



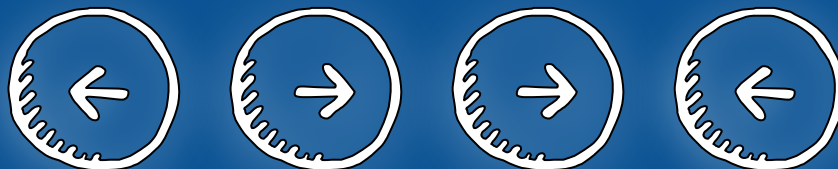
Алгоритмы и Алгоритмические Языки

Семинар #6:

1. Память программы и указатели.
2. Передача возвращаемых значений из функций.
3. Одномерные массивы.
4. Арифметика указателей и операция sizeof.
5. Контекст №0: тестовый контекст.



Память программы и указатели



Адресация памяти компьютера

Память компьютера – адресуемый набор ячеек.

Размер одной ячейки – байт (часто – 8 бит).

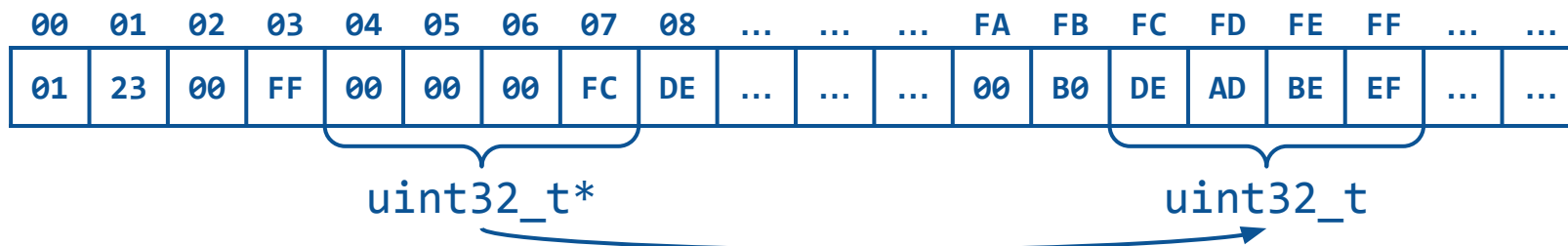
00	01	02	03	04	05	06	07	08	FA	FB	FC	FD	FE	FF
01	23	00	FF	DE	AD	BE	EF	DE	00	B0	1D	FA	CE	CA

В программе память выделена под:

- Исполняемый код вашей программы.
- Исполняемый код стандартной библиотеки.
- Локальные переменные.
- Глобальные переменные.
- Константные строки.

Указатели в языке Си

Тип данных, хранящий адрес ячейки памяти – **указатель**.
Обозначается добавлением звёздочки к типу.



```
uint32_t val = 0xDEADBEEF;  
uint32_t* ptr = &val;
```

Взятие адреса переменной

```
*ptr = 0xB01DFACE;  
*ptr += 0xBA1BEC;
```

Разыменование указателя

Операции с указателями: задачи

В программе объявлены следующие переменные:

```
int x; int* p; int** q;
```

Определить, какие из выражений являются корректными:

1) *p

6) x = p

11) *q

2) *x

7) p = x

12) **q

3) p = &x

8) q = &x

13) ***q

4) &p

9) q = &p

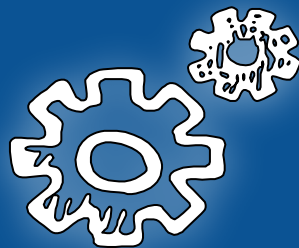
14) &*p

5) &(&x)

10) **p



Передача возвращаемых значений из функций



Применение указателей в функциях

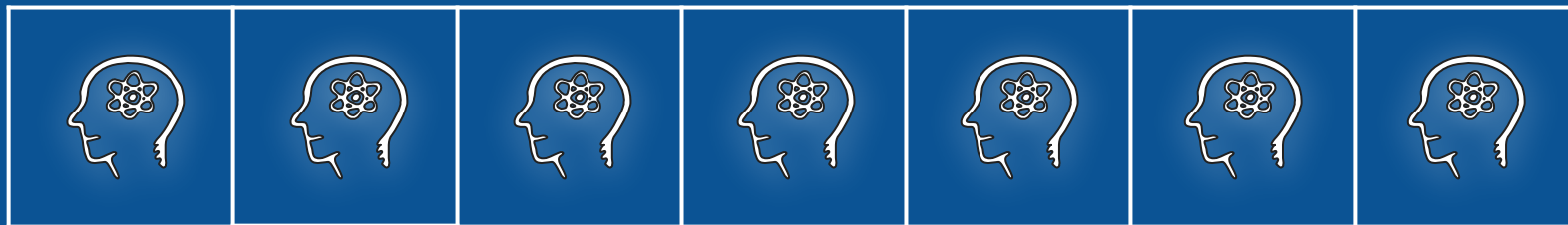


См. пример [06_square_equation](#):

- Ключевое слово **const** в указателях.
- Обращение по указателю **NULL**.
- Проверка указателей с помощью **assert**.



Одномерные массивы



Массивы в языке Си



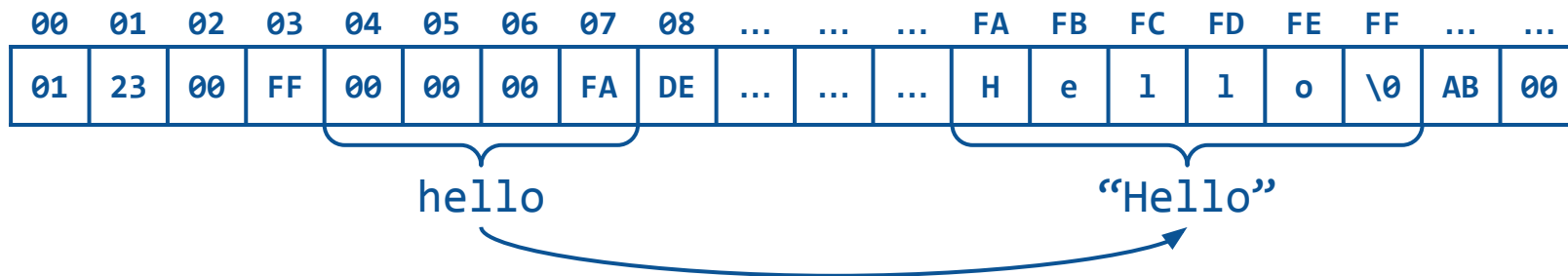
Декларация массива

```
char hello[6] = "Hello";
```

Доступ к элементам

```
hello[0] = 'Y';  
hello[1] += 10;
```

Нижележащее представление в памяти программы:



См. пример [06_sum](#).

Способы инициализации массивов



Способы инициализации массивов

```
// Uninitialized:  
int a[5];
```

```
// Zero-initialized:  
int d[5] = {0};
```

```
// List-initialized:  
int b[5] = {1, 2, 3, 4, 5};
```

```
// Element-wise initialized:  
int c[] = {  
    [0] = 0,  
    [4] = 4  
};
```

```
// 2D-array:  
int matrix[4][4] = {  
    {0, 0, 0, 0},  
    {0, 1, 1, 1},  
    {0, 1, 2, 2},  
    {0, 1, 2, 3}  
};
```

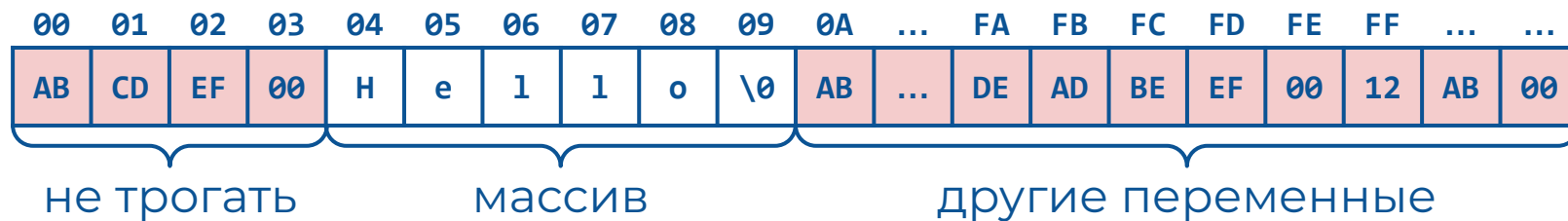
Передача массива в функцию

При передаче массива в функцию необходимо передавать длину:

```
long sum(const int* array, unsigned length) {  
    long sum = 0;  
  
    for (int i = 0; i < length; i++) {  
        sum += array[i];  
    }  
  
    return sum;  
}
```

Выход за границы массива

Доступ за границы массива – **Undefined Behavior (зло)**.



Способ поиска выхода за границы массива:

```
for (int i = start_i; continue_cycle(i); i = step(i))  
{  
    assert(0 <= i && i < array_size);  
    ...  
}
```



Арифметика указателей и операция sizeof



Операция `sizeof`

Операция **`sizeof`** возвращает кол-во байт, которое тип или результат выражения будет занимать в памяти:

```
printf("sizeof(char)      = %lu\n", sizeof(char));    --> 1
printf("sizeof(int)       = %lu\n", sizeof(int));     --> 4
printf("sizeof(long)      = %lu\n", sizeof(long));    --> 8
printf("sizeof(float)     = %lu\n", sizeof(float));   --> 4
printf("sizeof(double)    = %lu\n", sizeof(double));  --> 8

printf("sizeof(int*)      = %lu\n", sizeof(int*));    --> 8
printf("sizeof(void*)     = %lu\n", sizeof(void*));   --> 8

int array[5];
printf("sizeof(array)     = %lu\n", sizeof(array));   --> 20
```



Арифметика указателей

Основное правило арифметики указателей:

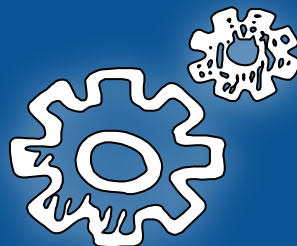
- При увеличении указателя типа `type*` на 1 происходит увеличение адреса на `sizeof(type)`.

Следствия:

- Конструкции `*(array + 4)` и `array[4]` эквивалентны.
- При вычитании адресов элементов одного массива получается разность индексов.
- Конструкции типа `ptr++` могут быть опасными!



Контекст №0: тестовый контекст



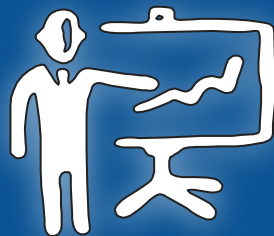


Алгоритм прохождения контеста

Студент:

1. Получает **логин и пароль** для входа в систему.
2. Заходит на **<https://contest.solutions/hse/>**.
3. Выбирает “Разминочное задание”.
4. **Меняет пароль** на собственный (поле “Настройки”).
5. Отправляет задачу “**Problem 00-1: Hello world!**”.
6. Убеждается у преподавателя, что задача засчитана.
7. Свободен.

Вопросы?



Красивые иконки взяты с сайта handdrawngoods.com