

## Latihan Soal 9 2022

### 1. Array

Diberikan kode assembly berikut:

```
local_array:
    pushl %ebp
    movl %esp, %ebp
    subl $32, %esp
    movl $0x1, 0xffffffff0(%ebp)
    movl $0x3, 0xffffffff4(%ebp)
    movl $0x5, 0xffffffff8(%ebp)
    movb $0x69, 0xffffffffeb(%ebp)
    movb $0x66, 0xffffffffec(%ebp)
    movb $0x69, 0xffffffffed(%ebp)
    movb $0x74, 0xffffffffee(%ebp)
    movb $0x62, 0xffffffffef(%ebp)
    movl -8(%ebp), %eax
    andl 8(%ebp), %eax
    movl %eax, -4(%ebp)
    leal -21(%ebp), %edx
    movl -4(%ebp), %eax
    addl %edx, %eax
    movzbl (%eax), %eax
    movsbl %al, %eax
    movl %ebp, %esp
    popl %ebp
    ret
```

Apabila diketahui bahwa huruf 'a' merupakan karakter ASCII ke 97 dan 'b' merupakan karakter ASCII ke 98, dan seterusnya. **Lengkapilah** kode bahasa C berikut berdasarkan kode assembly di atas:

```
_____ local_array(int i)
{
    _____ A[3] = { _____ };

    _____ B[5] = { _____ };

    int idx = _A[2] & i____;

    return _____B[idx] ____;
}
```

Jika argumen i dalam fungsi local\_array diberi nilai 15, nilai apa yang akan

dikembalikan oleh fungsi local\_array tersebut? 0101 & 1111 = 5 (0101) = B[5]= A[0] byte yang terendah = 1\_\_\_\_\_

## 2. Cache (dikerjakan setelah pembahasan cache)

Diketahui sebuah komputer memiliki memori 32 byte, byte addressable, dengan isi memori sebagai berikut

Address	Value	Address	Value	Address	Value	Address	Value
0	0xF0	8	0xED	16	0x32	24	0x8A
1	0xBC	9	0x03	17	0x3E	25	0xA0
2	0xBC	10	0x54	18	0x9E	26	0xB6
3	0xBE	11	0x2F	19	0x7E	27	0xC0
4	0x7E	12	0x32	20	0x21	28	0x8A
5	0x98	13	0xA9	21	0x76	29	0x54
6	0x38	14	0x5D	22	0x4D	30	0xBC
7	0x8A	15	0x04	23	0x2A	31	0x9E

Untuk meningkatkan kinerja pembacaan memori, komputer tersebut memiliki **two way set associative cache** seukuran 8 byte dengan dua buah set.

S = \_\_\_\_\_ s = \_\_\_\_\_  
 E = \_\_\_\_\_ e = \_\_\_\_\_  
 B = \_\_\_\_\_

Jika dilakukan pengaksesan memori 1 byte word dengan urutan di bawah ini, tentukan apakah miss (m) atau hit (h), jika diasumsikan pada awalnya cache kosong (*cold cache*):

Address	0	16	1	8	15	25	31	24	19	11	10
Miss/hit	m										

Gambarkan kondisi cache terakhir setelah selesainya pengaksesan memori.

Set Idx	Line 0				Line 1			
	Tag	V	Bytes		Tag	V	Bytes	
			0	1			0	1
0								
1								

## 4. Struktur Data

Diberikan struktur data sebagai berikut pada mesin IA32:

```

struct s1 {
    char a[3];
    union u1 b;
    int c;
};

struct s2 {
    struct s1 *d;
    char e;
    int f[4];
    struct s2 *g;
};

union u1 {
    struct s1 *h;
    struct s2 *i;
    char j;
};

```

Lengkapilah kode C yang kosong pada pasangan kode assembly – bahasa C di bawah ini

<pre> proc1: pushl %ebp movl %esp,%ebp movl 8(%ebp),%eax movl 12(%eax),%eax movl %ebp,%esp popl %ebp ret </pre>	<pre> int proc1(struct s2 *x) { Return      ; } </pre>
<pre> proc2: pushl %ebp movl %esp,%ebp movl 8(%ebp),%eax movl 4(%eax),%eax movl 20(%eax),%eax movl %ebp,%esp popl %ebp ret </pre>	<pre> int proc2(struct s1 *x) { return      ; } </pre>
<pre> proc3: pushl %ebp movl %esp,%ebp movl 8(%ebp),%eax movl (%eax),%eax movsbl 4(%eax),%eax movl %ebp,%esp popl %ebp ret </pre>	<pre> char proc3(union u1 *x) { } </pre>
<pre> proc4: pushl %ebp movl %esp,%ebp movl 8(%ebp),%eax movl (%eax),%eax movl 24(%eax),%eax movl (%eax),%eax movsbl 1(%eax),%eax movl %ebp,%esp popl %ebp ret </pre>	<pre> char proc4(union u1 *x) { return      ; } </pre>

5. Diberikan kode C rekursif berikut:

```
int silly(int n, int *p)
{
    int val, val2;
    if (n > 0)
        val2 = silly(n << 1, &val);
    else
        val = val2 = 0;
    *p = val + val2 + n;
}
```

dengan hasil assembly sebagai berikut

```
silly:
    pushl %ebp
    movl %esp,%ebp
    subl $20,%esp
    pushl %ebx
    movl 8(%ebp),%ebx
    testl %ebx,%ebx
    jle .L3
    addl $-8,%esp
    leal -4(%ebp),%eax
    pushl %eax
    leal (%ebx,%ebx),%eax
    pushl %eax
    call silly
    jmp .L4
.p2align 4,,7
.L3:
    xorl %eax,%eax
    movl %eax,-4(%ebp)
.L4:
    movl -4(%ebp),%edx
    addl %eax,%edx
    movl 12(%ebp),%eax
    addl %edx,%ebx
    movl %ebx,(%eax)
    movl -24(%ebp),%ebx
    movl %edx,%eax
    movl %ebp,%esp
    popl %ebp
    ret
```

- a. Apakah variabel val disimpan pada stack? Jika iya, pada byte offset berapakah (relatif terhadap %ebp) variabel tersebut disimpan? Mengapa perlu disimpan pada stack?
- b. Apakah variabel val2 disimpan pada stack? Jika iya, pada byte offset berapakah (relatif terhadap %ebp) variabel tersebut disimpan? Mengapa perlu disimpan pada stack?

- c. Apakah ada nilai yang disimpan (jika ada) pada posisi `-24(%ebp)`? Jika ada yang disimpan, mengapa nilai tersebut perlu disimpan?
- d. Apakah ada nilai yang disimpan (jika ada) pada posisi `-8(%ebp)`? Jika ada yang disimpan, mengapa nilai tersebut perlu disimpan?