

Teoria Kompilacji

Kompilator okrojonej wersji języka C do bytecode'u - wizja projektu

Filip Buszko
Artur Orzeł
III rok Informatyka IEiT

1 Ogólna wizja rozwiązania

W ramach projektu zostanie stworzony kompilator okrojonej wersji języka C. Zakładamy, że na wejście programu trafiać będzie plik po preprocessingu - zostały rozwinięte zdefiniowane makra.

Rozwiązanie będzie składać się z dwóch części: generacji czytelnego dla człowieka zdeasemblowanego bytecode'u (takiego, jaki zwraca polecenie **javap -c**), a następnie translacja otrzymanego kodu na właściwy kod bajtowy zgodnie z tabelą kodów http://en.wikipedia.org/wiki/Java_bytecode_instruction_listings.

2 Opis kompilowanego języka

Ze względu na trudności z tym związane zdecydowaliśmy nie obsługiwać instrukcji **include** i podobnych, powodujących odniesienie do innych plików. Będziemy obsługiwać tylko znaki ASCII w stałych znakowych. Zakładamy, że wszystkie liczby posiadają znak. Będziemy obsługiwać liczby całkowite w formacie dziesiętnym, binarnym, ósemkowym oraz szesnastkowym, a także liczby zmiennoprzecinkowe (również w postaci eksponencjalnej).

Uwzględnimy wszystkie podstawowe operatory dwuargumentowe oraz jednoargumentowe prefixowe, także operatory łączone (wykonujące działania i podstawienie wyniku). Pominiemy operatory rzutowania oraz sizeof. Zignorujemy kwalifikatory **volatile**, **inline**, **restrict**, **const**, **static**.

Typami obsługiwanymi będą:

- char,
- short,
- int,
- long,
- float,
- double

oraz typy złożone:

- struct,
- enum.

Obsługiwanymi instrukcjami sterującymi będą:

- if-else
- switch
- while
- for
- do-while
- break
- continue
- return

3 Planowane technologie

Rozwiązanie zostanie wykonane w języku python z wykorzystaniem biblioteki **PLY**, będącej implementacją narzędzi **lex** oraz **yacc** w języku python. Za zgodą prowadzącego możemy również rozważyć wykorzystanie gotowych narzędzi do parsingu języka C, np biblioteki **pyparser** (<https://bitbucket.org/eliben/pyparser>).