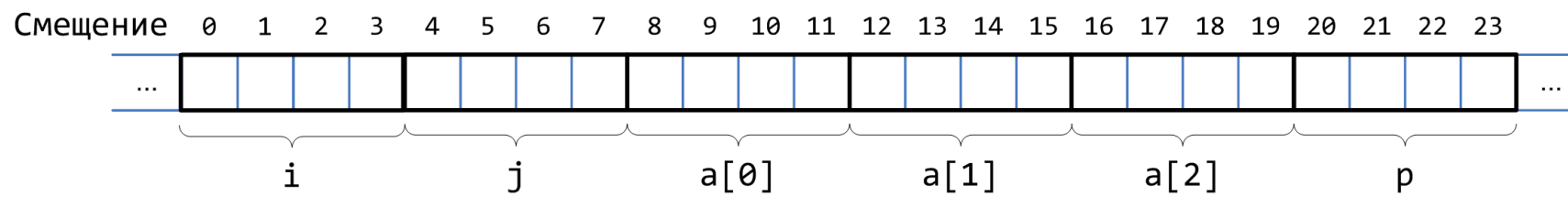


Лекция 6

26 февраля



```
struct rec {  
    int i;  
    int i;  
    int a[3];  
    int *p;  
}  
...  
struct rec *x;  
x->j = x->i;
```

```
mov    edx, dword [x]      ; (1)  
mov    eax, dword [edx]   ; (2)  
mov    dword [edx + 4], eax ; (3)
```

```
struct rec {  
    int i;  
    int i;  
    int a[3];  
    int *p;  
};
```

```
struct rec *x;  
int i;
```

```
&(r->a[i]);
```

```
mov    edx, dword [i]          ; (1)  
mov    eax, dword [x]          ; (2)  
lea    eax, [eax + 4 * edx + 8] ; (3)
```

```
struct rec {  
    int i;  
    int i;  
    int a[3];  
    int *p;  
};
```

```
struct rec *r;
```

```
r->p = &r->a[r->i + r->j];
```

```
mov    edx, dword [r]           ; (1)  
mov    eax, dword [edx + 4]     ; (2)  
add    eax, dword [edx]        ; (3)  
lea    eax, [edx + 4 * eax + 8] ; (4)  
mov    dword [edx + 20], eax    ; (5)
```

```
struct NODE_S {  
    struct NODE_S *left;  
    struct NODE_S *right;  
    double data;  
};
```

```
union NODE_U {  
    struct {  
        union NODE_U *left;  
        union NODE_S *right;  
    } internal;  
    double data;  
};
```

```
typedef enum {N_LEAF, N_INTERNAL} nodetype_t;
```

```
struct NODE_T {  
    nodetype_t type;  
    union NODE_U {  
        struct {  
            struct NODE_T *left;  
            struct NODE_T *right;  
        } internal;  
        double data;  
    } info;  
};
```

```
unsigned float2bit(float f) {  
    union {  
        float f;  
        unsigned u;  
    } temp;  
    temp.f = f;  
    return temp.u;  
}
```

```
unsigned copy(unsigned u) {  
    return u;  
}
```

```
global float2bit  
float2bit:  
    push    ebp  
    mov     ebp, esp  
    mov     eax, dword [ebp + 8]  
    mov     esp, ebp  
    pop     ebp  
    ret
```


Смещение 0 1 2 3 4 5 6 7 8 9



i c j

(1)

Смещение 0 1 2 3 4 5 6 7 8 9 10 11 12



i c j

(2)

Смещение 0 1 2 3 4 5 6 7 8 9 10 11 12



i j c

(3)