

# Posudek bakalářské práce

Matematicko-fyzikální fakulta Univerzity Karlovy

<b>Autor práce</b>	Michal Jireš	
<b>Název práce</b>	Incremental link-time optimization in GNU Compiler Collection	
<b>Rok odevzdání</b>	2023	
<b>Studijní program</b>	Informatika	
<b>Specializace</b>	Systémové programování	
<b>Autor posudku</b>	Jan Hubička	Vedoucí
<b>Pracoviště</b>	Katedra aplikované matematiky	

## K celé práci

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Obtížnost zadání	X			
Splnění zadání	X			
Rozsah práce <i>... textová i implementační část, zohlednění náročnosti</i>	X			
<p>GNU Compiler Collection (GCC) je produkční překladač s více než třicetiletou historií. Jakákoliv netriviální práce na takovém projektu vyžaduje důkladné studium zdrojových kódů, formátovacích konvencí a seznámení se s postupy práce na takovém projektu. Původní zadání vyžadovalo proniknout do relativně složité části link-time optimizeru. Proto určitě patří ke složitějším zadáním v kontextu bakalářských prací.</p>				

## Textová část práce

lepší OK horší nevyhovuje

	lepší	OK	horší	nevyhovuje
Formální úprava <i>... jazyková úroveň, typografická úroveň, citace</i>	X			
Struktura textu <i>... kontext, cíle, analýza, návrh, vyhodnocení, úroveň detailu</i>	X			
Analýza	X			
Vývojová dokumentace	X			
Uživatelská dokumentace	X			
<p>Hlavním cílem projektu bylo implementovat cache, která umožňuje znovu použít výsledky minulých překladů. Samotná cache je implementována spolehlivě s přihlednutím na paralelní kompilaci (zamykání souborů) i úklid starých souborů.</p> <p>GCC automaticky rozděljuje program do 128 částí, které překládá paralelně. Aby cache efektivně pracovala, je třeba maximalizovat počet částí programu, které se nezmění po malém zásahu do zdrojových kódů. Aktuální algoritmus pro rozdělení programu v tomto ohledu není příliš stabilní a proto volitelnou částí zadání byl návrh nového algoritmu. Michal Jireš našel zajímavé řešení, které v praktických testech funguje velmi dobře.</p> <p>Zcela nad rámec původního zadání sám sestavil metodu testování inkrementální kompilace pomocí aplikace změn z git repozitáře GCC, Clangu a dalších projektů. Na jejím základě identifikoval několik složitých situací, kdy GCC bez většího důvodu výrazně mění výsledek kompilace. U většiny problému navrhl i řešení kde většina už byla akceptována do oficiální verze překladače. Prokázal tím také, že jeho řešení je praktické a jde využít k překladu velkých projektů.</p>				

## Implementační část práce

lepší OK horší nevyhovuje

Kvalita návrhu . . . <i>architektura, struktury a algoritmy, použité technologie</i>	X			
Kvalita zpracování . . . <i>jmenné konvence, formátování, komentáře, testování</i>	X			
Stabilita implementace	X			
Michal Jireš pracoval velmi samostatně a jím zvolená řešení jsou funkční a v odpovídající kvalitě pro zahrnutí v oficiálním vydání GCC.				

**Celkové hodnocení**    Výborně  
**Práci navrhuji na zvláštní ocenění**    Ano

Datum

Podpis