# Architektury systemów komputerowych

Lista zadań nr 6

Na zajęcia 16 kwietnia 2020
(świąteczna, skrócona)

Przy tłumaczeniu kodu w asemblerze x86-64 do języka C należy trzymać się następujących wytycznych:

- Używaj złożonych wyrażeń minimalizując liczbę zmiennych tymczasowych.
- Nazwy wprowadzonych zmiennych muszą opisywać ich zastosowanie, np. `result` zamiast `rax`.
- Instrukcja `goto` jest zabroniona. Należy używać instrukcji sterowania `if`, `for`, `while` i `switch`.
- Jeśli to ma sens pętle `while` należy przetłumaczyć do pętli `for`.

**UWAGA!** W trakcie prezentacji rozwiązań należy zdefiniować i wyjaśnić pojęcia, które zostały oznaczone **wytłuszczoną** czcionką.

**Zadanie 1.** Poniższy wydruk otrzymano w wyniku deasemblacji rekurencyjnej procedury zadeklarowanej następująco: «`long puzzle(long n, long *p)`». Zapisz w języku C kod odpowiadający tej procedurze. Następnie opisz zawartość jej **rekordu aktywacji** (ang. *stack frame*). Wskaż **rejestry zapisane przez funkcję wołaną** (ang. *callee-saved registers*), zmienne lokalne i adres powrotu.

```
1  puzzle:
2          push  %rbp
3          xorl  %eax, %eax
4          mov   %rsi, %rbp
5          push  %rbx
6          mov   %rdi, %rbx
7          sub   $24, %rsp
8          test  %rdi, %rdi
9          jle   .L1
10         lea   8(%rsp), %rsi
11         lea   (%rdi,%rdi), %rdi
12         call  puzzle
13         add   8(%rsp), %rax
14         add   %rax, %rbx
15 .L1:    mov   %rbx, (%rbp)
16         add   $24, %rsp
17         pop   %rbx
18         pop   %rbp
19         ret
```

**Uwaga!** Wskaźnik wierzchołka stosu w momencie wołania procedury musi być wyrównany do adresu podzielnego przez 16.

**Zadanie 2.** Skompiluj poniższy kod źródłowy kompilatorem `gcc` z opcjami «`-Og -fomit-frame-pointer`» i wykonaj deasemblację **jednostki translacji** przy użyciu programu «`objdump`». Wytłumacz co robi procedura `alloca(3)`, a następnie wskaż w kodzie maszynowym instrukcje realizujące przydział i zwalnianie pamięci.

```
#include <alloca.h>

long aframe(long n, long idx, long *q) {
  long i;
  long **p = alloca(n * sizeof(long *));
  p[n-1] = &i;
  for (i = 0; i < n; i++)
    p[i] = q;
  return *p[idx];
}
```

**Wskazówka:** Przeczytaj również co robi instrukcja «`leave`».

**Zadanie 3.** Poniżej widnieje kod procedury o sygnaturze «`long puzzle5(void)`». Podaj rozmiar i składowe rekordu aktywacji procedury «`puzzle5`». Procedura «`readlong`», która wczytuje ze standardowego wejścia liczbę całkowitą, została zdefiniowana w innej jednostce translacji. Jaka jest jej sygnatura? Przetłumacz procedurę «`puzzle5`» na język C i wytłumacz jednym zdaniem co ona robi.

```
1  puzzle5:
2        subq   $24, %rsp
3        movq   %rsp, %rdi
4        call   readlong
5        leaq   8(%rsp), %rdi
6        call   readlong
7        movq   (%rsp), %rax
8        cqto
9        idivq  8(%rsp)
10       xorl   %eax, %eax
11       testq  %rdx, %rdx
12       sete   %al
13       addq   $24, %rsp
14       ret
```

**Zadanie 4.** Procedurę ze zmienną liczbą parametrów używającą pliku nagłówkowego `stdarg.h`[1] skompilowano z opcjami «`-Og -mno-sse`». Po jej deasemblacji otrzymano następujący wydruk. Co robi ta procedura i jaka jest jej sygnatura? Jakie dane są przechowywane w rekordzie aktywacji tej procedury? Prezentację zacznij od przedstawienia definicji struktury «`va_list`».

```
1  puzzle7:
2        movq  %rsi, -40(%rsp)
3        movq  %rdx, -32(%rsp)
4        movq  %rcx, -24(%rsp)
5        movq  %r8, -16(%rsp)
6        movq  %r9, -8(%rsp)
7        movl  $8, -72(%rsp)
8        leaq  8(%rsp), %rax
9        movq  %rax, -64(%rsp)
10       leaq  -48(%rsp), %rax
11       movq  %rax, -56(%rsp)
12       movl  $0, %eax
13       jmp .L2
14 .L3:  movq  -64(%rsp), %rdx
15       leaq  8(%rdx), %rcx
16       movq  %rcx, -64(%rsp)
17 .L4:  addq  (%rdx), %rax
18 .L2:  subq  $1, %rdi
19       js  .L6
20       cmpl  $47, -72(%rsp)
21       ja  .L3
22       movl  -72(%rsp), %edx
23       addq  -56(%rsp), %rdx
24       addl  $8, -72(%rsp)
25       jmp .L4
26 .L6:  ret
```

**Wskazówka:** Przeczytaj rozdział §3.5.7 dokumentu opisującego ABI dostępnego na stronie przedmiotu.

[1] https://en.wikipedia.org/wiki/Stdarg.h