МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Санкт-Петербургский государственный технологический институт  
(технический университет)»  
СПбГТИ(ТУ)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| УГНС | | 09.00.00 | | Информатика и вычислительная техника | | |
| Направление подготовки | | 09.03.01 | | Информатика и вычислительная техника | | |
| Направленность (профиль) | |  | | Системы автоматизированного проектирования | | |
| Форма обучения | |  | | очная | | |
|  | |  | |  | | |
| Факультет | |  | | Информационных технологий и управления | | |
| Кафедра | |  | | Систем автоматизированного  проектирования и управления | | |
| Учебная дисциплина | |  | **ОПЕРАЦИОННЫЕ СИСТЕМЫ** | | | |
| Курс | III | | | | Группа | 494 |

**КУРСОВОЙ ПРОЕКТ**

|  |  |
| --- | --- |
| **Тема:** | **Создание приложения, реализующего одну из базовых стратегий операционной системы** |
| **Задача:** | **Реализация моделей алгоритмов замены страниц** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент |  |  |  | Казанцев А.М. |
| Руководитель,  доцент, к.т.н. |  |  |  | Макарук Р.В. |
| Оценка за курсовой проект |  |  |  |  |

Санкт-Петербург

2021-2022 уч. год

Содержание

[ВВЕДЕНИЕ 3](#_Toc92722825)

[1 Аналитический обзор 4](#_Toc92722826)

[1.1 Обзор имеющихся на рынке систем сравнения алгоритмов замещения страниц 4](#_Toc92722827)

[1.2 Анализ технологии разработки программного обеспечения в среде RAD (RAD - rapid application development) 5](#_Toc92722828)

[1.3 Обзор основных стратегий ОС по управлению основной памятью 6](#_Toc92722829)

[1.4 Распределение памяти по алгоритмам замены страниц 7](#_Toc92722830)

[1.5 Установка программного продукта в операционной системе и анализ возможностей для переноса приложения на другой компьютер 10](#_Toc92722831)

[1.6 Обзор и обоснование выбора инструментальных средств разработки программного комплекса. 10](#_Toc92722832)

[2 Основная часть 12](#_Toc92722833)

[2.1 Определение структурной (иерархической) схемы решения задачи 12](#_Toc92722834)

[2.2 Определение основных этапов проектирования 12](#_Toc92722835)

[2.3 Анализ ограничений и исключительных ситуаций для алгоритмов 13](#_Toc92722836)

[2.4 Разработка основных алгоритмов задачи 13](#_Toc92722837)

[2.5 Разработка архитектуры программы 13](#_Toc92722838)

[2.6 Разработка дисплейных фрагментов 14](#_Toc92722839)

[2.7 Отладка программного комплекса 19](#_Toc92722840)

[2.8 Создание справочной системы приложения 19](#_Toc92722841)

[2.9 Создание инсталляционного пакета 21](#_Toc92722842)

[2.10 Тестирование разработанного программного продукта 23](#_Toc92722843)

[ВЫВОДЫ ПО РАБОТЕ 28](#_Toc92722844)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 29](#_Toc92722845)

[ПРИЛОЖЕНИЕ А 30](#_Toc92722846)

ВВЕДЕНИЕ

В настоящей время очень часто применяется метод виртуальной памяти, позволяющий выполнять программы, требующие больше оперативной памяти, чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем (например, жёстким диском). Для выполняющейся программы данный метод полностью прозрачен и не требует дополнительных усилий со стороны программиста, однако реализация этого метода требует как аппаратной поддержки, так и поддержки со стороны операционной системы.

В большинстве современных операционных систем виртуальная память организуется с помощью страничной адресации. Оперативная память делится на страницы: области памяти фиксированной длины (например, 4096 байт), которые являются минимальной единицей выделяемой памяти (то есть даже запрос на 1 байт от приложения приведёт к выделению ему страницы памяти).

Процессор преобразует номер виртуальной страницы в адрес соответствующей ей физической страницы при помощи буфера ассоциативной трансляции. Если ему не удалось это сделать, то требуется до заполнения буфера путём обращения к таблице страниц, что может сделать либо сам процессор, либо операционная система. Если страница была выгружена из оперативной памяти, то операционная система подкачивает страницу с жёсткого диска в ходе обработки события Page fault. При запросе на выделение памяти операционная система может «сбросить» на жёсткий диск страницы, к которым давно не было обращений. Критические данные обычно находятся в оперативной памяти [1].

Чтобы решить какую из страниц можно заменить, а какие лучше оставить в оперативной памяти используются алгоритмы замещения страниц. Алгоритмы могут как использовать статистические данные на момент “страничного сбоя”, так и не использовать их вовсе [2].

Целью данного курсового проекта является разработка программного комплекса для реализующего и сравнивающего между собой такие популярные алгоритмы замещения страниц, как LRU, NRU, FIFO, “вторая попытка” и часы.

Необходимо ознакомится с имеющимися на рынке системами сравнивающим между собой разнообразные алгоритмы вытеснения страниц, проанализировать имеющиеся возможности для реализации поставленной цели в Microsoft Windows 10, а также обеспечить возможности для переноса реализуемого приложения на другой компьютер.

1 Аналитический обзор

* 1. Обзор имеющихся на рынке систем сравнения алгоритмов замещения страниц

Программа, иллюстрирующая работу алгоритмов FIFO и LRU (рисунок 1, 2).

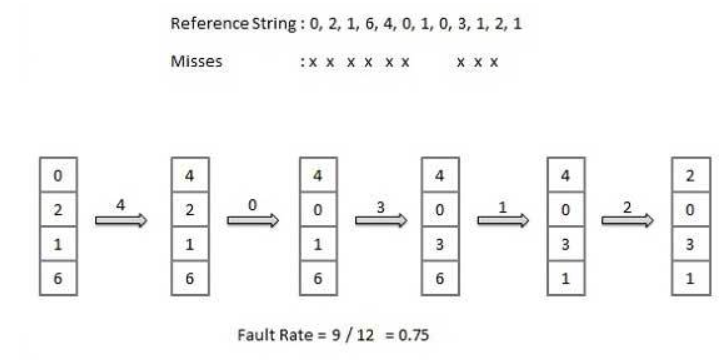


Рисунок 1 — Программ наглядно иллюстрирующая работу алгоритмов замещения страниц, в данном случае FIFO

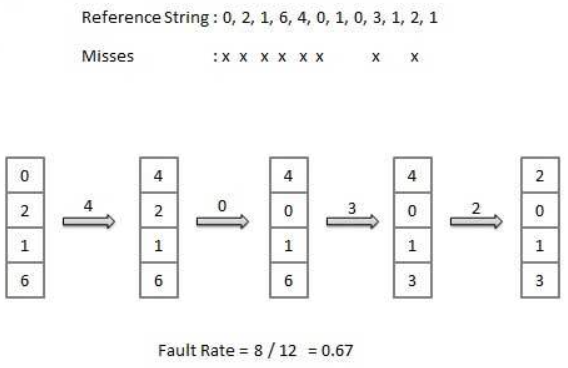


Рисунок 2 — Программ наглядно иллюстрирующая работу алгоритмов замещения страниц, в данном случае LRU

В данной программе при вводе строки, обозначающей последовательность номеров страниц, происходит расчёт алгоритмом и наглядный вывод изменения заполненности страничных блоков, также точно указывается в каких местах произошли прерывания [2].

1.2 Анализ технологии разработки программного обеспечения в среде RAD (RAD - rapid application development)

RAD (от англ. Rapid application development — быстрая разработка приложений) — концепция создания средств разработки программных продуктов, уделяющая особое внимание быстроте и удобству программирования, созданию технологического процесса, позволяющего программисту максимально быстро создавать компьютерные программы. С конца XX века RAD получила широкое распространение и одобрение. Концепцию RAD также часто связывают с концепцией визуального программирования [2].

Основные принципы RAD:

* инструментарий должен быть нацелен на минимизацию времени разработки.
* создание прототипа для уточнения требований заказчика.
* цикличность разработки: каждая новая версия продукта основывается на оценке результата работы предыдущей версии заказчиком.
* минимизация времени разработки версии, за счёт переноса уже готовых модулей и добавления функциональности в новую версию.
* команда разработчиков должна тесно сотрудничать, каждый участник должен быть готов выполнять несколько обязанностей.
* управление проектом должно минимизировать длительность цикла разработки [4].

Причины популярности RAD вытекают из тех преимуществ, которые обеспечивает эта технология (рисунок 3).

Наиболее существенными из них являются:

* высокая скорость разработки;
* низкая стоимость;
* высокое качество.



Рисунок 3 — Преимущества технологии RAD

Последнее из указанных свойств подразумевает полное выполнение требований заказчика как функциональных, так и нефункциональных, с учетом их возможных изменений в период разработки системы, а также получение качественной документации, обеспечивающей удобство эксплуатации и сопровождения системы. Это означает, что дополнительные затраты на сопровождение сразу после поставки будут значительно меньше. Таким образом, полное время от начала разработки до получения приемлемого продукта при использовании этого метода значительно сокращается [3].

1.3 Обзор основных стратегий ОС по управлению основной памятью

**Менеджер памяти** - часть операционной системы, отвечающая за управление памятью.

Основные методы распределения памяти:

* + без использования внешней памяти;
  + с использованием внешней памяти.

Методы без использования внешней памяти бывают:

1. однозадачными без подкачки на диск. Память разделяется только между программой и операционной системой. Такой вариант используется в MS-DOS;
2. распределение памяти с фиксированными разделами. Память просто разделяется на несколько разделов (возможно, не равных). Процессы могут быть разными, поэтому каждому разделу необходим разный размер памяти. Такие системы могут иметь общую очередь ко всем разделам или отдельную к каждому;
3. распределение памяти динамическими разделами. В такой системе сначала память свободна, потом идет динамическое распределение памяти. Основными недостатками такой системы являются: сложность реализации и фрагментация памяти [4].

Методы с использованием внешней памяти чаще всего берут диск за основу, при этом основными методами работы с диско являются:

* + Свопинг (подкачка) - процесс целиком загружается в память для работы при её нехватке;
  + Виртуальная память - процесс может быть частично загружен в память для работы. Основная идея заключается в разбиении программы на части, и в память эти части загружаются по очереди. Программа при этом общается с виртуальной памятью, а не с физической.

Одним из основных способов организации виртуальной памяти является её страничное представление.

Страницы — это части, на которые разбивается пространство виртуальных адресов.

Страничные блоки - единицы физической памяти.

Страницы всегда имеют фиксированный размер. Передача данных между ОЗУ и диском всегда происходит в страницах [5].

Страничное прерывание - происходит, если процесс обратился к странице, которая не загружена в ОЗУ. Процессор передается другому процессу, и параллельно страница загружается в память.

Таблица страниц - используется для хранения соответствия адресов виртуальной страницы и страничного блока (рисунок 4).

Таблица может быть размещена:

* + в аппаратных регистрах;
  + в ОЗУ.

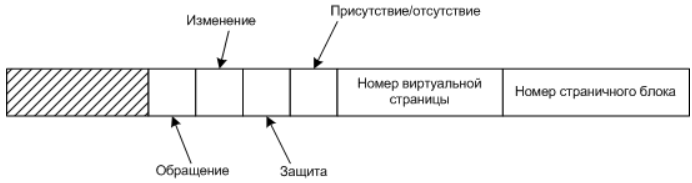


Рисунок 4 — Типичная запись таблицы страниц

Присутствие/отсутствие - загружена или не загружена в память;

Защита - виды доступа, например, чтение/запись;

Изменение - изменилась ли страница, если да то при выгрузке записывается на диск, если нет, просто уничтожается;

Обращение - было ли обращение к странице, если нет, то это лучший кандидат на освобождение памяти.

Информация о адресе страницы, когда она хранится на диске, в таблице не размещается.

Для ускорения доступа к страницам в диспетчере памяти создают буфер быстрого преобразования адреса, в котором хранится информация о наиболее часто используемых страниц.

Страничная организация памяти используется, и в UNIX, и в Windows [4].

1.4 Распределение памяти по алгоритмам замены страниц

При обработке страничного прерывания, если в ОП нет свободной физической страницы, какая-то из них должна быть освобождена. Иными словами, должна быть вытеснена занимающая ее математическая страница для «наложения» на эту физическую страницу требуемой математической. Если вытесняемая страница не является объектным кодом (программой), то она может содержать измененную при счете информацию, т.е. в общем случае эта страница должна быть записана (возвращена) на свое место во внешней памяти.

При замещении страниц сценарий их будущего использования заранее не известен, поэтому для выбора кандидатуры на замещение можно использовать лишь статистические данные на момент «страничного сбоя», либо не использовать их вовсе.

Идеальный алгоритм заключается в том, чтобы выгружать ту страницу, которая будет запрошена позже всех, но этот алгоритм не осуществим, т.к. нельзя знать какую страницу, когда запросят. Можно лишь набрать статистику использования [2].

**Алгоритм NRU**

Используется информация бит изменения и обращения. Важно реализовать изменение этих бит при каждом обращении к памяти, поэтому необходимо, чтобы они задавались аппаратно. Когда процесс запускается оба бита всех его страниц равны нулю. Периодически, например, по таймеру, бит обращения очищается, чтобы отличить страницы, к которым давно не было обращений от используемых. Когда возникает страничное прерывание, система проверяет все страницы. Они делятся на 4 класса:

0 – не было изменений и обращений

1 – не было обращений, но страница изменена

2 – было обращение, страница не изменилась

3 – были и изменения, и обращения.

Алгоритм NRU (not recently used) – замещает страницу в непустом классе с наименьшим номером, если их количество больше 1, то замещаемая страница выбирается случайно из таковых. Считается, что лучше выгрузить измененную страницу, к которой не было обращений хотя бы в течение последнего тика таймера, чем страницу, к которой такие обращения были [4].

**Алгоритм FIFO**

Система поддерживает список всех страниц, которые хранятся в памяти, причем в порядке их поступления. Поэтому самая первая страница является и самой старой. В результате удаляется страница, находящаяся в начале списка, а новая добавляется в его конец (рисунок 5).

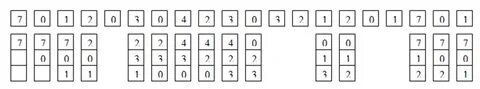


Рисунок 5 — Пример выполнения алгоритма FIFO

**Алгоритм LRU**

Дольше всего не использовавшаяся страница. Замещается та страница, к которой дольше всего не было обращений. Для реализации алгоритма необходимо в таблице добавить поле, хранящее счётчик номера входа. При страничном прерывании ищется страница с наименьшим значением и заменяется на новую, её счётчик увеличивается на 1 от максимального, тоже самое происходит с счётчиком страницы при обращении, если не было прерывания [4].

**Алгоритм “вторая попытка”**

Алгоритм «вторая попытка» ищет в списке самую старую страницу, к которой не было обращений в предыдущем временном интервале.

Подобен FIFO, но если R=1, то страница переводится в конец очереди, если R=0, то страница выгружается (рисунок 6).

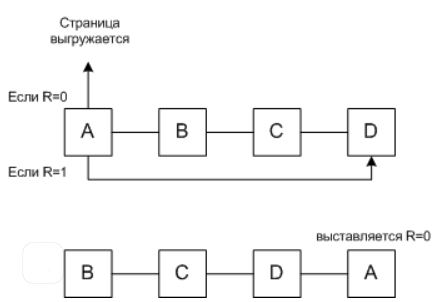


Рисунок 6 — Алгоритм “вторая попытка”

В таком алгоритме часто используемая страница никогда не покинет память. В этом алгоритме приходится часто перемещать страницы по списку [2, 4].

**Алгоритм “часы”**

Хотя алгоритм «вторая попытка» является корректным, он неэффективен, потому что постоянно передвигает страницы по списку. Поэтому лучше хранить все страничные блоки в кольцевом списке в форме часов. Стрелка указывает на старейшую страницу.

Когда происходит страничное прерывание, проверяется та страница, на которую направлена стрелка. Если ее бит R равен 0, страница выгружается, на ее место в часовой круг встает новая страница, а стрелка сдвигается вперед на одну позицию. Если бит R равен 1, то он сбрасывается, стрелка перемещается к следующей странице. Этот процесс повторяется до тех пор, пока не находится та страница, у которой бит R = 0 (рисунок 7).

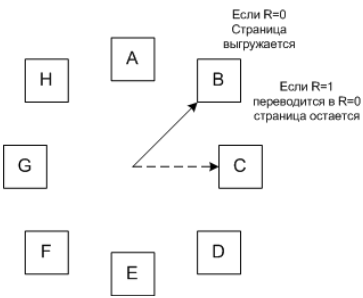


Рисунок 7 — Алгоритм “часы”

Чтобы избежать перемещения страниц по списку, можно использовать указатель, который перемещается по списку [5].

1.5 Установка программного продукта в операционной системе и анализ возможностей для переноса приложения на другой компьютер

Установка программного обеспечения — процесс установки программного обеспечения на компьютер конечного пользователя. Выполняется особой программой (пакетным менеджером), присутствующей в операционной системе (например, Установщик Windows в Microsoft Windows), или же входящим в состав самого программного обеспечения средством установки.

Большинство программ поставляются для распространения в сжатом виде. Для нормальной работы они должны быть распакованы, а необходимые данные правильно размещены на компьютере, учитывая различия между компьютерами и настройками пользователя. В процессе установки выполняются различные тесты на соответствие заданным требованиям, а компьютер необходимым образом настраивается для хранения файлов и данных, необходимых для правильной работы программы.

Установка, включает в себя размещение всех необходимых программе файлов в файловой системе, а также модификацию и создание конфигурационных файлов. Пакетные менеджеры также выполняют при установке контроль зависимостей, проверяя, есть ли в системе необходимые для работы данной программы пакеты, а в случае успешной установки регистрируя новый пакет в списке доступных [5].

Одной из задач данного курсового проекта является обеспечение возможности переноса программного продукта на другие компьютеры. Для выполнения данной задачи был выбран Microsoft Visual Studio Installer Projects, так ка средой разработки программы выбрана программа Visual Studio 2019, а также в силу того, что для данного проекта не требуются специализированные функции и возможности, предоставляемые другими пакетами программ.

1.6 Обзор и обоснование выбора инструментальных средств разработки программного комплекса.

Реализация поставленной задачи требует разработки программного продукта для мониторинга контролируемых технологических параметров. Для этого было выбрано 3 языка программирования C#, C++ и Python и составлена их сравнительная характеристика, представленная в таблице ниже:

Таблица 1 — Сравнительная характеристика языков программирования

|  |  |  |  |
| --- | --- | --- | --- |
| **Критерий сравнения** | **C#** | **C++** | **Python** |
| Поддержка ООП | + | + | + |
| Сетевое программирование | + | + | + |
| Статическая типизация | + | + | - |
| Автоматический сбор мусора | + | - | + |
| Компилируемый язык | + | + | - |

По результатам сравнения (таблица 1) и учитывая, что для реализации программного комплекса не требуется возможности более низкоуровневого программирования, был выбран язык C#, располагающий автоматическим сборщиком мусора. Также данный язык программирования имеет сравнительно невысокий порог вхождения.

В качестве сред разработки были отобраны Visual Studio 2019 и Sublime Text. Sublime Text является простым редактором кода с возможностями автодополнения и выбора используемого языка, но без возможности отладки программ без установки дополнительных расширений.

Для данного курсового проекта в качестве среды разработки была выбрана Visual Studio 2019 от компании Microsoft, так как она имеет максимальную совместимость с выбранным языком программирования, позволяет отлаживать программы, располагает множеством инструментов для быстрой и качественной разработки, а также включает в себя все возможности Sublime Text.

2 Основная часть

2.1 Определение структурной (иерархической) схемы решения задачи

В рамках данного курсового проекта необходимо спроектировать программный комплекс для изучения и сравнения различных алгоритмов замещения страниц. Необходимо реализовать требуемые возможности, изображенные в виде иерархической схемы (рисунок 8).

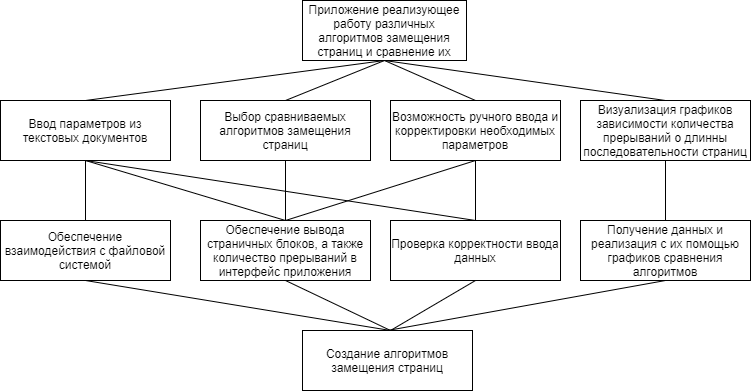


Рисунок 8 — Иерархическая схема решения задачи

2.2 Определение основных этапов проектирования

Среди этапов проектирования приложения выделены следующие:

* анализ требований и составление иерархической схемы решения задачи;
* анализ алгоритмов замещения страниц;
* разработка программного обеспечения;
  + разработка алгоритмов замещения страниц;
  + разработка интерфейсной составляющей проекта;
  + разработка взаимодействия с файловой системой для обеспечения ввода данных, а также сохранения последовательности номеров входящих страниц;
  + создание графиков и подключение к ним получаемых в ходе работы алгоритмов данных;
  + обеспечение проверки корректности входных данных:
  + обработка исключительных ситуаций;
* тестирование программного продукта;
* составление технической документации по проекту.

2.3 Анализ ограничений и исключительных ситуаций для алгоритмов

В процессе разработки программного продукта возникал ряд исключительных ситуаций, требующих обработки. Основные ограничения введены для обеспечения корректности ввода данных.

Формат файлов, из которых производится ввод последовательности номеров страниц — строго текстовый .txt. Внутри читаемого .txt файла для корректной работы программы должны содержаться строки, включающие в себя числовой номер одной страницы. Таким образом числа разделены между собой строками, строки — знаком возврата каретки (перевода строки). Допустимый диапазон значений: целочисленный и не выше максимального страничного номера, заданного пользователем. Наличие любых символов, отличных от десятичных цифр, приведут к возникновению исключительной ситуации и невозможности ввести параметры из данного файла до устранения ошибки в них. Обеспечение корректности ввода входных параметров позволяет избежать некоторых исключительных ситуаций при работе основного алгоритма программы.

2.4 Разработка основных алгоритмов задачи

Основным алгоритмом программы являются работа и сравнение на графике алгоритмов замещения страниц FIFO, LRU, NRU, “часы”, “вторая попытка”, про принципы работы которых рассказывается в аналитическом обзоре.

Результатами работы алгоритма программы являются две таблицы и график:

1. первая таблица включает в себя выбранные для сравнения алгоритмы замещения страниц, страничные блоки, содержащие номера страниц, а также количество прерываний после поступления набора страниц (рисунок 9);
2. вторая таблица содержит в себе данные для построения графика: количество посланных страниц и общее количество прерываний для выбранных алгоритмов сортировки (рисунок 17). Заполняется таблица по мере вхождения новых номеров страниц и отчищается при изменении настроек программы (увеличения количества страничных блоков, выбора других алгоритмов для равнения и т.п.);
3. график сравнения строится на основе выбранных пользователем алгоритмов замещения страниц и второй таблицы, пример графика показан на рисунке 18.

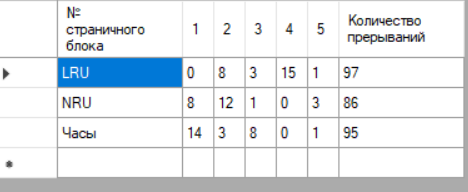


Рисунок 9 — таблица сравнения выбранных алгоритмов сортировки (LRU, NRU, “часы”)

2.5 Разработка архитектуры программы

Программа разбита на 2 формы:

* MainForm – отвечает за взаимодействие с пользователем, работы алгоритмов сортировки страниц, а также сохранение данных в таблицу для построения графика;
* AnalisAlgoritmsSortingPageForm – отвечает за построение графика и показание таблицы прерываний при обработки набора номеров страниц, заданных пользователем вручную или через файл.

Основные функции алгоритмов замещения страниц находятся внутри класса MainForm и показаны в таблице 2.

Таблица 2 — Основные функции, реализующие работу алгоритмов замещения страниц

|  |  |
| --- | --- |
| **Название функции** | **Назначение** |
| FIFONewPage | Функция, реализующая алгоритм сортировки FIFO |
| LRUNewPage | Функция, реализующая алгоритм сортировки LRU |
| NRUNewPage | Функция, реализующая алгоритм сортировки NRU |
| SecondAttemptNewPage | Функция, реализующая алгоритм сортировки “вторая попытка” |
| ClockNewPage | Функция, реализующая алгоритм сортировки “часы” |
| timerNRY\_Tick | Вспомогательная функция алгоритма NRU для обнуления битов обращения к странице, если его не происходило на протяжении отведённого времени. Функция срабатывает каждые 20 мс. |

Также помимо создания перечисленных функций использованы методы для обработки визуального взаимодействия пользователя и программы.

2.6 Разработка дисплейных фрагментов

При первом запуске программного продукта появляется рабочая область, ожидающая ввода исходных данных и выбора алгоритмов (рисунок 10).

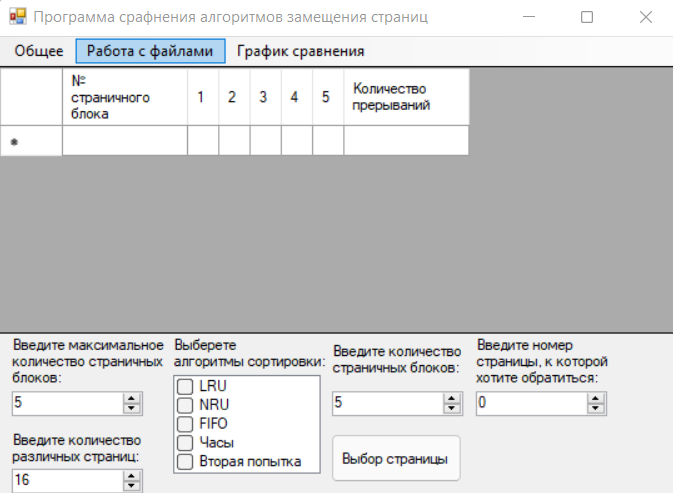


Рисунок 10 — Главное окно приложения при первом запуске

Возможности построения графиков на данном этапе недоступны в силу отсутствия данных для обработки. Пользователю предлагается ввести данные из файла при помощи выбора в меню «Работа с файлами» → «Открыть» (рисунок 11).

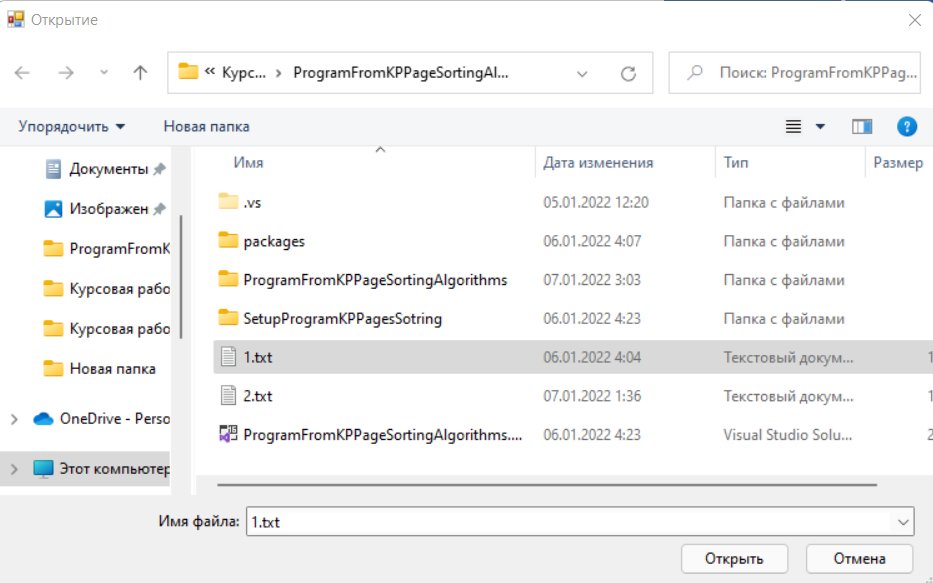


Рисунок 11 — Процесс открытия файла с данными

В случае, если файл содержит некорректные данные, то пользователь получит соответствующее сообщение об ошибке (рисунок 12).

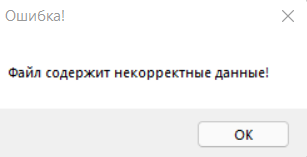


Рисунок 12 — Ошибка ввода некорректных данных

В случае, если данные в файле не подходят под заданные пользователем параметры высветиться другое сообщение об ошибке (рисунок 13), так же на рисунке 14 показано, каким будет это сообщение, если пользователь не выберет хотя бы один из алгоритмов.

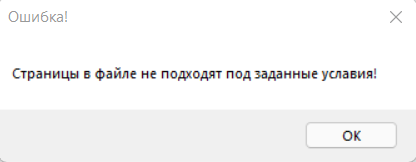


Рисунок 13 — Ошибка ввода данных неподходящих под заданные параметры

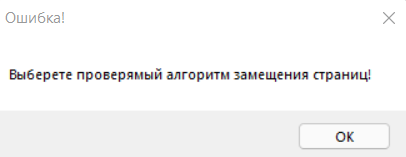


Рисунок 14 — Ошибка не был выбран алгоритм замещения страниц

Дополнительно имеются сообщения об ошибках (рисунок 15) при попытке задать неверные параметры для максимумов

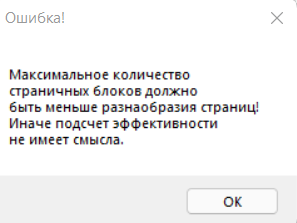


Рисунок 15 — Ошибка ввода некорректных максимумов

После корректного ввода последовательности номеров страниц, происходит их поочередный ввод в страничные блоки, а в середине главного окна отображается результат (рисунок 16). Одновременно создаётся таблица значений (рисунок 17) для графика сравнения (рисунок 18), после обработки файла показывается соответствующее окно (рисунок 19).

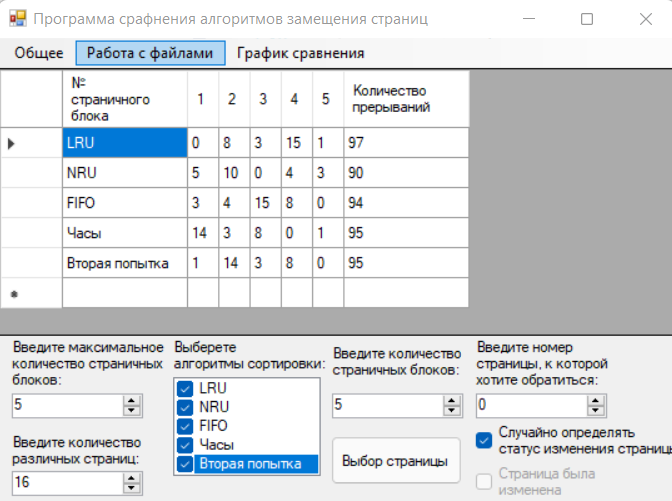


Рисунок 16 — Результат считанных данных из файла

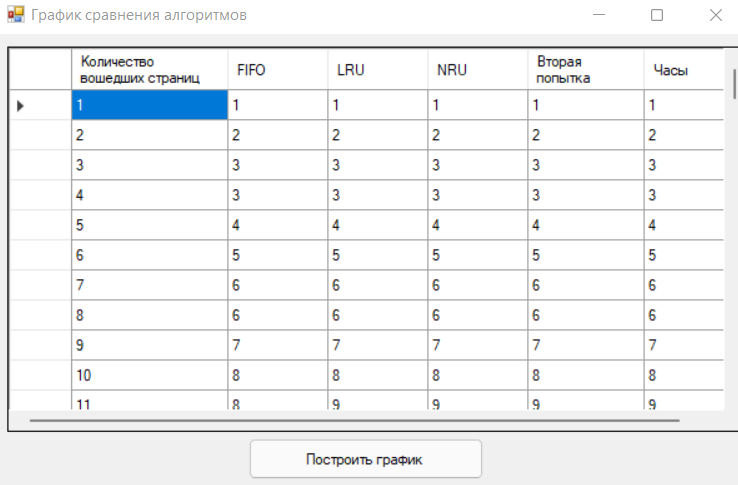


Рисунок 17 — Таблица значений для составления графика

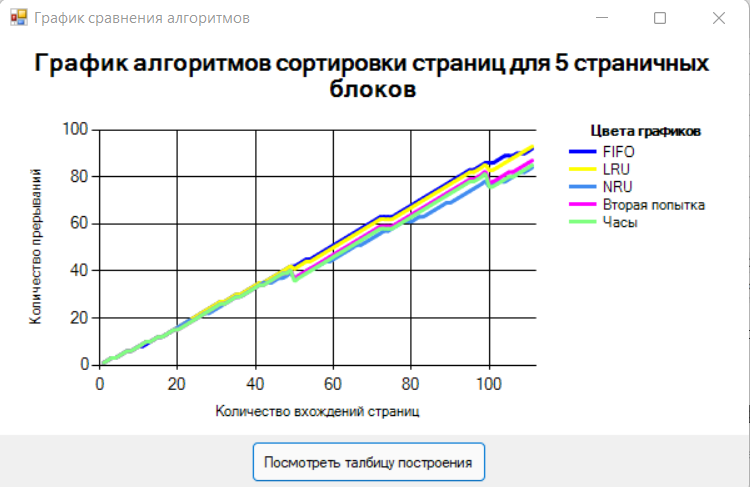


Рисунок 18 — График сравнения алгоритмов

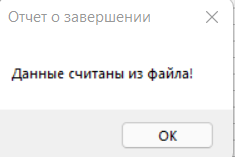


Рисунок 19 — Отчет о завершении обработки файла

Также у пользователя есть возможность сохранить последовательность номеров страниц (рисунок 20), если будет нечего сохранять, то выведется сообщение об ошибке (рисунок 21), при удачном сохранении выведется сообщение о завершении этого действия (рисунок 22) и будет создан сохраненный файл (рисунок 23).

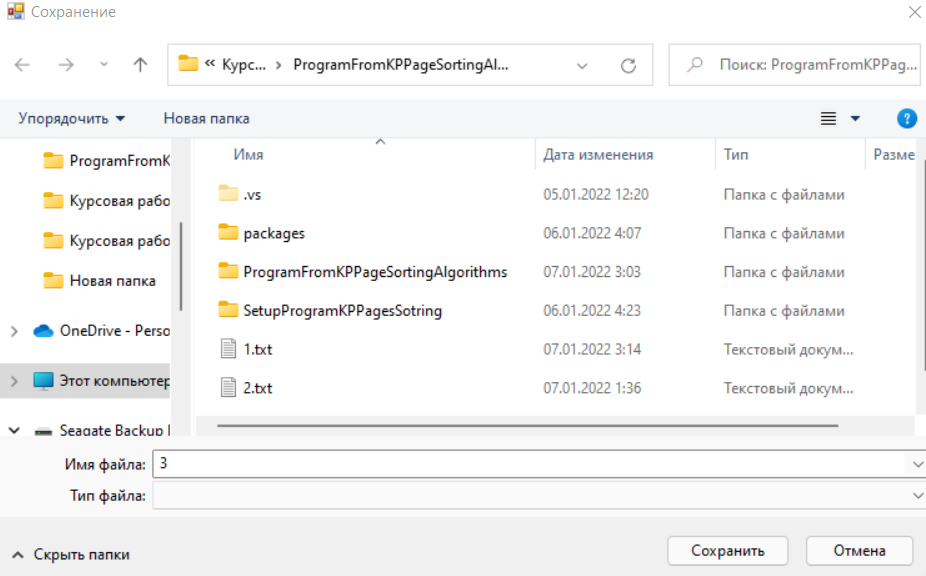


Рисунок 20 — Процесс выбора или создания файла для сохранения данных

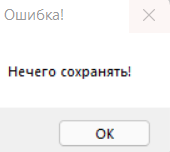


Рисунок 21 — Ошибка набор номер страниц пуст

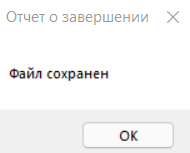


Рисунок 22 — Сообщение о завершении сохранения данных

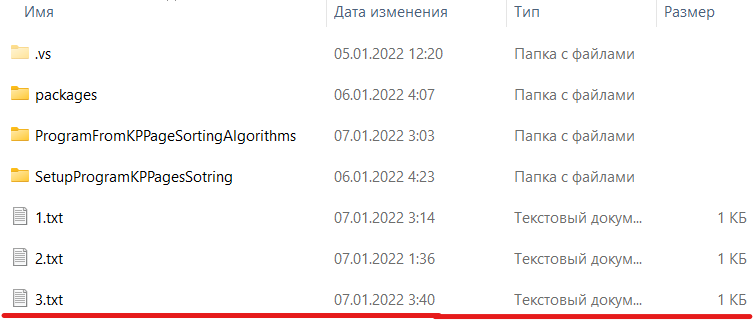
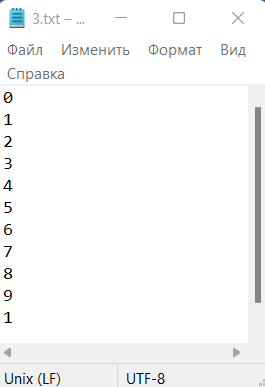
 

Рисунок 23 — Файл с данными и его местонахождение

В меню в верхней части окна приложения при наведении на «Общее» появиться две кнопки: «Справка» и «О программе», открывающие информационные окна с краткой справочной информацией и сведениях о программном продукте и его авторе (рисунок 24, 25).

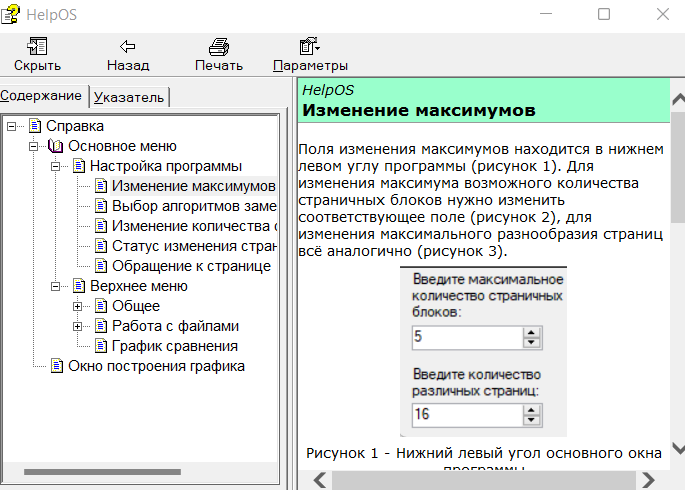


Рисунок 24 - Справочная система приложения

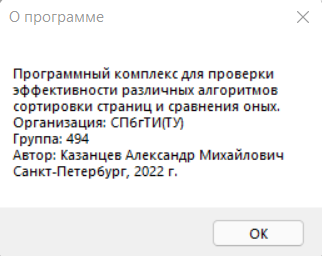


Рисунок 25 — Информационное окно «О программе»

2.7 Отладка программного комплекса

Отладка программного комплекса производилась путем проверки поведения различных частей программного кода при изменяющихся настройках программы: максимального и общего количеств страничных блоков, а также вариаций самих страниц. В процессе отладки были выявлены, локализованы и устранены следующие ошибки:

* построение графика невыбранного алгоритма замещения страниц;
* ошибки, возникающие при открытии недопустимых файлов с исходными данными;
* ошибки, возникающие при открытии неподдерживаемого формата файлов;
* некорректная работа алгоритма NRU из-за невозможности задать были ли изменения внутри входящей страницы, особенно эта проблема выражена при взятии набора номеров страниц из файла.

2.8 Создание справочной системы приложения

Для создания справочной системы использовался пакет программ CHM Editor (рисунок 26), позволяющий редактировать файлы с расширением .chm, добавлять текстовую и графическую информацию на страницы справки, а также структурировать ее содержимое.

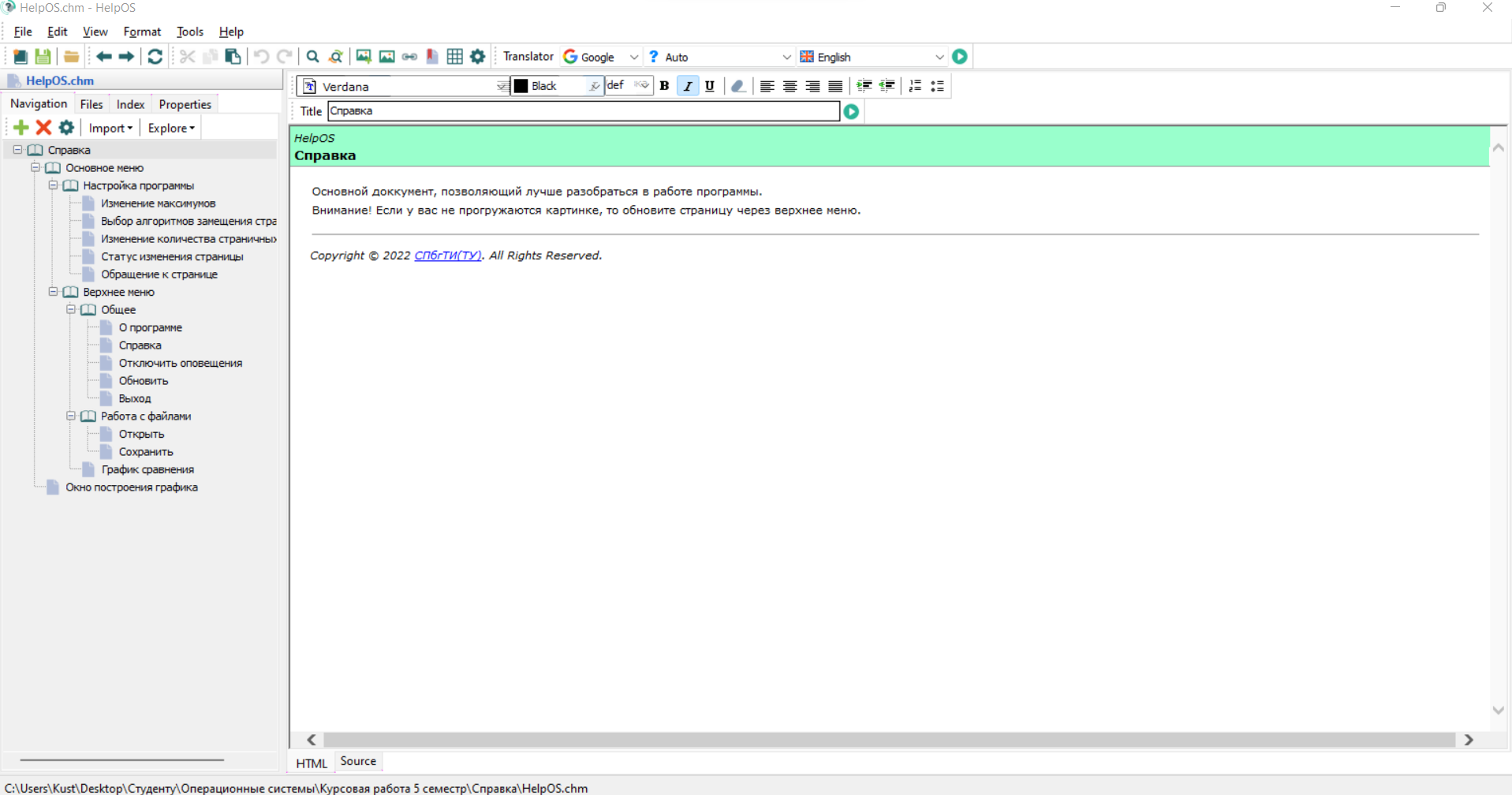


Рисунок 26 — Основное окно программы для создания справочной системы

В процессе создания справки были описаны основные возможности программного продукта и способы взаимодействия с приложением [6]. Шаги и способы работы с программным комплексом для полноты восприятия сопровождаются рисунками (рисунок 27).

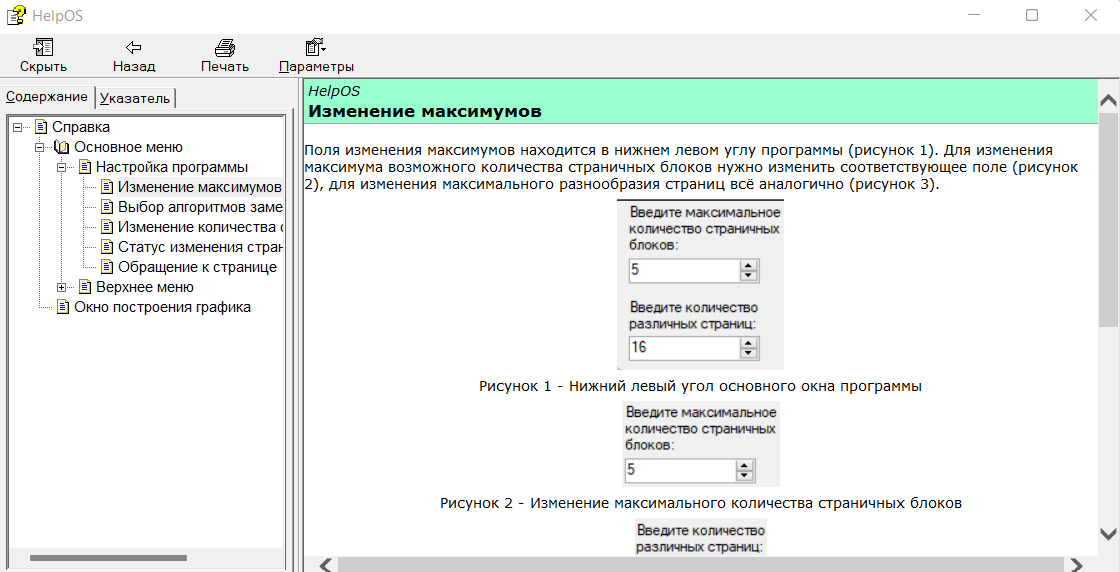


Рисунок 27 — Фрагмент справочной системы

Также в справочной системе присутствует поиск по ключевым словам для более быстрого взаимодействия и поиска необходимой информации.

2.9 Создание инсталляционного пакета

Для создания инсталляционного пакета использовался программный комплекс Microsoft Visual Studio Installer Projects.

Первым шагом создания установочного пакета является создание проекта установочного файла внутри решения нашей программы (рисунок 28).

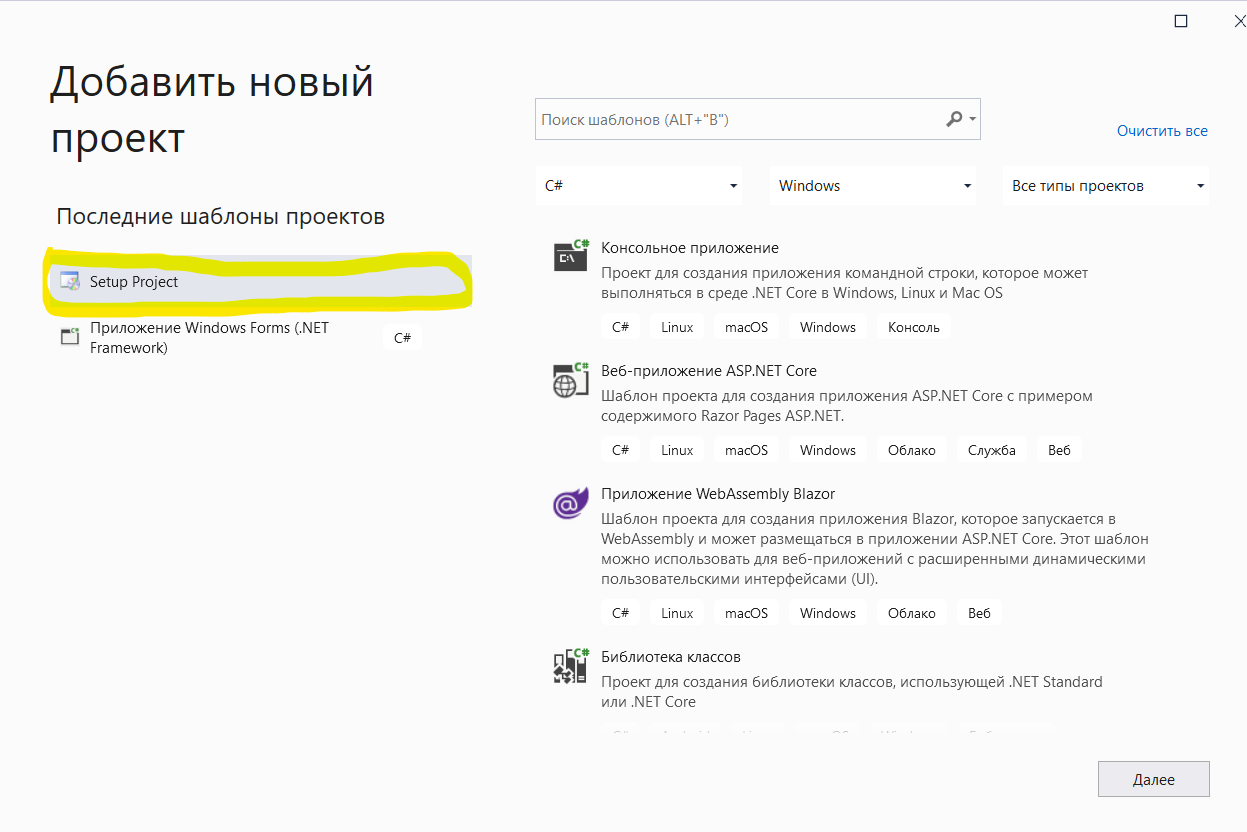


Рисунок 28 — Начало создания инсталляционного пакета

Далее необходимо указать название проекта (рисунок 29, 30).

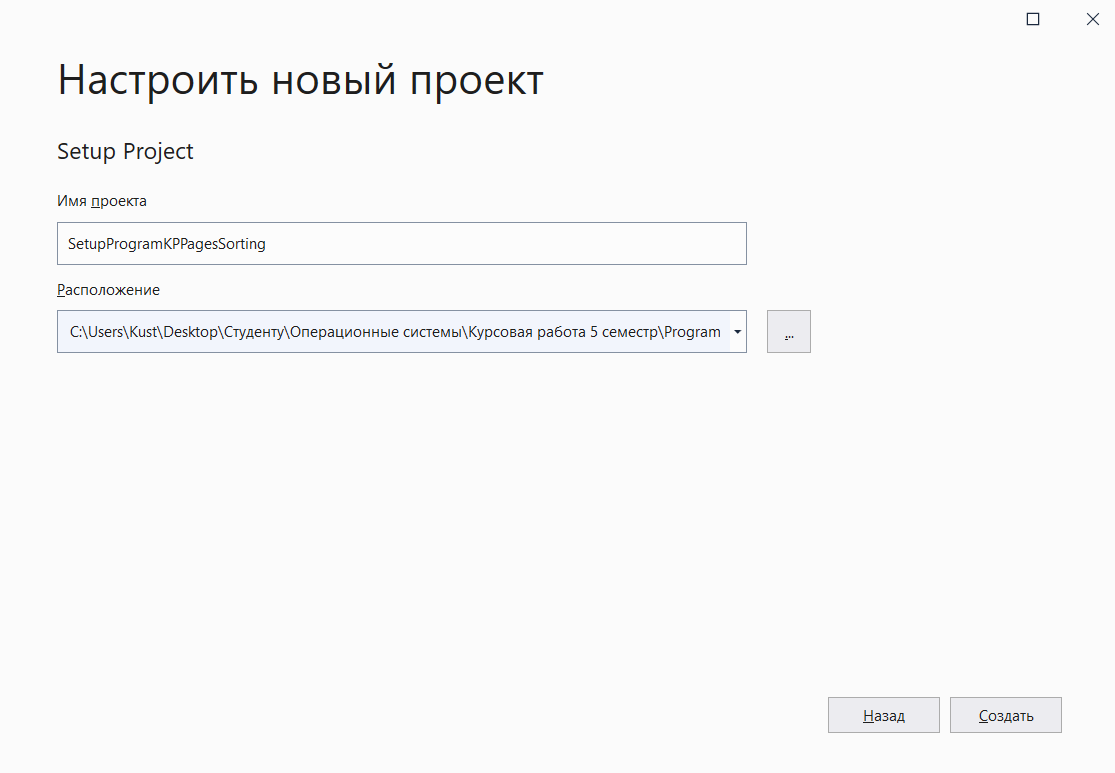


Рисунок 29 — Создание установочного проекта

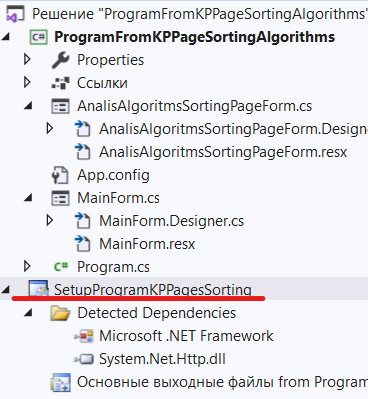


Рисунок 30 — Созданный проект

На следующем шаге необходимо создать Shortcut проекта и добавить его в папки User’s Desktop и App, созданный внутри User’s Programs Menu (рисунки 31, 32, 33).

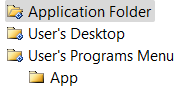
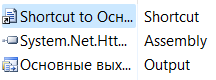
 

Рисунок 31 — Создание Shortcut

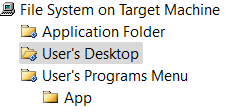
 

Рисунок 32 — Перемещение в User’s Desktop и переименование Shortcut

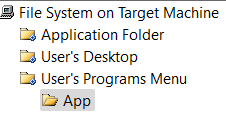
 

Рисунок 33 — Перемещение в App и переименование Shortcut

Далее в свойствах установочного проекта (рисунок 34) можно изменить автора приложения, компанию и другие конфигурации.

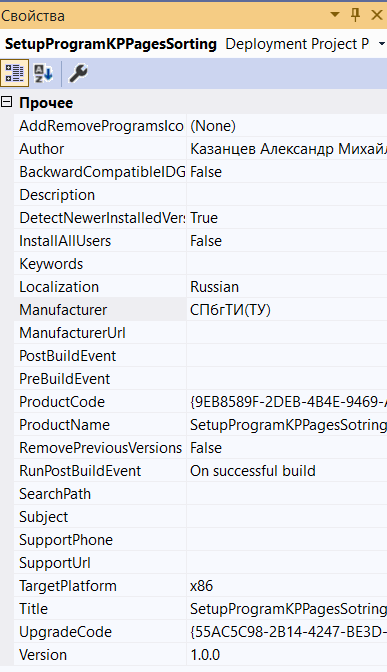


Рисунок 34 — Изменение некоторых конфигураций установочного проекта

Последним шагом является сборка установочного проекта, далее, если посмотреть в файлы проекта, то можно найти его установщик (рисунок 35).

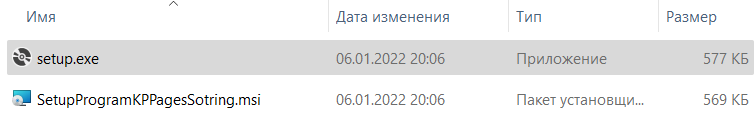


Рисунок 35 — Файл установки программы и его дополнительные файлы

2.10 Тестирование разработанного программного продукта

Для установки программного продукта необходимо открыть мастер установки и выполнить ряд шагов (рисунок 36).

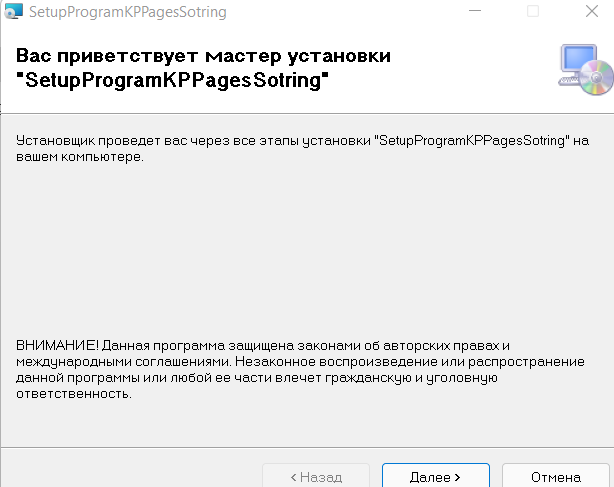


Рисунок 36 — Мастер установки приложения

Выбрав меню «Далее», доступен выбор папки, куда будет установлено приложение, а также выбор для кого будет доступно это приложение после установки (рисунок 37).

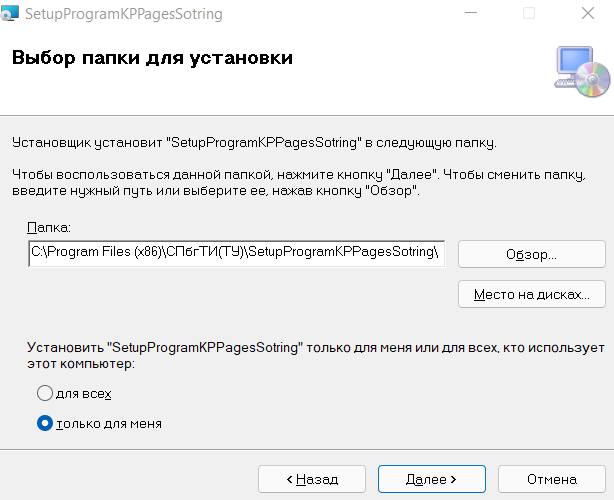


Рисунок 37 — Опции установки

После нажатие кнопки “Долее” происходит подтверждение установки (рисунок 38), мы снова выбираем “Далее” и начинается установка (рисунок 39).

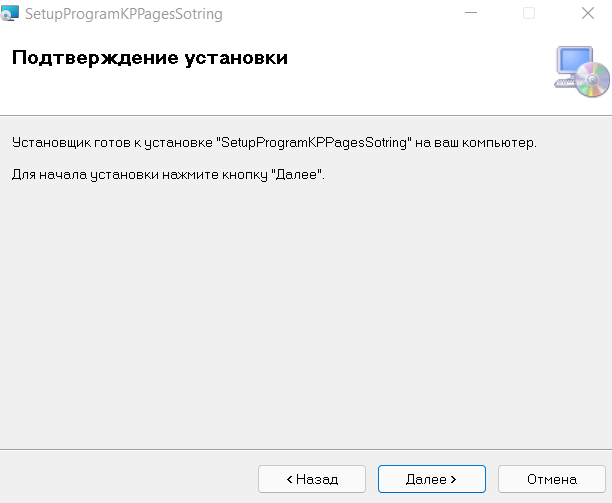


Рисунок 39 — Подтверждение установки

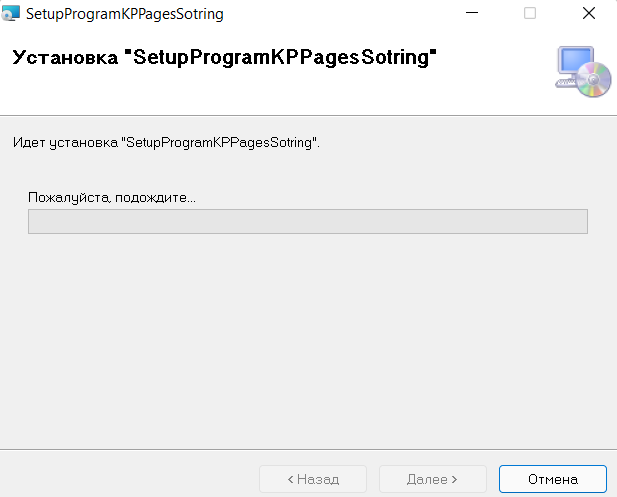


Рисунок 39 — Процесс установки

После завершение на компьютере пользователя появляется каталог, содержащий все необходимые данные и ярлык главного исполняемого файла (рисунок 40).

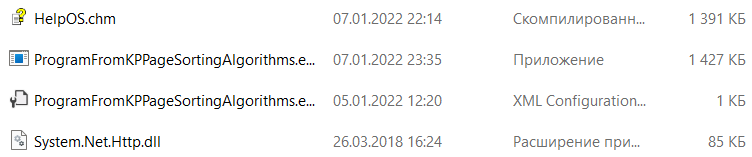


Рисунок 40 — Результат установки приложения

Тестирование программного комплекса, реализующего работу алгоритмов замещения страниц, происходило на тестовых данных, представленных на рисунке 41, и настройках программы (рисунок 43).



Рисунок 41 — Входящая последовательность номеров страниц

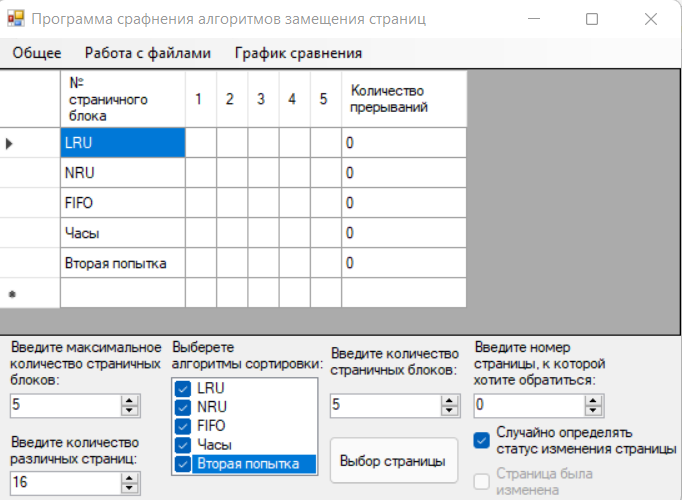


Рисунок 42 — Основное окно программы с определенными настройками тестирования

Результат вхождения данных и работы алгоритмов замещения страниц изображен на рисунках 43, 44, 45

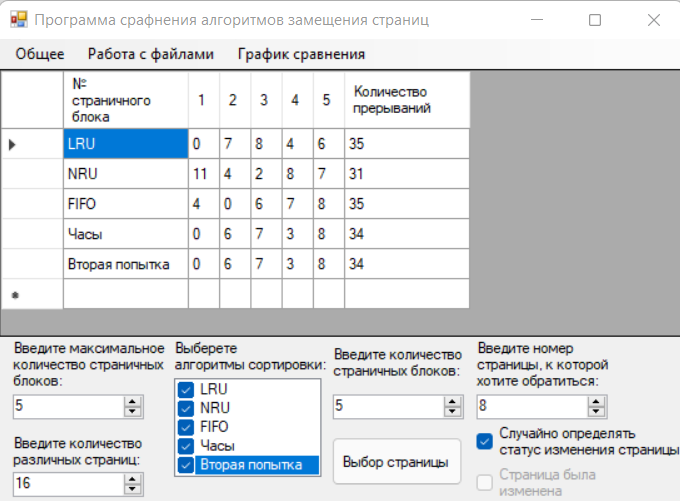


Рисунок 43 — Основное окно программы с результатом вхождения последовательности

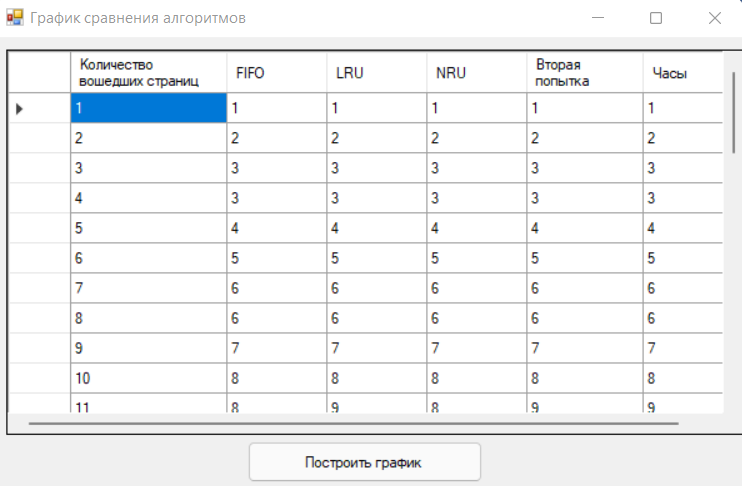


Рисунок 44 — Таблица данных для построения графика

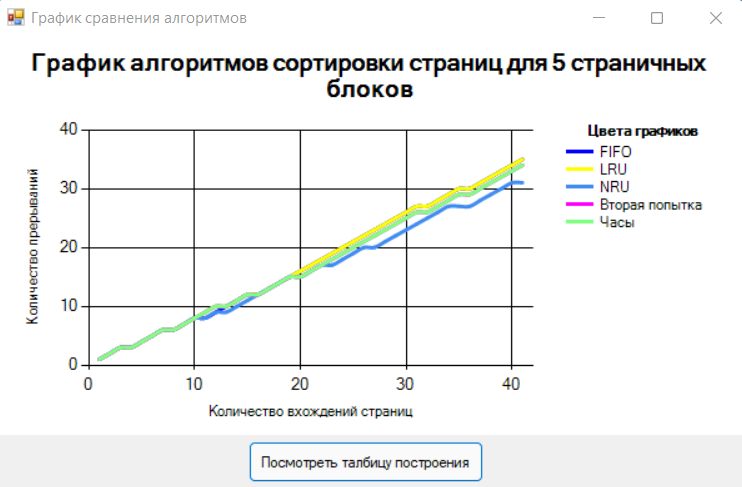


Рисунок 45 — График сравнения алгоритмов сортировки

ВЫВОДЫ ПО РАБОТЕ

В процессе выполнения данной курсовой работы была создана программа реализующая алгоритмы замещения страниц и сравнивающая их, получены навыки взаимодействия с файловой системой компьютера. В ходе разработки были выполнены следующие задачи:

* обоснование выбора инструментальных средств разработки;
* разработка структурной схемы решения задачи;
* разработка структуры данных, определение диапазона изменений;
* анализ ограничений и исключительных ситуаций;
* разработка основных алгоритмов программы;
* разработка архитектуры и дисплейных фрагментов приложения;
* разработка блок-схем реализации алгоритмов замещения страниц в программе;
* создание и отладка модулей программного комплекса;
* создание справочной системы программного продукта;
* создание инсталляционного пакета программы;
* тестирование разработанного программного продукта;
* разработка эксплуатационного документа «Руководство системного программиста»;
* оформление отчетной документации по проекту.

Перспективы развития приложения предполагают обеспечения большего разнообразия алгоритмов замещения страниц, а также создание графика сравнения общего количества прерываний при разнообразном количестве страничных блоков, это поможет лучше понять, когда лучше использовать тот или иной алгоритм.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вахалия, Ю. UNIX изнутри : учебное пособие / Ю. Вахалия — Санкт-Петербург : Питер, 2003. — 844 с. ISBN 5-94723-013-5
2. Барский, А. Б. Теория цифрового компьютера: учебное пособие / А.Б. Барский, В.В. Шилов — Москва : Издательский Дом ФОРУМ, 2019. — 301 с. — ISBN 978-5-8199-0774-0 // StudRef : электронно-библиотечная система : сайт. — URL: https://studref.com (дата обращения 08.01.2022). — Режим доступа : свободный
3. Солонин, Е. Б. Современные методики разработки информационных систем: учебное пособие / Е.Б. Солонин; Минобрнауки России, Уральский федеральный университет имени первого Президента России Б. Н. Ельцина, Кафедра вычислительной техники. — Екатеринбург: УрФУ, 2015. — 45 с. // Информационный портал УрФУ — URL: <http://study.urfu.ru> (дата обращения 08.01.2022) — Режим доступа: свободный
4. [Олифер, В.](http://irb.technolog.edu.ru/CGI/irbis64r_12/cgiirbis_64.exe?LNG=&Z21ID=&I21DBN=EK&P21DBN=EK&S21STN=1&S21REF=1&S21FMT=fullwebr&C21COM=S&S21CNR=20&S21P01=0&S21P02=1&S21P03=A=&S21STR=%D0%A0%D1%83%D1%81%D1%81%D0%B8%D0%BD%D0%BE%D0%B2%D0%B8%D1%87,%20%D0%9C%D0%B0%D1%80%D0%BA) Сетевые операционные системы / В. Олифер, Н. Олифер. — 6-е изд. — СПб.: Питер, 2001. — 544 с.
5. Таненбаум, Э. Современные операционные системы / Э. Таненбаум. Пер. с англ. — Санкт-Петербург : Питер, 2015. — 1120 с. — ISBN: 978-5-4461-1155-8
6. Гультяев, А.К. Help. Разработка справочных систем. Учебный курс / А.К. Гультяев. Санкт-Петербург : Питер, 2004. — 272 с. — ISBN 5-94723-921-3

ПРИЛОЖЕНИЕ А

(обязательное)

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Санкт-Петербургский государственный технологический институт  
(технический университет)»  
СПбГТИ(ТУ)

СОГЛАСОВАНО УТВЕРЖДАЮ

проф., д.т.н. \_\_\_\_\_\_\_\_\_\_\_\_\_Т.Б.Чистякова

\_\_\_\_\_\_\_\_\_\_\_\_\_

**«СИСТЕМА СРАВНЕНИЯ РАЗЛИЧНЫХ АЛГОРИТМОВ ЗАМЕЩЕНИЯ СТРАНИЦ»**

Руководство системного программиста

ЛИСТ УТВЕРЖДЕНИЯ

460.2.030. 00001–01 34 01–ЛУ

Руководитель курсового проекта

Р. В. Макарук

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_2021

Исполнитель:

А.М. Казанцев

“\_\_\_”\_\_\_\_\_\_\_\_\_\_\_2021

2021

Продолжение приложения А

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение   
высшего образования   
«Санкт-Петербургский государственный технологический институт  
(технический университет)»  
СПбГТИ(ТУ)

УТВЕРЖДЕН

460.2.030. 00001–01 34 01–ЛУ

**«СИСТЕМА СРАВНЕНИЯ РАЗЛИЧНЫХ АЛГОРИТМОВ ЗАМЕЩЕНИЯ СТРАНИЦ»**

Руководство системного программиста

460.2.030.00001–01 34 01

Листов 10

2021

Продолжение приложения А

2

460.2.030.00001–01 34 01

**АННОТАЦИЯ**

В данном программном документе приведено руководство системного программиста по настройке и использованию программы «ProgramFromKPPageSortingAlgorithms.exe», предназначенной для мониторинга работы алгоритмов замещения страниц: LRU, NRU, FIFO, “вторая попытка”, “часы”.

В данном программном документе, в разделе «Общие сведения о программе» указаны назначение и функции программы и сведения о технических и программных средствах, обеспечивающих выполнение данной программы, а также требования к персоналу.

В разделе «Структура программы» приведены сведения о структуре программы, ее составных частях, о связях между составными частями и о связях с другими программами.

В данном программном документе, в разделе «Настройка программы» приведено описание действий по настройке программы на условия конкретного применения (настройка на состав технических и программных средств, выбор функций и др.).

В разделе «Проверка программы» приведено описание способов проверки, позволяющих дать общее заключение о работоспособности программы (контрольные примеры, методы прогона, результаты).

В данном программном документе, в разделе «Сообщения системному программисту» указаны тексты сообщений, выдаваемых в ходе выполнения настройки, проверки программы, а также в ходе выполнения программы, описание их содержания и действий, которые необходимо предпринять по этим сообщениям.

Продолжение приложения А

3

460.2.030.00001–01 34 01

СОДЕРЖАНИЕ

Аннотация 2

Содержание 3

1 Общие сведения о программе 4

1.1. Назначение программы 4

1.2. Функции программы 4

1.3. Минимальный состав технических средств 5

1.4. Минимальный состав программных средств 5

1.5. Требования к персоналу (системному программисту) 5

2 Структура программы 6

2.1. Сведения о структуре программы 6

2.2. Сведения о составных частях программы 6

2.3. Сведения о связях между составными частями программы 6

2.4. Сведения о связях с другими программами 6

3 Настройка программы 7

3.1. Настройка на состав технических средств 7

3.2. Настройка на состав программных средств 7

4 Проверка программы 7

4.1. Описание способов проверки 7

4.2. Методы прогона 7

4.2.1. Проверка работоспособности программы 7

4.2.2. Проверка на сообщение об ошибке 7

5 Сообщения системному программисту 8

Продолжение приложения А

4

460.2.030.00001–01 34 01

1. **ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММЕ**

1.1 Назначение программы

В настоящей время очень часто применяется метод виртуальной памяти, позволяющий выполнять программы, требующие больше оперативной памяти, чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем. Для выполняющейся программы данный метод полностью прозрачен и не требует дополнительных усилий со стороны программиста, однако реализация этого метода требует как аппаратной поддержки, так и поддержки со стороны операционной системы.

В большинстве современных операционных систем виртуальная память организуется с помощью страничной адресации. Оперативная память делится на страницы: области памяти фиксированной длины, которые являются минимальной единицей выделяемой памяти.

Процессор преобразует номер виртуальной страницы в адрес соответствующей ей физической страницы при помощи буфера ассоциативной трансляции. Если ему не удалось это сделать, то требуется до заполнения буфера путём обращения к таблице страниц, что может сделать либо сам процессор, либо операционная система. Если страница была выгружена из оперативной памяти, то операционная система подкачивает страницу с жёсткого диска в ходе обработки события Page fault. При запросе на выделение памяти операционная система может «сбросить» на жёсткий диск страницы, к которым давно не было обращений. Критические данные обычно находятся в оперативной памяти.

Чтобы решить какую из страниц можно заменить, а какие лучше оставить в оперативной памяти используются алгоритмы замещения страниц. Данная программа разработана для мониторинга работы и сравнения эффективности различных алгоритмов таких как: LRU, NRU, FIFO, “вторая попытка”, “часы”.

Программа «ProgramFromKPPageSortingAlgorithms.exe» работает под управлением ОС Windows 10.

* 1. Функции программы

Основной функцией программы «ProgramFromKPPageSortingAlgorithms.exe» является реализация и визуализация работы алгоритмов сравнения страниц, для которых программа получает, сохраняет данные путем взаимодействия с файловой системой, проверяя на их корректность, или с помощью ручного ввода пользователя. В случае некорректности входных данных программа выведет сообщение об ошибки с текстом возникшей проблемы.

Продолжение приложения А

5

460.2.030.00001–01 34 01

Основная задача - обеспечить работу выбранных пользователем алгоритмов сортировки.

Данные функции программы ProgramFromKPPageSortingAlgorithms позволяют производить корректировку входных и выходных данных, мониторинг работы алгоритмов (заполнения страничных блоков), а также просматривать графики сравнения различных алгоритмов сортировки.

* 1. Минимальный состав технических средств

Минимальный состав используемых технических средств:

PC на базе процессора с тактовой частотой 1 ГГц или выше;

ОЗУ более 1 Гб;

512 МБ видеопамяти и выше.

1.4. Минимальный состав программных средств

Системные программные средства, используемые программой ProgramFromKPPageSortingAlgorithms.exe, должны быть представлены локализованной версией операционной системы Windows 10.

1.5 Требования к персоналу (системному программисту)

Системный программист должен иметь минимум среднее техническое образование. В перечень задач, выполняемых системным программистом, должны входить:

а) задача поддержания работоспособности технических средств;

б) задача установки (инсталляции) и поддержания работоспособности системных программных средств – операционной системы;

в) задача установки (инсталляции) и поддержания работоспособности программы ProgramFromKPPageSortingAlgorithms.exe.

**2 СТРУКТУРА ПРОГРАММЫ**

2.1 Сведения о структуре программы

Программа «ProgramFromKPPageSortingAlgorithms.exe» состоит из одной запускаемой формы.

Продолжение приложения А

6

460.2.030.00001–01 34 01

2.2 Сведения о составных частях программы

Программа «ProgramFromKPPageSortingAlgorithms.exe» состоит из одной запускаемой формы, а также отдельно запускаемой справочной системой.

2.3 Сведения о связях между составными частями программы

Программа «ProgramFromKPPageSortingAlgorithms.exe» способна запускать справочную систему и форму с графиком при нажатии определенной кнопки интерфейса.

2.4 Сведения о связях с другими программами

Программа «ProgramFromKPPageSortingAlgorithms.exe» в ходе своей работы не требует связи с другими программами.

**3 НАСТРОЙКА ПРОГРАММЫ**

3.1 Настройка на состав технических средств

Программа «ProgramFromKPPageSortingAlgorithms.exe» не требует каких-либо настроек на состав технических средств.

3.2 Настройка на состав программных средств

Программа «ProgramFromKPPageSortingAlgorithms.exe» устанавливает необходимые для работы компоненты при установке программного продукта при помощи инсталляционного пакета.

Продолжение приложения А

7

460.2.030.00001–01 34 01

**4 ПРОВЕРКА ПРОГРАММЫ**

4.1 Описание способов проверки

Работоспособность программы проверяется следующим способом. Необходимо взять уже проверенный набор номеров страниц и сравнить результаты работы программы с уже известными (сложнее проверить алгоритм NRU). Для проверки работы с файлами необходимо открыть данный файл внутри программного комплекса. В случае корректности входных данных будет построена таблица прерываний, станет доступна возможность построения графика зависимостей и будет построен соответствующий график. В случае некорректности входных данные внутри указанного файла программа «ProgramFromKPPageSortingAlgorithms.exe» выдаст сообщение об ошибке, описанное в разделе 5. «Сообщения системному программисту» данного программного руководства.

4.2 Методы прогона

4.2.1 Проверка работоспособности программы

Проверьте наличие читаемого .txt файла, содержащего корректные исходные данные, в соответствии с требованиями корректности.

Проверьте наличие файла HelpOS.chm для корректного отображения файловой системы

При работе с большим набором страниц проверьте наличие достаточного количества свободной оперативной памяти и процессорного времени

4.2.2 Проверка на сообщение об ошибке

В случае некорректности входных данных или неверной, с точки зрения формата, корректировки данных внутри таблицы значений программа «ProgramFromKPPageSortingAlgorithms.exe» должна выдать сообщение об ошибке, описанное в разделе 5. «Сообщения системному программисту» данного программного руководства.

Продолжение приложения А

8

460.2.030.00001–01 34 01

**5 СООБЩЕНИЯ СИСТЕМНОМУ ПРОГРАММИСТУ**

Программа «ProgramFromKPPageSortingAlgorithms.exe» выдает: сообщения об ошибке, показанные на рисунках 1, 2, 3, 4, 5 и сообщения о завершении рисунки 6 и 7.

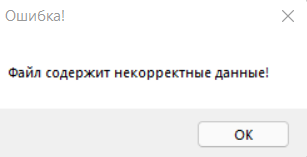


Рисунок 1 – Сообщение системному программисту

ПРИЧИНА. Выбранный файл с исходными данными содержит некорректные для обработки программой данные.

ДЕЙСТВИЯ ПРОГРАММЫ. Программа «ProgramFromKPPageSortingAlgorithms.exe» не производит регистрацию введенных данных, работа выбранных алгоритмов сортировке не начинается.

ДЕЙСТВИЯ ПРОГРАМИСТА. Проверить по указанному пути правильность указанных в файле данных.

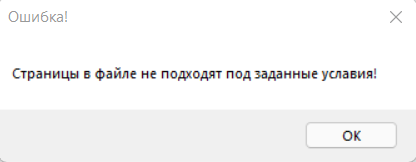


Рисунок 2 – Сообщение системному программисту

ПРИЧИНА. Выбранный файл с исходными данными содержит данные, которые могут выходить за пределы заданных программистов условий.

ДЕЙСТВИЯ ПРОГРАММЫ. Программа «ProgramFromKPPageSortingAlgorithms.exe» не производит регистрацию введенных данных, работа выбранных алгоритмов сортировке не начинается.

ДЕЙСТВИЯ ПРОГРАМИСТА. Проверить по указанному пути правильность указанных в файле данных или изменить параметры страничного разнообразия в программе.

Продолжение приложения А

9

460.2.030.00001–01 34 01

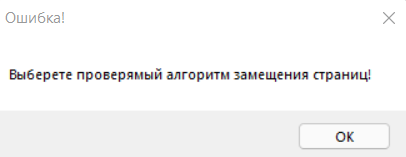


Рисунок 3 – Сообщение системному программисту

ПРИЧИНА. В настройках программы не был выбран ни один из алгоритмов.

ДЕЙСТВИЯ ПРОГРАММЫ. Программа «ProgramFromKPPageSortingAlgorithms.exe» не производит регистрацию введенных данных, работа выбранных алгоритмов сортировке не начинается.

ДЕЙСТВИЯ ПРОГРАМИСТА. Проверить по указанному пути правильность указанных в файле данных или изменить параметры страничного разнообразия в программе.

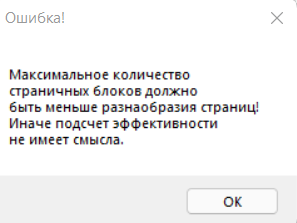


Рисунок 4 – Сообщение системному программисту

ПРИЧИНА. В настройках программы максимальное число страничных блоков должно быть меньше разнообразия страниц.

ДЕЙСТВИЯ ПРОГРАММЫ. Программа «ProgramFromKPPageSortingAlgorithms.exe» не производит регистрации изменения настроек.

ДЕЙСТВИЯ ПРОГРАМИСТА. Увеличить число разнообразия страниц или уменьшить максимально возможное количество страничных блоков, и только после этого изменять второе условие.

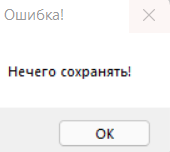


Рисунок 5 – Сообщение системному программисту

ПРИЧИНА. Последовательность набора страниц ещё не была введена пользователем программы.

Продолжение приложения А

10

460.2.030.00001–01 34 01

ДЕЙСТВИЯ ПРОГРАММЫ. Программа «ProgramFromKPPageSortingAlgorithms.exe» не производит сохранения данных в файл.

ДЕЙСТВИЯ ПРОГРАМИСТА. Ввести нужную для сохранения последовательность номеров страниц вручную.

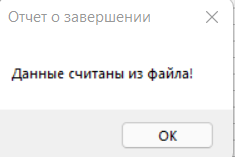


Рисунок 6 – Сообщение системному программисту

ПРИЧИНА. Данные из файла были считаны и проверены на похождения всем заданным условиям.

ДЕЙСТВИЯ ПРОГРАММЫ. Программа «ProgramFromKPPageSortingAlgorithms.exe» производит регистрацию введенных данных, работа выбранных алгоритмов сортировки начинается.

ДЕЙСТВИЯ ПРОГРАМИСТА. Продолжить работу с программой.

Рисунок 7 – Сообщение системному программисту

ПРИЧИНА. Данные были сохранены в файл в корректной для использования в будущем форме.

ДЕЙСТВИЯ ПРОГРАММЫ. Программа «ProgramFromKPPageSortingAlgorithms.exe» ожидает действий пользователя.

ДЕЙСТВИЯ ПРОГРАМИСТА. Продолжить работу с программой.