Алимова Ольга Викторовна

Методические указания по выполнению практической работы на тему "Очередь и ее использование"

учебно-методическое пособие

для студентов прикладных
математических специальностей

Санкт-Петербург

2024

Оглавление

Очередь, варианты реализации очереди.

Очередь (queue) — пример абстрактного типа данных с правилом доступа к элементам «первый пришёл — первый вышел» (FIFO, англ. first in, first out).

У очереди имеется голова (англ. head) и хвост (англ. tail). Когда элемент ставится в очередь, он занимает место в её хвосте. Из очереди всегда выводится элемент, который находится в ее голове.

Для реализации очереди достаточно реализовать следующие операции:

- *empty* — проверка очереди на наличие в ней элементов,
- *push* (запись в очередь) — операция вставки нового элемента в хвост очереди,
- *pop* (снятие с очереди) — операция удаления элемента из головы очереди,
- *size* — операция получения количества элементов в очереди.

Можно выделить два способа программной реализации очереди. Первый из них основан на базе массива, а второй на базе указателей (связного списка). Первый способ – статический, т. к. очередь представляется в виде простого статического массива, второй – динамический.

Реализация очереди с помощью массива позволяет организовать и впоследствии обрабатывать очередь, имеющую фиксированный размер. Переменные `head` и `tail` — индексы первого и последнего элементов очереди.

Ниже представлен пример реализации функций для очереди целых чисел на циклическом массиве.

```
struct queue {
    int value[n];
    int head = 0, tail = 0, count = 0;
};

void push(queue *q, int x) {
    if (q->count < n) {
        q->value[(q->tail)++] = x;
        q->tail %= n;
        (q->count)++;
    }
}

int pop(queue *q) {
    if (q->count) {
        int x = q->value[q->head++];
        q->head  %=  n;
        (q->count)--;
        return x;
    }
    return INT_MIN;
}

int size(queue *q) {
    return q->count;
}

bool empty(queue *q) {
    return q->count == 0;
}
```

Очевидным плюсом такой реализации очереди является простота разработки, к минусам стоит отнести ограничение на количество элементов находящихся в очереди одновременно.

Реализация очереди на базе указателей позволяет организовать и впоследствии обрабатывать очередь с произвольным количеством элементов. К минусам стоит отнести гораздо более сильную фрагментацию памяти и, возможно, более медленную последовательную итерацию по такой очереди, нежели итерация по очереди, реализованной на массиве. Переменные `head` и `tail` — указатели на первый и последний элементы двусвязного списка.

Пример реализации очереди на базе двунаправленного связного списка:

```
struct link {
    int value;
    link *next,
         *prev;
};

struct queue {
    link* head = NULL;
    link* tail = NULL;
    int count = 0;
};

void push(queue *q, int x) {
    link *tmp = (link*)malloc(sizeof(link));
    tmp->value = x;
    tmp->next = NULL;
    tmp->prev = q->tail;
    if (!q->tail)
        q->head = tmp;
    else
        q->tail->next= tmp;
    q->tail = tmp;
    q->count++;
}

int pop(queue *q) {
    if (q->count > 0) {
        link* tmp = q->head;
        q->head = q->head->next;
        if (q->head)
            q->head->prev = NULL;
        else
            q->tail = q->head;
        q->count--;
        int x = tmp->value;
        free(tmp);
        return x;
    }
    return INT_MIN;
}
```

Код функций `size` и `empty` отличаться не будет.

Следует отметить, что представленный выше пример избыточен. Для очереди с правилом доступа к элементам «первый пришёл — первый вышел» достаточно использовать однозвязный список. Двусвязный список больше подходит для абстрактного типа данных Дек[1].

Постановка задачи.

С системами массового обслуживания мы встречаемся повседневно. Любому из нас приходилось когда-то ждать обслуживания в очереди (например, в магазине, на авто заправке, в библиотеке, кафе и т. д.). Более того, любое производство можно представить как последовательность систем обслуживания. К типичным системам обслуживания относят также ремонтные и медицинские службы, транспортные системы, аэропорты, вокзалы и другие. Особое значение приобрели такие системы при изучении процессов в информатике. Это, прежде всего, компьютерные системы, сети передачи информации, ОС, базы и банки данных. Системы обслуживания играют значительную роль в повседневной жизни.

Система массового обслуживания состоит из обслуживающих аппаратов (ОА) и очередей запросов одного или двух типов, различающихся временем поступления и временем обработки.

Запросы поступают в «хвост» очереди с различными интервалами времени распределенными по случайному закону от минимального значения (иногда от нуля) до максимального значения количества единиц времени (в зависимости от варианта задания). В ОА запросы поступают из «головы» очереди по одной и обслуживаются за указанное в задании время. Время обработки каждого запроса – случайная величина, распределенная равновероятно от минимального до максимального значений (все времена – вещественного типа).

Требуется смоделировать процесс обслуживания первых 1000 запросов первого типа, выдавая после обслуживания каждых 100 запросов и в конце процесса информацию в соответствии с заданием.

Очередь необходимо представить в виде двухсвязного списка. Все операции должны быть оформлены подпрограммами.

Предусмотреть возможность демонстрации работы программы в двух вариантах:

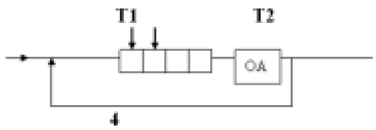
- для 1000 запросов первого типа в соответствии с заданием;
- для 10 – 15 запросов первого типа с отображением состояния очереди (очередей) на каждой итерации цикла.

Сделать оценку полученных результатов.

[1] Дек (deque — double ended queue) — структура данных, представляющая из себя список элементов, в которой добавление новых элементов и удаление существующих производится с обоих концов. Эта структура поддерживает как FIFO, так и LIFO, поэтому на ней можно реализовать как стек, так и очередь.

Варианты заданий.

Задание 1. Система массового обслуживания состоит из обслуживающего аппарата (ОА) и очереди запросов.



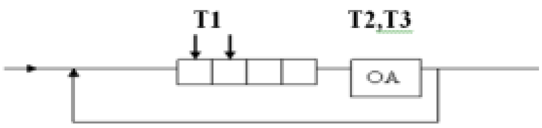
Запросы поступают в "хвост" очереди по случайному закону с интервалом времени **T1,** равномерно распределенным от **a1 до a2** единиц времени (е.в.), обслуживаются запросы в течении времени **T2** равномерно распределенным по случайному закону в интервале от **b1 до b2** е. в.

Каждый запрос после обслуживания в аппарате возвращается в «хвост» очереди для повторной обработки и после 4-х циклов обслуживания покидает систему.

После выхода из системы обработки каждого 100-го запроса выводится текущее состояние очереди, а именно: количество вошедших запросов и текущая длина очереди, а так же для запроса с максимальным временем ожидания вывести его номер и цикл обслуживания, вывести максимальное и среднее время пребывание запроса в очереди.

Система завершает свою работу по выходу из нее 1000 запросов. На экране при этом отображается общее время моделирования, время простоя ОА, количество обработанных запросов, а так же количество вошедших и вышедших из очереди запросов.

Задание 2. Система массового обслуживания состоит из обслуживающего аппарата (ОА) и очереди запросов двух типов .



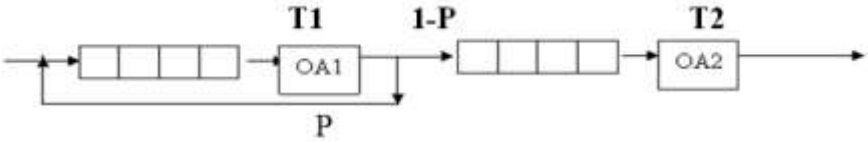
Запросы первого типа поступают в "хвост" очереди по случайному закону с интервалом времени **T1,** равномерно распределенным от **a1 до a2** единиц времени (е.в.). В ОА они поступают из "головы" очереди по одной и обслуживаются также равновероятно за время **T2** от **b1 до b2** е.в., после чего покидают систему.

Единственный запрос второго типа постоянно обращается в системе, обслуживаясь в ОА равновероятно за время **T3** от **c1 до c2** е.в. и возвращаясь в очередь не далее 4-й позиции от "головы". В начале процесса запрос второго типа входит в ОА, оставляя пустую очередь.

Требуется смоделировать процесс обслуживания первой партии из 1000 запросов **первого типа**, выдавая после обслуживания каждых 100 запросов первого типа информацию о текущей длине очереди, количестве вошедших и вышедших запросов, а так же для запроса с максимальным временем ожидания вывести его номер и его тип, вывести максимальное и среднее время пребывание запроса в очереди.

Система завершает свою работу по выходу из нее 1000 запросов первого типа. На экране при этом отображается общее время моделирования, время простоя ОА, количество обработанных запросов первого типа, количество обращений запроса второго типа, а так же количество вошедших и вышедших из очереди запросов обоих типов.

Задание 3. Система массового обслуживания состоит из двух обслуживающих аппаратов (ОА1 и ОА2) и двух очередей запросов. Всего в системе обращается 1000 запросов.



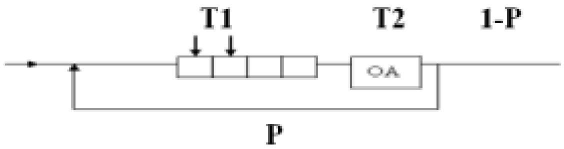
Запросы поступают в "хвост" каждой очереди; в ОА запросы поступают из "головы" очереди по одному и обслуживаются за интервалы времени **Т1** и **Т2,** равномерно распределенные по случайному закону от **a1** до **a2** и от **b1** до **b2** единиц времени (е.в.) соответственно.

Каждый запрос после обработки в ОА1 с вероятностью **Р = 0.7** вновь поступает в "хвост" первой очереди, совершая новый цикл обслуживания, а с вероятностью **1 – Р** входит во вторую очередь. В начале процесса все запросы уже находятся в первой очереди.

Требуется смоделировать процесс обслуживания до выхода из ОА2 партии из 1000 запросов, выдавая после обслуживания каждых 100 запросов информацию о количестве принятых запросов и текущей длине каждой очереди. Для запроса с максимальным временем ожидания в каждой очереди вывести его номер, максимальное и среднее время пребывание запросов в каждой очереди.

Система завершает свою работу по выходу из нее 1000 запросов. На экране при этом отображается общее время моделирования, время простоя ОА1 и ОА2, количество обработанных запросов в каждом аппарате, а так же количество вошедших и вышедших из каждой очереди запросов.

Задание 4. Система массового обслуживания состоит из обслуживающего аппарата (ОА) и очереди запросов.

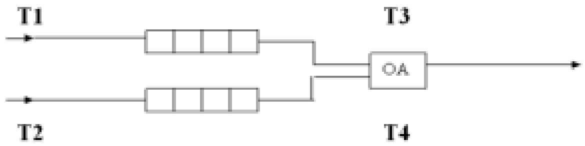


Запросы поступают в "хвост" очереди с интервалом времени **Т1,** равномерно распределенным по случайному закону от **a1 до a2** единиц времени (е.в.). В ОА запросы поступают из "головы" очереди по одной и обслуживаются также равновероятно за время **Т2** от **b1 до b2** е.в., Каждый запрос после ОА с вероятностью **Р = 0.8** вновь поступает в "хвост" очереди, совершая новый цикл обслуживания, а с вероятностью **1–Р** покидает систему. В начале процесса в системе запросов нет.

Требуется смоделировать процесс обслуживания партии из 1000 запросов, выдавая после обслуживания каждых 100 запросов информацию о количестве принятых запросов и текущей длине очереди. Для запроса с максимальным временем ожидания вывести его номер, максимальное и среднее время пребывания запроса в очереди.

Система завершает свою работу по выходу из нее 1000 запросов. На экране при этом отображается общее время моделирования, время простоя ОА, количество обработанных запросов в аппарате, а так же количество вошедших и вышедших из очереди запросов.

Задание 5. Система массового обслуживания состоит из обслуживающего аппарата (ОА) и двух очередей запросов двух типов.



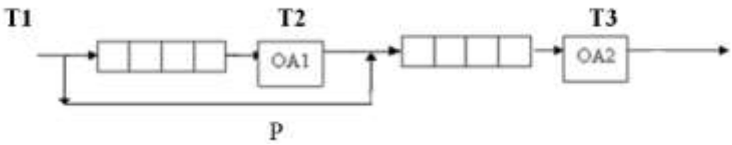
Запросы первого и второго типов поступают в "хвосты" своих очередей с интервалами времени **T1** и **T2**, равномерно распределенными от **a1** до **a2** и от **b1** до **b2** единиц времени (е.в.) соответственно. В ОА запросы поступают из "головы" очереди по одному и обслуживаются за времена **T3** и **T4**, также равновероятно распределенные от **c1** до **c2** е.в. и от **d1** до **d2** е.в. соответственно, после чего покидают систему. В начале процесса в системе запросов нет.

Запрос второго типа может войти в ОА, если в системе нет запросов первого типа. Если в момент обслуживания запроса второго типа в пустую очередь входит запрос первого типа, то он ждет первого освобождения ОА и далее поступает на обслуживание (система с относительным приоритетом).

Требуется смоделировать процесс обслуживания первой партии из 1000 запросов **первого типа**, выдавая после обслуживания каждых 100 запросов обоих типов информацию о количестве принятых запросов и текущей длине каждой очереди. Для запроса с максимальным временем ожидания в каждой очереди вывести его номер, максимальное и среднее время пребывания запросов в каждой очереди.

Система завершает свою работу по выходу из нее 1000 запросов первого типа. На экране при этом отображается общее время моделирования, время простоя ОА, количество обработанных запросов первого и второго типа, а так же количество вошедших и вышедших из каждой очереди запросов обоих типов.

Задание 6. Система массового обслуживания состоит из двух обслуживающих аппаратов (ОА1 и ОА2) и двух очередей запросов одного типа. Всего в системе обращается 1000 запросов.



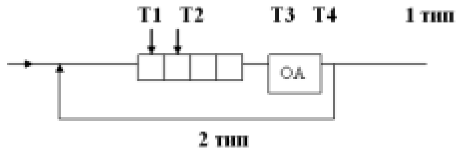
Запросы поступают в "хвост" каждой очереди с интервалом времени **T1,** равномерно распределенным по случайному закону от **a1 до a2** единиц времени(е.в.); в ОА запросы поступают из "головы" очереди по одному и обслуживаются за интервалы времени **T2** и **T3,** равномерно распределенные по случайному закону от **b1** до **b2** и от **c1** до **c2** е.в. соответственно.

Каждый запрос с вероятностью **P = 0.4** поступает в "хвост" второй очереди, пропуская цикл обслуживания в ОА1, а с вероятностью **1 – P** входит в первую очередь. В начале процесса обе очереди пустые.

Требуется смоделировать процесс обслуживания до выхода из ОА2 партии из 1000 заапросов, выдавая после обслуживания каждых 100 запросов в ОА2 информацию о количестве принятых запросов в ОА1 и ОА2, текущей длине каждой очереди. Для запроса с максимальным временем ожидания в каждой очереди вывести его номер, максимальное и среднее время пребывание запроса в каждой очереди.

Система завершает свою работу по выходу из нее 1000 запросов. На экране при этом отображается общее время моделирования, время простоя ОА1 и ОА2, количество обработанных запросов в каждом аппарате, а так же количество вошедших и вышедших из каждой очереди запросов.

Задание 7. Система массового обслуживания состоит из обслуживающего аппарата (ОА) и очереди запросов двух типов.

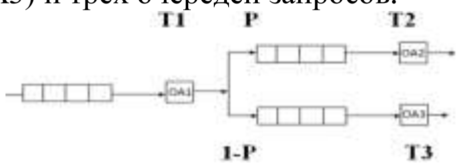


Запросы двух типов поступают в "хвост" очереди с интервалом времени **T1** и **T2,** равномерно распределенными по случайному закону от **a1** до **a2** и от **b1** до **b2** единиц времени (е.в.). В ОА запросы поступают из "головы" очереди по одному и обслуживаются за время **T3** (1 типа) и **T4** (2 типа) распределенными равновероятно от **c1** до **c2** и от **d1** до **d2** е.в., запрос второго типа после ОА вновь поступает в "хвост" очереди, совершая новый цикл обслуживания, а запрос первого типа покидает систему. В начале процесса в системе запросов нет.

Требуется смоделировать процесс обслуживания до выхода из ОА партии из 1000 запросов первого типа, выдавая после обслуживания каждых 100 запросов первого типа информацию о количестве принятых запросов каждого типа, текущей длине очереди. Для запроса с максимальным временем ожидания в очереди вывести его номер, максимальное и среднее время пребывание запроса в очереди.

Система завершает свою работу по выходу из нее 1000 запросов первого типа. На экране при этом отображается общее время моделирования, время простоя аппарата, количество обработанных запросов каждого типа, а так же количество вошедших и вышедших из очереди запросов.

Задание 8. Система массового обслуживания состоит из обслуживающих аппаратов (ОА1, ОА2 и ОА3) и трех очередей запросов.



В начале процесса все запросы одного типа находятся в первой очереди. В ОА1 запросы поступают из "головы" очереди по одному и обслуживаются за время **T1** равномерно распределенное по случайному закону от **a1** до **a2** единиц времени (е.в.).

Каждый запрос с вероятностью **P = 0.7** поступает в "хвост" второй очереди, попадая на обслуживание в ОА2, а с вероятностью **1 – P** поступает в "хвост" третьей очереди, попадая на обслуживание в ОА3. В начале процесса вторая и третья очереди пустые.

В ОА2 и ОА3 запросы поступают из "головы" своей очереди по одной и обслуживаются за время **T2** и **T3** равновероятно распределенные от **b1** до **b2** и от **c1** до **c2** е.в.

Требуется смоделировать процесс обслуживания партии из 1000 запросов, выдавая после обслуживания каждых 100 запросов (суммарно в ОА2 и ОА3) информацию о количестве принятых запросов для каждого ОА и текущей длине каждой очереди. Для запроса с максимальным временем ожидания в каждой очереди вывести ее номер, максимальное и среднее время пребывание запросов в очереди.

Система завершает свою работу по выходу из нее 1000 запросов (суммарно из ОА2 и ОА3). На экране при этом отображается общее время моделирования, время простоя каждого аппарата, количество обработанных запросов в каждом аппарате, а так же количество вошедших и вышедших из каждой очереди запросов.

Указания к выполнению работы.

Каждый запрос характеризуется номером, типом/счетчиком/вероятностью (зависит от задания), временем входа в очередь, временем выхода из очереди и временем обработки в ОА.

Длительности обработки запросов и интервалы между их поступлением (единицы времени – е.в.) – случайные равномерно распределенные числа *вещественного типа* в указанном диапазоне (например, от `t1` до `t2`). Для получения случайной величины в указанном диапазоне значений можно использовать генератор случайных чисел, возвращающий значение от 0 до 1:

```
double t = (double) rand() / RAND_MAX;
```

Используя линейное преобразование можно получить необходимое время.

```
time = (t2 – t1) * t + t1
```

В один и тот же момент времени один запрос может прийти в очередь, а другой – начать обрабатываться или выйти из системы.

Допустим, пришел второй запрос, а в этот же момент первый запрос закончил обработку и вышел из системы (см. рисунок 1). Процессы обработки одного запроса и прихода другого запроса идут одновременно, т.е. протекают во времени параллельно, а не последовательно.

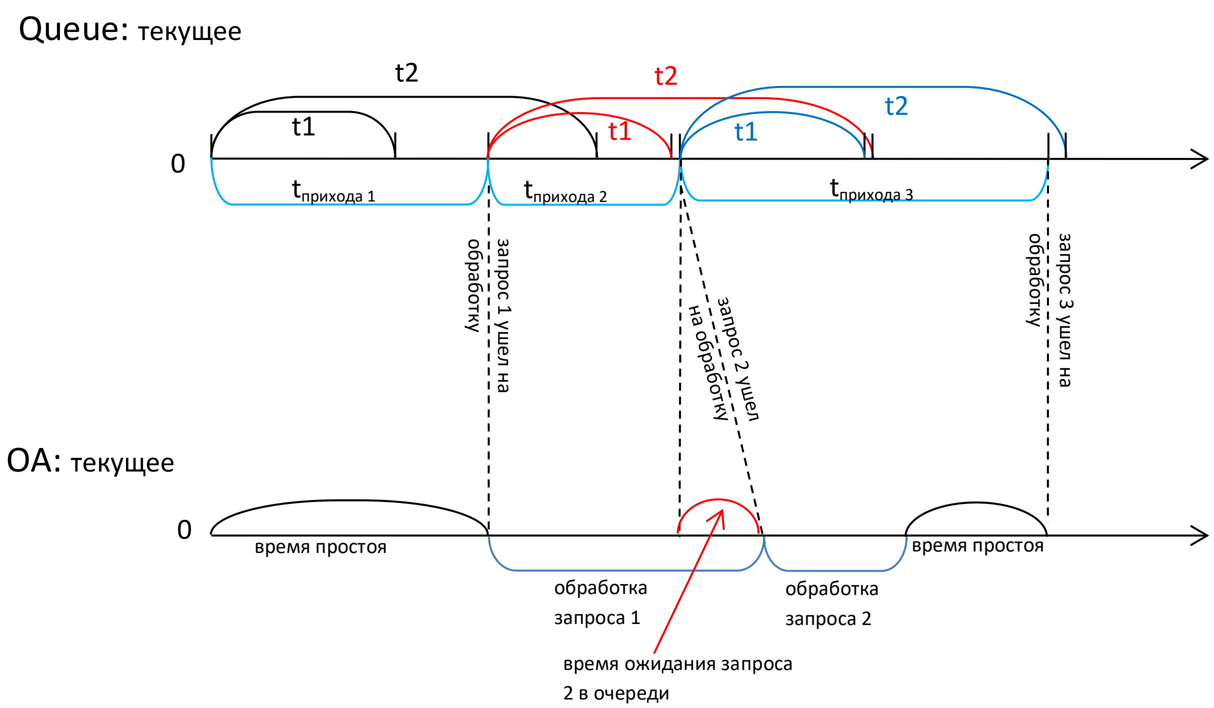


Рисунок 1 – Параллельные процессы поступления и обработки запросов

При возврате запросов после ОА в очередь время прихода нового запроса отсчитывается от времени прихода предыдущего запроса. На рисунке 2 запрос 1 и запрос 2 успеют вернуться в очередь раньше, чем поступит запрос 3. Запрос 3 попадет на обработку только после повторной обработки запросов 1 и 2.

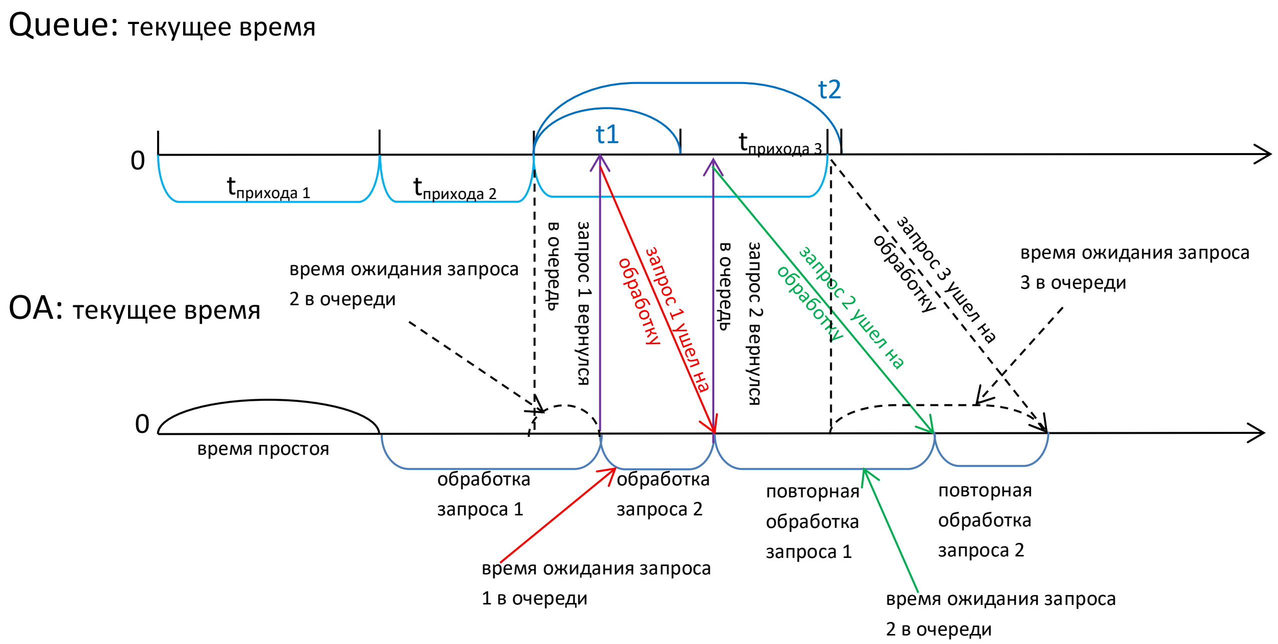


Рисунок 2 – Параллельные процессы при возврате запросов в очередь после обработки в ОА.

Оценивание результатов работы программы.

Имея время прихода первого запроса, допустим, $t_{прихода1}$, можно определить время прихода следующего запроса как $t_{прихода2} = t_{прихода1} + time$. Следовательно, время прихода 1000 запросов будет равно:

$$t_{прихода} = \sum_{i=1}^{1000} t_{прихода\_i}$$

Среднее время прихода или обработки запросов можно подсчитать как среднее арифметическое временного диапазона (например, если t1 = 1, а t2 = 4, то среднее время будет равно $t_{ср} = (1 + 4) / 2 = 2.5$).

Так как используются случайные величины, то точные результаты получить нельзя, но порядок значений $t_{прихода}$ и $1000 * t_{ср}$ должен быть одинаковым.

Аналогичным образом рассчитывается время обработки запросов.

При оценивании работы программы по времени поступления запросов ожидаемое время моделирования равно среднему интервалу между приходом запросов в систему, умноженному на количество вошедших запросов. Если есть две очереди, то проверка ведется по каждой из очередей.

При оценивании работы программы по времени выхода запросов из системы ожидаемое время обработки в ОА должно быть равно среднему времени обработки запроса, умноженному на количество обработанных запросов. Если есть две очереди и один аппарат, то это будет сумма времен обслуживаний запросов каждого типа.

Если среднее время обработки запросов больше среднего интервала между их поступлением, то очередь будет расти и время моделирования будет определяться временем обработки запросов. При этом количество вошедших запросов должно быть равно времени моделирования, деленному на средний интервал между приходом запросов.

Если средний интервал между поступлением запросов больше или равен среднему времени их обработки, то длина очереди стабилизируется (может стремиться к нулю, единице или к какой-то другой величине) и время моделирования будет определяться временем поступления запросов.

При этом ОА может работать с простоем, если время окончания обработки запроса будет меньше, чем время прихода очередного запроса. Тогда время простоя будет определяться разницей этих времен, т. е. $t_{простоя} = t_{обработки} - t_{прихода.}$ Общее время простоя ОА будет равно сумме простоя перед обслуживанием каждого запроса, если простой имел место.

Если есть одна очередь и два аппарата, соединенных последовательно, то время моделирования будет определяться временем обработки в наиболее загруженном аппарате.

Необходимо привести тестирование программы, при тестировании необходимо проверить правильность работы программы при различном заполнении очередей, т.е., когда время моделирования определяется временем обработки запросов и когда определяется временем поступления запросов.

Примеры оценивания результатов работы.

Задание 1: a1=0, a2=10, b1=0, b2=2

Время обработки запроса лежит в интервале от 0 до 2 е.в., значит среднее значение времени обработки одного запроса 1 е.в. Так как каждый запрос обрабатывается 4 раза до выхода из системы, то система обработает 4000 запросов и общее время обработки запросов будет 4000 е.в.

Для поступления 1000 запросов, если каждый приходит в среднем за 5 е.в., потребуется 5000 е.в. Следовательно, ОА работает с простоем и время моделирования будет определяться временем поступления запросов, т.е. оно должно быть равно 5000 е.в. Время простоя будет равно разнице между временем обработки запросов и временем их обслуживания: 5000 – 4000 = 1000 е.в.

Задание 2: a1=0, a2=5, b1=0, b2=4, c1=0, c2=4

Для прихода 1000 запросов, если каждый приходит в среднем за 2.5 е.в., потребуется 2500 е.в.

Время обработки запроса лежит в интервале от 0 до 4 е.в., значит среднее значение времени обработки одного запроса 2 е.в., а общее время обработки 1000 запросов будет 1000 * 2 = 2000 е.в.

Запрос 2 типа обрабатывается в среднем 2 е.в., он может встать в очередь не менее 1000 / 4 = 250 раз, значит общее время обработки запроса 2 типа будет более 500 е.в.

Т.к. время прихода и время обработки сравнимо, то ОА может работает с простоем и без. Время моделирования должно быть около 2500 е.в. Время простоя будет равно разнице между временем обработки запросов и временем их поступления и практически должно быть равно 0.

Задание 3: a1=0, a2=6, b1=1, b2=8

Среднее значение времени обработки одного запроса в ОА1 3 е.в. Так как из каждых 1000 запросов 700 возвращается в очередь, из которых 700 * 0.7 = 490 еще раз возвращаются в очередь и т.д. ОА1 обработает около 3310 запросов и общее время обработки запросов будет: 9930 е.в. В начале работы системы все запросы находятся в очереди, поэтому простоя ОА1 не будет.

Поскольку в ОА2 запросы поступают после обработки в ОА1, то среднее время поступления 1000 запросов будет равно времени моделирования ОА1. Среднее значение времени обработки одного запроса в ОА2 4.5 е.в. и в итоге все 1000 запросов попадут на обработку в ОА2, общее время обработки запросов будет 4500 е.в. А время простоя ОА2 будет 9930 – 4500 = 5430 е.в. Время моделирования ОА2 определяется временем поступления запросов и равно 9930 е.в.

Задание 4: a1=0, a2=6, b1=0, b2=1

Среднее значение времени обработки одного запроса 0.5 е.в. Так как из каждых 1000 запросов 800 возвращается в очередь, из которых 800 * 0.8 = 640 еще раз возвращаются в очередь и т.д. ОА обработает 4944 запроса и общее время обработки всех запросов будет 2472 е.в.

Для поступления 1000 запросов, если каждый поступает в среднем за 3 е.в., потребуется 3000 е.в. Следовательно, ОА работает с простоем и время моделирования будет определяться временем поступления запросов и должно быть равно 3000 е.в. Время простоя будет равно разнице между временем обработки запросов и временем их обслуживания 3000 – 2472 = 528 е.в.

Задание 5: a1=1, a2=5, b1=0, b2=3, c1=0, c2=4, d1=0, d2=1

Для поступления 1000 запросов первого типа в очередь 1, если каждый приходит в среднем за 3 е.в., потребуется 3000 е.в.

Среднее значение времени обработки одного запроса первого типа 2 е.в., а общее время обработки 1000 запросов будет 2000 е.в.

Для поступления 1000 запросов второго типа в очередь 2, если каждый приходит в среднем за 1.5 е.в., потребуется 1500 е.в., вторая очередь будет заполняться в два раза быстрее, и за 3000 е.в. в нее поступит 2000 запросов второго типа.

Среднее значение времени обработки одного запроса второго типа 0,5 е.в., а общее время обработки 1000 запросов будет 500 е.в.

Если в системе были бы запросы только первого типа, то время простоя ОА было бы 3000 – 2000 = 1000 е.в., но за это время система может обработать 2000 запросов второго типа.

Итого система обработает 1000 запросов первого и 2000 запросов второго типа, время моделирования будет 3000 е.в., а время простоя практически будет равно 0.

Задание 6: a1=1, a2=3, b1=3, b2=5 , c1=0, c2=2

Из 1000 поступивших в систему запросов 600 попадают в очередь 1, и 400 попадают сразу в очередь 2.

Для 600 запросов, если каждый приходит в среднем за 2 е.в., потребуется 1200 е.в. Среднее значение времени обработки одного запроса в ОА1 – 4 е.в., а общее время обработки 600 запросов будет 2400 е.в. Следовательно ОА1 работает без простоя, и время моделирования будет определяться временем обработки запросов, т.е. оно должно быть равно 2400 е.в.

В очередь 2 попадут 600 запросов после обработки в ОА1, их время прихода определяется временем моделирования ОА1 и равно 2400 е.в.

400 запросов поступают сразу в очередь 2, в среднем за 2 е.в., но, так как их время проступления меньше времени прихода запросов после обработки в ОА1, то они будут попадать в очередь между запросами, прошедшими обработку в ОА1, таким образом среднее время прихода 1000 запросов в очередь 2 будет 2400 е.в.

Среднее значение времени обработки одного запроса в ОА2 1 е.в., а общее время обработки 1000 запросов будет 1000 е.в. следовательно время моделирования ОА2 будет определятся временем прихода запросов во вторую очередь и будет равно 2400 е.в, среднее время простоя ОА2 будет 2400 – 1000 = 1400 е.в.

Задание 7: a1=1, a2=3, b1=1, b2=5, c1=0, c2=2, d1=1, d2=3

Так как появление запросов разного типа равновероятно, то будет сгенерировано 1000 запросов первого типа и 1000 запросов второго типа. После поступления в систему 1000 запросов первого типа, новые запросы второго типа тоже поступать в систему не будут.

Для поступления 1000 запросов первого типа в очередь, если каждый приходит в среднем за 2 е.в., потребуется 2000 е.в. Для прихода 1000 запросов второго типа в очередь, если каждый приходит в среднем за 3 е.в., потребуется 3000 е.в. Общее время поступления запросов обоих типов в очередь будет 5000 е.в.

Поскольку все запросы второго типа возвращаются в очередь, то каждый запрос второго типа в среднем успеет трижды пройти обработку в ОА.

Среднее значение времени обработки одного запроса первого типа 1 е.в., а общее время обработки 1000 запросов первого типа будет 1000 е.в. Среднее значение времени обработки одного запроса второго типа 2 е.в., а общее время обработки 3000 запросов второго типа будет 6000 е.в.

Итого система обработает 1000 запросов первого и 3000 запросов второго типа, время моделирования будет 7000 е.в., а время простоя практически будет равно 0.

Задание 8: a1=0, a2=0.02, b1=1, b2=5, c1=3, c2=9

Так как все запросы в начале работы находятся в очереди 1, то время моделирования ОА1 будет зависеть только от времени их обработки, и для 1000 запросов будет равно 10 е.в., а среднее время выхода 1000 запросов из ОА1 будет 5 е.в. (для первых запросов время ожидания в очереди будет 0 е.в., а для последних 10 е.в.)

После обработки в ОА1 700 запросов поступают во вторую очередь и 300 запросов в третью очередь, соответственно, время поступления запросов в очереди будет 700 * 5 = 3500 е.в. и 300 * 5 = 1500 е.в.

Среднее значение времени обработки одного запроса в ОА2 – 3 е.в., в ОА3 – 6 е.в. Для обработки 700 запросов в ОА2 требуется 2100 е.в., для обработки 300 запосов в ОА3 требуется 1800 е.в.

Таким образом время моделирования ОА2 зависит от времени поступления запросов, ОА2 работает с простоем равным 3500 – 2100 = 1400 е.в.

Время моделирования ОА3 зависит от времени обработки запросов, ОА3 работает без простоя.

Источники информации

- Очередь — Викиконспекты.
- Кормен Е., Лейзерсон Ч., Ривест Р. «Алгоритмы: построение и анализ. » – М.:МЦНМО. 2000
- Кнут Д. «Искусство программирования для ЭВМ.» Т. 3. – М.: Вильямс. 2001.