**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФГБОУ ВО «БРЯНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Кафедра «Информатика и программное обеспечение»

**Расчетно-графическая работа**

по курсу

**«Операционные системы»**

**Тема:** *Алгоритм банкира для одного вида ресурсов*

Руководитель

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ к.т.н., доц. Гулаков К. В.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Студент гр. О-18-ПРИ-РПС-Б

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Подгорняк А.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

**БРЯНСК 2020**

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc41684444)

[Взаимоблокировка процессов 3](#_Toc41684445)

[Цель расчетно-графической работы 3](#_Toc41684446)

[1. Описание алгоритма 4](#_Toc41684447)

[1.1. Общее описание алгоритма 4](#_Toc41684448)

[1.2. Детальное описание алгоритма 4](#_Toc41684449)

[1.3. Блок-схема 5](#_Toc41684450)

[1.4. Листинги 6](#_Toc41684451)

[2. Демонстрация работы 7](#_Toc41684452)

[Заключение 9](#_Toc41684453)

[Список литературы 10](#_Toc41684454)

# Введение

## Взаимоблокировка процессов

Взаимоблокировка процессов может происходить, когда несколько процессов борются за один ресурс. Ресурсы бывают выгружаемые и невыгружаемые, аппаратные и программные.

Выгружаемый ресурс - этот ресурс безболезненно можно забрать у процесса.

Невыгружаемый ресурс - этот ресурс нельзя забрать у процесса без потери данных.

Проблема взаимоблокировок процессов возникает при борьбе за невыгружаемый ресурсы.

**Условия необходимые для взаимоблокировки:**

1. Условие взаимного исключения - в какой-то момент времени, ресурс занят только одним процессом или свободен.
2. Условие удержания и ожидания - процесс, удерживающий ресурс может запрашивать новые ресурсы.
3. Условие отсутствия принудительной выгрузки ресурса.
4. Условие циклического ожидания - должна существовать круговая последовательность из процессов, каждый, из которого ждет доступа к ресурсу, удерживаемому следующим членом последовательности.

## Цель расчетно-графической работы

Целью данной работы является изучение алгоритма банкира для одного вида ресурсов для избегания взаимоблокировок, а также создание программы, которая будет имитировать работу этого алгоритма.

# Описание алгоритма

## Общее описание алгоритма

Алгоритм банкира для безопасного распределения ресурсов операционной системой был предложен Э. Дейкстрой и впервые реализован в операционной системе THE в конце 1960-х гг. Происхождение названия связано с тем, что поведение алгоритма напоминает осторожную стратегию банкира при проведении банковских операций:

* Банкиру поступает запрос от клиента на получение кредита.
* Банкир проверяет, приводит ли этот запрос к небезопасному состоянию.
* Банкир в зависимости от этого дает или отказывает в кредите.

## Детальное описание алгоритма

Система имеет ограниченное количество запасов. Каждый процесс должен априорно обозначить свои потребности в ресурсах по максимуму. Перед выдачей резервов производится оценка возможной взаимоблокировки и определяется безопасная последовательность для выдачи ресурсов, поэтому некоторым процессам, возможно, придется подождать. Когда процедура получает требуемые ресурсы, она должна их вернуть системе за ограниченный период времени. Всё это продолжается до тех пор, пока процессы не будут выполнены

## Блок-схема

Ниже представлена блок-схема выполнения алгоритма (Рис. 1)

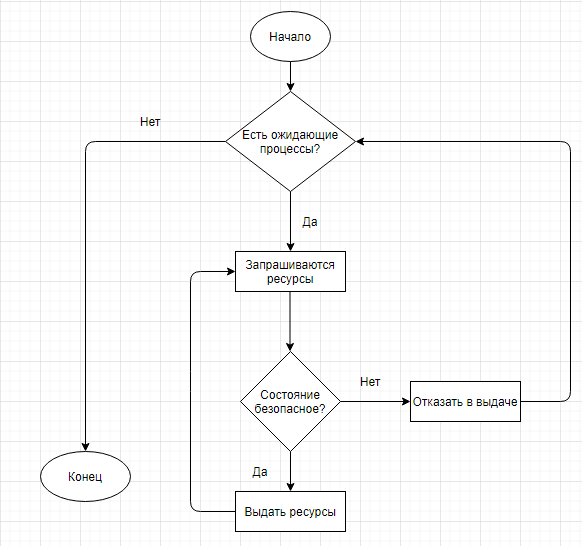


Рис. 1. Блок схема

## Листинги

Листинг 1

*Функции для генерации случайного состояния системы*

|  |
| --- |
| public void FillList()  {  for (int i = 0; i < 4; i++)  {  Processes tmp = new Processes();  tmp.name = (char)(65 + i);  RandomNum = rnd.Next(1, 10);  tmp.max = RandomNum;  RandomNum = rnd.Next(1, 4);  tmp.has = RandomNum;  List.Add(tmp);  tmp = null;  }  }  public void FillTable()  {  dataGridView1.Rows.Add(3);  for (int i = 0; i < 4; i++)  {    dataGridView1.Rows[i].Cells[0].Value = List.ElementAt(i).name;  dataGridView1.Rows[i].Cells[1].Value = List.ElementAt(i).has;  dataGridView1.Rows[i].Cells[2].Value = List.ElementAt(i).max;  dataGridView1.Rows[i].Cells[3].Value = List.ElementAt(i).max - List.ElementAt(i).has;  List.ElementAt(i).need =(int)dataGridView1.Rows[i].Cells[3].Value;  }  SumAll();  Available -= sum;  sum = 0;  label2.Text = "Available: " + Available;  } |

Листинг 2

*Функция обработки запроса на выдачу ресурсов*

|  |
| --- |
| public int Next() |
| { |
| for (int i = 0; i < 4; i++) |
| { |
| if (Available >= List.ElementAt(i).need && List.ElementAt(i).done!= "true") |
| { |
| List.ElementAt(i).done = "true"; |
| sequence +=" "+ List.ElementAt(i).name; |
| return 1; |
|  |
|  |

# Демонстрация работы

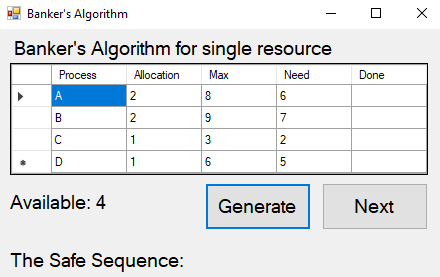


Рис. 2. Генерация состояния системы

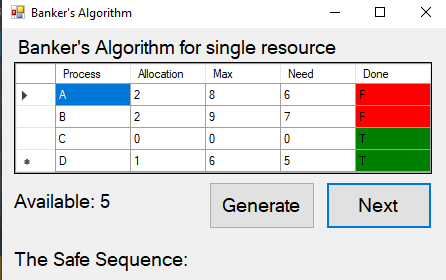


Рис. 3. Процесс поиска безопасной последовательности

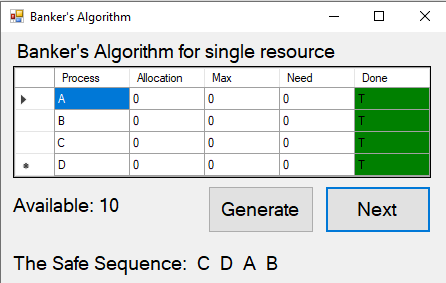


Рис. 4. Результат выполнения алгоритма

После генерации случайного состояния пользователю предлагается вручную пройти все шаги по поиску безопасной последовательности с помощью кнопки “Next”, которая запускает работу алгоритма.

# Заключение

В результате работы была изучена проблема взаимоблокировки процессов, а также ее решение с помощью алгоритма банкира, была написана программа, имитирующая данное решение.

Ссылка на источник с кодом проекта:

<https://drive.google.com/drive/folders/1d51HJG6_W2jMD-HBIaz7Wi0mwcjKwipU?usp=sharing>

# Список литературы

1. Руссинович М. Внутреннее устройство Microsoft Windows, 6-е издание. / Д. Соломон, М. Руссинович. — СПб.: Питер, 2016. – 800 С.
2. Олифер В.Г., Олифер Н.А. Сетевые операционные системы: Учебник для вузов. 2-е изд. / В.Г. Олифер, Н.А. Олифер. – СПб.: Питер, 2016. – 669 с.
3. Дейтел Х., Дейтел П., Чофнес Д.Р. Операционные системы. Основы и принципы [пер. с англ.] / Х. Дейтел, П. Дейтел, Д.Р. Чофнес– М.: БиномПресс, 2017. – 1204 с.
4. Таненбаум Э., Современные операционные системы [пер. с англ.] / Э. Таненбаум. – СПб.: Питер, 2016. – 1120 с.
5. А. Ю. Молчанов Системное программное обеспечение: /. - СПб.; М.; Нижний Новгород: Питер: Питер принт, 2017. - 395 с.
6. Албахари Дж., Албахари Б., «C# 7.0. Карманный справочник.» / Албахари Дж., Албахари Б. – СПб.: Питер, 2017. – 224 с.