

- #1: 06/10/4

*Na první a tvrté p ednášce jsem si (celkem zbyte n ) psal skoro všechno, dál jsou jen ty zajímavé v ci nebo nic (viz komentá u druhé p ednášky).*

- **Úvod (Principy počítačů SWI087)**

- RNDr. David Obdržálek, david.obdrzalek@mff.cuni.cz predmet [SWI087]
- osobně 2. patro 209
- témata
  - historie
  - architektura
  - reprezentace dat
  - procesory
  - paměti
  - propojovací systémy
  - periferie
  - novinky
- literatura
  - Internet
    - také [download.matfyz.info](http://download.matfyz.info)—SWI087
    - [www.ksi.ms.mff.cuni.cz](http://www.ksi.ms.mff.cuni.cz)—pp jen z mff.cuni.cz (neveřejné)
  - W. Stallings: Computer Organisation and Architecture
  - A. Tanenbaum: Structured Computer Organisation
  - N. J. Davis: EE 4504 Computer Organisation
    - Přednáška, která se nedá najít, prima...
  - (L. Novelli: Můj první počítač)
  - technika
    - J. Hlavička: Architektura počítačů
    - J. Douša&al.: Introduction
  - matfyzpress
    - Principy počů – nedoporučuje
- zkouška
  - písemná

- **Historie**

*slajdy 01-historie1*

- (pre)historie
  - 1971 – mikroprocespr 4004
  - 1958 – integrovaný obvod
  - 1947 – transistor
  - 1904 – elektronky

## Principy počítačů

- 1822 – Difference Engine C. Babbage
- 1642 – Pascalina B. Pascala
- ? – abakus
- základy
  - formalizace
    - formální logika – Aristoteles
  - matematizace
    - čísla
  - mechanizace
    - jemná mechanika 17. st., elektřina
- dnešní chápání počítače
  - Jacquard – programovatelný stav (1752-1834)
  - de Vaucanson (1709-1782)
- praotcové
  - Pascal (1623-1662)
  - Babbage (1792-1871)
    - nedokončené
  - Leibniz (1646-1716)
    - navrhl užití dvojkové soustavy
  - Thomas de Colmar (1785-1870)
    - počítadlo pro použití banky
- Charles Babbage
  - filosof
    - 1832: On the Economy of Manufacturers and Machinery
    - 1864: Passages from the Life of a Philosopher
  - tato díla mj. popisují stroje
    - 1822 Difference Engine
      - hodnoty polynomů 6. st.
      - paměť (sklad), řídicí jednotka (mlýn), vstup/výstup
      - dostal grant, ale nikdy to nedodělal, zabralo by to fotbalové hřiště, k problém nepřesnosti
    - 1842 Analytical engine
      - lepší nápad, přednesl ve šv. král. akademii
      - univerzální stroj řízený programem na „děrných štítcích“
      - objevuje se podmíněný skok
      - opět nedokončeno
      - dekadická soustavva, až 50 míst
      - paměť pro 1000 čísel
      - děrné štítky (jako Vaucanson)

## Principy počítačů

- Ada Byron (zájem o stroj, 38 let zemřela)
- Herman Hollerith
  - tabulátor
  - vstupní data na štítcích
  - soutěž o vylepšení sčítání lidu: 6 týdnů místo x let (62 mil. obyvatel)
  - 1924 založil/převzal firmu, ze které se později stalo IBM
  - tvrdil, že neznal Babbage atd.
- 20. století – armáda
  - Hollertihovy tabulátory
  - mezi válkami a během 2. s. v.
    - Konrad Zuse
      - 36-38 počítač Z1 – mechanický, dvojkový
      - Z2
      - 38-41 Z3
        - první použitelný počítač
        - relé
        - děrná páska
        - 64 slov po 22 bitech; plovoucí řadová čárka
        - 2400 relé
        - 5-10 Hz, násobení 3 s, 4kW, 1 tuna
      - 41-45 Z4
        - paměť: 1024 slov
        - 2 registry
        - instrukce
          - + - \* / sqrt
          - čtení z klávesnice
          - zobrazení registru
          - atd.
- Alan Turing 1912–1954
  - základy AI, Turingův test
  - 1936 hypotetický **Turingův stroj**
    - příklad T. s.
      - čtení sekvence čísel → změny stavů → výstup sekvence čísel
      - ~ stavová tabulka (zkusit optimalizovat čtyři stavy)
  - během II. Sv. války v Británii, pokusy o dešifrování Enigmy
    - 1943 Colossus
- John von Neuman

- #2: 06/10/18

*Bohužel z princip mám poněkud idšší zápisky. Párkrát jsem chybl a nikdy jsem neměl moc chuť si to zapisovat. Od osmé přednášky se snažím psát to, co říká, a není to z mého ze slajdu. Takže to berte jako doplněk k těm slajdům.*

- #3: 06/10/25

- **Reprezentace dat**

- prý velmi důležité a praktické:)
- Literatura
  - Goldberg: What every computer scientist should know about floating-point arithmetic
    - účastnil se tvorby normy IEEE pro FPA
    - IA-32 Intel Architecture Software Developer's Manual (Vol.1 – Basic Architecture)
      - lehce se odchyluje od normy
- Rozdělení dat
  - literály
    - logické hodnoty
      - binární → dvouhodnotová logika
      - reprezentace
        - bitem (buď adresování po bitech nebo přes masku)
        - datovou jednotkou (slovem, často cokoli nenulového je 1)
    - znaky
      - EBCDIC
        - 8bitové
        - vychází z děrných štítků (aby bylo zjevné, co je vyděrováno – prý)
        - 00-3F řídicí
        - 40-FF tisknutelné
        - volné pozice
        - abeceda není v jednom bloku, právě pro dobrou čitelnost štítků
      - ASCII
        - 7bitové, 7stopá páska
        - 8bitová rozšíření pro pseudografiku a národní znaky
      - Unicode
        - součást normy původně pro telekomunikace z roku 1993
        - pozn. blok pro uživatelská rozšíření
        - problémy „se staršími programy“
          - délka znaků
        - pozor na implementace
    - grafické symboly

## Principy počítačů

- podobně jako znaky
- jazyky pro popis grafických symbolů (to má jako být třeba postscript)
- nečíselná data
- čísla
- instrukce
- Číselné soustavy
  - Polyadické
    - Reprezenace pomocí koeficientů základům jednotlivých řádů (vč. zlomkové části), základy nemusí být mocninami jednoho čísla
  - Nepolyadické
    - římské číslice
      - může být více reprezentací, pokud se nedefinuje jen zkrácený zápis
      - blbě se s ní počítá
    - soustava zbytkových tříd (residue number system)
      - k-tice různých základů – prvočísel
      - číslo vyjádřeno k-ticí zbytků po dělení příslušným základem
      - jednoznačné jen pro čísla menší než součin základů
      - špatně se porovnává, dělí, ale rychle se sčítá, dělí, násobí
      - používá se při zpracování signálu (DSP)
  - Přepis **celého** čísla do soustavy s jiným základem
    - $A = a(n-1)z^{(n-1)} + \dots + a(0)z^0$  – Hornerovo schéma
    - převod je jasný (dělím celočíselně novým základem a zapisuju zbytky)
    - převod do desítkové
      - může být jednodušší – pro nás (opět Horner)
  - Přepis čísla se **zlomkovou** částí
    - Hledáme koeficienty pro  $z^i$ ,  $-m \leq i \leq n-1$
    - celá část viz výše
    - zlomková část podobně, ale *násobím*, dokud ve zlomkové části nemám nulu, a zapisuju si, co my vyleze před řádovou čárku
      - to ale nemusí být konečné!, takže je třeba včas skončit:)
- Čísla
  - přirozená (vč. nuly)
    - přímo
    - BCD (4 bity na desítkovou číslici, někdy se pakuje)
  - celá
    - se znaménkem
      - dvě reprezentace nuly, složité operace
    - s posunutím
      - konstanta reprezentuje nulu
    - (dvojkový) doplněk

- modul (TODO)
- postup ve dvojkové
  - zneguj bity
  - přičti 1

- TODO zkus

- v jiných soustavách jinak (v desítkové 0→9 2→8 atd., ale nejde všechno)
- racionální
- reálná
  - čísla s pevnou řadovou čárkou
  - floating point

- ! rozmyslet posunutí exponentu (proč není kulaté)
- rozmyslet počty bitů mantisy a exponentu v IEEE754

- #4: 06/11/1
- #5: 06/11/8
- #6: 06/11/15
- #7: 06/11/22
- #8: 06/11/29

*07-pamet.pdf – co není ve slajdech*

- **Cache / vyrovnávací paměť**

- v cachi se ukládá, i zda jsou data platná či změněná (tedy čekají na zápis do paměti)
- direct mapping: levnější než asociativní, ale ne tak výkonný
- set associative mapping: (má nějaké výhody, ale jaké to jsem nepochopil)
- nyní je tendence používat spíš jednodušší algoritmy
- Write-Back: problémy třeba, když je více procesorů
  - možné řešení: při první změně se zapíše i do paměti, pak (případně) až při odstranění z cache (ostatní procesory tedy díky změně paměti vidí, že se něco změnilo a mohou ověřit, zda neproběhla ještě jedna změna)
  - MESI = modified, exclusive, shared, invalid = stavy datových položek
- přístupy se mohou kombinovat nebo konfigurovat (někdy i na programátorské úrovni)
- Strategie přidělování paměti
  - First Fit:
    - Jedu paměti a přidělím první dostatečně velký blok (a co z něj zbude, bude tvořit nový menší volný blok)
      - Problém: moc malých zbytků
  - Next Fit:
    - Jako first fit, ale začnu až od minule přiděleného místa.

- (Stejná časová náročnost.)
  - Best fit
    - Snaha přidělit nejmenší dostatečný blok.
  - Worst fit
    - Naopak bere z největšího bloku (řeší stejný problém jako best fit, ale jiným způsobem) – i zbytky jsou pak relativně velké.
  - Každý je lepší pro jiné účely. Best/Worst jsou časově náročnější.
- **Virtuální paměť**
    - Zahrnuje (resp. může zahrnovat) ochranu paměti procesů. (Překlad adres, nelze přistoupit na adresu jiného procesu [v těch případech kdy by to jít nemělo].)
    - Snížená fragmentace (také daná tím, že se překládají adresy).
    - Nutná hardwarová podpora.
    - Page fault obvykle vznikne v hardwaru a zpracuje ho operační systém.
      - Během jedné instrukce může dojít k několika (třeba i hodně moc, když má běžet na úseku paměti) page faultům.
      - OS pak hledá volný rámec, případně uvolňuje.
      - Fyzická paměť se pak vlastně chová jako cache virtuální paměti.
    - Stránkovací tabulky
      - typicky jedna na proces → oddělení paměti procesů, ale je možné sdílení paměti mezi procesy (pokud si to vyžádají)
    - Interní fragmentace
      - přidělují se větší bloky než proces chce (velikost stránek)
    - Externí fragmentace
      - nevyužitá místa mezi bloky
    - Víceúrovňové stránkování
      - Tabulky druhé úrovně nemusí existovat, případně mohou být odložené.
    - Asociativní paměť
      - Zvláštní stránkovací paměť
    - Inverzní stránkování
      - Využívá toho, že rámců je méně než stránek, takže tabulky jsou menší
    - Příznaky v tabulce
      - dirty – změněná data
      - accessed – byla čtená
      - read/write – ochrana proti zápisu

• #9: 06/12/06

*Co není ve slajdech.*

- **Algoritmy výměny stránek**

- „Clock“ – reálně se používá, podle analogie s pohybem po ciferníku

- varianta se dvěma ručičkami (svírají jakoby úhel a jedna nuluje, takže neoběhnu celé kolo)
- základ stránkování Linuxu, na Windows složitější
  - obecně se tyto základní algoritmy modifikují, aby to lépe fungovalo v reálných situacích

- **Segmentace**

*Jiný algoritmus než stránkování.*

- Velikost segmentu se může případně i měnit.
- Narozdíl od stránkování se dotýká programátora.
- Page Segmentation
  - staré, už na Multicsu, ale používá třeba Pentium
  - každý segment stránkovaný zvlášť
  - ? na Multicsu, co segment to soubor
- Intel 286 měl jen segmentování.
- Segmented Pages
  - Narozdíl od stránkování explicitně řečeno kolik nejvyšších bitů je číslo segmentu.
- Segmentace
  - segmenty mohou mít různé způsoby přístupu (vykonávání, čtení, ...), u stránkování je to těžko
  - můžu třeba segment s knihovnou dát dvěma různým procesům
- **TODO: Co a jak se používá dnes na běžných OS**

- **Mikroprocesory**

- čip není mikroprocesor
  - mikroprocesor je obecný, programovatelný, čipy byly už dříve
- Z80 potřeboval méně podpůrných obvodů než 8080, jinak byly kompatibilní na úrovni kódu

- #10: 06/12/13

*Co není ve slajdech.*

- **Hard-wired řadič**

- Moc se nepoužívá (v počítačích). Velmi rychlé, ale neflexibilní.
- V prvních RISCových procesorech: chytře navržená jednoduchá instrukční sada – výhodné použití hard-wired řadiče

- **Mikroprogramování**

- Příklad: Na nějakých (jakých?) starých IBM se obsah instrukční sady načítal z pásky, na VAXu mohly běžet při určitém nastavení řídicí paměti instrukce PDP-11. Jiný stroj zase bez instrukční sady (prázdná řídicí paměť). Nebo se při přepínání kontextu daly přepínat instrukční sady (optimalizace výkonu).



- Výhody: přehlednější, obecnější, možnost změnit sadu (nemusí se předělat celý řadič).
- Nevýhody: složitost, pomalost. Poč v počí.
- Dekodéry při vertikálním kódování nejdou přeprogramovat.

- **Nanoprogramování**

- Často u CISC procesorů

- **K obrázku Simple CPU**

- PC – program counter
- MAR/MDR (memory address/data register) – paměťové registry (adresa a data)
- ACC – akumulátor
- IR – instrukční registr
- Příklad: Zápis adresy do adresového registru, v dalším kroku, až budou data v datového registru, otevřou brány, aby data protekla do akumulátoru.
- logický popis řídicích instrukcí dává obsah řídicí paměti

- **K obrázku PIC12C5xx**

- File registers je něco jako datová paměť.
- Není vonneumanovská architektura (oddělená data a paměť, sběrnice).

• #11: 06/12/20

*Co není ve slajdech.*

- **Pipelining**

- V rámci procesoru, ale i FPU apod.
- Myšlenka pipeliningu se používala již dříve (před RISC procesory).
- Omezené možnosti: jednak obnáší režie, jednak to prostě nemá nad určitou úroveň cenu (už nelze o moc víc paralelizovat). (Viz bubliny.)
- Bubliny: Vznikají, když nemůžu zparalelizovat dvě za sebou jdoucí instrukce, musí se čekat, až první zcela proběhne.
  - Buď seřadím (já kompilátor) instrukce chytrě, aby se to nestalo. (Přinejhorším prázdné instrukce.)
  - Poradí si s tím jinak daná architektura (nedojde k nekonsistentním výsledkům)? – jak kdy.
- Skoky (podmíněné samozřejmě):
  - Můžu si dělat dvě pipeline a jednu zahazuju. (Malé zpoždění, pipeline jsou krátké.) – multiple streams
  - Paměť skoku – podle toho, zda se skákalo minule.
    - To může poradit i kompilátor, pokud to umožňuje instrukční sada (viz dále).
  - Zpoždění skoku – závislé na kompilátoru.
  - Predikce skoku

- Statická: Náповěda od kompilátoru, zřejmé. Hardware – podle toho, jak daleko to skáče (krátký – asi cyklus).
- Dynamická: Podobné jako paměť skoku (výše), trochu fikanější. (Čtyři stavy, viz obrázek – když to nevyjde jednou, neměním názor.)

- #12: 07/01/03

*Co není ve slajdech.*

- **Pojmy, které se vyskytly během přednášky bez vysvětlení**

- superskalární
- data flow

- (obojí lze najít poměrně dobře vysvětlené na [en.wikipedia.org](http://en.wikipedia.org))

- #13: 07/01/10

*Co není ve slajdech.*

- **Máme si promyslet (zjistit):**

- Co přesně je sudá a lichá parita (čeho je sudý/lichý počet)
- Vyzkoušet Hammingův kód. (Opraví se i (1) chyba ve vlastní dopočítané paritě?)
- ?? CRC to máme chápat nějak hlouběji, nebo ne ??
- (fyz./log. struktura sítě)

- **Flynnova klasifikace**

- MISD prý skoro neexistuje
- MIMD prý vytlačuje SIMD (huh!?)

- **Četba**

- Heinlein: Domek jako klíčka (?? postavil si domeček)

- **Sítě**

- důležité: Řízení dynamické sítě (ne ty šílené věci před tím)