

17. Указатели, void*, указатели на функции (на примере функции qsort).

План ответа:

- 1 понятие «указатель»;
- 2 void*, особенности операций с ним;
- 3 приведение указателей разных типов к void* и обратно;
- 4 определение указателя на функцию;
- 5 присваивание значения указателю на функцию;
- 6 вызов функции по указателю;
- 7 использование указателей на функции.

1 Указатель – это объект, содержащий адрес объекта или функции, либо выражение, обозначающее адрес объекта или функции.

2 void* -- бестиповой указатель, указывает на произвольное место в памяти.
Особенности

Указателя типа void* нельзя разыменовывать.

К указателям типа void* не применима адресная арифметика.

Доп: Основная миссия – упрощение преобразований указателей, универсализация структур и формальных параметров функций.

Тип указателя void* используется, если тип объекта неизвестен.

позволяет передавать в функцию указатель на объект любого типа;

полезен для ссылки на произвольный участок памяти, независимо от размещенных там объектов.

3 В языке C допускается присваивание указателя типа void* указателю любого другого типа (и наоборот) без явного преобразования типа указателя.

```
double d = 5.0;
double *pd = &d;
void *pv = pd;
pd = pv;
```

4 Объявление указателя на функцию

```
double trapezium(double a, double b, int n, double (*func)(double));
```

5 присваивание значения указателю на функцию

```
result = integrate(0, 3.14, 25, sin);
```

6 Вызов функции по указателю

```
y = (*func)(x); // y = func(x);
```

7 Использование указателей

Например, функция void qsort(void* base, size_t memb, size_t size, int (*compar)(const void*, const void*))

из заголовочного файла <stdlib.h> имеет в качестве формального параметра указатель на функцию, сравнивающие некие 2 значения, задаваемые ссылками на них, без ограничения на типы параметров.