

Dobrý den, rád bych vás krátce seznámil se svojí diplomovou prací a přiblížil její aktuální stav.

NEXT

V rámci prezentace představím pojem datově orientované programování, popíšu jak byly splněny body zadání, v čem spočíval můj přínos a v jakém stavu se aktuálně práce nachází.

NEXT

Datově orientované programování je programovací paradigma, které vzniklo díky identifikování nedostatků objektově orientovaného návrhu v určitých oblastech. Jak vyplývá z názvu, považuje data za to nejpodstatnější. Staví na tvrzení, že primární úkol aplikací je transformace dat z jedné formy do druhé. Nesnaží se hardware zatěžovat abstrakcemi v podobě modelů, ale snaží se ho respektovat a využívat co nejefektivněji. Právě na využívané optimalizace se soustředí tato práce. Optimalizace se týkají zejména správného využití vyrovnávacích pamětí, nepletení se do cesty instrukční pipeline procesoru a snaží se vytěžit maximum z paralelního potenciálu hardware. Pojem datově orientované programování byl poprvé použit v článku pro herní vývojáře. Z této sféry toto paradigma patrně pochází.

NEXT

Zde jsou body zadání, kterým jsem se měl věnovat. První bod je pokryt v teoretické části, kde uvádím definice tohoto pojmu. Druhý bod řeším v obou částech, jednak uvedením rozdílů v hlavních myšlenkách obou směrů a také v praktické části v příkladech. Třetímu bodu se extenzivně věnuju v teoretické části, kde dopodrobna popisuju informace například o vyrovnávacích pamětech nebo instrukční pipeline a také demonstruju na několika praktických příkladech. Čtvrtý bod je pokryt praktickou částí, kde prezentuju sadu ukázkových programů, které mají ukazovat problémový kód a jeho vylepšení v podobě optimalizací. Pátý bod je obsažen v praktické části v podobě zhodnocení výkonu těchto ukázkových programů pomocí nástrojů měřících výkon a profilovacích nástrojů. Výsledky měření jsou prezentovány v tabulkách a následná analýza je zachycena v textové a obrázkové podobě. Na základě výsledků pátého bodu byla sestavena sada doporučení pro programátory, kteří by chtěli aplikovat tento způsob tvorby programů v praxi.

NEXT

Vlastní přínos spočíval ve stanovení a implementaci ukázkových příkladů, které demonstrují různé aspekty tvorby programů. Každý program má různé varianty. Jedna varianta vždy trpí nějakým nedostatkem a další varianty tento nedostatek odstraňují. Pro implementaci byl použit jazyk C++ a byly porovnány překladače GCC, Clang a MSVC spolu s různými stupni optimalizace. Nástroj pro měření je Google Benchmark, který poskytuje informace o časování. Následně byly výsledky analyzovány a některé z nich byly více dopodrobna popsány za podpory profilovacích nástrojů vTune a Tracy. Na základě předchozích bodů byla sestavena sada doporučení, která zmiňuje zejména významné dosažené zlepšení ukázkových programů.

NEXT