

# List 6

Saturday, 1 April 2023 12:38



ask23\_liste\_

6

## Architektury systemów komputerowych

Lista zadań nr 6

Narzędzia 5 i 18 kwietnia 2023

Należy być przygotowanym do wyjaśnienia semantyki każdej instrukcji, która pojawia się w treści zadania. W tym celu posłuży się dokumentacją [x86 and amd64 instruction reference](#). W szczególności trzeba wiedzieć jak dana instrukcja korzysta z rejestrów flag, gdzie obliczenia zależą od jej wartości.

W trakcie tłumaczenia kodu z członka ABI x86-64 do języka C należy trzymać się następujących wytycznych:

- Używać złożonych wyrażeń minimalizujących liczby zmienionych tymczasowych.
- Nazwy wprowadzonych zmiennej muszą opisywać ich znaczenie na result zamianie rax.
- Instrukcja goto jest zabroniona. Należy używać instrukcji sterowania if, while lub switch.
- Polecenie voids nie należy przetłumaczyć do pętli if, for, while lub switch.

Uwaga! Przedstawienie rozwiązań nieostosującego się powyższych zasad może skutkować negatywnymi konsekwencjami.

Zadanie 1. Ponizszy wydruk oznaczający wynik dekomplikacji rekurencyjnej procedury zdefiniowanej następująco: `long pointless(long n, long *p);` Zapisz w języku C kod odpowiadający tej procedurze. Następnie opisz zawartość jej rekordu aktywacji (ang. stack frame). Wskaz rejestr zapisane przez funkcję `wofang` (ang. callee-saved registers), zmienne lokalne i adres powrotu. Następnie uzasadni, że wartość rejestrów rax w wierszu 10 jest podzielna przez 16 – zgodnie z [1, 3.2.2]. Zastanów się czemu autorzy ABI zdecydowali się na taką konwencję?

```
1: pointless:
2: {
3:     long local_18;
4:     long lVar1;
5:     long _var1;
6: 
7:     if (n < 1) {
8:         lVar1 = 0;
9:     }
10:    else {
11:        local_18 = __RAX;
12:        lVar1 = pointless(n, &local_18);
13:        lVar1 = Var1 + local_18;
14:    }
15:    *p = n + lVar1;
16:    return;
17: }
```

Zadanie 2. Ponizszy zamieszczono kod procedury o sygnaturze `struct T puzzle2(long *a, long n);`. Na jego podstawie podaj typ struct `T`. Przetłumacz tą procedurę na język C, co czym jednym zdaniem zakończy Twoje zajęcia. Działanie instrukcji `if` i `else`. Gdyby sygnatura procedury nie była zrozumiała, mani i jaką nazwą powinniśmy ją zapisywać? Zauważ, że wynik procedury nie mieści się w rejestrach `rax` i `rdi`, zatem zostanie umieszczone w pamięci. Wskaz rejestr w [1, 3.2.3], który wymusza takie zachowanie kompilatora.

```
1: puzzle2:
2: {
3:     long rdx, Rdx1, L1;
4:     long xorl, Xor104, Xor10;
5:     long addq, Add1, Add2;
6:     long movq, Mov1, Mov2;
7:     long novaq, Novaq1, Novaq2;
8:     long leaq, Leaq1, Leaq2;
9:     long idivq, Idivq1, Idivq2;
10:    long cmpq, Cmp1, Cmp2;
11:    long leaq, Leaq3, Leaq4;
12:    long ret;
13: }
```

<http://www.felixcloutier.com/x86/>

## STACK FRAME

redukcyjny adres

lvar1

lvar1