1. **Datově orientovaný přístup při vývoji software**

**Vedoucí**: Ing. Peter Janků, Ph.D.

1. Definujte pojem datově orientovaný návrh a seznamte se s touto problematikou.
2. Porovnejte tento způsob návrhu s objektově orientovaným návrhem.
3. Popište vliv mikroarchitektury počítače na rychlost běhu programu.
4. Demonstrujte jednotlivé principy na příkladech.
5. Ověřte efektivitu programů pomocí nástrojů pro výkonnostní testy a profilování.
6. Sestavte sadu doporučení pro využití datově orientovaného přístupu.
7. **Seznam kapitol**

* Teoretická část
  + Datově orientované programování
    - Definice
    - Historický výskyt
    - Hlavní myšlenky
      * Je to o datech
      * Data nejsou problémová doména
      * Statistika
      * Data se mění
    - Využití
    - Další paradigmata
      * Objektově orientované programování
      * Funkcionální programování
      * Porovnání
  + Hardware jako platforma
    - Vyrovnávací paměť
      * Struktura a adresování
      * Pojmy
      * Vyrovnávací paměť s přímým mapováním
      * Set associative vyrovnávací paměť
      * Plně asociativní vyrovnávací paměť
      * Substituční strategie
      * Způsoby zapisování
    - Operační paměť
    - CPU pipelining
      * Fáze instrukčního kanálu
      * Sekvenční zpracování
      * Skalární pipeline
      * Superskalární pipeline
      * Ideální vs reálné řešení
      * Datové a kontrolní závislosti
    - Prefetching
      * Softwarový prefetching
      * Hardwarový prefetching
      * Execution based prefetching
      * Address correlation based prefetching
      * Content directed prefetching
    - Lokalita
    - Jazyk symbolických adres
      * Vnitřní architektura x86-32
      * Adresovací módy
      * Instrukce pro přesun dat
      * Instrukce pro aritmetické operace
      * Instrukce pro bitovou rotaci a posuv
      * Instrukce pro předání kontroly
    - Překladače
      * GCC
      * Clang
      * MSVC
    - SIMD
    - Paralelizace
  + Měření výkonu aplikací
    - Benchmarking
    - Profilování
      * Druhy profilerů
* Praktická část
  + Příklady optimalizací
    - Iterace polem
    - Předvídatelnost operací
    - Hot vs cold data
    - SoA vs AoS
    - Využití pipeliningu
    - Zarovnání dat
    - Datové a kontrolní závislosti
    - Falešné sdílení
  + Ověření výkonu pomocí nástrojů
  + Souhrn optimalizací

1. **Progres v teoretické části**

Ve zmíněných kapitolách od „Datově orientované programování“ až „SIMD“ v „Hardware jako platforma“ se nachází informace nezbytné pro pochopení funkce příkladů uvedených v praktické části. Tyto uvedené kapitoly jsou zcela vypracované. Nechávám samozřejmě prostor pro doplnění jakýchkoliv připomínek vedoucího. V teoretické části zbývají dopracovat kapitoly „Paralelizace“ až „Druhy profilerů“.

**Zdroje**

CppCon 2014: Mike Acton "Data-Oriented Design and C++". *Youtube* [online]. 30.9.2014 [cit. 2022-12-21]. Dostupné z: https://www.youtube.com/watch?v=rX0ItVEVjHc. Kanál uživatele CppCon

FABIAN, Richard. Data-Oriented Design: Software engineering for limited resources and short schedules [online]. Richard Fabian, 2018, 307 s. ISBN 9781916478701.

LLOPIS, Noel. Data-Oriented Design (Or Why You Might Be Shooting Yourself in The Foot With OOP) [online]. 2009-12-04 [cit. 2022-12-23]. Dostupné z: <https://gamesfromwithin.com/data-oriented-design>

Object-oriented programming [online]. 2022-09-28 [cit. 2022-12-30]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming>

MITCHELL, Brad. What Is Functional Programming and Why Use It? [online]. 2022-07-13 [cit. 2022-12-30]. Dostupné z: <https://www.codingdojo.com/blog/what-is-functional-programming>

BRYANT, Randal a David O'HALLARON. Computer Systems: A Programmer's Per-spective. 3rd Edition. Pearson, 2015, 1128 s. ISBN 013409266X.

Lecture 25: Prefetching - Carnegie Mellon - Computer Architecture 2015 - Onur Mutlu. *Youtube* [online]. 30.9.2014 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=ibPL7T9iEwY. Kanál uživatele Carnegie Mellon Computer Architecture.

Lecture 26. More Prefetching and Emerging Memory Technologies - CMU - Comp. Arch. 2015 - Onur Mutlu. *Youtube* [online]. 30.9.2014 [cit. 2022-12-30]. Dostupné z: https://www.youtube.com/watch?v=TUFins4z6o4. Kanál uživatele Carnegie Mellon Computer Architecture.

HENNESSY, John L. a David A. PATTERSON. Computer Architecture: A Quantitative Approach. 4th Edition. 2006, 704 s. ISBN 0123704901.

KUSSWURM, Daniel. Modern X86 Assembly Language Programming: 32-bit, 64-bit, SSE, and AVX. Apress, 2014, 700 s. ISBN 1484200659.

STALLMAN, Richard M. Using the GNU Compiler Collection: For gcc version 12.2.0 [online]. GNU Press, 2022 [cit. 2023-01-16]. Dostupné z: https://gcc.g nu.org/onlinedocs/gcc-12.2.0/gcc.pdf

Clang Compiler User’s Manual [online]. [cit. 2023-01-16]. Dostupné z: <https://clang.llvm.org/docs/UsersManual.html>

Compiler Options [online]. [cit. 2023-01-16]. Dostupné z: https://learn.microsoft.com/en-us/cpp/build/reference/compiler-options

1. **Progres v praktické části**

Postup v praktické části je výrazně menší oproti části teoretické. V současné chvíli má podobu několika testovacích programů psaných v C/C++, kde testuji jednotlivé principy nastolené v teoretické části. Po kompletaci teoretické části, ve které se ještě sám seznamuji s různými problematikami, se naplno pustím do vypracování ukázkových příkladů. Mám nachystané prostředí pro kompilaci programů pomocí překladačů GCC, Clang a MSVC. Pro sestavování využívám CMake. Pro benchmarking jsem zvolil knihovnu Google Benchmark. Vývojové prostředí je Visual Studio Code. Jako dodatečný disassembler používám webový nástroj <https://godbolt.org/> . Jako profiler zvolím Intel VTune a v rámci porovnání různých druhů nejspíš ještě nějaké další.

V této části se budu věnovat vypracování větších či menších příkladů poukazujících na aplikaci několika konceptů představených v teoretické části. Také zde bude porovnán způsob vývoje pomocí datově a objektově orientovaného přístupu. Poté proběhne srovnání časů běhu odpovídajících programů pomocí benchmarkovacích nástrojů a využití zdrojů pomocí profilovacích nástrojů. Na závěr poskytnu přehled významných optimalizací a doporučení programátorům pro vývoj software.

1. **Vyjádření vedoucího**