

# Kompilatory

## Laboratorium 10

Piotr Błaszyński

3 marca 2021

Zadania (wyjaśnienie w dalszej części dokumentu):

- dodać kompilację tworzenia i używania statycznych tablic jednowymiarowych,
  - reguły gramatyki,
  - generowanie kodu wynikowego,

Obsługa **tablic jednowymiarowych** składa się z dwóch elementów: zapamiętania rozmiaru tablicy i odwoływania się do elementu tablicy (uwaga: przypominam, że odwołania mogą występować po lewej i prawej stronie wyrażeń). Wyjściowe reguły gramatyki dla deklaracji (uproszczone w stosunku do C - w C rozmiar można określić wyrażeniem stałym i deklarować wiele zmiennych):

```
arr_decl
: INT ID '[' LC ']' ';'
    { tablica_symboli[$2], typ=ARRI, rozmiar=$4; }
: FLOAT ID '[' LC ']' ';'
    { tablica_symboli[$2], typ=ARRF, rozmiar=$4; }
```

Wyjściowe reguły gramatyki dla indeksowania:

```
arr_expr
: ID '[' wyr ']' '
```

Rozmiar tablicy w MIPS podajemy po dwukropku (lepiej użyć *.word*). Przy odwoływaniu się do tablicy należy pomnożyć przez 4 wartość przesunięcia względem adresu początkowego, adres początku tablicy ładujemy przy pomocy *la*. Wartość pod adres wpisujemy lub pobieramy przy pomocy nawiasów okrągłych (np. *sw \$t0, (\$t4)*)

Dla kodu:

```
int a[10];  
a[3] = 2;  
x = a[2+1];
```

Należy wygenerować następujący kod (mnemoniki):

```
.data  
    a: .word 0:10  
    x: .word 0  
    result1: .word 0  
.text  
li $t0, 2  
la $t4, a  
li $t5, 3  
mul $t5, $t5, 4  
add $t4, $t4, $t5  
sw $t0, ($t4)  
  
li $t0, 2  
li $t1, 1  
add $t0, $t0, $t1  
sw $t0, result1  
la $t4, a  
lw $t5, result1  
mul $t5, $t5, 4  
add $t4, $t4, $t5  
lw $t0, ($t4)  
sw $t0, x
```