

## Zad 1.

niedziela, 23 kwietnia 2023 10:29

**Zadanie 1.** Posługując się ABI dla architektury x86-64 wyznacz rozmiar struktury «node». Dla każdej składowej typu «node» podaj: wymaganie na wyrównanie (funkcja alignof), przesunięcie względem początku struktury node (funkcja offsetof), rozmiar (funkcja sizeof). Następnie zoptymalizuj rozmiar struktury zmieniając kolejność pól. Wyznacz rozmiar struktury po optymalizacji.

```
1 struct node {
2   char id[2];
3   int (*hashfn)(char *);
4   short flags;
5   union {
6     struct {
7       short n_key;
8       int n_data[2];
9       unsigned char n_type;
10    } s;
11    unsigned l_value[2];
12  } u;
13 };
14
```

Wskazówka: Zapoznaj się z dodatkowymi slajdami do wykładu 7.

	alignof	offsetof	sizeof
id	1	0	2
hashfn	8	8	8
flags	2	16	2
u	4	20	16
↓			
s	4	0 [20]	16
↓			
n_key	2	0 [20]	2
n_data	4	4 [24]	8
n_type	1	12 [32]	1
l_value	4	0 [20]	8

przed opt.  
 sizeof(node) = 40  
 alignof(node) = 8

optymalizacja

```
11 struct node {
12   int (*hashfn)(char*);
13   union {
14     struct {
15       int n_data[2];
16       short n_key;
17       unsigned char n_type;
18     } s;
19     unsigned l_value[2];
20   } u;
21   short flags;
22   char id[2];
23 };
24
```

sizeof(struct node) = 24

### Częściowa definicja alignof:

- alignof( $t$ ) = sizeof( $t$ ) dla  $t \in \{\text{char}, \text{short}, \text{int}, \text{long}, \text{float}, \text{double}\}$
- alignof( $t *$ ) = sizeof( $t *$ )
- alignof( $t []$ ) = alignof( $t$ )
- alignof(struct  $s \{t_1 f_1; t_2 f_2; \dots; t_n f_n;\}$ ) =  $\max_i \{\text{alignof}(t_i)\}$
- alignof(union  $u \{t_1 f_1; t_2 f_2; \dots; t_n f_n;\}$ ) =  $\max_i \{\text{alignof}(t_i)\}$

### Definicja offsetof:

Dla struct  $s \{t_1 f_1; t_2 f_2; \dots; t_n f_n;\}$ :

offsetof( $s, f_1$ ) = 0

offsetof( $s, f_i$ ) = alignup(offsetof( $s, f_{i-1}$ ) + sizeof( $f_{i-1}$ ), alignof( $t_i$ ))

### Definicja sizeof dla typów maszynowych dla x86-64:

- sizeof(char) = 1,
- sizeof(short) = 2,
- sizeof(int) = 4,
- sizeof(long) = 8,
- sizeof(float) = 4,
- sizeof(double) = 8,
- sizeof( $t *$ ) = 8

### Definicja sizeof dla typów złożonych:

- sizeof( $t [n]$ ) =  $n * \text{sizeof}(t)$
- sizeof(struct  $s \{t_1 f_1; t_2 f_2; \dots; t_n f_n;\}$ ) = alignup(offsetof( $s, f_n$ ) + sizeof( $f_n$ ),  $\max_i \{\text{alignof}(t_i)\}$ )
- sizeof(union  $u \{t_1 f_1; t_2 f_2; \dots; t_n f_n;\}$ ) = alignup( $\max_i \{\text{sizeof}(t_i)\}$ ,  $\max_i \{\text{alignof}(t_i)\}$ )

### alignup( $p, n$ ) = $q$

Wprowadźmy na dodatkową funkcję, która będzie wyznaczać najmniejszą liczbę  $q$  większą równą  $p$  i podzielną przez  $n$ , będącą potęgą dwójki.

```
uintptr_t alignup(uintptr_t p, size_t n) {
    return (p + n - 1) & -n;
}
```