|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ \_\_\_\_\_\_\_\_Информатика и системы управления\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КАФЕДРА \_\_\_\_\_\_\_Программное обеспечение ЭВМ и информационные технологии\_\_\_\_\_\_\_\_

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ ПО ПРЕДМЕТУ "ТИПЫ И СТРУКТУРЫ ДАННЫХ" №4**

Студент \_Золотухин Алексей Вячеславович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*фамилия, имя, отчество*

Группа \_\_\_\_\_\_ИУ7-34Б\_\_\_\_\_\_\_\_\_

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_Золотухин А. В.\_\_\_\_

*подпись, дата фамилия, и.о.*

Принял \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_Силантьева А. В. \_

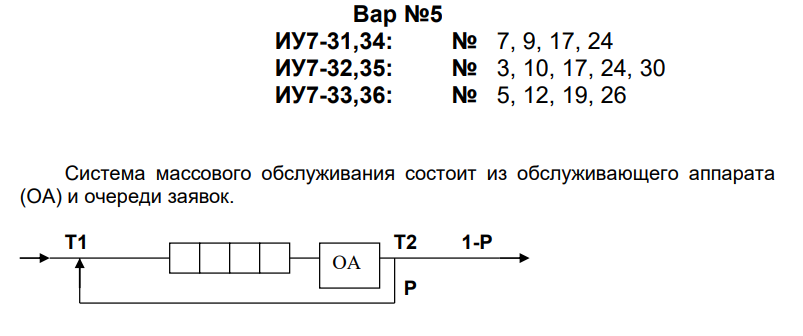
*подпись, дата фамилия, и.о.*

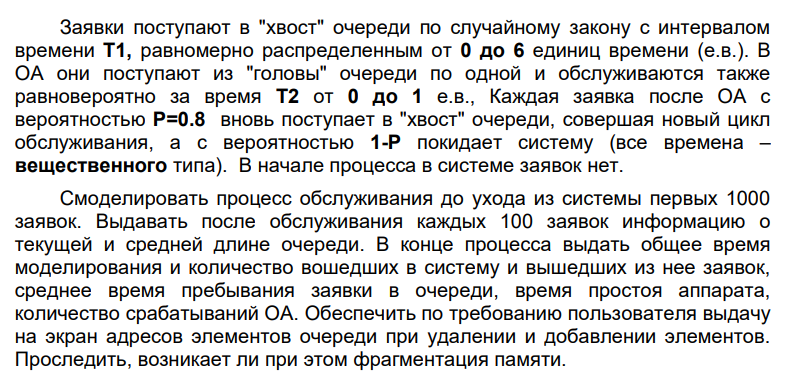
Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Москва, 2021**

**Цель работы:** отработка навыков работы с типом данных «очередь», представленным в виде одномерного массива и односвязного линейного списка. Сравнительный анализ реализации алгоритмов включения и исключения элементов из очереди при использовании двух указанных структур данных. Оценка эффективности программы (при различной реализации) по времени и по используемому объему памяти.

**Условия задачи:**



****

**Входные данные**

На входе программа получает номер процесса, интересующего пользователя:

1 - 3 — демонстрация работы очереди-массива

4 — решение поставленной задачи (обработка тысячи элементов выбранным дальше способом при вводимых t1, t2)

5 - 7 — демонстрация работы очереди-списка.

**Выходные данные**

Программа выводит результат выполнения выбранного пункта.

**Аварийные ситуации**

В случае переполнения очереди будет выведено сообщение об ошибке.

При попытке достать элемент из пустой очереди будет выведено сообщение об ошибке.

Реализованная программа представляет собой консольное приложение, предоставляющее пользователю возможность изучить работу структуры очередь, а также различные способы её представления.

**Структуры данных**

В программе было использовано 2 реализации очереди — с помощью массива и с помощью списка.

Структура очереди на массиве содержит в себе массив элементов, индекс начального элемента, индекс конечного, длину массива.

Структура очереди на списке содержит в себе указатель на первый и на последний элементы. Каждый элемент содержит значение, а также указатель на следующий элемент (next).

***Реализация на массиве:***

#define MAX\_N\_A\_I 5

typedef struct

{

    int queue[MAX\_N\_A\_I];

    int n;

    int pout;

    int pin;

} queue\_a\_i\_t;

***Реализация на списке:***

typedef struct node\_t

{

    int val;

    struct node\_t \*next;

} node\_t;

typedef struct

{

    node\_t \*head;

    node\_t \*tail;

} queue\_l\_t;

Также в программе существуют две вспомогательные структуры для решения задачи.

Структура «заявка»

typedef struct

{

    double stime;

    double wtime;

    double etime;

} request\_t;

Очередь «заявок» на массиве

#define MAX\_N\_A\_R 10000

typedef struct

{

    request\_t queue[MAX\_N\_A\_R];

    int n;

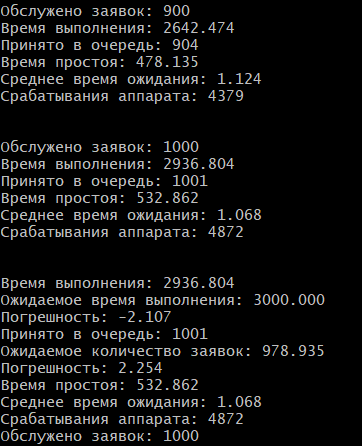
    int pout;

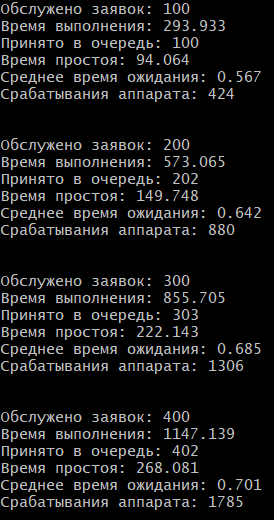
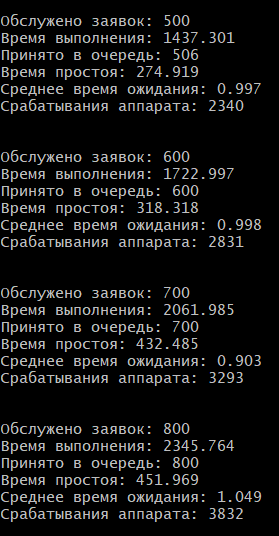
    int pin;

} queue\_a\_r\_t;

Очередь заявок на списке

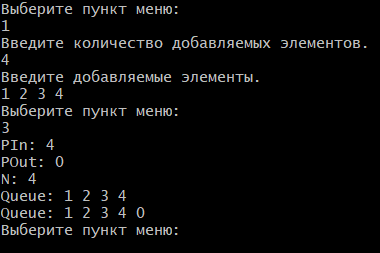
**Тесты**

****1) Обработка тысячи элементов первого типа на массиве с временами прихода и обработки t1, t2, расположенными от 0 до 6, от 0 до 1 единиц времени соответственно.

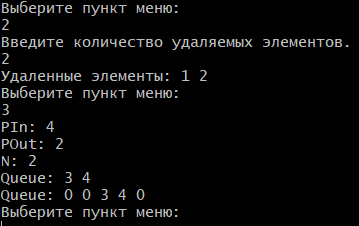


**Демонстрация работы массива:**

Добавление:

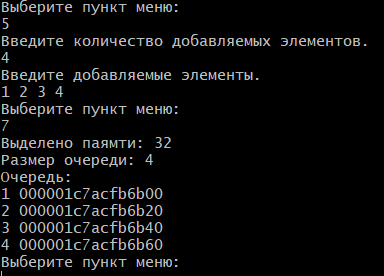


Удаление:

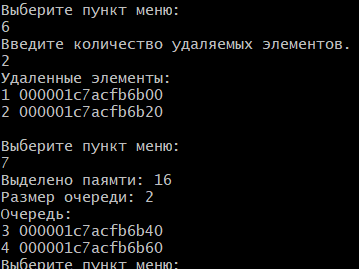


**Демонстрация работы списка:**

Добавление:

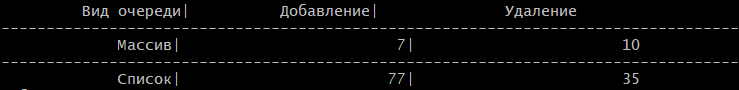


Удаление:

****

**Заключение**

**Сравнение структур но времени**

****

**Сравнение стуктур по памяти (int)**

|  |  |  |  |
| --- | --- | --- | --- |
| Структура | 10 элементов | 100 элементов | 1000 элементов |
| Очередь-массив | 52 | 412 | 4012 |
| Очередь-список | 176 | 1616 | 16016 |

В результате работы были реализованы функции работы с очередью, а также были сделаны две реализации очереди - на массиве и на списке.

При работе программы фрагментация памяти почти не происходит.

С точки зрения временных затрат и затрат памяти очередь, реализованная массивом выгодней. Однако не стоит забывать, что очередь, реализованная на списке, позволяет хранить любое (ограниченное оперативной памятью) количество элементов. Таким образом, выбор структуры зависит от поставленной задачи.

**Ответы на контрольные вопросы**

***1.***

***2. Что такое очередь?***

Очередь – это последовательный список переменной длины, включение элементов в который идет с одной стороны (с «хвоста»), а исключение – с другой стороны (с «головы»). Принцип работы очереди: первым пришел – первым вышел, т. е. First In – First Out (FIFO).

***2. Каким образом, и какой объем памяти выделяется под хранение очереди при***

***различной ее реализации?***

При моделировании простейшей линейной очереди на основе одномерного массива выделяется последовательная область памяти из m мест по L байт, где L – размер поля данных для одного элемента размещаемого типа. В каждый текущий момент времени выделенная память может быть вся свободна, занята частично или занята полностью.

Реализация очереди в виде линейного списка. Большинство проблем, возникающих при реализации очереди в виде массива, устраняется при реализации очереди на основе односвязного линейного списка, каждый элемент которого содержит информационное поле и поле с указателем «вперед» (на следующий элемент). В этом случае в статической памяти можно либо хранить адрес начала и конца очереди, либо – адрес начала очереди и количество элементов.

***3. Что происходит с элементами очереди при ее просмотре?***

При классической реализации — он удаляется из очереди.

***4. Каким образом эффективнее реализовывать очередь. От чего это зависит?***

От поставленных задач и требований. Массив быстрее и экономичней. Но его размер ограничен.

***5. В каком случае лучше реализовать очередь посредством указателей, а в каком – массивом?***

Если максимальное количество элементов неизвестно или велико — списком. В остальных случаях разумней использовать массив.

***6. Что такое фрагментация памяти?***

Фрагментация это выделение не последовательных адресов в памяти

***7. На что необходимо обратить внимание при тестировании программы?***

При тестировании программы необходимо:

-проверить правильность работы программы при различном заполнении очередей, т.е., когда время моделирования определяется временем обработки заявок и когда определяется временем прихода заявок;

-отследить переполнение очереди, если очередь в программе ограничена.

***8. Каким образом физически выделяется и освобождается память при динамических запросах?***

Память представляет собой бинарную кучу с признаком — занятость-незанятость ячейки. Динамический запрос меняет признак ячейки.