Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy v Praze

Autor práce	Roman Kápl		
Název práce	Tracing function calls in Windows NT kernel		
Rok odevzdání	2015		
Studijní program	Informatika	**Studijní obor**	Programování
Autor posudku	Mgr. Pavel Ježek, Ph.D.	**Role**	Vedoucí
Pracoviště	KDSS MFF UK		

Prosím vyplňte hodnocení křížkem u každého kritéria. Hodnocení *OK* označuje práci, která kritérium vhodným způsobem splňuje. Hodnocení *lepší* a *horší* označují splnění nad a pod rámec obvyklý pro bakalářskou práci, hodnocení *nevyhovuje* označuje práci, která by neměla být obhájena. Hodnocení v případě potřeby doplňte komentářem. Komentář prosím doplňte všude, kde je hodnocení jiné než *OK*.

K celé práci	lepší	OK	horší	nevyhovuje
Obtížnost zadání	☒	☐	☐	☐
Splnění zadání	☒	☐	☐	☐
Rozsah práce *... textová i implementační část, zohlednění náročnosti*	☒	☐	☐	☐

Komentář Autor si zvolil na bakalářskou práci extrémně obtížné zadání (myslím, že kolonka „lepší“ viz výše, zde ani z daleka nestačí na popis skutečného stavu), které vyžaduje detailní pozorumění fungování jádra Windows (včetně neveřejných a nedokumentovaných částí, a mnoha způsobů „hackování“), a jak se nakonec ukázalo i odlazení netriviálního ovladače běžícího v režimu jádra.

Výsledkem práce je nástroj na profesionální úrovni, který je využitelný nejen ve výuce, ale jistě ho využije i mnoho programátorů ovladačů ve Windows. Nástroj umí sbírat širokou škálu událostí (včetně přerušení a DPC), a hlavně dokáže spolehlivě určit jejich společný kontext, a na základě toho zobrazit průběh volání (a návaznost i asynchronních událostí generovaných i různými vlákny nebo přerušeními!) na zjištěné struktuře driver stacku (která není čtena z konfigurace, ale přesně odpovídá reálnému stavu a zachyceným voláním!), případně sequence a dalších diagramech – z tohoto pohledu se jedná o zcela unikátní nástroj, a zatím mi není na světě známo libovolné konkurenční řešení, které by bylo schopné takového detailu a přesnosti!!! Navíc práce funguje v 32-bit i 64-bit Windows, což je na úrovni jádra a dalších technických detailů obrovská výzva.

Zdrojové kódy práce zabírají přes 450 kB kódu v jazycích C (ovladač v jádře pro sběr událostí) a C++ (analyzační a vizualizační aplikace), což je zvlášť při obrovské technické složitosti kódu vysoce nadstandardní pro běžné bakalářské práce.

Výsledná práce je jako celek zcela bez diskuze na úrovni excelentní diplomové práce, a je škoda, že autor je teprve v první fázi studia na MFF, a tedy práci odevzdává „jen“ jako bakalářskou.

Textová část práce	lepší	OK	horší	nevyhovuje
Formální úprava *... jazyková úroveň, typografická úroveň, citace*	☒	☒	☐	☐
Struktura textu *... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu*	☒	☐	☐	☐
Analýza	☒	☐	☐	☐
Vývojová dokumentace	☒	☐	☐	☐
Uživatelská dokumentace	☒	☐	☐	☐

Komentář Text práce je dobře formátovaný, obsahuje přehledné obrázky, a je psán stylisticky i odborně velmi dobrou angličtinou! Bohužel se práci nevyhnuly občasné překlepy a chybějící písmenka.
Text je také příkladně strukturován, autor čtenáře dobře provádí i složitými koncepty, a vždy drží adekvátní úroveň detailu nutnou pro pochopení. Zároveň autor čtenáře neutápí v nepodstatných detailech, což je zvlášť u takto technicky náročného tématu zcela zásadní pro pochopení hlavních myšlenek a neztracení se v širším kontextu.
Více než příkladná je též analytická část práce, kde autor hlavní netriviální problémy řeší se čtenářem, důkladně hledá a zkoumá možná řešení, a vždy velmi dobře argumentuje svoji výslednou volbu. Nadstandardně dobře je v kontextu práce provedeno důkladné srovnání existujících nástrojů, výběr vlastností nového nástroje, a problém získávání informací o událostech v jádře. Zde je důležité si uvědomit, že k danému tématu není příliš mnoho snadno dohledatelných informací (resp. často lze nalézt informace zavádějící, nebo nepravdivé), proto pro tak kvalitní analýzu musel autor provést nadstandardně širokou rešerši. Navíc se autor nedrží jen domény Windows, ale pokud je to možné tak se snaží inspirovat i postupy a nástroji z jiných systémů (např. Linuxu).
Vývojové i uživatelské dokumentace jsou též zcela kompletní, každá je velmi dobře zaměřena na cílovou skupinu čtenářů, a dávají velmi dobrý přehled o všech důležitých aspektech práce. Vývojová dokumentace je podpořena kvalitními dokumentačními komentáři. V uživatelské dokumentaci autor velmi dobře prezentuje reálné a velmi vhodně zvolené příklady zachycených komunikací v jádře – vzorové „trace“ jsou i přiložené na doprovodném CD.

Implementační část práce	lepší	OK	horší	nevyhovuje
Kvalita návrhu *... architektura, struktury a algoritmy, použité technologie*	☒	☐	☐	☐
Kvalita zpracování *... jmenné konvence, formátování, komentáře, testování*	☒	☐	☐	☐
Stabilita implementace	☒	☐	☐	☐

Komentář Z práce je vidět, že autor je vynikajícím vývojářem: tam kde je to možné, tak používá vhodné existující knihovny, tam kde by kompromisy v řešení nebo návrhu byly příliš vysoké, tak se nebojí přijít s vlastním originálním řešením. Celkově je kód velmi dobře navržen. Nadstandardní je též to, že monitorované funkce jádra OS nejsou napevno zabudovány v aplikaci, ale autor navrhl vlastní šablonový systém, kterým lze snadno do aplikace dodat sledování širší množiny událostí.
Stejně tak je kód kvalitní i po řemeslné stránce, jsou dobře připravené nástroje pro překlad a generování kódu, kód je dobře komentován a používá vhodné jmenné konvence.
Aplikace je stabilní ve všech režimech a všech verzích Windows, což je, zvlášť u kódu, který běží v jádře OS a navíc přepisuje části jádra vlastním kódem, skoro až zázrak!

Celkové hodnocení Výborně (spíše lepší)
Práci navrhuji na zvláštní ocenění Ano

Datum 28. srpna 2015 **Podpis**