MASARYKOVA UNIVERZITA

Informační technologie

Jaroslav Šmarda

Informační technologie

- Historie počítačů
- Koncepce počítače
- Procesor
- Operační systém
- Programovací jazyky

Historie počítačů

- 1833 - Angličan Charles Babage tzv."Analytical Engine" pracující s děrnými štítky
- 1938- německý stavební inženýr Konrád Zuse sestrojil první mechanický počítač Z1
- 1941 - Z3 od Konrada Zuse první reléový (zničen 1944) – dvojková soustava
- 1946 - americký ENIAC - desítková soustava
- 1952 - počítač IAS vzor pro první velkosériově vyráběný počítač IBM 701.
- 1958 - v Československu první reléový počítač SAPO
- 1948 - poprvé předveden tranzistor, komerčně v roce 1952

Generace počítačů

- Generace počítačů podle použitých s

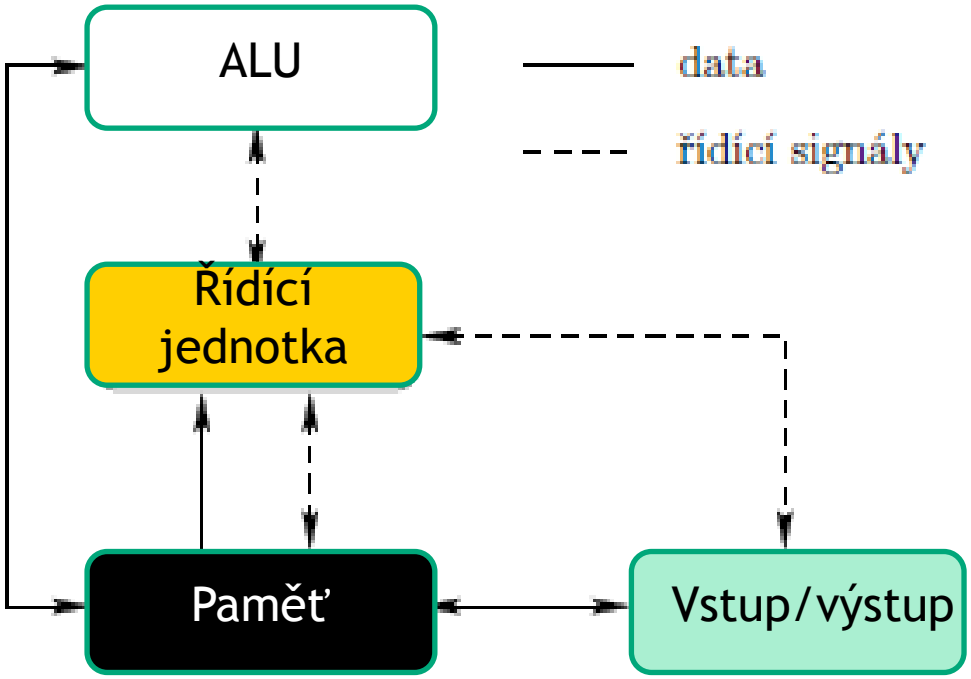
Generace	Rok	Použité součástky
1.	1950	Elektronky
3.	1964	Integrované obvody
4.	1981	VLSI (Very LSI)

- Počítače 5. generace jsou charakterizovány kvalitativně odlišnými prostředky komunikace s uživatelem, operační rychlostí a prvky umělé

Koncepce počítače (John von Neumanna z let 1946-1952)

- Počítač obsahuje:
  - lineárně organizovanou **operační paměť**, která je rozdělena na stejně velké buňky
  - **procesor s ALU** (Arithmetic Logic Unit) , **řadičem** a **datovými registry**
  - **V/V zařízení**

Koncepce počítače (John von Neumanna z let 1946-1952)



Koncepce počítače: data a instrukce

- Data a instrukce
  - jsou vyjádřeny **binárně**, nejsou explicitně označeny (speciálně platí i pro datové typy)
- Data a instrukce
  - se uchovávají v jedné paměti na místech označených **adresami** (Podle výpisu paměti nelze poznat, jestli jde o data, či instrukce)
- Koncepce platí dodnes
  - změna jen v počtu procesorů

Paměť počítače

- Nejmenší jednotka paměti:
  - Bit (Binary Digit) – 0 nebo 1
- Základní jednotka paměti:
  - Byte = 8 bitů
  - 0 1 0 0 1 1 1 0
  - 2x2x2x2x2x2x2x2 = 2^8 = 256 možností

Paměť počítače

- 4 bity (1/2 bytu)
  - 0 1 1 1
  - 2x2x2x2 = 16 možností
  - šestnáctková soustava (hexadecimální)

Paměť počítače

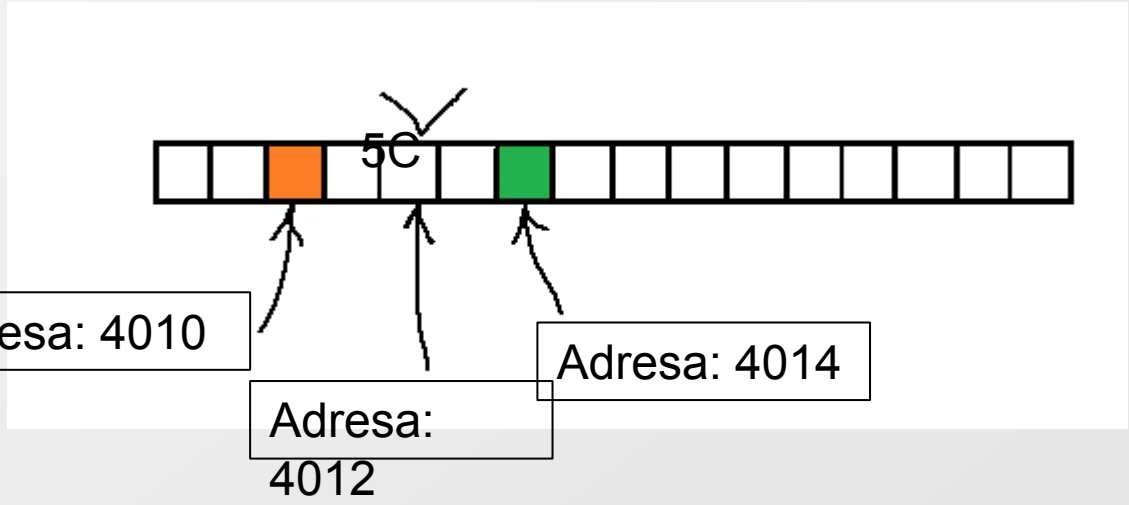
7D 30 FF 4A

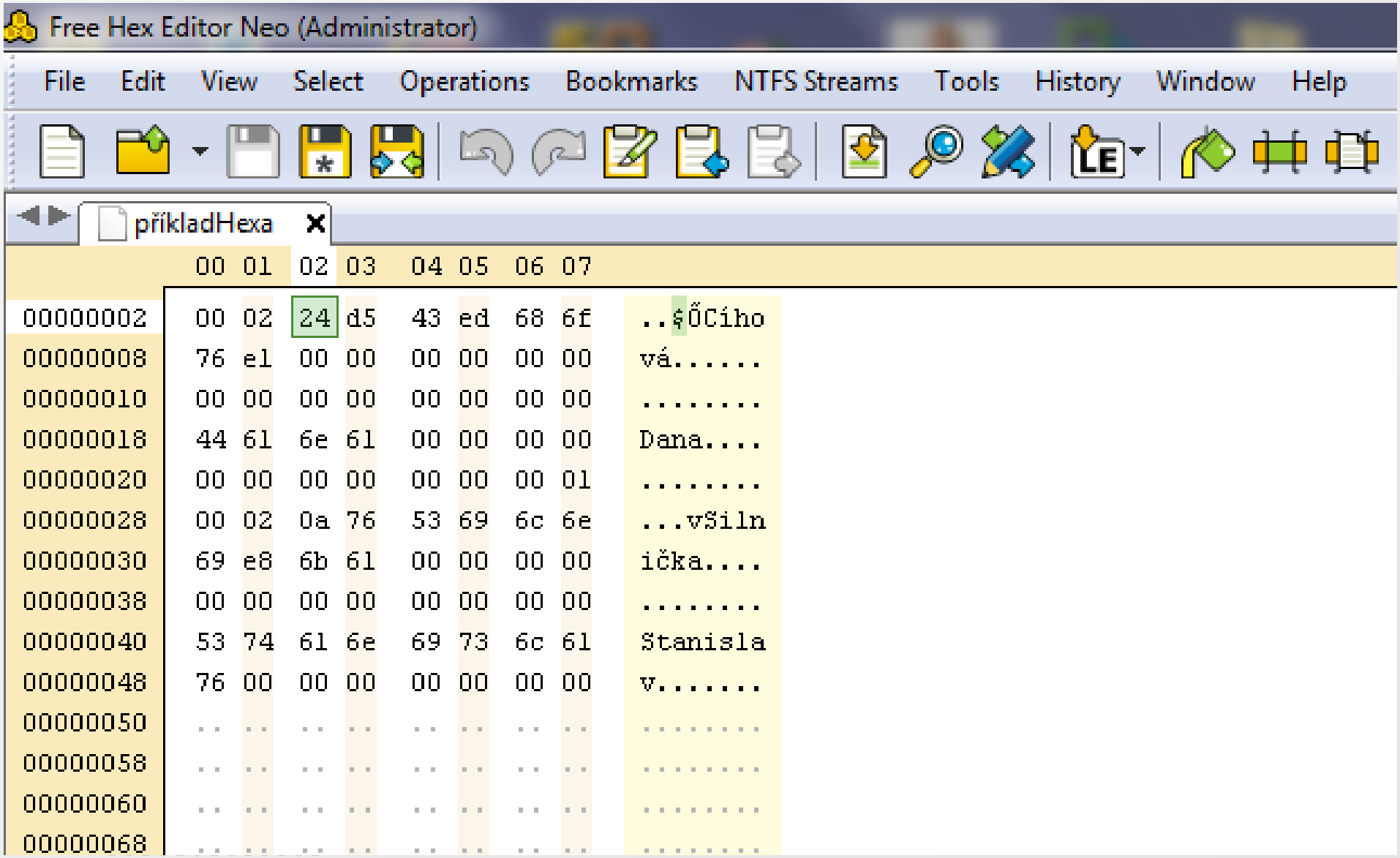
0111 1101
0011 0000
1111 1111
0100 1010

4 bity	Hexadecimální číslice
0 0 0 1	1
0 0 1 1	3
0 1 0 1	5
0 1 1 1	7
1 0 0 1	9
1 0 1 1	B
1 1 0 1	D
1 1 1 1	F

Paměť počítače

- Paměť s přímým přístupem (RAM – Random Access Memory)
- Všechna paměťová místa kdykoliv dostupná přes adresu




Free Hex Editor Neo (Administrator)
File Edit View Select Operations Bookmarks NTFS Streams Tools History Window Help
příkladHexa
00 01 02 03 04 05 06 07
00000002 00 02 24 d5 43 ed 68 6f ..$ŐCího
00000008 76 e1 00 00 00 00 00 00 vá.....
00000010 00 00 00 00 00 00 00 00 ....
00000018 44 61 6e 61 00 00 00 00 Dana....
00000020 00 00 00 00 00 00 00 01 ....
00000028 00 02 0a 76 53 69 6c 6e ...vSiln
00000030 69 e8 6b 61 00 00 00 00 ička....
00000038 00 00 00 00 00 00 00 00 ....
00000040 53 74 61 6e 69 73 6c 61 Stanisla
00000048 76 00 00 00 00 00 00 00 v.......
00000050 .. ..
00000058 .. ..
00000060 .. ..
00000068 .. ..

Program – posloupnost instrukcí

- Předpis pro řešení úlohy
- Procesor interpretuje (postupně provádí) instrukce uložené v paměti v pořadí, ve kterém jsou uloženy v paměti, pokud toto pořadí není změněno speciálními instrukcemi
- Změna pořadí provádění instrukcí:
  - instrukce podmíněného a nepodmíněného skoku

Rozdělení instrukcí procesoru

- Přesunové
  - mezi registry procesoru nebo mezi operační pamětí a registry
- Aritmetické
  - sčítání, odčítání, …
- Logické
  - log. součet, log. součin, rotace a posuvy, …
- Skoku
  - při rozhodování, …
- vstupně výstupní
  - pro práci s periferními zařízeními, …
- Ostatní
  - řídicí, …

Příklady instrukcí procesoru Intel

Assembler	operandy	název instrukce
mov	2	move
add	2	add
sub	2	subtract
mul	1	multiply (unsigned)
imul	1,2,3	multiply (signed)
div	1	divide (unsigned)
idiv	1	divide (signed)
inc	1	increment
dec	1	decrement
shr	2	shift right (unsigned)
sar	2	shift arithmetic right (signed)
shl	2	shift left

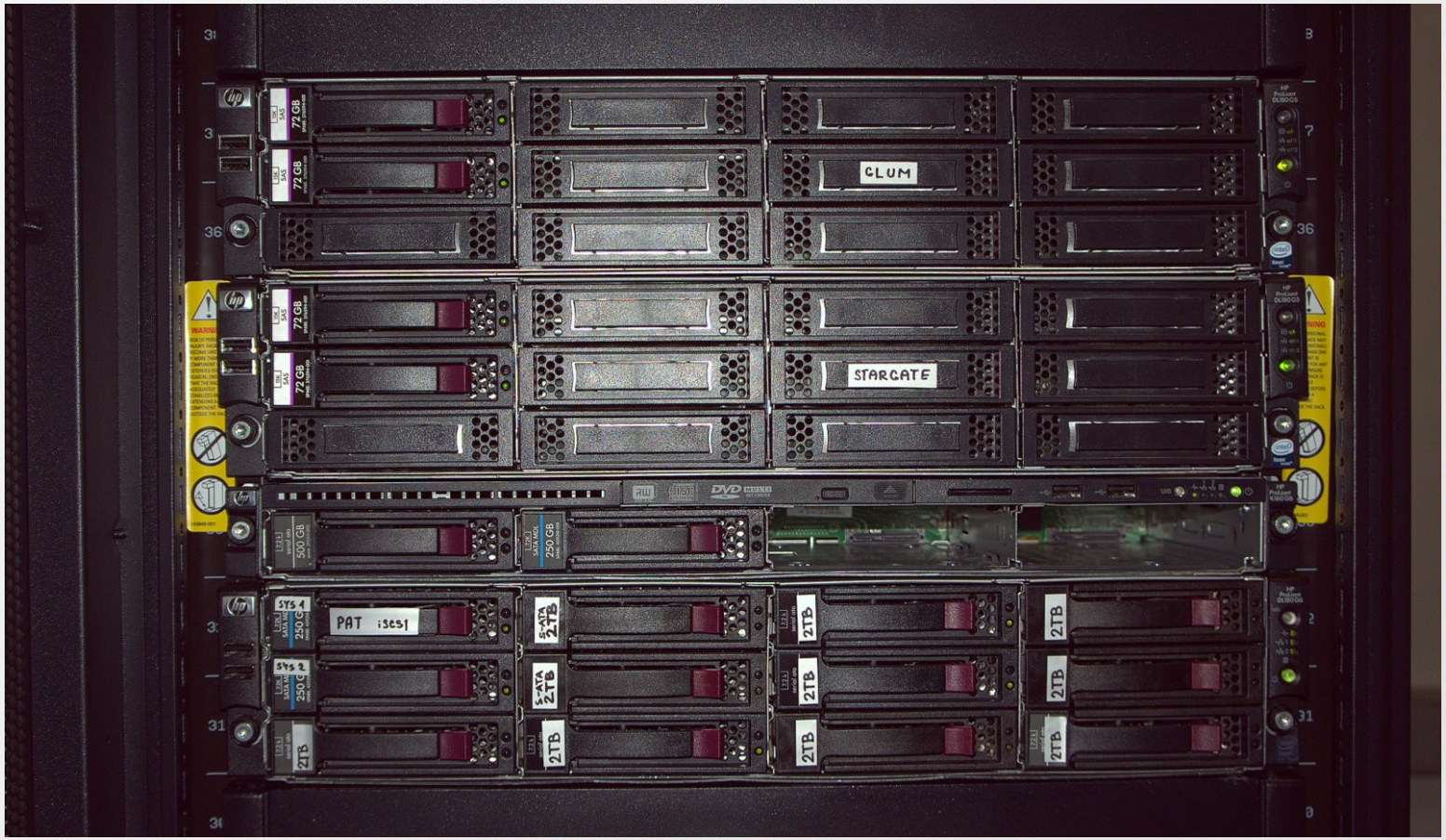
Dvě základní koncepce procesorů

- CISC (Complete Instruction Set Computer)
  - úplná instrukční sada (Intel)
- RISC (Reduced Instruction Set Computer)
  - redukovaná instrukční sada.
  - pro vykonání 80 % operací je zapotřebí cca 20% instrukcí
  - méně časté instrukce nahrazeny programem ze základních instrukcí
  - Zjednodušení procesoru, zvýšení rychlosti (A4 pro iPhone a iPad)

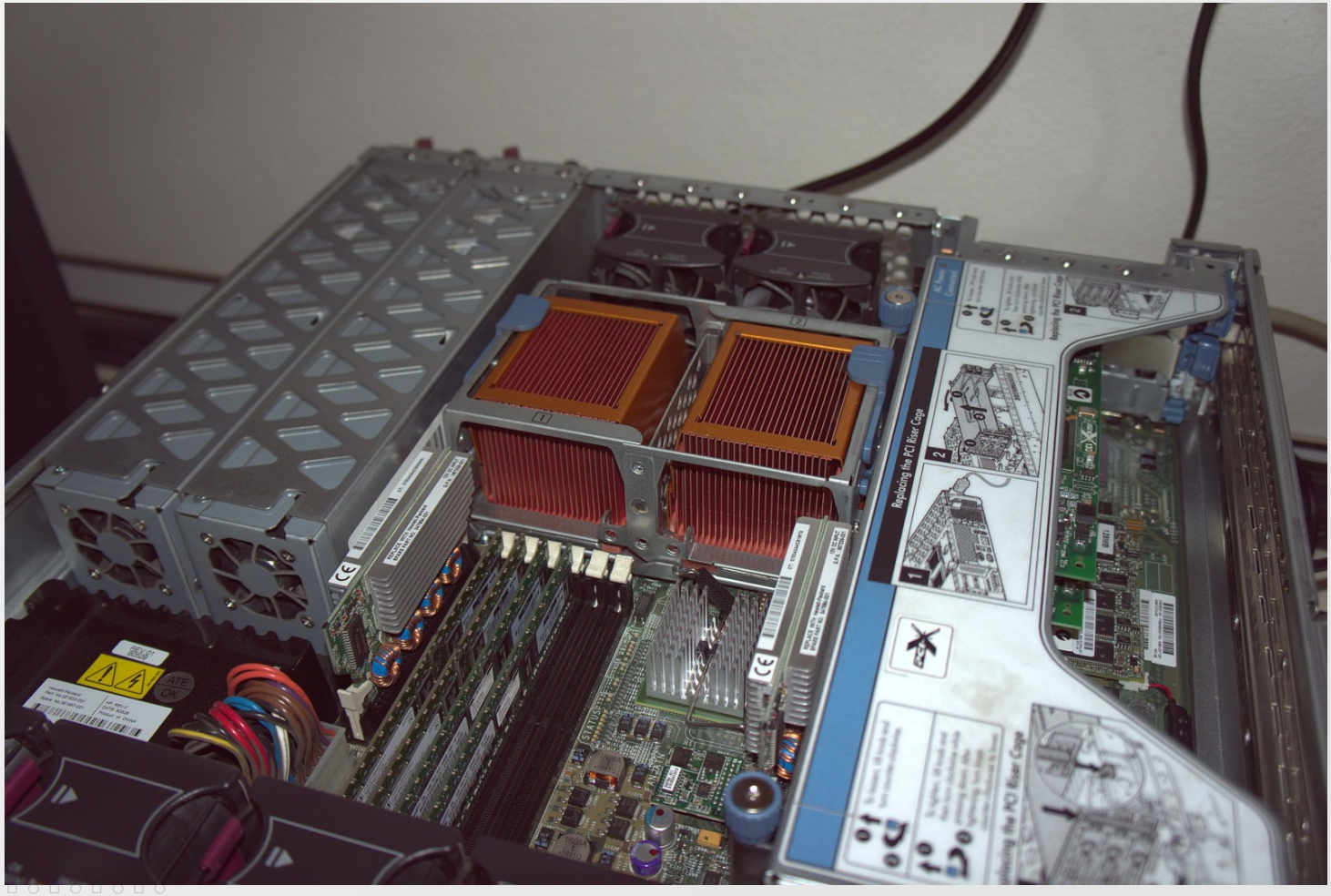
Procesor A5 (RISC ARM)

Základní deska notebooku (Intel – CISC)

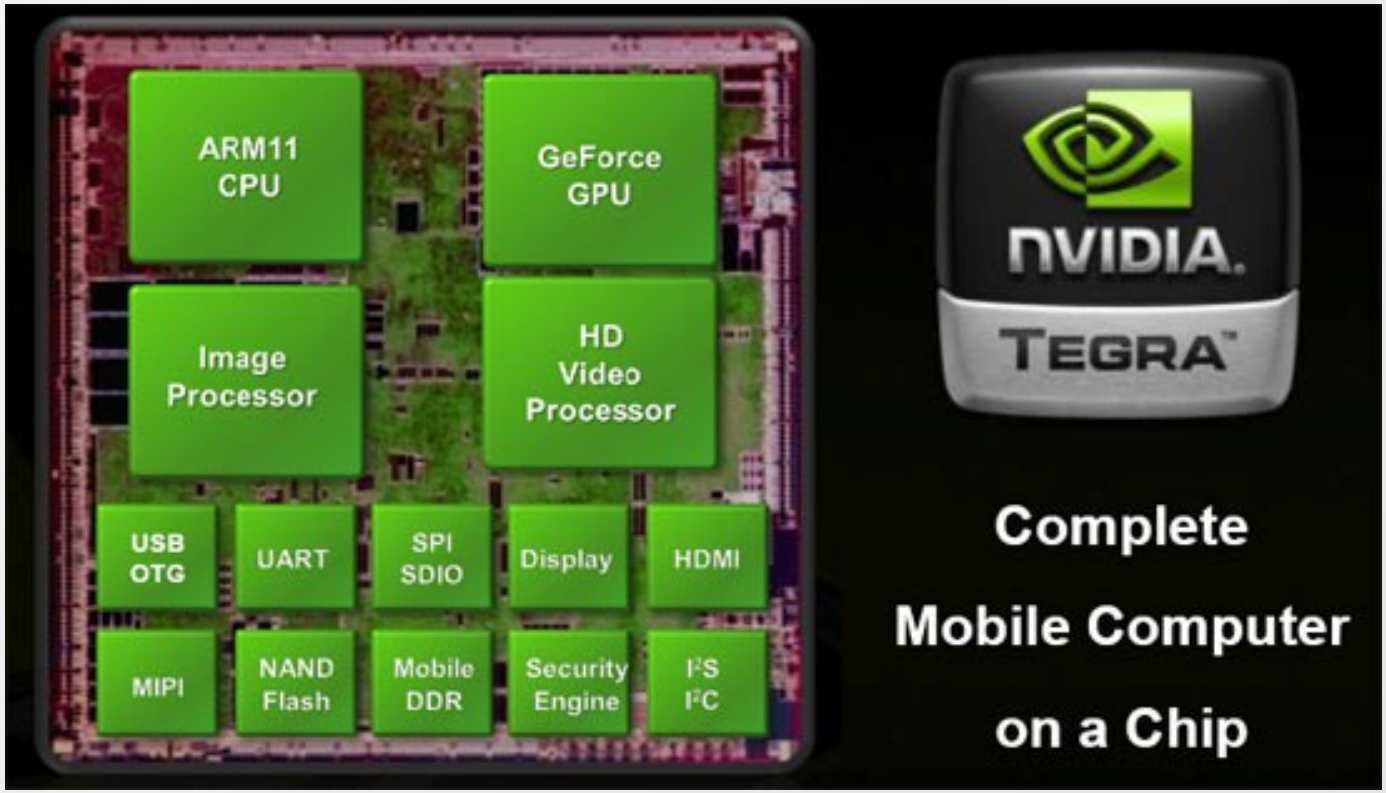
3 servery



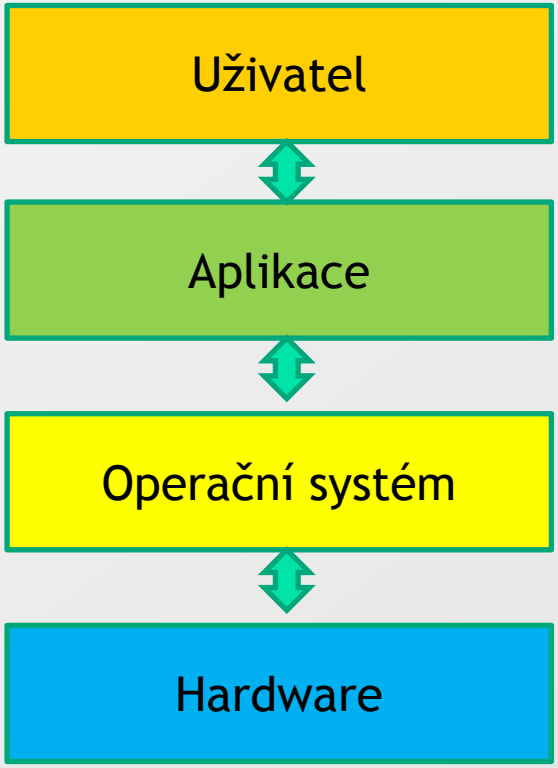
Server se dvěma procesory (Intel – CISC)



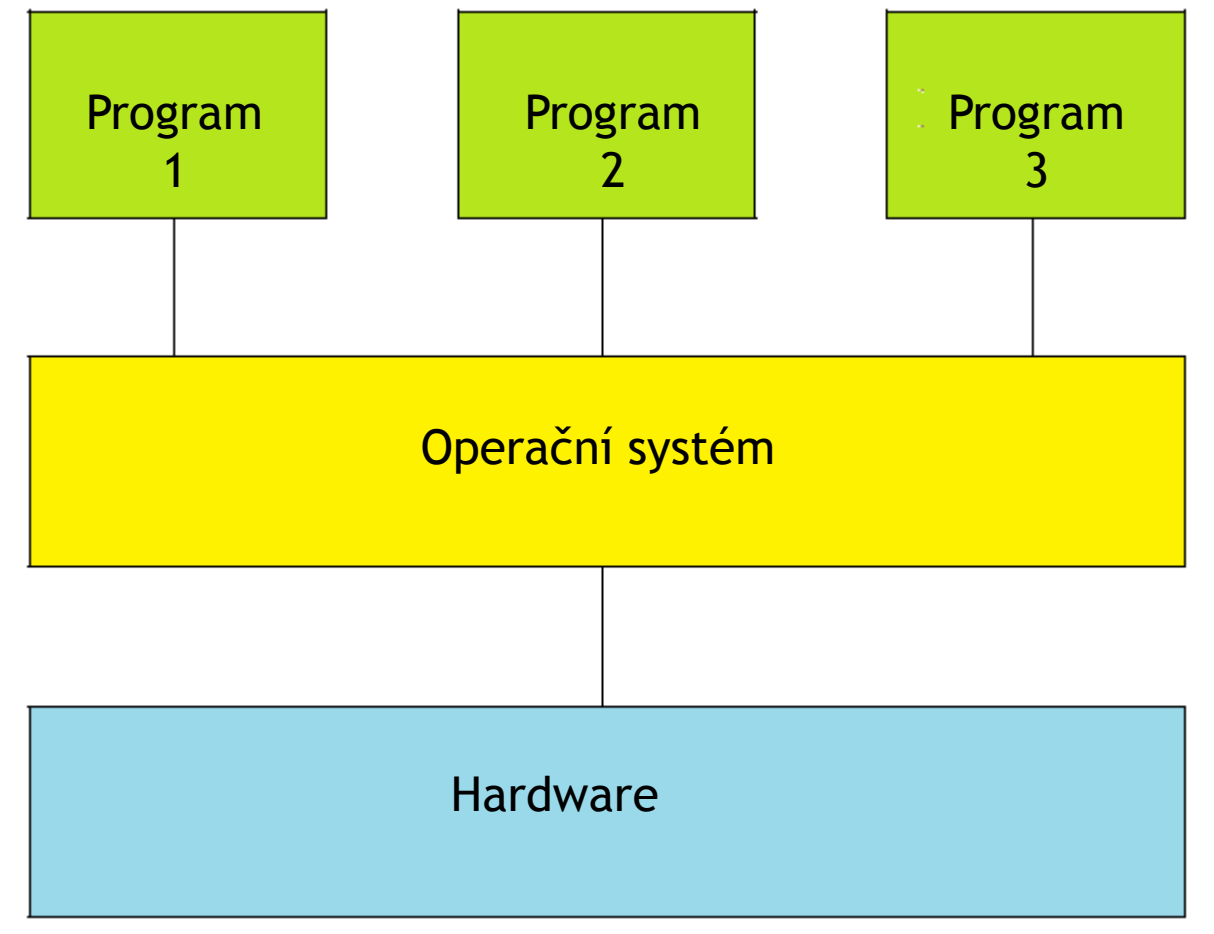
Nvidia Tegra



Operační systém



Softwarová struktura počítače



Systémový software – operační systém (OS)

- Operační systém (OS)
  - prostředník mezi HW a aplikačními programy
  - rozhraní:
    - pro hardware
    - pro aplikační programy
  - běží v každém počítači (i v iPhone)
  - abstrakce pro ovládání hardware (zjednodušení aplikačních programů)

Operační systém

- více programů paralelně
- správa zdrojů (procesor, paměť, obrazovka, klávesnice, tiskárny, další V/V zařízení)
- efektivní využití hardwaru, který je k dispozici

Příklady operačních systémů

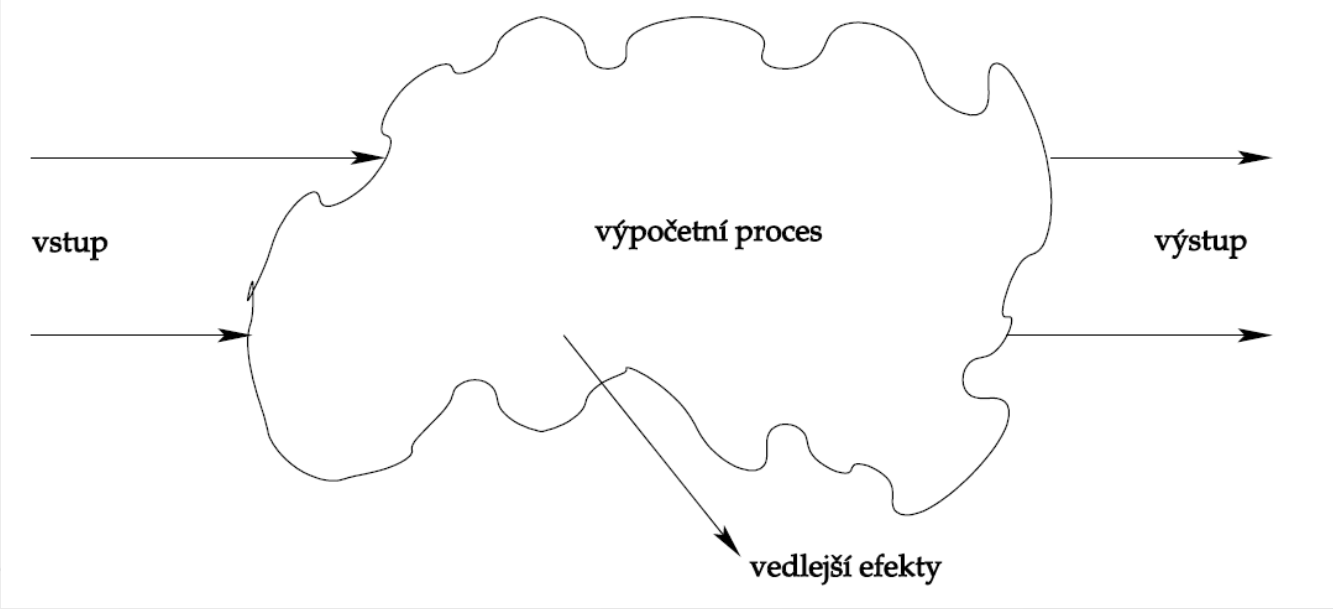
- Windows 7
- Linux
- Android
- Mac OS X
- Mac iOS
- Chrome OS

Aplikační programy

- **Aplikační program** nebo **aplikace:**
  - řeší jednu nebo několik vzájemně propojených úloh
  - pomáhá uživateli řešit problémy reálného světa
  - naplňuje potřeby uživatele
  - Příklady:
    - podnikový software, účetní software, kancelářský program, grafický program, počítačová hra
- Systémový software (komponenta OS):
  - řídí a integruje počítačové zdroje

Výpočetní proces v počítači

- posloupnost elementárních kroků s počátkem a koncem přetvářející vstupní data (vstupní parametry na výstupní (řešení)
- je důsledkem vykonávání programu



Programovací jazyky

- Program
  - předpis s přesně daným tvarem a významem, podle kterého vzniká výpočetní proces
  - je pasivní entita – text
- Programovací jazyk
  - smluvená pravidla, v souladu se kterými je program vytvořen
  - program je napsán v programovacím jazyku
- Algoritmus
  - přesný návod či postup, kterým lze vyřešit daný typ úlohy
  - teoretický princip řešení problému

2 etapy programování

- éra neprogramovatelných počítačů –jednoúčelové, program hardwarovou součástí počítače
  - Například dešifrovací počítače k dešifrování šifer německého stroje Enigma
- éra programovatelných počítačů –univerzální počítač
  - Prvním byl Z3

Programovací jazyky

- nižší
  - přímá vazba na hardware =>
    plné využití počítače
  - žádné prostředky abstrakce =>
    zdlouhavé programování, chyby, které se špatně opravují
- vyšší
  - není přímá vazba na hardware
  - vyšší prostředky abstrakce=>
    efektivnější programování, méně chyb

Nižší programovací jazyky

- Kódy stroje

001	01000	vlož číslo 8 do paměťové buňky
011	00101	přičti k obsahu buňky číslo 5
100	00000	vytiskni obsah buňky na obrazovku
111	00000	ukonči program

- Assemblery
  - pro zápis instrukcí použity mnemotechnické zkratky

load	8	vlož číslo 8 do paměťové buňky
add	5	přičti k obsahu buňky číslo 5
print		vytiskni obsah buňky na obrazovku
stop		ukonči program

Nižší programovací jazyky

- Bajtkódy
  - Bajtkódy (anglicky bytecode) lze chápat jako jazyky stroje pro virtuální procesory (nejsou hardwarové, ale softwarové)

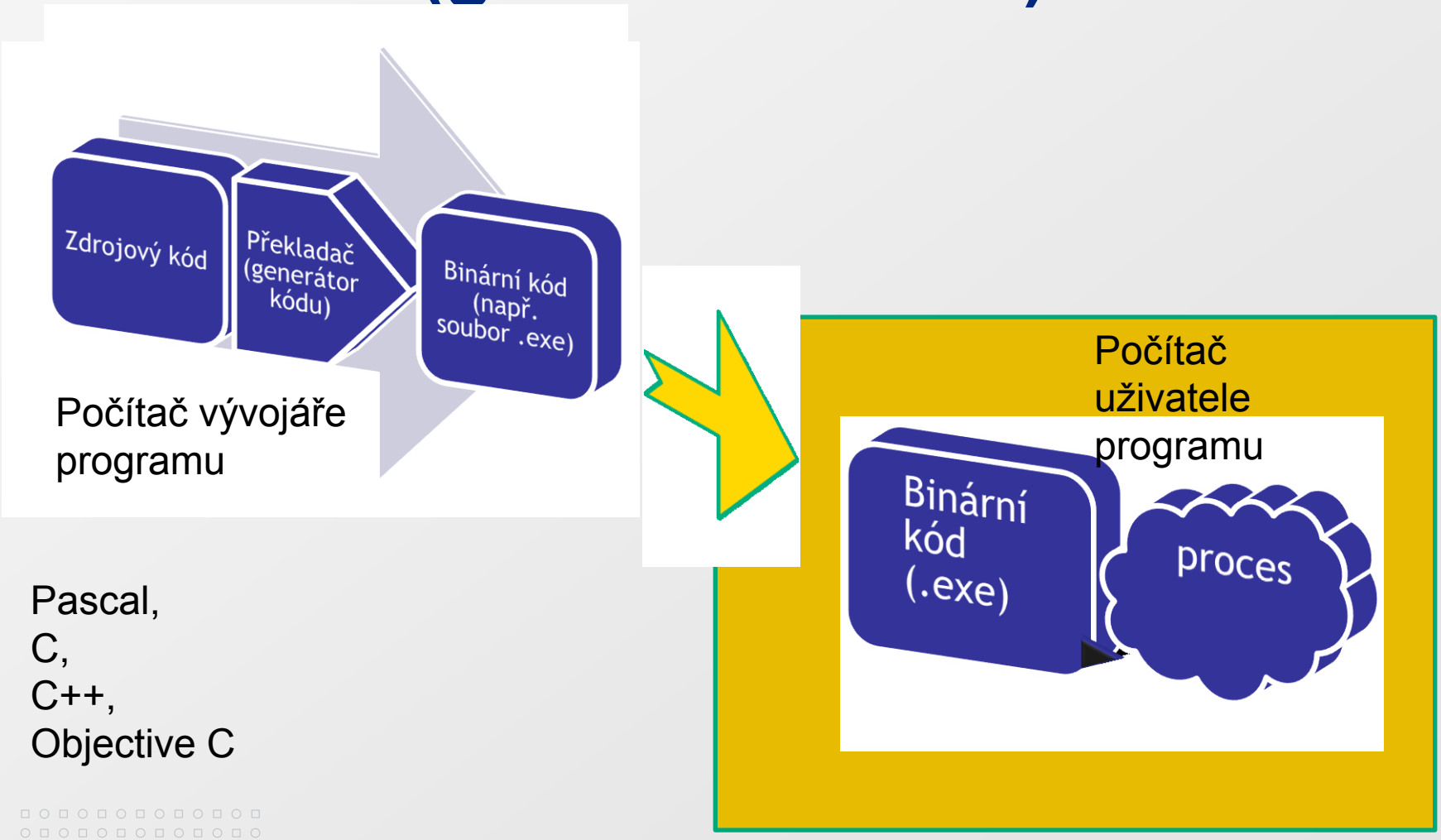
Vyšší programovací jazyky

- Nejsou vázány na konkrétní hardware
- Historie:
  - FORTRAN (1953)
  - ALGOL (1958)
  - COBOL (1959)

Vyšší programovací jazyky

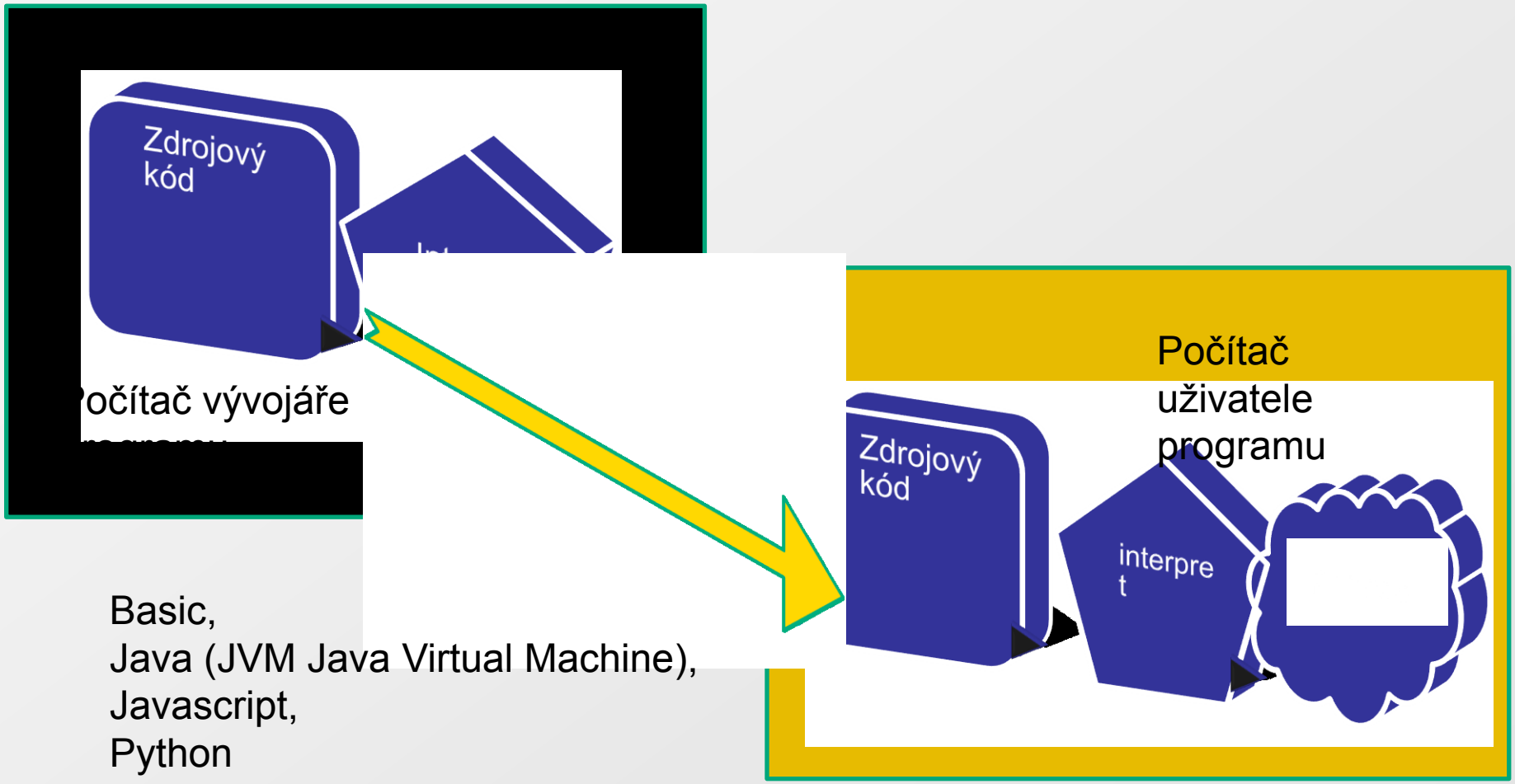
- PASCAL
  - původně pro výuku, strukturované programování
- C
  - Unix
- Java
  - Internet
- C++
- Objective C
  - Apple
- C#
  - Windows .NET
- Python
- Basic

Překladač (generátor kódu)



Pascal,
C,
C++,
Objective C

Překladač (interpret)



Basic,
Java (JVM Java Virtual Machine),
Javascript,
Python

Syntax a sémantika programu

- Dva úhly pohledu na programy:
  - Syntax
    - způsob nebo tvar, ve kterém se programy zapisují
    - soubor přesně daných pravidel definujících, jak zapisujeme programy v daném programovacím jazyku
  - Sémantika
    - význam programu

Program: Hello World!

V jazyce Pascal:

```
program HelloWorld;
begin
  writeln('Hello World');
end.
```

V jazyce Python:

```
print "Hello, World!"
```

Program: Hello World!

V jazyce C:

```
#include <stdio.h>
main() {
    printf ("Hello World!\n");
}
```

V jazyce Java:

```
public class HelloWorld {
  public static void main (String[]
    args) {
      System.out.println("Hello,
    world!\n");
  }
}
```

Program: Hello World!

V jazyce Objective C:

```
#import <stdio.h>
int main ( int argc, const char *argv[]
    ){ printf( "hello world\n" );
    return 0;
}
```

Syntax a sémantika

- Syntaktická chyba je chybou v zápisu programu:
  - printf ("Hello World!\n);
- Sémantická chyba je chybou ve významu programu:
  - a= 10;
    b=0;
    c=a/b;