

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии
Дисциплина: «Архитектура вычислительных систем»

Задача об инвентаризации по рядам

Пояснительная записка

Выполнил:
Попов Андрей Эдуардович,
студент гр. БПИ197.

Москва
2020

1. Текст задания

Задача об инвентаризации по рядам. После нового года в библиотеке университета обнаружилась пропажа каталога. После поиска и наказания виноватых, ректор дал указание восстановить каталог силами студентов. Фонд библиотека представляет собой прямоугольное помещение, в котором находится M рядов по N шкафов по K книг в каждом шкафу. Требуется создать многопоточное приложение, составляющее каталог. При решении задачи использовать метод «портфель задач», причем в качестве отдельной задачи задается составление каталога одним студентом для одного ряда.

2. Применяемые расчетные методы

2.1. Теория решения задания

Для решения поставленной задачи использовался метод «Портфель задач»:

Каждый поток обрабатывает ряд, к которому ещё ни один другой поток не приступал. Как только поток обработает ряд, он переходит к следующему, не обработанному ряду. Достигается такая обработка с помощью OpenMP. В частности используется разбиение выполнения цикла на несколько потоков.

Названия книг создаются случайным образом из букв английского алфавита и пробелов. Был выбран вариант реализации подразумевающий, что потоки сортируют обрабатываемые ряды и каталог содержит информацию об отсортированных рядах.

2.2. Организация входных данных

Программа запрашивает параметры M, N, K из условия задачи, а также число студентов. $1 \leq M, N, K \leq 100$.

2.3 Организация выходных данных

Программа сообщает содержимое каталога по рядам. А также сообщается том, что какой-то студент нашёл какую-то книгу на какой-то позиции. (пример: рисунок 1)

```
student 0 has found |Kuco| at the position [0][0][0]
student 1 has found |Pgmy divxbfd| at the position [1][0][0]
student 0 has found |Be vzmn | at the position [0][0][1]
student 1 has found |Xnzzjeg r| at the position [1][0][1]
student 0 has found |Joinsmeaegy| at the position [0][1][0]
student 1 has found |Y c cek | at the position [1][1][0]
student 0 has found |Qxu rcer| at the position [0][1][1]
student 1 has found |Mikvrz f gs| at the position [1][1][1]
CATALOG:
row 0:
|Be vzmn | at the position [0][0][1]
|Joinsmeaegy| at the position [0][1][0]
|Kuco| at the position [0][0][0]
|Qxu rcer| at the position [0][1][1]
row 1:
|Mikvrz f gs| at the position [1][1][1]
|Pgmy divxbfd| at the position [1][0][0]
|Xnzzjeg r| at the position [1][0][1]
|Y c cek | at the position [1][1][0]
```

Рисунок 1.

2.4. Дополнительный функционал программы

Программа обрабатывает ситуацию неверного ввода, если были посланы не корректные числа или числа выходили за допустимый диапазон.

3. Тестирование программы

3.1. Корректные значения

Рисунок 2. M=1, N=1, K=1,. Каталог созданся корректно

```
Enter the number of rows(M)(Not less than 1, not more then 100): 1
Enter the number of bookcases(N)(Not less than 1, not more than 100): 1
Enter the number of books(K)(Not less than 1, not more than 100): 1
student 0 started working with row 0
student 0 finished working with row 0
CATALOG:
row 0:
|Ficgggiej x| at the position [0][0][0]:
```

Рисунок 3 M=100, N=20, K=5. Сразу после ввода данных

```
Enter the number of rows(M)(Not less than 1, not more then 100): 100
Enter the number of bookcases(N)(Not less than 1, not more than 100): 20
Enter the number of books(K)(Not less than 1, not more than 100): 5
student 0 has found |As u| at the position [0][0][0]
student 2 has found |Nek lb aaeio| at the position [50][0][0]
student 1 has found |Oflx c bbf| at the position [25][0][0]
student 1 has found |Vmtc us df | at the position [25][0][1]
student 0 has found |Aedbojggxba| at the position [0][0][1]
student 2 has found |Sd trnn| at the position [50][0][1]
student 3 has found |Mdj ka zz| at the position [75][0][0]
student 1 has found |Mx b b| at the position [25][0][2]
student 0 has found |Gdhp rbyy | at the position [0][0][2]
student 2 has found |E pqul at the position [50][0][2]
student 3 has found |Dtkrc | at the position [75][0][1]
student 1 has found |Oh lhjuqy| at the position [25][0][3]
student 0 has found |Soevswm| at the position [0][0][3]
student 2 has found |Wwng| at the position [50][0][3]
student 3 has found |Wogbd op| at the position [75][0][2]
student 1 has found |Xe f zn| at the position [25][0][4]
student 0 has found |J plcc m rc o| at the position [0][0][4]
student 2 has found |Ckgitpx| at the position [50][0][4]
student 3 has found |Jv ns| at the position [75][0][3]
student 1 has found |Dyntu fd | at the position [25][1][0]
student 0 has found |Cwe oera | at the position [0][1][0]
student 2 has found |Wd e y| at the position [50][1][0]
student 3 has found |Hck x kv| at the position [75][0][4]
student 1 has found |Oouhzd| at the position [25][1][1]
student 0 has found |Uhj z rhwjbm| at the position [0][1][1]
student 2 has found |N v st| at the position [50][1][1]
student 3 has found |Ugd | at the position [75][1][0]
student 1 has found |Vpotkuv| at the position [25][1][2]
student 0 has found |Rycx_gk| at the position [0][1][2]
student 2 has found |Zac | at the position [50][1][2]
student 2 has found |Nntgycopkh t| at the position [50][1][3]
student 1 has found |Wlb j ummieim| at the position [25][1][3]
student 0 has found |lvlcvm| at the position [0][1][3]
student 3 has found |C lu r| at the position [75][1][1]
student 2 has found |Mvka i| at the position [50][1][4]
student 1 has found |Jmymypjk nfg| at the position [25][1][4]
student 0 has found |Cwhcek| at the position [0][1][4]
student 3 has found |Mydbckgs| at the position [75][1][2]
student 2 has found |Dtvldhnl| at the position [50][2][0]
student 1 has found |Gth xo oiua| at the position [25][2][0]
student 0 has found |Wnzocw| at the position [0][2][0]
student 3 has found |Mj nojg st j| at the position [75][1][3]
student 2 has found |Ilvxloi| mep| at the position [50][2][1]
student 1 has found |Zuxs oqrg n| at the position [25][2][1]
student 0 has found |Myomifzlkz c| at the position [0][2][1]
student 3 has found |K lcsukg uthku| at the position [75][1][4]
student 2 has found |Bsg wnaqjedmx| at the position [50][2][2]
student 1 has found |Ye icwbxc| at the position [25][2][2]
student 0 has found |Peuyza p| at the position [0][2][2]
student 3 has found |Qidgelgdoal at the position [75][2][0]
```

Рисунок 4. Вывод ответа к тесту (рисунок 3). Лишь часть полного ответа. Каталог созданся корректно

```
|Vwav h| at the position [98][11][1]
|Vwvpzjf kkc| at the position [98][15][3]
|Vy k flxbxher| at the position [98][5][3]
|W opzb| at the position [98][13][0]
|W dsn x keh| at the position [98][17][0]
|Wbdiipbwse| at the position [98][3][1]
|Wp vad| at the position [98][7][0]
|Wphdmpao nk z | at the position [98][14][3]
|X m | at the position [98][13][1]
|Xc wtsj | at the position [98][19][0]
|Xpmxsurxpc| at the position [98][6][3]
|Xrilcmdnv| at the position [98][14][1]
|Xrpzuqknzhcz | at the position [98][16][2]
|Yda stvp| at the position [98][0][1]
|Yffs qbn| at the position [98][1][2]
|Ymkfluo| at the position [98][2][3]
row 99:
|Ae brh| at the position [99][9][2]
|Ajk e fi rvg| at the position [99][19][2]
|Amughwn ygd| at the position [99][8][1]
|Ar j uiu| at the position [99][16][2]
|Be wj| at the position [99][5][4]
|Bzu wgfuu| at the position [99][11][2]
|C p iyw| at the position [99][1][4]
|C suzok otp| at the position [99][15][2]
|Cdglvjqurev| at the position [99][5][3]
|Cqhbhiwzexcq| at the position [99][11][0]
|Dblss ja | at the position [99][13][1]
|Dlyhgd| at the position [99][9][1]
|Dqtq| at the position [99][4][1]
|Ecza clsk | at the position [99][10][1]
|Egprvbbt zs l| at the position [99][13][3]
```

3.2. Некорректные значения

```

Enter the number of rows(M)(Not less than 1, not more then 100): asd
Wrong number. Try again.
Enter the number of rows(M)(Not less than 1, not more then 100): -1
Wrong number. Try again.
Enter the number of rows(M)(Not less than 1, not more then 100):
asd
Wrong number. Try again.
Enter the number of rows(M)(Not less than 1, not more then 100): 101
Wrong number. Try again.
Enter the number of rows(M)(Not less than 1, not more then 100): 0
Wrong number. Try again.
Enter the number of rows(M)(Not less than 1, not more then 100): 2
Enter the number of bookcases(N)(Not less than 1, not more than 100): 101
Wrong number. Try again.
Enter the number of bookcases(N)(Not less than 1, not more than 100): 2
Enter the number of books(K)(Not less than 1, not more than 100): 1
Enter the number of students(not less that 1 not more then M): -1
Wrong number. Try again.
Enter the number of students(not less that 1 not more then M): 20
Wrong number. Try again.
Enter the number of students(not less that 1 not more then M): 3
Wrong number. Try again.
Enter the number of students(not less that 1 not more then M): 2
CATALOG:
row 0:
  bookcase 0:
    [0][0][0]: Ctzsozyugx
  bookcase 1:
    [0][1][0]: Czbgt tk
row 1:
  bookcase 0:
    [1][0][0]: Uqql xdo
  bookcase 1:
    [1][1][0]: Amvss

```

Рисунок 5. Множество попыток вводить строку или переносы строки, а также попытки выйти за допустимый диапазон значений. Программа предлагает юзеру заново ввести данные.

ПРИЛОЖЕНИЕ 1**Список литературы**

1. Заголовок. [Электронный ресурс] // URL: <https://habr.com/ru/post/259153/>
2. Заголовок. [Электронный ресурс] // URL:
<http://www.softcraft.ru/edu/comparch/>

Код программы

```
#include <iostream>
#include <string>
#include <vector>
#include <ctime>
#include <future>
#include <omp.h>

//Попов Андрей
//БПИ197
//Вариант 16
//16. Задача об инвентаризации по рядам.После нового года в
//библиотеке университета обнаружилась пропажа каталога.После поиска и
//наказания виноватых, ректор дал указание восстановить каталог силами
//студентов.Фонд библиотека представляет собой прямоугольное помещение,
//в котором находится М рядов по N шкафов по К книг в каждом шкафу.
//Требуется создать многопоточное приложение, составляющее каталог.При
//решении задачи использовать метод «портфель задач», причем в качестве
//отдельной задачи задается составление каталога одним студентом для одного
//ряда.

/// <summary>
/// ввод целого числа, вводимое пользователем
/// </summary>
/// <param name="message">Сообщение, котороре будет показано пользователю</param>
/// <param name="min_value">Минимальное значение вводимого числа</param>
/// <param name="max_value">Максимальное значение вводимого числа</param>
/// <returns>возвращает вводимое число</returns>
int get_int(std::string message, int min_value = INT32_MIN, int max_value = INT32_MAX) {
    while (true) {
        int integer;
        std::cout << message;
        std::cin >> integer;
        if (std::cin.fail() || min_value > integer || integer > max_value) {
            std::cin.clear();
            std::cin.ignore(INT32_MAX, '\n');
            std::cout << "Wrong number. Try again.\n";
        }
        else {
            std::cin.ignore(INT32_MAX, '\n');
            return integer;
        }
    }
}
```



```

    }

}

/// <summary>
/// Возвращает случайное целое число включая обе границы
/// </summary>
/// <param name="min_value">левая граница</param>
/// <param name="max_value">правая граница</param>
/// <returns>Возвращает случайное целое число включая обе границы</returns>
int next_int(int min_value, int max_value, int my_seed = 0) {
    return min_value + ((rand()+my_seed) % (max_value - min_value + 1));
}

/// <summary>
/// создаёт случайное название книги
/// </summary>
/// <param name="min_length">минимальная длина</param>
/// <param name="max_length">максимальная длина</param>
/// <returns>случайное название книги</returns>
std::string get_random_title(int min_length = 3, int max_length = 30, int my_seed = 0) {
    std::string str;
    str += (char)next_int('A', 'Z', my_seed);
    for (int i = 0; i < next_int(min_length, max_length, my_seed) - 1; i++) {
        if (next_int(1, 1000, my_seed) < 175) //обусловлено приблизительной частотой
        пробела в текстах на английском языке
            str += " ";
        else
            str += (char)next_int('a', 'z', my_seed);
    }
    return str;
}

/// <summary>
/// Создаёт ряд в каталоге
/// </summary>
/// <param name="library">каталог</param>
/// <param name="library_row">конкретный ряд этого каталога</param>
/// <param name="M"></param>
/// <param name="N"></param>
/// <param name="K"></param>
void make_row(std::string** library_row, int* catalog_row, int N, int K, int student_number, int
row_number) {
    std::vector<std::string> prototype_library_row;

```

```

std::vector<int> prototype_catalog_row;
for (int j = 0; j < N; j++) {
    for (int k = 0; k < K; k++) {
#pragma omp critical (searching)
        {
            library_row[j][k] = get_random_title(3, 30, row_number);
            std::cout << "student " << student_number << " has found "
                << '|' << library_row[j][k] << " | at the position " << '[' <<
row_number << "]" << j << "]" << k << "]" << "\n";
        }
        prototype_library_row.push_back(library_row[j][k]);
        prototype_catalog_row.push_back(j * K + k);
    }
}
std::sort(prototype_catalog_row.begin(), prototype_catalog_row.end(),
    [prototype_library_row](int a, int b) {return prototype_library_row[a] <
prototype_library_row[b]; });
for (int j = 0; j < K * N; j++)
    catalog_row[j] = prototype_catalog_row[j];
}

/// <summary>
/// Составляет каталог в нескольких потоках
/// </summary>
/// <param name="library">библиотека</param>
/// <param name="catalog">каталог</param>
/// <param name="M"></param>
/// <param name="N"></param>
/// <param name="K"></param>
/// <returns></returns>
void student_thread(std::string*** library, int** catalog, int M, int N, int K) {
    srand(time(NULL));
#pragma omp parallel for
    for (int i = 0; i < M; i++) { //реализация "портфеля зада" - поиск рядов, к которым ещё
никто не приступал и заполнение этих рядов.
        int student_number = omp_get_thread_num();
        //std::cout << "student " << student_number << " started working with row " << i <<
'\n'; //для проверок, что потоки работают как надо
        make_row(library[i], catalog[i], N, K, omp_get_thread_num(), i);
        //std::cout << "student " << student_number << " finished working with row " << i <<
'\n';

    }
}

```

```

/// <summary>
/// Выписывает все книги каталога
/// </summary>
/// <param name="library">библиотека</param>
/// <param name="catalog">каталог</param>
/// <param name="M"></param>
/// <param name="N"></param>
/// <param name="K"></param>
void write_answer(std::string*** library, int** catalog, int M, int N, int K) {
    std::cout << "CATALOG:\n";
    for (int i = 0; i < M; i++) {
        std::cout << "row " << i << ":\n";
        for (int j = 0; j < N * K; j++) {
            int n = catalog[i][j] / K;
            int k = catalog[i][j] % K;
            std::cout << '|' << library[i][n][k] << " | at the position " << '[' << i << "]" << n
<< "]" << k << "]" << "\n";
        }
    }
}

int main() {
    int M = get_int("Enter the number of rows(M)(Not less than 1, not more then 100): ", 1,
100); // ввод входных данных
    int N = get_int("Enter the number of bookcases(N)(Not less than 1, not more than 100): ", 1,
100);
    int K = get_int("Enter the number of books(K)(Not less than 1, not more than 100): ", 1,
100);
    int** catalog = new int* [M];
    std::string*** library = new std::string * * [M]; //генерация библиотеки
    for (int i = 0; i < M; i++) {
        catalog[i] = new int[N * K];
        library[i] = new std::string * [N];
        for (int j = 0; j < N; j++)
            library[i][j] = new std::string[K];
    }
    student_thread(library, catalog, M, N, K);
    write_answer(library, catalog, M, N, K); //вывод ответа на задачу
}

```