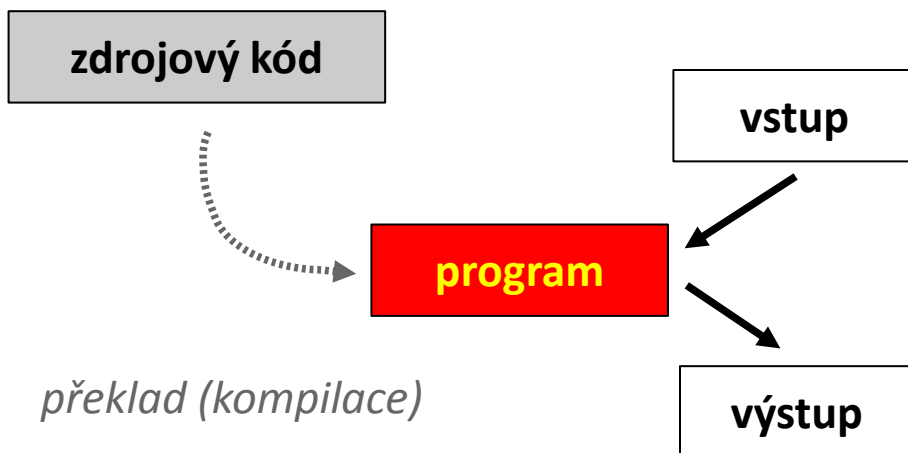
**Překládané jazyky:**

- C/C++
- Fortran

**Skript** je textový soubor obsahující příkazy a řídicí sekvence, které jsou vykonávány **interpretem** použitého **skriptovacího jazyka**.

**Skriptovací jazyky:**

- bash
- gnuplot
- awk
- JavaScript
- PHP



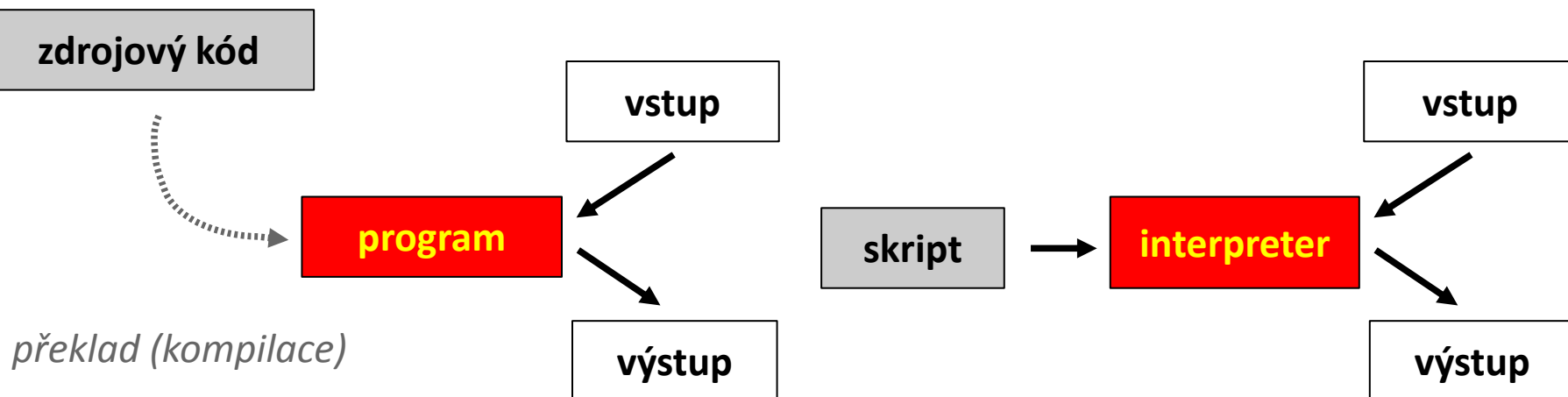
# Programy vs Skripty

- snadná optimalizace
- rychlé vykonávání

- nutnost rekompilace
- nelze vytvářet samospustitelný kód

- nevyžaduje rekompilaci
- vytváření samospustitelného kódu

- špatná optimalizovatelnost
- pomalejší vykonávání



# Programy vs Skripty

Programy určené pro **náročné vědeckotechnické výpočty** jsou vždy psané v **kompilovatelných programovacích jazycích**. Mezi tyto jazyky patří:

- Fortran
- C/C++

**Skriptovací jazyky** se pro náročné výpočty buď vůbec nepoužívají nebo se používají v **podpůrných částech výpočtů**, které nejsou výpočetně náročné.



# Cvičení 2

1. Určete procentuální zastoupení programů napsaných v jazyku Fortran, C/C++ a jiný, které jsou uvedeny na následující stránce:

[http://en.wikipedia.org/wiki/List\\_of\\_quantum\\_chemistry\\_and\\_solid\\_state\\_physics\\_software](http://en.wikipedia.org/wiki/List_of_quantum_chemistry_and_solid_state_physics_software)

Výsledek znázorněte ve formě výsečového grafu.