

Kompilatory

Laboratorium 14

Piotr Błaszyński

3 marca 2025

Zadania (wyjaśnienie w dalszej części dokumentu):

- dodać kompilację instrukcji do obsługi struktur (ew. klas),
 - reguły gramatyki,
 - generowanie kodu wynikowego,

Struktury w MIPS realizuje się poprzez odpowiednie umieszczenie zmiennej będącej strukturą w pamięci. Dla każdego elementu należącego do struktury, w tym także dla struktur wchodzących w skład struktury należy określić liczbę zajmowanych komórek. W przypadku potrzeby obsługi dynamicznie alokowanych struktur konieczne jest skorzystanie z przerwania *sbrk*, jak przy dynamicznie alokowanych tablicach. W przypadku statycznej zmiennej będącej strukturą, należy w bloku danych zarezerwować blok pamięci o odpowiednim rozmiarze (dla każdej zmiennej będącej strukturą). Ponadto, przydatne jest zachowanie przesunięcia poszczególnych pól względem początku struktury (przykładowa realizacja: mapa odwzorowująca nazwy elementów struktury na przesunięcie). Przy odwołaniu do elementu struktury należy postępować podobnie jak przy tablicach: do adresu bazowego dodajemy przesunięcie i odwołujemy się do komórki pod adresem będącym wynikiem tej operacji.

Dla przykładowego kodu:

```
struct example
{
    int x;
    float z;
    int y;
};
struct example e1;
e1.y = 42;
e1.x = 10;

struct example e2;
e2.y = 242;
e2.x = 210;
```

Należy wygenerować kod:

```
.data
    e1:      .space 12
    e2:      .space 12

.text

    li $t0, 42
    la $t4, e1
    add $t4, $t4, 8
    sw $t0, ($t4)

    li $t0, 10
    la $t4, e1
    add $t4, $t4, 0
    sw $t0, ($t4)

    li $t0, 242
    la $t4, e2
    add $t4, $t4, 8
    sw $t0, 8($t4)

    li $t0, 210
    la $t4, e2
    add $t4, $t4, 0
    sw $t0, ($t4)
```

W przypadku struktur można odwołanie do elementu uprościć (ponieważ nie ma potrzeby indeksowania wyrażeniem):

```
.data
    e1:      .space 12
    e2:      .space 12
.text
    li $t0, 42
    la $t4, e1
    sw $t0, 8($t4)

    li $t0, 10
    la $t4, e1
    sw $t0, 0($t4)

    li $t0, 142
    la $t4, e2
    sw $t0, 8($t4)

    li $t0, 210
    la $t4, e2
    sw $t0, 0($t4)
```