

Лекция 15. Разное

Работа с многомерными массивами

Многомерные массивы в языке C.

```
double a[10][20], b[100][30][50]; // a[i][j] b[i][j][k]
```

Многомерные массивы размещаются в памяти "по строкам". Т.е. первый элемент соответствует индексам $0, \dots, 0$, потом последний индекс пробегает все свои значения, далее меняется следующий с конца индекс и т.д.

для размерностей n_1, n_2, n_3 и индексов i_1, i_2, i_3 линейная позиция s элементов $a[i_1][i_2]$ и $b[i_1][i_2][i_3]$ вычисляется как

двумерный: $s = i_1 * n_2 + i_2$

трехмерный $s = i_1 * n_2 * n_3 + i_2 * n_3 + i_3$

и т.п.

проблема с передачей матриц в функции

```
void f(double x[10][20]);
```

Пример с передачей двумерного массива фиксированной размерности.

Массивы и указатели. Есть операция индексирования указателя.

```
double *r; r[i]
double **p, ***q; // p[i][j] q[i][j][k] но надо выделять память !!!
```

для матрицы:

p — адрес начала массива указателей на строки

$p[i]$ — адрес начала массива элементов строки

$p[i][j]$ — конкретный элемент строки

Т.е. для матрицы размерности $n_1 * n_2$ получаем массив указателей на строки длины n_1 и n_1 массивов строк длины n_2 каждый. Можно все строки хранить подряд в массиве длины $n_1 * n_2$. Итого: n_1 памяти под указатели и $n_1 * n_2$ под элементы.

Нужно выделять память под все эти массивы и потом аккуратно ее освободить.

Пример матрицы с двойным указателем.

Используется пересчет индексов в линейную позицию по вышеприведенным формулам. Для удобства используется инструкция `define`.

Пример матрицы на одномерном массиве.

В любом варианте ошибка по размерности, как правило, фатальна.

Метод с одномерным массивом предпочтительнее с точки зрения быстродействия и памяти. Метод с массивом указателей более универсален (например, строки разной длины).

Функции с переменным числом аргументов

Тип `va_list` в качестве параметра.

| | |
|------------------------------------|-----------------------------------|
| <code>va_start(param, list)</code> | - инициализация списка параметров |
| <code>va_arg(list, type)</code> | - получение очередного параметра |
| <code>va_end(list)</code> | - разгрузка списка параметров |

Пример `arg.c` функций с переменным числом параметров