

Posudek bakalářské práce

předložené na Matematicko-fyzikální fakultě
Univerzity Karlovy v Praze

posudek oponenta

Autor/ka: Jakub Hájek

Název práce: Persistentní datové struktury v C

Studijní program a obor: Informatika, Správa počítačových systémů

Rok odevzdání: 2010/2011

Jméno a tituly vedoucího/opponenta: Mgr. Zbyněk Falt

Pracoviště: KSI MFF UK

	e x c e l e n t n í	o d p o v í d a j í c í	s l a b š í	n e v y h o v u j í c í
Náročnost zadaného tématu		✓	✓	
Míra splnění zadání		✓	✓	
Rozsah práce		✓		
Struktura textové části práce		✓		
Analýza		✓	✓	
Vývojová dokumentace		✓		
Uživatelská dokumentace		✓		
Jazyková a typografická úroveň		✓	✓	
Návrh a design implementace		✓		
Kvalita zpracování softwarové části			✓	
Stabilita aplikace			✓	

Nejvýznamnější klady:

Student vytvořil knihovnu pro jazyk C, která poskytuje služby pro persistentní uchování stromových struktur. Knihovna navíc poskytuje implementaci persistentních AVL, červeno-černých a B-stromů, čímž její užitná hodnota dále roste. Kladně hodnotím i snadné zprovoznění knihovny a testů podle přiložené dokumentace.

Nejzávažnější nedostatky:

Téma práce není příliš náročné, neboť knihovna poskytuje pouze relativně tenkou sjednocující persistentní vrstvu nad třemi typy databází a ještě tenčí vrstvu nazvanou vrstva uzlů. Implementace vyhledávacích stromů v uživatelské vrstvě je již nad rámec samotného zadání.

Student téma splnil, ale mám dvě výhrady:

- 1) Nedovedu si představit, jak by se knihovna používala pro „struktury o velikosti větší, než je velikost dostupné paměti“ (viz kapitola 1), neboť
 - a. Např. funkce `pl_restore_node_byid` neumožňuje rozlišit, zda selhala proto, že uzel nebyl nalezen, nebo proto, že se nepodařilo naalokovat paměť pro uzel. Tím není možné detekovat situaci, že dochází paměť a případně nějakou uvolnit.
 - b. Na mnoha místech se předpokládá, že volání funkce `malloc` neselže, takže knihovna rozhodně nebude stabilní v případě, že aplikace má nedostatek paměti.
 - c. Očekával bych, že knihovna bude implementovat alespoň jednoduchou strategii pro automatické uvolňování nepoužívaných uzlů z paměti v případě jejího nedostatku.
- 2) V případě měření výkonu mi chybí podrobnější analýza, proč se výkon jednotlivých úložišť tak výrazně liší pro různé typy vyhledávacích stromů.

Ukázky zdrojových textů v textu mají nejednotný styl formátování a jsou v nich syntaktické chyby, např. řádek: `val=(int*)malloc(sizeof(int));`. Text je ale srozumitelný a kromě několika drobných gramatických chyb či překlepů je v pořádku. Z typografického hlediska je v práci několik nedostatků: osamocené znaky na konci řádků, přetékané řádky, krátké výpisy jsou občas rozděleny koncem stránky, chybné rozdělování slov (persite-nční). Ale není jich příliš mnoho. Místo wikipedie by bylo vhodnější citovat původní články.

Zdrojové kódy jsou nejednoduše formátovány, některé soubory jsou okomentovány stylem `doxygen` (`dg_persistence.h`), některé jednořádkovým neformátovaným komentářem (`node.h`) a některé vůbec (`btree.h`). Volání funkcí nad databází nejsou ošetřena pro případ chyby, stejně tak volání `malloc`, s čímž souvisí nedostatečná stabilita a robustnost knihovny. Ve zdrojových kódech je množství zakomentovaných ladicích výpisů či celých funkcí, které by měly být spíše `#ifdef`ované.

Další poznámky:

Nevím, na kolik formálnímu odevzdání vadí skutečnost, že datum odevzdání (5.8.2011 – dle Harmonogramu akademického roku 2010/2011) předchází datu zadání (8.8.2011). Nechávám to na posouzení komise.

	v ý b o r n ě	v e l i d o b ř e	d o b ř e	n e p r o s p ě l / a
Návrh známky		✓		

Datum: 23.8.2011

Podpis:

Poučení k formuláři pro hodnocení infromatických bakalářských prací

Tento formulář je určen pro hodnocení vedoucího i oponenta bakalářské práce, která má formu softwarového projektu. Bakalářské projekty jiných typů (teoretická práce, srovnávací studie apod.) budou hodnoceny pomocí standardních textových posudků.

Jednotlivá políčka vyplňte nejlépe elektronicky (lze případně i ručně), je možné zaškrtnout i dvě sousední políčka (např. pro hodnocení typu 'něco mezi odpovídající a slabší'), a to i u návrhu výsledné známky. Pokud některá položka nemá vzhledem k práci smysl (např. stabilita aplikace u práce bez vlastní implementace), položku nevyplňujte. Výsledná navrhovaná známka nemusí být žádným 'průměrem' hodnocení jednotlivých kritérií. Pokud některé položky hodnotíte jako slabší nebo nevyhovující, v sekci Nejzávažnější nedostatky popište důvody vašeho hodnocení a zjištěné nedostatky.

Výklad stupňů hodnocení:

- excelentní znatelně lepší/rozsáhlejší/dokonalejší než je pro Bc práci požadováno
- odpovídající přiměřené Bc práci, student splnil to, co měl
- slabší výhrady ke kvalitě, rozsahu, hloubce nebo zpracování
- nevyhovující neodpovídá požadavkům na Bc práci, práce nemá být obhájena

Vyplněné a ručně podepsané (i v případě elektronického vyplňování) hodnocení odevzdejte na sekretariát KSI, elektronickou verzi pošlete na sekretariat@ksi.ms.mff.cuni.cz. Pokud máte emailový kontakt na autora práce, pošlete posudek i jemu.