

**NAME**

**c2tc** - Miniaturní překladač pro jazyk C2. Momentálně pouze vypisuje informace o kódu.

**SYNOPSIS**

**c2tc** [-heidofatA?] [--help] [--usage] [--experiment] [--ast0] [--ast1] [--test] [-f [-o NAME]] [-d DIR] [--dir DIR] [TARGETS]

**DESCRIPTION**

**c2tc** je miniaturní překladač pro jazyk C2. Používá parsovací knihovnu **mpc** Daniela Holdena (orange-duck) (odkaz: <https://github.com/orangeduck/mpc> for more info). Verze obsažená v **c2tc** je modifikována k vypisování indexu uzlů stromu relativně k jejich rodičům a produkování barevného výstupu. Protože **c2tc** nepoužívá ad hoc parser, ale tuto knihovnu, je parsovací efektivita někde mezi  $O(n)$  a  $O(n^2)$ . To je obzvlášť znatelné při parsování velkých souborů (10k řádků a více). Parsování momentálně zabírá nejvíc času ze všech úloh prováděných překladačem.

Pro sbírky dynamické velikosti **c2tc** používá vlastní implementaci vektoru podle tutoriálu Edda Manna (link: <http://eddmann.com/posts/implementing-a-dynamic-vector-array-in-c/>). Verze vektoru v **c2tc** byla původně implementována jako vkladatelná knihovna pro C2. Minimalistická implementace lineárního spojového seznamu je využívána vloženou knihovnou ooc. Knihovna ooc poskytuje základní prostředky pro objektově-orientované programování, ale zůstává z většiny nevyužitá, protože **c2tc** ještě nezpracovává vnitřky funkcí.

Veškeré zpracovávání chyb je zprostředkováno miniaturní knihovnou **'throw'** Josepha Werleho <[joseph.werle@gmail.com](mailto:joseph.werle@gmail.com)>. V **c2tc** je **'throw'** zabudováno do souborů *error.c/h* z důvodu ušetření místa a snížení počtu zdrojových souborů v repozitáři. Poskytnutá verze je modifikována k tomu, aby využívala knihovnu **'log.h'** Thorstena Lorenze k produkování výstupu do stderr.

Logging je zpracován dříve-jmenovanou knihovnou **'log.h'**. Momentálně **c2tc** vypisuje veškeré informace do stdout, což se pravděpodobně nezmění. Logging do souboru je jednoduše zprostředkován pomocí přesměrování výstupu v shellu. **c2tc** se snaží být co nejmenší, a proto logging do souborů v **c2tc** není vítaným rozšířením.

**c2tc** má velmi malé nároky na paměť. Mnoho ukazatelů ale není úmyslně uvolněno, a proto dochází k únikům paměti. Paměť využitá **c2tc** pravděpodobně nikdy nedosáhne velikostí, které by byly ohrožující pro současné počítače, a proto malé úniky paměti nejsou akutním problémem. Kvůli únikům paměti není **c2tc** doporučen na vestavěné systémy, přestože jej lze zkompileovat a spustit na jakémkoliv systému, který splňuje závislosti.

**OPTIONS****-h, --help**

Zobrazí text pro pomoc.

**-, --usage**

Zobrazí text pro užití.

**-i, --info**

Zobrazí extra informace při běhu programu.

**-d, --dir path**

Přejít do složky *path* před hledáním recipe souboru.

**-a, --ast0**

Vypíše syntaxní strom před tím, než proběhne jakékoliv čištění/zjednodušování

**-A, --ast1**

Vypíše syntaxní strom po jeho zjednodušení

**-o, --output name**

Změní jméno vygenerovaného cíle v souborovém módu.

**-e, --experiment**

Zapne experimentální funkce. Změna pravděpodobně nebude postřehnuta uživatelem. Experimentální funkce se s vývojem rychle mění.

**-f, --file**

Přepíná na souborový mód. Souborový mód znamená, že jména, která c2tc dostal jako vstup, jsou brána jako jména souborů spíše než jména cílů. c2tc vygeneruje ad hoc cíl pro tyto soubory, jehož jméno může být změněno přepínačem -o.

**-t, --test**

Vypíše testovací informace. To také znamená zacházení se soubory jako s testy spíše než kompilačními jednotkami.

**MODUS OPERANDI**

- 1 c2tc přečte parametry z příkazového řádku. Pokud není spuštěný souborový mód, začne c2tc iterativně hledat recipe.txt, dokud ho nenajde nebo nedosáhne kořenového adresáře. Složka se souborem recipe.txt se stane pracovním adresářem programu.
- 2 c2tc přečte soubor recipe.txt a odebere komentáře a prázdné řádky. Zbytek je potom parsován čtečkou recipe souborů, která, na rozdíl od C2 parseru, je ad hoc parser, a vytvoří cíle na základě dat.
- 3 na základě informací poskytnutých v souboru recipe.txt c2tc najde soubory a zkontroluje, jestli jsou čitelné. Pokud c2tc narazí na nečitelný soubor, vypíše chybu.
- 4 soubory jsou pak předány parsovací části programu. Nejprve jsou ze zdrojového kódu odebrány komentáře, poté je text předán C2 parseru.
- 5 C2 parser parsuje soubor a vrátí nezpracovaný syntaxní strom. Ten je potom předán čistícím funkcím.
- 6 Nejdříve se ze stromu vyhází nepotřebné tokeny jako jsou závorky, uvozovky nebo přebytečná klíčová slova. Dále se zjednoduší značky jednotlivých uzlů stromu - odeberou se z nich speciální znaky, zkombinují se jména parsovacích pravidel a zruší se nepotřebná jména zabudovaných prvků (např. 'head|module|>' -> 'module', 'ident|regex' -> 'ident').
- 7 Vyčištěné stromy jsou poté předány analyzáru. Analyzář a spol. jsou momentálně jediná část c2tc, která není napsána v C, ale v Rustu. Analyzář poté přečte každý strom a vytvoří objekty (resp. struktury), které obsahují data o některých typech symbolů nalezených ve stromech. Typy podporované ve vydání pro Informatiku 2016/2017 jsou: import příkazy, uživatelem definované typy a funkce.
- 8 Objekty jsou poté přečteny a je z nich vytvořen barevný výstup do stdout.

**LIMITACE**

Přestože c2tc dokáže parsovat veškerý kód v jazyce C2, provádí ho i několik limitací. Za prvé, c2tc nedokáže zpracovat makra. Jazyk C2 momentálně nemá vůbec makra, ale standardní překladač C2C je schopen zpracovat makra jazyka C, pokud jsou vložena do souborů C2. Protože v jazyce C2 oficiálně makra nejsou, nemá je ani c2tc.

c2tc používá k parsování knihovnu mpc. To na jednu stranu přináší velké výhody v podobě čitelné gramatiky jazyka a bezchybnosti parseru, ale na druhou stranu způsobuje i nevýhody v podobě delších parsovacích časů a špatného zobrazování syntaxních chyb. V současné době mpc pouze prozradí, kde (resp. u jakého bloku kódu) se chyba nachází, a co se mpc pokoušelo parsovat, když narazilo na chybu.

**MAPA KÓDU**

**arg.h** Modifikovaný header pro zpracování přepínačů z příkazové řádky. Původně napsáno Christopherem Lohmannem. Změny zahrnují možnost dát přepínače kamkoliv do příkazu, přepínače psané celým slovem (např. **--test**), makra pro jednoduché přepínače a integrace s c2tc pro lepší čtení cílů/souborů určených ke kompilaci. Kvůli integraci s c2tc je verze 'arg.h' v c2tc nepoužitelná v jiných programech.

**errors.c/h**

Obsahuje definice chyb a dříve-jmenovanou knihovničku 'throw'. Verze 'throw' v c2tc byla modifikována tak, aby používala knihovnu 'log.h' k logování a neukončovala program při první chybě, ale místo toho zaznamenávala počet chyb.

**log.h** Malá logovací knihovna

**main.c** Hlavní soubor. Obsahuje funkci `main()` a funkce pro zpracování přepínačů a pro zobrazení pomocného textu.

**mpc.c/h**

Relativně malá (cca 3k řádků) parsovací knihovna. `c2tc` ji využívá k vlastnímu parsování jazyka

**ooc.c/h** Malá knihovna zprostředkující minimální, neúplnou objektovou orientaci. `ooc` je zatím jediná moje knihovna, která je (ne)licencovaná jako volné dílo.

**parsers.c**

Soubor obsahující jak parser pro C2 tak pro `recipe.txt` soubory.

**recipe.c**

Obsahuje funkce pro zpracování `recipe.txt` souborů.

**shared.h**

Společný header obsahující definice veřejných symbolů ze souborů, kde by jich bylo moc málo na to, aby bylo potřeba vytvořit vlastní header. Také obsahuje deklarace globálních proměnných.

**symbol\_types.h**

Header obsahující definice typů symbolů jazyka C2. Momentálně nevyužitý, jelikož zpracované symboly zatím neopouští část programu napsanou v Rustu.

**tests.c**

Obsahuje funkci pro testování pomocných funkcí pro práci s řetězci a funkcí pro práci s implementací vektorů podle Edda Manna.

**tree\_transform.c/h**

Obsahuje funkce pro vyčištění a zjednodušení stromu.

**types.h** Obsahuje definice struktur pro cíle, `recipe.txt` atd. Také obsahuje výčtové typy.

**types.rs**

Obsahuje definice typů pro část programu napsanou v Rustu.

**analyse.rs**

Obsahuje část programu, která analyzuje kód.

**util.rs**

Pomocné funkce pro Rustovou část `c2tc`.

## PŘÍKLADY

Příklady parsovatelné s `c2tc` je možné najít ve složkách `/build/orig` a `/build/tests`. Ve složce `/build/orig` jsou delší a komplexnější příklady, původně pro překladač C2C, na kterých jsem se podílel s Basem van den Bergem, v `/build/tests` jsou jednoduché příklady kódu, vytvořené specificky pro testování `c2tc`.

## LICENSE

`c2tc` je licencováno licencí 'ISC license':

ISC License

Copyright (c) 2016-2017, Lukáš Hozda <luk.hozda@gmail.com>

Permission to use, copy, modify, and/or distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.