

# Разработка компонентов электронного задачника, связанных с обработкой структур данных

Цюй Жуйчэнь

Научный руководитель — к.ф.-м.н, доцент

Абрамян Михаил Эдуардович

# Постановка задачи

- Разработать вариант конструктора учебных заданий для языка C++ (в дополнение к существующему для Free Pascal).
- Перенести в новую версию задачника существующие задания по структурам данных из стандартной библиотеки C++ и технологии параллельного программирования MPI с помощью нового конструктора.
- Адаптировать к ОС Linux все компоненты задачника Programming Taskbook версии 5, которые реализованы для ОС Windows.

# Прототип новой версии электронного задачника

- Прототип новой версии электронного задачника был разработан в рамках выполнения выпускных работ в 2022-23 учебном году. Особенностью этой версии является ориентация на **консольный вывод**.
- В имеющемся прототипе была реализована **новая архитектура**, обеспечивающая более простое взаимодействие учебных программ и ядра задачника. При этом было сохранено разделение задачника на **ядро и комплекс подключаемых динамических библиотек**, содержащих реализации групп учебных заданий.
- Был разработан новый вариант **конструктора учебных заданий**, с помощью которого удалось легко адаптировать ранее разработанные группы заданий к новой архитектуре. Созданный прототип позволял выполнять **задания базового уровня**, а также начальные группы заданий из **задачника по параллельному программированию** для языков C и C++.

# Конструктор задач для C++

- CreateGroup: **Создание группы задач**

```
inline void CreateGroup(const char* GroupName, const char*  
GroupDescription, const char* GroupAuthor, const char*  
GroupKey, int TaskCount, TInitTaskProc InitTaskProc);
```

- UseTask: **Импорт существующей группы задач**

```
inline void UseTask(const char* GroupName, int TaskNumber);
```

# Конструктор задач для C++

- CreateTask: **Создание задач с различными параметрами**

```
inline void CreateTask(const char* SubgroupName, int*);
```

- TaskText: **Задание текста задачи**

```
inline void TaskText(const char* S, int X, int Y);
```

- DataN: **Добавление целочисленных данных**

```
inline void DataN(const char* Cmt, int N, int X, int Y, int W);
```

- ResultN: **Добавление результатов целочисленных данных**

```
inline void ResultN(const char* Cmt, int N, int X, int Y, int W);
```

# Перенос компонентов PT5 и конструктора в Linux

В современных операционных системах Dynamic Link Library (DLL) и Shared Object (SO) - две широко используемые технологии для модульности и повторного использования кода.

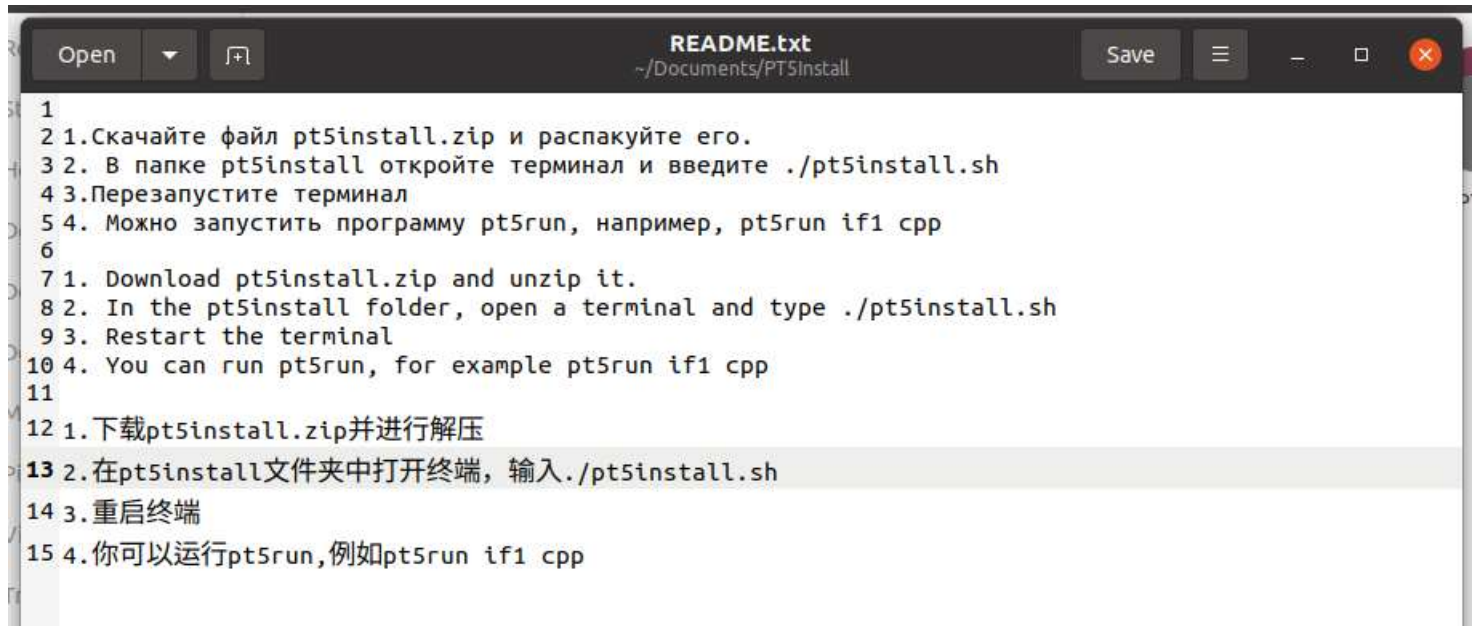
- В Windows: LoadLibrary и GetProcAddress
- В Linux: dlopen и dlsym

# Адаптация для Linux

Проблемы, возникшие при переносе:

- Проблемы с загрузкой динамических библиотек. Если имя файла библиотеки не совпадает по регистру, Unix-системы не могут его найти. В Windows такой проблемы нет.
- Проблемы с использованием имен программ.  
Неправильный регистр в имени файла может привести к тому, что программа не будет найдена и запущена в Unix-системах.

# Инструкция по установке



A screenshot of a text editor window titled "README.txt" with the path "~/Documents/PT5install". The window contains a list of instructions in Russian and English, numbered 1 through 15. The instructions are as follows:

```
1
2 1. Скачайте файл pt5install.zip и распакуйте его.
3 2. В папке pt5install откройте терминал и введите ./pt5install.sh
4 3. Перезапустите терминал
5 4. Можно запустить программу pt5run, например, pt5run if1 cpp
6
7 1. Download pt5install.zip and unzip it.
8 2. In the pt5install folder, open a terminal and type ./pt5install.sh
9 3. Restart the terminal
10 4. You can run pt5run, for example pt5run if1 cpp
11
12 1. 下载pt5install.zip并进行解压
13 2. 在pt5install文件夹中打开终端，输入./pt5install.sh
14 3. 重启终端
15 4. 你可以运行pt5run,例如pt5run if1 cpp
```



# Реализация новых групп заданий

Включает:

- Демонстрационная группа для конструктора C++
- Реализация для задачника РТ5 групп заданий по изучению библиотеки STL
- Реализация для задачника РТ5 группы заданий по изучению параллельных матричных алгоритмов

# Описание новых групп заданий

- DemoCrr(7 заданий) :задачи на обработку **простых числовых данных, массивов, строк и файлов**
- STL1Iter (знакомство с итераторами и алгоритмами, 24 задания)
- STL2Seq (последовательные контейнеры, 34 задания)
- STL3Alg (обобщенные алгоритмы, 64 задания)
- STL4Str (строки как последовательные контейнеры, 28 заданий)
- STL5Assoc (ассоциативные контейнеры, 36 заданий)
- STL6Func (функциональные объекты, 14 заданий)
- MPI9Matr(44 заданий) :группа для **изучения параллельных матричных алгоритмов**

# Реализация одного из заданий демо-группы ДемоСрр

```
void MakerDemo4()
{
    CreateTask("Двумерные массивы (матрицы): вывод элементов");
    TaskText("Дана матрица размера~{M}\\;\\x\\;{N} и целое число~{K} (1~\\l~{K}~\\l~{M}).", 0, 2);
    TaskText("Вывести элементы {K}-й строки данной матрицы.", 0, 4);
    int m = RandomN(2, 5), n = RandomN(4, 8), k = 1;
    if (m == 5)
        k = 0;
    DataN("M = ", m, 3, 1, 1);
    DataN("N = ", n, 10, 1, 1);
    double a[5][8];
    for (int i = 0; i < m; i++)
        for (int j = 0; j < n; j++)
        {
            a[i][j] = RandomR(-9.99, 9.99);
            DataR(a[i][j], Center(j+1, n, 5, 1), i + k + 1, 5);
        }
    k = RandomN(1, m);
    DataN("K = ", k, 68, 5, 1);
    for (int j = 0; j < n; j++)
        ResultR(a[k-1][j], Center(j+1, n, 5, 1), 3, 5);
    SetTestCount(5);
}
```

- Задание генерируется с помощью конструктора для языка C++

# Пример задания из группы DemoCprr

```
guruichen@ubuntu:~/Documents/PT5RunMini$ ./pt5run DemoCprr3 cpp
Programming Taskbook [C++]
Ввод и вывод данных, оператор присваивания
    Даны стороны прямоугольника a и b.
    Найти его площадь  $S = a \cdot b$  и периметр  $P = 2 \cdot (a + b)$ .
Input data
    a = 1.80                                b = 1.40
Example of right solution
                                S = 2.52
                                P = 6.40
DemoCprr3 (1/3)
Acquaintance with the task.
[ENTER - quit, ? - info] PT5Run>
```

- Задание связано с обработкой строковых данных

# Пример задания из группы STL1Iter

```
guruichen@ubuntu:~/Documents/PT5RunMini$ ./pt5run STL1Iter6 cpp
```

Programming Taskbook [C++]

Знакомство с итераторами и алгоритмами

Дана строка name и набор символов. Записать в текстовый файл с именем name исходный набор символов в том же порядке, добавляя после каждого символа пробел. Использовать итераторы `ptin_iterator`, `ostream_iterator` и алгоритм `copy`.

Input data

name = "kuqzlxzo.tst"  
(22): 'A' '4' 'Z' 'P' 'R' 'V' '2' 'c' 'o' 'b' 'e' 'w' 'm' 'K' 'X' 'q' 'C'  
'b' 'o' 'Y' 'q' 'x'

Example of right solution

1: "A 4 Z P R V 2 c o b e w m K X q C b o Y q x "

STL1Iter6

Acquaintance with the task.

(1/5)

```
[ENTER - quit, ? - info] PT5Run>
```

- Задание связано с изучением итераторов и базовых алгоритмов

# Пример задания из группы MPI9Matr

```
guruichen@ubuntu:~/Documents/PT5RunMini$ ./pt5run MPI9Matr8 cpp
"/usr/bin/mpiexec" -n 3 /home/guruichen/Documents/PT5RunMini/ptprj
--Programming Taskbook [C++]
--Band algorithm 1 (horizontal bands)
  Integers M, P, Q and two file names are given in the master process.
  The given files contain elements of a matrix A of the size M x P
  and a matrix B of the size P x Q. Modify the initial stage of the band
  algorithm of matrix multiplication (see the MPI9Matr2 task) as follows:
  each process should read the corresponding bands of the matrices A and B
  directly from the given files using the MPI_File_seek and MPI_File_read_all
  collective functions (a new file view is not required).
  To send the sizes of matrices and file names, use the MPI_Bcast
  collective function.
  Include all these actions in a MatrisScatterFile function (without parameters).
  As a result of the call of this function, each process will receive
  the values NB, P, NB, Q, as well as one-dimensional arrays filled
  with the corresponding bands of the matrices A, B, C. Output all obtained data
  (that is, the numbers NB, P, NB, Q and the bands of the matrices A, B, C)
  in each process after calling the MatrisScatterFile function. Perform the input
  of initial data in the MatrisScatterFile function, perform the output
  of the results in the Solve function.
  Remark. For some bands, some of their elements (namely, the last rows)
  or even the entire bands should not be read from the source files
  and will remain zero-valued ones. However, this situation does not require
  special processing, since the MPI_File_read_all function automatically stops
  reading the data (without generating any error message)
  when the end of the file is reached.
--Input data
Process 0: M = 8 P = 6 Q = 7
File with matrix A: "Adlxmr.tst"
1: -7 3 -7 6 5 -1
7: 2 3 5 -3 -6 -5
13: 6 -4 -1 -7 -4 1
19: -2 2 -4 1 -5 -6
25: -2 6 -7 -6 4 1
31: -5 4 -3 -4 -4 1
37: -6 3 6 2 6 2
43: -7 -5 -7 6 -4 4
File with matrix B: "B2esld.tst"
1: 1 -1 2 5 4 4 4
8: 2 1 -4 7 6 -2 -2
15: 7 7 -7 -4 -2 -4 2
22: 2 5 -1 7 6 6 -4
29: 7 -3 0 -4 -3 3 5
36: 0 -6 -5 -1 0 1 3
--Example of right solution
```

- Задание связано с одним из этапов реализации ленточного алгоритма перемножения матриц.

# Результаты работы

- Реализован расширенный кроссплатформенный вариант электронного задачника **Programming Taskbook** версии 5.
- В дополнение к существующему конструктору учебных заданий для языка Free Pascal разработан **вариант конструктора для языка C++**.
- С помощью разработанного конструктора адаптированы для новой версии задачника существующие **группы заданий, связанные с изучением структур данных**, входящих в стандартную библиотеку шаблонов языка C++, а также **с изучением технологии параллельного программирования MPI**.
- Все компоненты задачника **Programming Taskbook** версии 5, реализованные для ОС Windows, **адаптированы к ОС Linux**.