Zad 7.

czwartek, 13 kwietnia 2023 21:02

Zadanie 7 (2). Procedurę ze zmienną liczbą parametrów używającą pliku nagłówkowego `stdarg.h`[3] skompilowano z opcjami «-Og -mno-sse». Po jej deasemblacji otrzymano następujący wydruk. Przetłumacz procedurę «puzzle7» na język C i wytłumacz jednym zdaniem co ona robi. Narysuj rekord aktywacji procedury, a następnie podaj jego rozmiar i składowe. Prezentację zacznij od przedstawienia definicji struktury «va_list» na podstawie [1, 3.5.7].

```
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
```

setup

dodajemy pobrany arg do wyniku

pętla po argum. w %rdi

pobieranie argumentu ze stosu (jak skończą się rejestry)

-Og wyłącza optymalizacje

-mno-sse (?)

sprawdzamy czy nie skończyły się rejestry

pobieranie argumentu z rejestrów

→ zapisywanie rejestrów do reg_save_area

→ inigalizacja struktury va_list

→ When a function taking variable-arguments is called, `%al` must be set to the total number of floating point parameters passed to the function

w tym pizypadku ustawiamy na 0

-72(%rsp) = gp_offset ustawiony na 8

-64(%rsp) = overflow_arg_area → wskaznik na arg ze stosu [8(%rsp)]

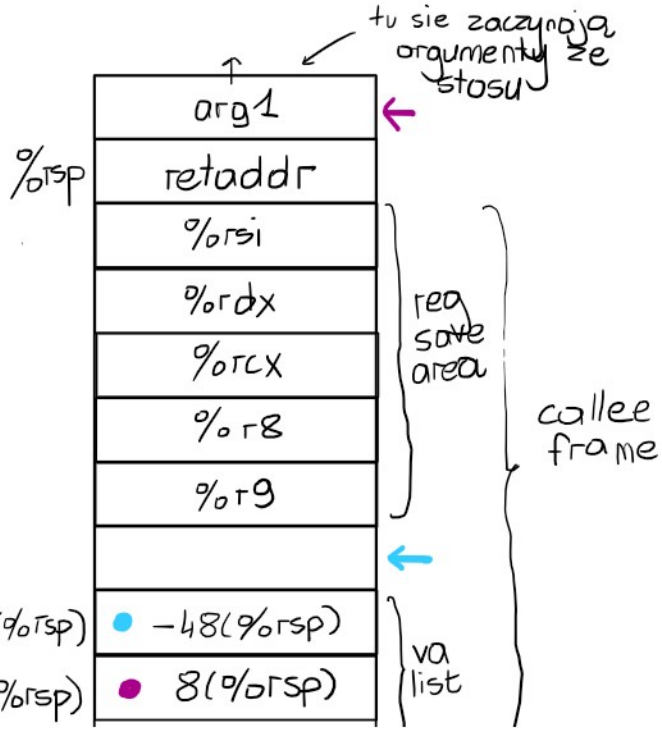
-56(%rsp) = wskaźnik na początek reg_save_area -48(%rsp)

```
11 #include <stdarg.h>
12 long sum(long n, ...) {
13     va_list args;
14     long result = 0;
15     va_start(args, n);
16     for (int i = 0; i < n; i++) {
17         result += va_arg(args, long);
18     }
19     va_end(args);
20     return result;
21 }
```

sumuje n argumentów

stack frame po 11 linijce

rozmiar: 72 bajtu



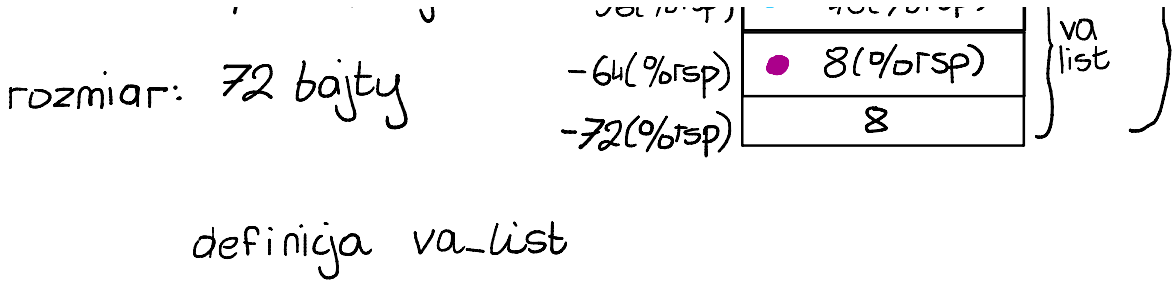


Figure 3.34: `va_list` Type Declaration

```
typedef struct {
   unsigned int gp_offset;
   unsigned int fp_offset;
   void *overflow_arg_area;
   void *reg_save_area;
} va_list[1];
```

The `va_list` Type

The `va_list` type is an array containing a single element of one structure containing the necessary information to implement the `va_arg` macro. The C definition of `va_list` type is given in figure 3.34.