



## Posudek oponenta závěrečné práce

**Student:** Martin Taibr  
**Oponent práce:** Ing. Jan Trávníček, Ph.D.  
**Název práce:** Překlad QuakeC do Rustu  
**Obor:** Teoretická informatika

**Datum vytvoření:** 10. 6. 2019

<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – následující škálou 1 až 4:</i>
<b>1. Splnění zadání</b>	<b>1=zadání splněno, 2=zadání splněno s menšími výhradami, 3=zadání splněno s většími výhradami, 4=zadání nesplněno</b>
<i>Popis kritéria:</i> Posuďte, zda předložená ZP dostatečně a v souladu se zadáním obsahově vymezuje cíle, správně je formuluje a v dostatečné kvalitě naplňuje. V komentáři uveďte body zadání, které nebyly splněny, posuďte závažnost, dopady a případně i příčiny jednotlivých nedostatků. Pokud zadání svou náročností vybočuje ze standardů pro daný typ práce nebo student případně vypracoval ZP nad rámec zadání, popište, jak se to projevilo na požadované kvalitě splnění zadání a jakým způsobem toto ovlivnilo výsledné hodnocení.	
<i>Komentář:</i> Zadání je dle mého názoru náročnější, respektive velmi pracné. I když je obecně možné tvrdit, že programy v jednom jazyce lze převést na programy v jiném jazyce při zachování funkcionality, reálný navíc strojový překlad je velmi náročný. Cílem práce bylo právě takový překladač vytvořit. Bohužel se to povedlo jen v omezené míře, pro podмноžinu konstrukcí jazyka QuakeC a jen některá makra.	
<i>Hodnotící kritérium:</i>	<i>Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):</i>
<b>2. Písemná část práce</b>	<b>65 (D)</b>
<i>Popis kritéria:</i> Zhodnoťte přiměřenost rozsahu předložené ZP vzhledem k obsahu, tj. zda všechny části ZP jsou informačně bohaté a ZP neobsahuje zbytečné části. Dále posuďte, zda předložená ZP je po věcné stránce v pořádku, případně vyskytují-li se v práci věcné chyby nebo nepřesnosti. Zhodnoťte dále logickou strukturu ZP, návaznosti jednotlivých kapitol a pochopitelnost textu pro čtenáře. Posuďte správnost používání formálních zápisů obsažených v práci. Posuďte typografickou a jazykovou stránku ZP, viz Směrnice děkana č. 26/2017, článek 3. Posuďte, zda student využil a správně citoval relevantní zdroje. Ověřte, zda jsou všechny převzaté prvky řádně odlišeny od vlastních výsledků, zda nedošlo k porušení citační etiky a zda jsou bibliografické citace úplné a v souladu s citačními zvyklostmi a normami. Zhodnoťte, zda převzatý software a jiná autorská díla, byly v ZP použity v souladu s licenčními podmínkami.	

#### *Komentář:*

Typografické nedostatky práce zahrnují nějaké překlepy "ochutu" vs "ochotu" hned v poděkování, chybný tvar slova "licencí" vs "licensí", a další.

Text práce je často zbytečně osobní. Některé věty textu nedávají smysl.

V pravo přetékáající text na straně 14 a dalších.

Text práce obsahuje anglicizmy, například i po definicích českého termínu s ekvivalentním anglickým termínem jen v závorkách jako doplněk.

Dopředný odkaz na sekci 4.2. a na kapitolu 7.

#### Faktické nedostatky:

Bylo by hezké uvést, co je zmiňovaný překládaný projekt Xonotic.

Makra jazyka QuakeC jsou popsána v rešerši jen zběžně a bez příkladů.

Rešerše by mohla více diskutovat i zamínutné implementace preprocesorů, chybí mi příklad chování GCC (v práci popsáno pouze v jednom odstavci), dále pak příklady informací, které poskytuje Clang.

Kapitola 6 Návrh obsahuje odstavec, který se zpětně vrací k již navrženým přístupům zpracování maker a přidává další možnost, jak informace o makrech získat. To by ale mělo být vše uvozeno na jednom místě, tedy v první zmínce jak přistupovat ke zpracování maker.

Tento později navržený způsob detekce maker je více rozepsán až v kapitole implementace, kde je jeho popis ale promíchán s překladem výrazů.

Některé sekce kapitoly 6 jsou psané jako chronologický popis prací. (Použitý fragment textu jako "touto dobou jsem komunikoval...")

Kapitola 6 stejně tak zmiňuje komplikace návrhu lexeru při existenci definování typů. Podle mého názoru, jazyky, které podporují typedef, nemusí už na úrovni lexeru vědět, jestli je identifikátor typ nebo něco jiného, toto by z pohledu teorie měla řešit až některá z dalších částí překladače.

Kapitola návrh by měla velmi podrobně diskutovat, jak sobě odpovídají konstrukce v QuakeC a Rustu a jak je na sebe přeložit. Bohužel jediný zdrojový kód je v kapitole 2, ten ale není ukázkou ekvivalentního kódu v QuakeC a Rustu.

Hned první věta kapitoly 7 ("Tato kapitola popisuje výsledný návrh...") je v kontradikci s jejím názvem "Implementace a testování". Sekce 7.4 by uvítala více příkladů.

Text práce tvrdí že Parser je schopný zpracovat celý projekt Xonotic, ale dále se píše, že překladač podporuje pouze velmi malou část QuakeC. Z toho logicky vyplývá, že překladač naparsuje kód v QuakeC, ale jen jeho malou podmnožinu zvládne korektně přeložit do Rustu.

Student se sám přiznává, že nesplnil některé body jím navrženého zadání. Viz odstavec 3 na straně 36.

Testování je zmíněno jen jedním odstavcem v sekci 7.7.

*Hodnotící kritérium:*

*Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):*

### **3. Nepísemná část, přílohy**

**80 (B)**

#### *Popis kritéria:*

Dle charakteru práce se případně vyjádřete k nepísemné části ZP. Například: SW dílo – kvalita vytvořeného programu a vhodnost a přiměřenost technologií, které byly využité od vývoje až po nasazení. HW – funkční vzorek – použité technologie a nástroje, Výzkumná a experimentální práce – opakovatelnost experimentů

#### *Komentář:*

Implementace překladače využívá knihovny závislé na nightly verzi kompilátoru Rust, navíc specifické verze, což není pohodlné ani udržitelné.

Překladač je schopný přeložit výrazy, deklarace builtin datových typů, příkaz printf, volání funkcí (nikoliv rekurzi, která ale zřejmě v QuakeC není) a makra, nezvládne přístup k objektu Entity. Bohužel objekt typu Entity je použitý na mnoha místech. Ukázalo se, že program přeloží i podmínky a cykly.

Překladač zachovává pouze použitá makra.

*Hodnotící kritérium:*

*Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):*

#### 4. Hodnocení výsledků, jejich využitelnost

80 (B)

**Popis kritéria:**

Dle charakteru práce zhodnoťte možnosti nasazení výsledků práce v praxi nebo uveďte, zda výsledky ZP rozšiřují již publikované známé výsledky nebo přinášející zcela nové poznatky.

**Komentář:**

Využití nástroje jako takového je vzhledem k nekompletní funkcionalitě omezené. Nástroj si dovedu představit jako inspiraci pro jiné překladače, které zatím nemají podporu překladu QuakeC maker. Případně jako základní implementace, do které by musela být funkcionalita překladu chybějících standardních konstrukcí doplněna.

*Hodnotící kritérium:*

*Způsob hodnocení – nehodnotí se*

#### 5. Otázky k obhajobě

**Popis kritéria:**

Uveďte případné dotazy, které by měl student zodpovědět při obhajobě ZP před komisí (body oddělte odrážkami).

**Otázky:**

V práci píšete, že pro jazyk QuakeC je použitý standardní preprocessor C. Dále také zmiňujete, že nástroj c2rust (překladač C do Rustu) začíná podporovat makra. Můžete porovnat rozsah maker, které přeloží Váš nástroj, s rozsahem maker, které přeloží c2rust?

*Hodnotící kritérium:*

*Způsob hodnocení – bodové hodnocení 0 až 100 bodů (známka A až F):*

#### 6. Celkové hodnocení

73 (C)

**Popis kritéria:**

Shrňte stránky ZP, které nejvíce ovlivnily Vaše celkové hodnocení. Celkové hodnocení nemusí být aritmetickým průměrem či jinou hodnotou vypočtenou z hodnocení v předchozích jednotlivých kritériích. Obecně platí, že bezvadně splněné zadání je hodnoceno klasifikačním stupněm A.

**Text hodnocení:**

Vytvoření kompletního překladače z jednoho jazyka do jiného jazyka, i když jsou oba relativně podobné, není nikdy jednoduché. Z toho důvodu hodnotím zadání jako složitější. Student se zaměřil především na překlad maker, které v době tvorby zadání nebyly podporované v žádném z nástrojů pro překlad QuakeC ani C do Rustu. Výsledný překladač, i když není plně funkční, přeloží některé konstrukce QuakeC. V textu mi především chybí podrobný popis, jak sobě mají odpovídat konstrukce jazyka QuakeC a jazyka Rust. Celkově hodnotím práci 73 body, tedy stupněm C (dobře).

Podpis oponenta práce: