Индивидуальное домашнее задание-4 (Гринченко Евгений, БПИ 236, вариант 34)

Небольшое предисловие перед началом - код программы я реализовывал сразу на оценку 8, поэтому не удалось соблюдать условия итеративности. Постараюсь отразить в этом отчёте выполняемость критериев на оценку 4-7.

Отчет о многопоточной программе "Сельская библиотека"

1. Постановка задачи

Концептуальная модель

Программа моделирует работу сельской библиотеки с ограниченным фондом книг, где множество читателей взаимодействуют с книжными ресурсами в условиях конкурентного доступа.

Ключевые сущности системы

1. `Library`
2. `Reader`
3. `Logger`

Основные требования к системе

- Управление книжным фондом
- Параллельный доступ читателей к книгам
- Корректная синхронизация потоков
- Протоколирование событий

2. Архитектура программы

Структура классов

Класс Library

```
class Library {
public:
```

```
  Library(int numBooks, Logger* loggerPtr);
  void takeBooks(int readerId,
                 const std::vector<int>& desiredBooks,
                 std::vector<int>& takenBooks,
                 std::vector<int>& missingBooks);
  void returnBooks(int readerId, const std::vector<int>& books);

private:
  int numBooks;
  std::vector<bool> bookAvailable;  // Состояние книг
  std::vector<pthread_mutex_t> bookMutexes;  // Мьютексы книг
  std::vector<pthread_cond_t> bookConds;  // Условные переменные
  Logger* logger;
};
```

Класс Reader

```
class Reader {
public:
  Reader(int id, Library* libraryPtr, Logger* loggerPtr);
  void run();  // Основной метод жизненного цикла читателя

private:
  std::vector<int> chooseBooks();  // Выбор книг
  void readBooks(const std::vector<int>& books);  // Чтение книг
  void collectMissingBooks(const std::vector<int>& missingBooks);  // Получение недостающих книг

  int id;
  Library* library;
  Logger* logger;
};
```

Класс Logger

```
class Logger {
public:
  Logger(std::ostream& consoleStream, std::ofstream& fileStream);
  void log(const std::string& message);  // Потокобезопасное протоколирование

private:
  std::ostream& console;
  std::ofstream& file;
  pthread_mutex_t logMutex;  // Мьютекс для синхронизации вывода
};
```

3. Механизмы параллельных вычислений

Синхронизационные примитивы

Мьютексы (`pthread_mutex_t`)

- Защита критических секций
- Предотвращение одновременного доступа к общим ресурсам
- Используются для:
  - Блокировки доступа к книгам
  - Синхронизации записи данных о ходе выполнения программы

Условные переменные (`pthread_cond_t`)

- Механизм ожидания освобождения книг
- Уведомление потоков о смене состояния ресурса
- Реализация сценария ожидания недоступной книги

Алгоритм синхронизации

1. Блокировка мьютекса книги
2. Проверка доступности книги
3. При недоступности - ожидание через `pthread_cond_wait()`
4. Освобождение книги через `pthread_cond_broadcast()`
5. Разблокировка мьютекса

4. Генерация случайных данных

Выбор книг

```
std::vector<int> Reader::chooseBooks() {
  int numBooks = rand() % 3 + 1;  // 1-3 книги
  std::vector<int> chosenBooks;

  while (chosenBooks.size() < numBooks) {
    int bookId = rand() % library->numBooks;
    // Проверка уникальности книги
    if (std::find(chosenBooks.begin(), chosenBooks.end(), bookId)
        == chosenBooks.end()) {
      chosenBooks.push_back(bookId);
    }
  }
```

```
  return chosenBooks;
}
```

Имитация чтения

```
void MySleep() {
#ifdef _WIN32
  Sleep((rand() % 3 + 1) * 1000);
#elif __linux__
  usleep((rand() % 3 + 1) * 100000);
#endif
}
```

5. Жизненный цикл читателя

Сценарий визита в библиотеку

1. Выбор случайных книг
2. Попытка получить книги
3. Чтение полученных книг
4. Возврат книг
5. Обработка недоступных книг
6. Повторение визитов

```
void Reader::run() {
  const int numVisits = 3;

  for (int visit = 0; visit < numVisits; ++visit) {
    std::vector<int> desiredBooks = chooseBooks();
    std::vector<int> takenBooks, missingBooks;

    library->takeBooks(id, desiredBooks, takenBooks, missingBooks);

    // Чтение и возврат полученных книг
    if (!takenBooks.empty()) {
      readBooks(takenBooks);
      library->returnBooks(id, takenBooks);
    }

    // Специальный визит для недостающих книг
    if (!missingBooks.empty()) {
      collectMissingBooks(missingBooks);
```

```
    }

    MySleep();  // Время между визитами
  }
}
```

6. Протоколирование событий

Механизм записи

- Потокобезопасная запись
- Синхронный вывод в консоль и файл
- Детальная информация о событиях

```
void Logger::log(const std::string& message) {
  pthread_mutex_lock(&logMutex);
  console << message;
  file << message;
  pthread_mutex_unlock(&logMutex);
}
```

7. Конфигурирование и запуск

Входные параметры

- Количество книг (N>2 (для корректной работы алгоритма))
- Количество читателей (M>0 (для корректной работы алгоритма))
- Файл вывода

Источники конфигурации

1. Интерактивный ввод
2. Аргументы командной строки
3. Конфигурационный файл

8. Возможные сценарии

1. **Успешное получение книг**
    - Все желаемые книги доступны
    - Быстрый цикл получения-чтения-возврата
2. **Частичное получение книг**

    - Некоторые книги недоступны
    - Ожидание освобождения
3. **Конкуренция за книги**
    - Несколько читателей претендуют на одну книгу
    - Корректная синхронизация доступа

Механика генерации случайных чисел

1. Инициализация генератора случайных чисел

В main():

```
srand(time(nullptr));
```

- Использует текущее системное время как seed
- Обеспечивает разные последовательности при каждом запуске программы

2. Инициализация в конструкторе Reader

```
Reader::Reader(int readerId, Library* libraryPtr, Logger* loggerPtr)
  : id(readerId), library(libraryPtr), logger(loggerPtr) {
  // Уникальный seed для каждого читателя
  srand(time(nullptr) + id);
}
```

- Добавление ID читателя к текущему времени
- Гарантирует уникальность последовательностей для разных потоков

3. Выбор книг

```
std::vector<int> Reader::chooseBooks() {
  // Случайное количество книг: 1-3
  int numBooks = rand() % 3 + 1;
  std::vector<int> chosenBooks;

  while (chosenBooks.size() < numBooks) {
    // Случайный ID книги в диапазоне библиотеки
    int bookId = rand() % library->numBooks;

    // Проверка уникальности книги
    if (std::find(chosenBooks.begin(), chosenBooks.end(), bookId)
```

```
      == chosenBooks.end()) {
    chosenBooks.push_back(bookId);
   }
  }

  return chosenBooks;
}
```

4. Имитация времени чтения и ожидания

```
static void MySleep() {
#ifdef _WIN32
  Sleep((rand() % 3 + 1) * 1000);
#elif __linux__
  usleep((rand() % 3 + 1) * 100000);
#endif
}
```

Диапазоны и особенности генерации

Количество книг для чтения

- Диапазон: 1-3 книги
- `rand() % 3 + 1` гарантирует:
  - Минимум 1 книга
  - Максимум 3 книги
  - Равномерное распределение

Выбор книг

- ID книг: от 0 до `numBooks - 1`
- Гарантия уникальности выбранных книг
- Равномерное распределение
  Конечно! В программе реализован многовариантный ввод параметров через командную строку с поддержкой различных ключей.

Варианты ввода параметров командной строки

Основные ключи

1. `-n` или `--books` : Количество книг
   - Пример: `-n 10`

    - Задает число книг в библиотеке
2. `-m` или `--readers` : Количество читателей
    - Пример: `-m 5`
    - Устанавливает число параллельных читателей
3. `-o` или `--output` : Файл для вывода результатов
    - Пример: `-o library_log.txt`
    - Определяет путь файла журнала
4. `-c` или `--config` : Путь к конфигурационному файлу
    - Пример: `-c config.txt`
    - Позволяет загрузить параметры из файла
5. `-h` или `--help` : Справка
    - Выводит подсказку по использованию программы

Код обработки аргументов командной строки

```
int main(int argc, char* argv[]) {
  int N = 0;  // Количество книг
  int M = 0;  // Количество читателей
  std::string outputFile = "";
  std::string configFile = "";

  for (int i = 1; i < argc; ++i) {
    std::string arg = argv[i];

    // Обработка ключа количества книг
    if ((arg == "-n" || arg == "--books") && i + 1 < argc) {
      N = std::atoi(argv[i + 1]);
      i++;
    }

    // Обработка ключа количества читателей
    else if ((arg == "-m" || arg == "--readers") && i + 1 < argc) {
      M = std::atoi(argv[i + 1]);
      i++;
    }

    // Обработка ключа выходного файла
    else if ((arg == "-o" || arg == "--output") && i + 1 < argc) {
      outputFile = argv[i + 1];
      i++;
    }

    // Обработка ключа конфигурационного файла
```

```
    else if ((arg == "-c" || arg == "--config") && i + 1 < argc) {
      configFile = argv[i + 1];
      i++;
    }

    // Вывод справки
    else if (arg == "-h" || arg == "--help") {
      std::cout << "Использование: "
                << argv[0]
                << " [-n количество_книг] "
                << "[-m количество_читателей] "
                << "[-o файл_вывода] "
                << "[-c конфигурационный_файл]\n";
      return 0;
    }
  }
}
```

Примеры запуска программы

1. Полный набор параметров:

```
./ConsoleApp_ABC.exe -n 10 -m 5 -o output.txt
```

2. С использованием конфигурационного файла:

```
./ConsoleApp_ABC.exe -c config1.txt
```

3. Краткая справка:

```
./ConsoleApp_ABC.exe -h
```

Конфигурационный файл

Пример содержимого `config1.txt` :

```
# Количество книг
N = 15

# Количество читателей
M = 7
```

```
# Файл для вывода
outputFile = library_results.txt
```

Преимущества реализации

1. Гибкость настройки
2. Возможность использования конфигурационного файла
3. Встроенная справочная информация
4. Кроссплатформенность
5. Простота расширения

Обработка входных данных

В коде предусмотрена многоуровневая проверка входных параметров:

- Приоритет командной строки
- Fallback к конфигурационному файлу
- Интерактивный ввод при отсутствии данных
- Валидация введенных значений

```
// Проверка корректности количества книг
if (N <= 2) {
  std::cout << "Enter the number of books (N): ";
  while (!(std::cin >> N) || N <= 2) {
    std::cout << "Incorrect input. Enter a positive integer for N and >2: ";
    std::cin.clear();
    std::cin.ignore(std::numeric_limits<std::streamsize>::max(), '\n');
  }
}
```

Такой подход обеспечивает:

- Удобство использования
- Отказоустойчивость
- Интуитивно понятный интерфейс
  Примеры работы программы:
  Тут тоже небольшое предисловие надо:
  Изначально писал код на Visual Studio под Windows, далее тестил работу программы на WSL(Ubuntu), которую поставил на Clion.
  Вот пример ввода и вывода, который осуществляется из консоли:

```
Enter the number of books (N): -1
Incorrect input. Enter a positive integer for N and >2: 4
Enter the number of readers (M): 0
Incorrect input. Enter a positive integer for M: 6
Enter the name of the output file: output.txt
The reader 2 took the book 0.
The reader 2 took the book 2.
The reader 1 couldn't take the book 0 (busy).
The reader 2 took the book 3.
The reader 1 couldn't take the book 2 (busy).
The reader 3 couldn't take the book 0 (busy).
The reader 2 started reading a book(-s): 0 2 3
The reader 1 couldn't take the book 3 (busy).
The reader 1 waits for some time before a special visit for missing books.
The reader 4 couldn't take the book 0 (busy).
The reader 3 couldn't take the book 2 (busy).
The reader 5 couldn't take the book 0 (busy).
The reader 3 couldn't take the book 3 (busy).
The reader 4 couldn't take the book 2 (busy).
The reader 6 couldn't take the book 0 (busy).
The reader 3 waits for some time before a special visit for missing books.
The reader 4 couldn't take the book 3 (busy).
The reader 5 couldn't take the book 2 (busy).
The reader 4 waits for some time before a special visit for missing books.
The reader 5 couldn't take the book 3 (busy).
The reader 6 couldn't take the book 2 (busy).
The reader 5 waits for some time before a special visit for missing books.
The reader 6 couldn't take the book 3 (busy).
The reader 6 waits for some time before a special visit for missing books.
The reader 2 returned the book 0.
The reader 2 returned the book 2.
The reader 2 returned the book 3.
The reader 1 received a notification about the availability of the book 0 and took it.
The reader 1 started reading a book(-s): 0
The reader 2 took the book 2.
The reader 2 started reading a book(-s): 2
The reader 1 returned the book 0.
The reader 5 received a notification about the availability of the book 0 and took it.
The reader 5 started reading a book(-s): 0
The reader 2 returned the book 2.
The reader 1 received a notification about the availability of the book 2 and took it.
The reader 1 started reading a book(-s): 2
The reader 5 returned the book 0.
The reader 6 received a notification about the availability of the book 0 and took it.
The reader 6 started reading a book(-s): 0
The reader 3 exceeded the book waiting time 0.
```

```
The reader 4 exceeded the book waiting time 0.
The reader 1 returned the book 2.
The reader 5 received a notification about the availability of the book 2 and took it.
The reader 1 received a notification about the availability of the book 3 and took it.
The reader 5 started reading a book(-s): 2
The reader 1 started reading a book(-s): 3
The reader 6 returned the book 0.
The reader 1 returned the book 3.
The reader 5 returned the book 2.
The reader 5 received a notification about the availability of the book 3 and took it.
The reader 6 received a notification about the availability of the book 2 and took it.
The reader 5 started reading a book(-s): 3
The reader 6 started reading a book(-s): 2
The reader 4 couldn't take the book 2 (busy).
The reader 4 waits for some time before a special visit for missing books.
The reader 3 couldn't take the book 2 (busy).
The reader 3 waits for some time before a special visit for missing books.
The reader 2 took the book 1.
The reader 2 couldn't take the book 2 (busy).
The reader 2 couldn't take the book 3 (busy).
The reader 2 started reading a book(-s): 1
The reader 6 returned the book 2.
The reader 5 returned the book 3.
The reader 6 received a notification about the availability of the book 3 and took it.
The reader 6 started reading a book(-s): 3
The reader 2 returned the book 1.
The reader 2 waits for some time before a special visit for missing books.
The reader 1 took the book 1.
The reader 1 took the book 2.
The reader 1 couldn't take the book 3 (busy).
The reader 1 started reading a book(-s): 1 2
The reader 6 returned the book 3.
The reader 1 returned the book 1.
The reader 1 returned the book 2.
The reader 1 waits for some time before a special visit for missing books.
The reader 3 received a notification about the availability of the book 2 and took it.
The reader 3 started reading a book(-s): 2
The reader 5 took the book 1.
The reader 5 couldn't take the book 2 (busy).
The reader 5 took the book 3.
The reader 5 started reading a book(-s): 1 3
The reader 6 couldn't take the book 1 (busy).
The reader 6 couldn't take the book 2 (busy).
The reader 6 couldn't take the book 3 (busy).
The reader 6 waits for some time before a special visit for missing books.
The reader 5 returned the book 1.
```

```
The reader 5 returned the book 3.
The reader 5 waits for some time before a special visit for missing books.
The reader 1 received a notification about the availability of the book 3 and took it.
The reader 1 started reading a book(-s): 3
The reader 6 received a notification about the availability of the book 1 and took it.
The reader 1 returned the book 3.
The reader 6 started reading a book(-s): 1
The reader 3 returned the book 2.
The reader 4 received a notification about the availability of the book 2 and took it.
The reader 4 started reading a book(-s): 2
The reader 1 took the book 0.
The reader 1 couldn't take the book 1 (busy).
The reader 1 couldn't take the book 2 (busy).
The reader 1 started reading a book(-s): 0
The reader 6 returned the book 1.
The reader 1 returned the book 0.
The reader 1 waits for some time before a special visit for missing books.
The reader 2 exceeded the book waiting time 2.
The reader 2 completed his visits to the library.
The reader 4 returned the book 2.
The reader 3 took the book 1.
The reader 6 received a notification about the availability of the book 2 and took it.
The reader 6 started reading a book(-s): 2
The reader 3 couldn't take the book 2 (busy).
The reader 3 took the book 3.
The reader 3 started reading a book(-s): 1 3
The reader 5 exceeded the book waiting time 2.
The reader 3 returned the book 1.
The reader 6 returned the book 2.
The reader 3 returned the book 3.
The reader 3 waits for some time before a special visit for missing books.
The reader 6 received a notification about the availability of the book 3 and took it.
The reader 6 started reading a book(-s): 3
The reader 1 received a notification about the availability of the book 1 and took it.
The reader 1 started reading a book(-s): 1
The reader 5 took the book 2.
The reader 5 started reading a book(-s): 2
The reader 6 returned the book 3.
The reader 5 returned the book 2.
The reader 3 received a notification about the availability of the book 2 and took it.
The reader 3 started reading a book(-s): 2
The reader 4 couldn't take the book 1 (busy).
The reader 4 couldn't take the book 2 (busy).
The reader 4 took the book 3.
The reader 4 started reading a book(-s): 3
The reader 3 returned the book 2.
```

```
The reader 4 returned the book 3.
The reader 4 waits for some time before a special visit for missing books.
The reader 1 returned the book 1.
The reader 1 received a notification about the availability of the book 2 and took it.
The reader 1 started reading a book(-s): 2
The reader 5 completed his visits to the library.
The reader 3 completed his visits to the library.
The reader 4 received a notification about the availability of the book 1 and took it.
The reader 6 took the book 0.
The reader 4 started reading a book(-s): 1
The reader 6 started reading a book(-s): 0
The reader 4 returned the book 1.
The reader 6 returned the book 0.
The reader 1 returned the book 2.
The reader 4 received a notification about the availability of the book 2 and took it.
The reader 4 started reading a book(-s): 2
The reader 6 completed his visits to the library.
The reader 4 returned the book 2.
The reader 1 completed his visits to the library.
The reader 4 completed his visits to the library.
The simulation is complete. The results are recorded in output.txt
```

Важное уточнение при стандартном запуске с консоли VS, все выходные файлы, при вводе имени файла как в примере, сохраняются рядом с .cpp и .h файлами.

ConsoleApp_ABC.vcxproj.filters	12.12.2024 23:50	VC++ Project Filte...	2 КБ
ConsoleApp_ABC.vcxproj.user	17.12.2024 1:14	Per-User Project O...	1 КБ
Library.cpp	17.12.2024 1:28	Файл "CPP"	3 КБ
Library.h	17.12.2024 1:26	Исходный файл C...	1 КБ
Logger.cpp	12.12.2024 23:29	Файл "CPP"	1 КБ
Logger.h	17.12.2024 1:27	Исходный файл C...	1 КБ
output.txt	17.12.2024 15:16	Текстовый докум...	8 КБ
Reader.cpp	17.12.2024 1:23	Файл "CPP"	4 КБ
Reader.h	17.12.2024 1:24	Исходный файл C...	1 КБ

При запуске кода из командой строки, все создаваемые программой файлы сохраняются рядом с файлом .exe

```
D:\C++\ConsoleApp_ABC\x64\Debug>.\ConsoleApp_ABC.exe -n 10 -m 5 -o output.txt
Output file from command line: output.txt
Output file is set to: output.txt
The reader 1 took the book 0.
The reader 2 couldn't take the book 0 (busy).
The reader 1 took the book 4.
The reader 3 couldn't take the book 0 (busy).
The reader 1 took the book 7.
The reader 2 couldn't take the book 4 (busy).
The reader 2 couldn't take the book 7 (busy).
The reader 1 started reading a book(-s): 0 4 7
The reader 4 couldn't take the book 0 (busy).
The reader 3 couldn't take the book 4 (busy).
The reader 2 waits for some time before a special visit for missing books.
The reader 5 couldn't take the book 0 (busy).
The reader 3 couldn't take the book 7 (busy).
The reader 4 couldn't take the book 4 (busy).
The reader 3 waits for some time before a special visit for missing books.
The reader 4 couldn't take the book 7 (busy).
The reader 5 couldn't take the book 4 (busy).
The reader 4 waits for some time before a special visit for missing books.
The reader 5 couldn't take the book 7 (busy).
The reader 5 waits for some time before a special visit for missing books.
The reader 1 returned the book 0.
The reader 1 returned the book 4.
The reader 2 received a notification about the availability of the book 0 and took it.
The reader 1 returned the book 7.
The reader 2 started reading a book(-s): 0
The reader 2 returned the book 0.
The reader 1 took the book 8.
The reader 1 started reading a book(-s): 8
The reader 2 received a notification about the availability of the book 4 and took it.
The reader 5 received a notification about the availability of the book 0 and took it.
The reader 2 started reading a book(-s): 4
The reader 5 started reading a book(-s): 0
The reader 2 returned the book 4.
The reader 2 received a notification about the availability of the book 7 and took it.
The reader 2 started reading a book(-s): 7
The reader 1 returned the book 8.
```

```
The reader 3 returned the book 8.
The reader 2 returned the book 2.
The reader 2 returned the book 7.
The reader 2 completed his visits to the library.
The reader 5 took the book 2.
The reader 1 returned the book 1.
The reader 5 took the book 7.
The reader 5 started reading a book(-s): 2 7
The reader 4 returned the book 5.
The reader 4 waits for some time before a special visit for missing books.
The reader 1 received a notification about the availability of the book 5 and took it.
The reader 1 started reading a book(-s): 5
The reader 5 returned the book 2.
The reader 5 returned the book 7.
The reader 3 took the book 1.
The reader 3 couldn't take the book 5 (busy).
The reader 3 took the book 7.
The reader 3 started reading a book(-s): 1 7
The reader 5 completed his visits to the library.
The reader 1 returned the book 5.
The reader 3 returned the book 1.
The reader 4 received a notification about the availability of the book 1 and took it.
The reader 4 started reading a book(-s): 1
The reader 3 returned the book 7.
The reader 3 waits for some time before a special visit for missing books.
The reader 1 received a notification about the availability of the book 7 and took it.
The reader 1 started reading a book(-s): 7
The reader 3 received a notification about the availability of the book 5 and took it.
The reader 3 started reading a book(-s): 5
The reader 4 returned the book 1.
The reader 3 returned the book 5.
The reader 1 returned the book 7.
The reader 4 received a notification about the availability of the book 7 and took it.
The reader 4 started reading a book(-s): 7
The reader 3 completed his visits to the library.
The reader 1 completed his visits to the library.
The reader 4 returned the book 7.
The reader 4 completed his visits to the library.
The simulation is complete. The results are recorded in output.txt

D:\C++\ConsoleApp_ABC\x64\Debug>
```

config1.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
config2.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
config3.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
ConsoleApp_ABC.exe	17.12.2024 15:11	Приложение	482 КБ
ConsoleApp_ABC.pdb	17.12.2024 15:11	Program Debug D...	3 500 КБ
pthreadVC3d.dll	12.12.2024 23:05	Расширение при...	133 КБ
output.txt	17.12.2024 17:16	Текстовый докум...	6 КБ

Пример выходного файла output.txt

```
The reader 1 took the book 0.
The reader 1 took the book 1.
The reader 2 couldn't take the book 0 (busy).
The reader 1 took the book 2.
The reader 2 couldn't take the book 1 (busy).
The reader 4 couldn't take the book 0 (busy).
The reader 1 started reading a book(-s): 0 1 2
The reader 2 couldn't take the book 2 (busy).
The reader 3 couldn't take the book 0 (busy).
The reader 4 couldn't take the book 1 (busy).
The reader 2 waits for some time before a special visit for missing books.
The reader 5 couldn't take the book 0 (busy).
The reader 4 couldn't take the book 2 (busy).
The reader 3 couldn't take the book 1 (busy).
The reader 4 waits for some time before a special visit for missing books.
The reader 3 couldn't take the book 2 (busy).
The reader 5 couldn't take the book 1 (busy).
The reader 3 waits for some time before a special visit for missing books.
The reader 5 couldn't take the book 2 (busy).
The reader 5 waits for some time before a special visit for missing books.
The reader 1 returned the book 0.
The reader 1 returned the book 1.
The reader 1 returned the book 2.
.................................................
The reader 5 returned the book 0.
The reader 5 received a notification about the availability of the book 1 and took it.
The reader 5 started reading a book(-s): 1
The reader 5 returned the book 1.
The reader 5 completed his visits to the library.
```

Также отлавливаются ситуации, когда из командной строки вводятся некорректные данные

```
D:\C++\ConsoleApp_ABC\x64\Debug>.\ConsoleApp_ABC.exe -n 2 -m 5 -o output.txt
Output file from command line: output.txt
Enter the number of books (N): 3
Output file is set to: output.txt
The reader 1 took the book 0.
The reader 4 couldn't take the book 0 (busy).
The reader 1 took the book 1.
The reader 2 couldn't take the book 0 (busy).
The reader 1 took the book 2.
The reader 4 couldn't take the book 1 (busy).
The reader 3 couldn't take the book 0 (busy).
The reader 1 started reading a book(-s): 0 1 2
The reader 4 couldn't take the book 2 (busy).
The reader 2 couldn't take the book 1 (busy).
The reader 5 couldn't take the book 0 (busy).
The reader 4 waits for some time before a special visit for missing books.
The reader 2 couldn't take the book 2 (busy).
The reader 3 couldn't take the book 1 (busy).
```

Есть вывод памятки об используемых параметров для ввода из командой строки

```
D:\C++\ConsoleApp_ABC\x64\Debug>.\ConsoleApp_ABC.exe -h
Usage: .\ConsoleApp_ABC.exe [-n number_of_books] [-m number_of_readers] [-o output_file] [-c config_file]

D:\C++\ConsoleApp_ABC\x64\Debug>
```

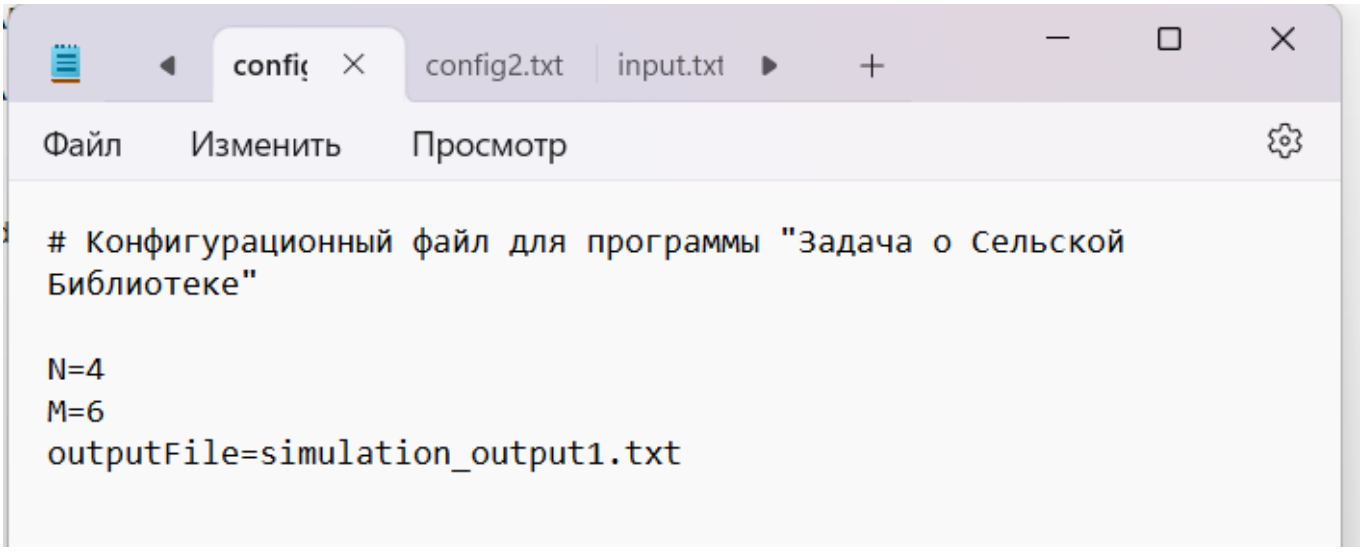
Альтернативный ввод параметров через командную строку

```
D:\C++\ConsoleApp_ABC\x64\Debug>.\ConsoleApp_ABC.exe --books 2 --readers 5 -o output.txt
Output file from command line: output.txt
Enter the number of books (N): 2
Incorrect input. Enter a positive integer for N and >2: 3
Output file is set to: output.txt
The reader 1 took the book 0.
The reader 1 took the book 1.
The reader 2 couldn't take the book 0 (busy).
The reader 1 took the book 2.
The reader 2 couldn't take the book 1 (busy).
The reader 4 couldn't take the book 0 (busy).
The reader 1 started reading a book(-s): 0 1 2
The reader 2 couldn't take the book 2 (busy).
The reader 3 couldn't take the book 0 (busy).
The reader 4 couldn't take the book 1 (busy).
The reader 2 waits for some time before a special visit for missing books.
The reader 5 couldn't take the book 0 (busy).
The reader 4 couldn't take the book 2 (busy).
The reader 3 couldn't take the book 1 (busy).
The reader 4 waits for some time before a special visit for missing books.
The reader 3 couldn't take the book 2 (busy).
The reader 5 couldn't take the book 1 (busy).
The reader 3 waits for some time before a special visit for missing books.
The reader 5 couldn't take the book 2 (busy).
The reader 5 waits for some time before a special visit for missing books.
The reader 1 returned the book 0.
```

Также было создано три файла config, каждый из которых хранит параметры для запуска

config1.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
config2.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
config3.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
ConsoleApp_ABC.exe	17.12.2024 15:11	Приложение	482 КБ

Как выглядит внутри сам файл



Пример работы

```
D:\C++\ConsoleApp_ABC\x64\Debug>.\ConsoleApp_ABC.exe -c config1.txt
The reader 1 took the book 0.
The reader 1 took the book 2.
The reader 2 couldn't take the book 0 (busy).
The reader 1 took the book 3.
The reader 3 couldn't take the book 0 (busy).
The reader 2 couldn't take the book 2 (busy).
The reader 1 started reading a book(-s): 0 2 3
The reader 4 couldn't take the book 0 (busy).
```

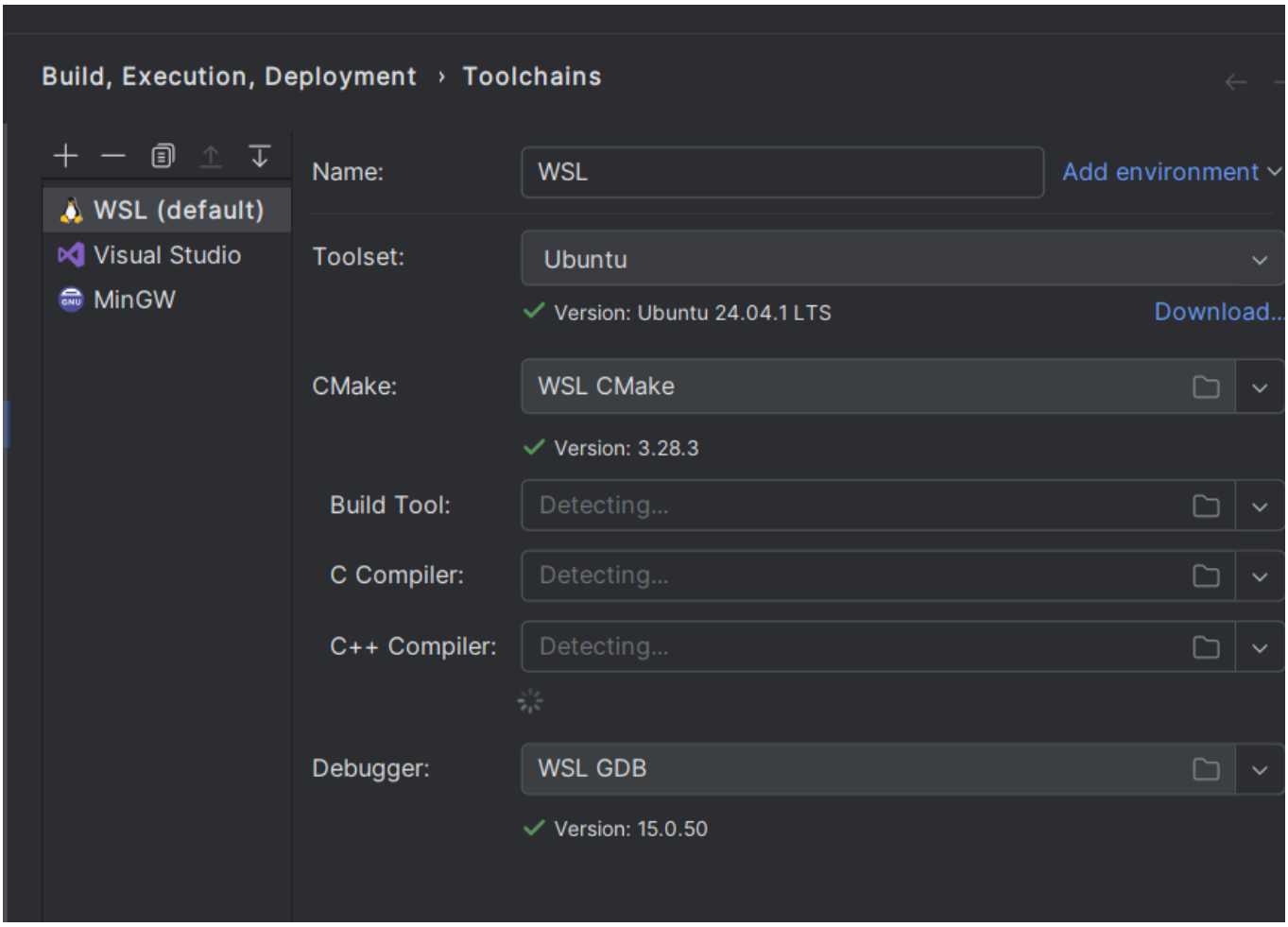
```
The reader 4 received a notification about the availability of the book 1 and took it.
The reader 4 started reading a book(-s): 1
The reader 3 exceeded the book waiting time 2.
The reader 5 completed his visits to the library.
The reader 6 exceeded the book waiting time 2.
The reader 4 returned the book 1.
The reader 2 exceeded the book waiting time 2.
The reader 3 completed his visits to the library.
The reader 6 completed his visits to the library.
The reader 2 completed his visits to the library.
The reader 4 exceeded the book waiting time 2.
The reader 4 completed his visits to the library.
The simulation is complete. The results are recorded in simulation_output1.txt
```

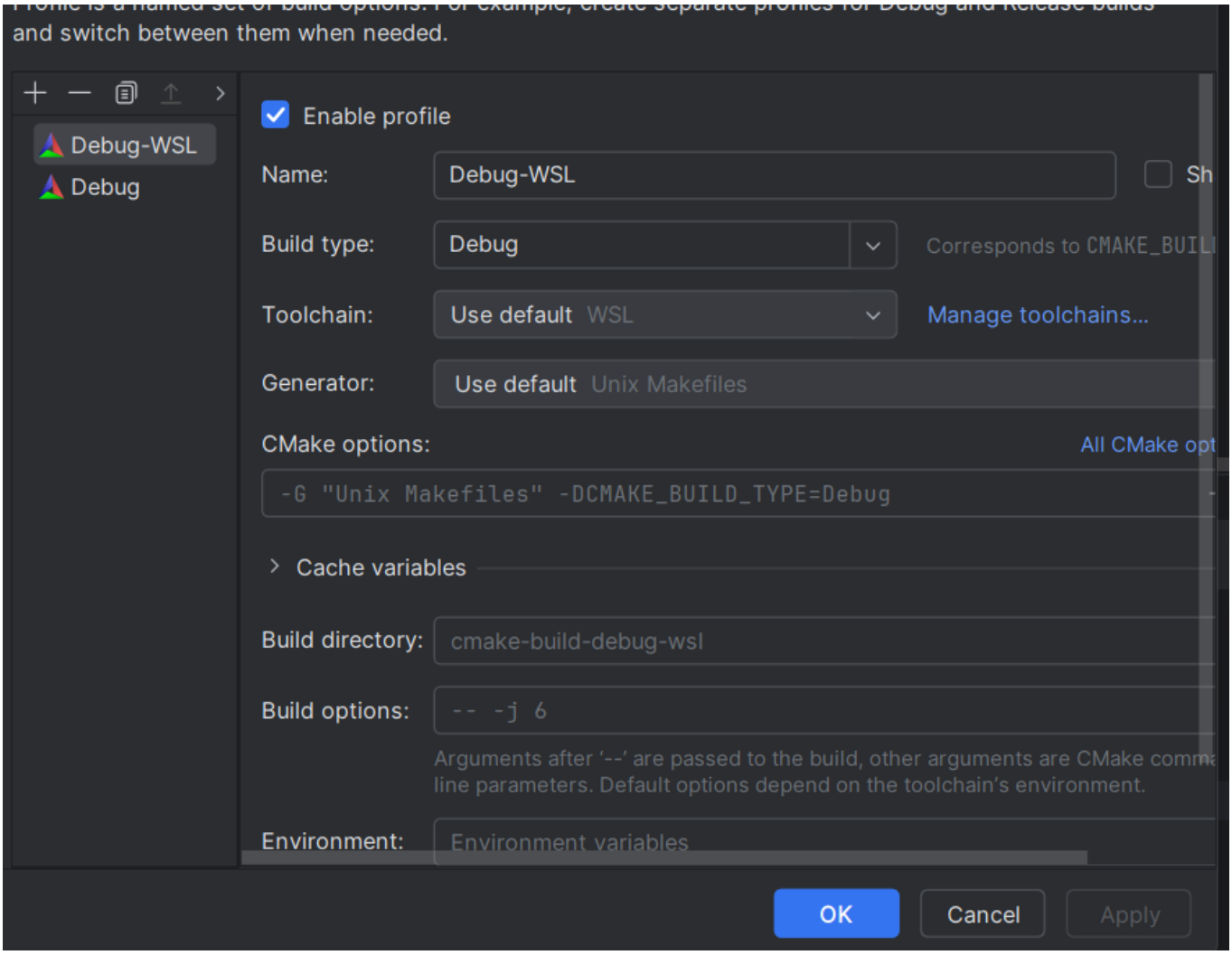
config1.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
config2.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
config3.txt	17.12.2024 1:17	Текстовый докум...	1 КБ
ConsoleApp_ABC.exe	17.12.2024 15:11	Приложение	482 КБ
ConsoleApp_ABC.pdb	17.12.2024 15:11	Program Debug D...	3 500 КБ
output.txt	17.12.2024 17:21	Текстовый докум...	6 КБ
pthreadVC3d.dll	12.12.2024 23:05	Расширение при...	133 КБ
simulation_output1.txt	17.12.2024 17:26	Текстовый докум...	8 КБ

Также запускал в CLion на WSL(Ubuntu)

```
====================[ Build | Console_ABC_4 | Debug ]==========================
/usr/bin/cmake --build /mnt/d/C++/Console_ABC_4/cmake-build-debug --target Console_ABC_4 -- -j 6
[100%] Built target Console_ABC_4

Build finished
```


Build, Execution, Deployment › Toolchains
WSL (default)
Visual Studio
MinGW
Name:
WSL
Add environment
Toolset:
Ubuntu
Version: Ubuntu 24.04.1 LTS
Download...
CMake:
WSL CMake
Version: 3.28.3
Build Tool:
Detecting...
C Compiler:
Detecting...
C++ Compiler:
Detecting...
Debugger:
WSL GDB
Version: 15.0.50




```
/mnt/d/C++/Console_ABC_4/cmake-build-debug/Console_ABC_4
Enter the number of books (N): -2
Incorrect input. Enter a positive integer for N and >2: 2
Incorrect input. Enter a positive integer for N and >2: 3
Enter the number of readers (M): 4
Enter the name of the output file: output.txt
The reader 1 took the book 1.
The reader 1 took the book 2.
The reader 2 couldn't take the book 1 (busy).
The reader 2 couldn't take the book 2 (busy).
The reader 2 waits for some time before a special visit for missing books.
The reader 1 started reading a book(-s): 1 2
The reader 3 took the book 0.
```

Заключение

Программа демонстрирует:

- Эффективное использование многопоточности
- Корректную синхронизацию доступа к общим ресурсам

- Гибкость настройки параметров
- Надёжность при параллельном выполнении

Всем добра!