Вывод последних K строк входного файла на C++

* 9 апреля 2015 в 12:59, [Задачки](https://tproger.ru/category/problems/)
* 2 минуты
* 2 865

Можно действовать прямо — подсчитать количество строк (N) и вывести строки с N-K до N. Для этого понадобится дважды прочитать файл, что очень неэффективно. Давайте найдем решение, которое потребует прочитать файл только один раз и выведет последние Kстрок.

Можно создать массив для K строк и прочитать последние K строк. В нашем массиве там будут храниться строки от 1 до K, затем от 2 до K+1, затем от 3 до K+2 и т.д. Каждый раз, считывая новую строку, мы будем удалять самую старую строку из массива.

Вы можете удивиться: разве может быть эффективным решение, требующее постоянного сдвига элементов в массиве? Это решение станет эффективным, если мы правильно реализуем сдвиг. Вместо того чтобы каждый раз выполнять сдвиг массива, можно «закольцевать» массив.

Используя такой массив, читая новую строку, мы всегда будем заменять самый старый элемент. Самый старый элемент будет храниться в отдельной переменной, которая будет меняться при добавлении новых элементов.

Пример использования закольцованного массива:

|  |  |
| --- | --- |
| 1  2  3  4 | шаг 1 (исходное состояние):    массив = {a, b, с, d, е, f}. р = 0  шаг 2 (вставка g):            массив = {g, b, с, d, е, f}. р = 1  шаг 3 (вставка h):            массив = {g, h, с, d, е, f}. р = 2  шаг 4 (вставка i):            массив = {g, h, i, d, e, f}. p = 3 |

Приведенный далее код реализует этот алгоритм:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | void printLast10Lines(char\* fileName) {        const int K = 10;      ifstream file (fileName);      string L[K];      int size = 0;        /\* читаем файл построчно в круговой массив \*/      while (file.good()) {          getline(file, L[size % K]);          size++;      }        /\* вычисляем начало кругового массива и его размер \*/      int start = size > K ? (size % K) : 0;      int count = min(K, size);        /\* выводим элементы в порядке чтения \*/      for (int i = 0; i < count; i++) {          cout << L[(start + i) % K] << endl;      }  } |

Мы считываем весь файл, но в памяти хранится только 10 строк.