

18. Указатели и многомерные статические массивы.

План ответа:

- 1 концепция многомерного массива как «массива массивов»;
- 2 определение многомерных массивов;
- 3 инициализация многомерных массивов;
- 4 «слои», составляющие многомерные массивы;
- 5 обработка многомерных массивов с помощью указателей;
- 6 передача многомерных массивов в функцию;
- 7 const и многомерные массивы.

1-2 ???

Отличие многомерного массива от одномерного состоит в том, что в одномерном массиве положение элемента определяется одним индексом, а в многомерном — несколькими. Примером многомерного массива является матрица.

Общая форма объявления многомерного массива
тип имя[размерность1][размерность2]...[размерностьm];

Количество размерностей массива практически не ограничено.

```
int a[3][2];
```

Компилятор Си располагает строки матрицы а в памяти одну за другой в плотную друг к другу.

3 Инициализация многомерных массивов

```
int a[3][3] = {  
    {1, 2, 3},  
    {4, 5}  
};
```

```
int d[][2] = { {1, 2} };
```

```
int e[][] = {  
    {1, 2},  
    {4, 5}
```

```
}; // error: array type has incomplete element type
```

```
int b[3][3] = {1, 2, 3, 4, 5, 6, 7, 8, 9};
```

```
// warning: missing braces around initializer [-Wmissing-braces]
```

```
// c99
```

```
int c[2][2] = {[0][0] = 1, [1][1] = 1};
```

4 Компоненты многомерного массива int a[2][3][5];

а — массив из двух элементов типа “int [3][5]”
int (*p)[3][5] = a;

a[i] — массив из трех элементов типа “int [5]” (i из [0, 1])
int (*q)[5] = a[i];

a[i][j] — массив из пяти элементов типа “int” (i из [0, 1], j из [0, 1, 2])
int *r = a[i][j];

a[i][j][k] — элемент типа “int” (i из [0, 1], j из [0, 1, 2], k из [0, 1, 2, 3, 4])
int s = a[i][j][k];

5 обработка многомерных массивов с помощью указателей

как одномерный.

```
int a[N][M];
```

```
int *p;
```

```
for (p = &a[0][0]; p <= &a[N-1][M-1]; p++)  
    *p = 0;
```

Обработка строки матрицы (обнуление i-ой строки)

```
// указатель на начало i-ой строки
```

```
int *p = &a[i][0];
```

```
// обнуление i-ой строки
```

```
for (p = a[i]; p < a[i] + M; p++)  
    *p = 0;
```

a[i] можно передать любой функции, обрабатывающей одномерный массив

Обработка столбца матрицы (обнуление j-го столбца)

// указатель на строку (строка — это массив из M элементов)

```
int (*q)[M]; //!!!Без скобок получится массив из M указателей.
```

Выражение q++ смещает указатель на следующую строку

Выражение (*q)[j] возвращает значение в j-ом столбце строки, на которую указывает q.

```
for (q = a; q < a + N; q++)
```

```
    (*q)[j] = 0;
```

6 Передача многомерных массивов в функцию

Пусть определена матрица

```
int a[N][M];
```

Для ее обработки могут быть использованы функции со следующими прототипами:

```
void f(int a[N][M], int n, int m);
```

```
void f(int a[][M], int n, int m);
```

```
void f(int (*a)[M], int n, int m);
```

Неверные прототипы:

```
void f(int a[][], int n, int m);
```

// не указана вторая размерность => компилятор не сможет

// выполнить обращение по индексу

```
void f(int *a[M], int n, int m);
```

```
void f(int **a, int n, int m);
```

// неверный тип — массив указателей

7 const и многомерные массивы.

Согласно C99 T (*p)[N] не преобразуется неявно в T const (*p)[N].

Способы борьбы

не использовать const;

использовать явное преобразование типа