

#### 14. Указатели, базовые операции.

План ответа:

- ???
1. организация памяти с точки зрения программиста;
  2. понятие «указатель»;
  3. разновидности указателей в языке Си;
  4. использование указателей;
  5. определение переменной-указателя;
  6. базовые операции над указателями (“&” и “\*”);
  7. модификатор const и указатель.

1. организация памяти с точки зрения программиста;

2. понятие «указатель»;

Указатель – это объект, содержащий адрес объекта или функции, либо выражение, обозначающее адрес объекта или функции. [Жешке Р. «Толковый словарь стандарта языка Си»]

```
int a = 2, *p = &a;      // p – указатель-объект, &a – указатель-выражение
```

3. разновидности указателей в языке Си;

типизированный указатель на данные (тип\* имя);

бестиповой указатель (void\* имя);

указатель на функцию.

4. использование указателей;

Передача параметров в функцию

Изменяемые параметры

«Объемные» параметры

Обработка областей памяти

Динамическое выделение памяти

Ссылочные структуры данных

5. определение переменной-указателя;

Адресом переменной того же типа

```
int a = 1;
```

```
double d = 5.0;
```

```
int *p = &a;
```

```
int *p_err = &d; // error: initialization from incompatible pointer type
```

Значением другого указателя того же типа

```
int *q = p;
```

```
double *q_err = p; // error: initialization from incompatible pointer type
```

Значением NULL

```
int *r = NULL;
```

```
int *r_ok = 0;
```

```
int *r_err = 12345; // error: initialization makes pointer from integer without
```

6. базовые операции над указателями (“&” и “\*”);

Доступ к данным: not\_ptr\_name = \*ptr1\_name;

Получение адреса: ptr2\_name = &not\_ptr\_name;

При определении переменной-указателя перед ее именем должен размещаться символ ‘\*’.

Для определения адреса данных используется операция ‘&’ (операция взятия адреса).

Для доступа к данным, на которые указывает указатель, используется операция ‘\*’ (операция разыменования)

Операции ‘&’ и ‘\*’ взаимно обратные.

```
int d = 5, *p = &d;
```

```
*&d == d
```

```
&*p == p
```

7. модификатор const и указатель.

Указатель на константу

```
int a = 5;
```

```
int b = 7;
const int *p = &a;      // <-

printf("%d\n", *p);
*p = 4; // error: assignment of read-only location '*p'
p = &b;
```

Константный указатель

```
int a = 5;
int b = 7;
int * const p = &a;      // <-

printf("%d\n", *p);
*p = 4;
p = &b; // error: assignment of read-only variable 'p'
```

Константный указатель на константу

```
int a = 5;
int b = 7;
const int *const p = &a;      // <-

printf("%d\n", *p);
*p = 4; // error: assignment of read-only location '*p'
p = &b; // error: assignment of read-only variable 'p'
```