

Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и
радиоэлектроники»

Кафедра электронных вычислительных машин

Лабораторная работа №5
по дисциплине ОСиСП

Тема: «Потоки исполнения, взаимодействие и
синхронизация»

Выполнил:
студент группы 150501
Ковальчук Д.И.

Проверил:
старший преподаватель
_____ Поденок Л. П.

Минск
2023

1 ПОСТАНОВКА ЗАДАЧИ

Задача производители-потребители для потоков. Основной поток создает очередь сообщений, после чего ожидает и обрабатывает нажатия клавиш, порождая и завершая потоки двух типов — производители и потребители.

Очередь сообщений представляет собой классическую структуру — кольцевой буфер, содержащий указатели на сообщения, и пара указателей на голову и хвост. Помимо этого очередь содержит счетчик добавленных сообщений и счетчик извлеченных.

Производители формируют сообщения и, если в очереди есть место, перемещают их туда.

Потребители, если в очереди есть сообщения, извлекают их оттуда, обрабатывают и освобождают память с ними связанную.

Для работы используются два семафора для заполнения и извлечения, а также мьютекс для монопольного доступа к очереди.

Сообщения имеют следующий формат (размер и смещение в байтах):

Имя	Размер	Смещение	Описание
type	1	0	тип сообщения
hash	2	1	контрольные данные
size	1	3	длина данных в байтах (от 0 до 256)
data	$((size + 3)/4)*4$	4	данные сообщения

Производители генерируют сообщения, используя системный генератор `rand(3)` для `size` и `data`. В качестве результата для `size` используется остаток от деления на 257.

Если остаток от деления равен нулю, `rand(3)` вызывается повторно. Если остаток от деления равен 256, значение `size` устанавливается равным 0, реальная длина сообщения при этом составляет 256 байт.

При формировании сообщения контрольные данные формируются из всех байт сообщения. Значение поля `hash` при вычислении контрольных данных принимается равным нулю. Для расчета контрольных данных можно использовать любой подходящий алгоритм на выбор студента.

После помещения значения в очередь перед освобождением мьютекса очереди производитель инкрементирует счетчик добавленных сообщений. Затем

после поднятия семафора выводит строку на stdout, содержащую помимо всего новое значение этого счетчика.

Потребитель, получив доступ к очереди, извлекает сообщение и удаляет его из очереди. Перед освобождением мьютекса очереди инкрементирует счетчик извлеченных сообщений. Затем после поднятия семафора проверяет контрольные данные и выводит строку на stdout, содержащую помимо всего новое значение счетчика извлеченных сообщений.

Следует предусмотреть задержки, чтобы вывод можно было успеть прочесть в процессе работы программы. Следует предусмотреть защиту от тупиковых ситуаций из-за отсутствия производителей или потребителей.

Также дополнительно обрабатывается еще две клавиши – увеличение и уменьшение размера очереди.

2 АЛГОРИТМ

В программе main нужно объявить все необходимые переменные (мьютекс для монопольного доступа к очереди, два семафора для количества сообщений в очереди и свободных мест в очереди, собственно саму очередь), проинициализировать очередь, открыть семафоры, проверяя при этом возвращаемые значения функций, и создать мьютекс. Далее нужно войти в бесконечный цикл считывания опций, при вводе опции вызвать соответствующую функцию.

При вводе опции „q” нужно закрыть семафоры, уничтожить мьютекс и удалить все созданные потоки.

При вводе опции „i” необходимо увеличить размер очереди на 1, при вводе опции „d” нужно уменьшить размер очереди на 1.

При вводе опции „1” необходимо создать новый поток для производителя и запустить функцию, внутри которой производитель будет генерировать сообщения и класть их в очередь, при вводе опции „2” нужно удалить последнего производителя и вернуть занятые им ресурсы.

При вводе опции „3” необходимо создать новый поток для потребителя и запустить функцию, внутри которой производитель будет проверять контрольную сумму сообщений и извлекать их из очереди, при вводе опции „4” нужно удалить последнего потребителя и вернуть занятые им ресурсы.

При вводе опции „m” нужно выводить меню со всеми возможными опциями.

При вводе опции „l“ необходимо вывести id главного потока, а также id потоков потребителей и производителей.

При вводе опции „s“ необходимо вывести текущий размер очереди.

3 ТЕСТ

```
darya@darya:~/osisp/lab05$ ./main
```

```
        MENU
1  CREATE PRODUCER
2  DELETE PRODUCER
3  CREATE CONSUMER
4  DELETE CONSUMER
i  INCREASE SIZE OF QUEUE
d  DECREASE SIZE OF QUEUE
m  SHOW MENU
l  SHOW THREADS INFO
s  SHOW SIZE OF QUEUE
q  EXIT PROGRAMM
```

```
1
```

```
PRODUCER CREATED. PID: 140182527407808
```

```
1
```

```
PRODUCER CREATED. PID: 140182519015104
```

```
3PROCESS (TID): 140182527407808 CREATED MESSAGE:
TYPE: 0
HASH: 637F
SIZE: 66
AMOUNT OF PUSHED MESSAGES: 1
```

```
CONSUMER CREATED. PID: 140182510622400
```

```
3
```

```
CONSUMER CREATED. PID: 140182502229696
```

```
PROCESS (TID): 140182519015104 CREATED MESSAGE:
TYPE: 0
HASH: 0A3C
SIZE: 188
AMOUNT OF PUSHED MESSAGES: 2
```

```
PROCESS (TID): 140182527407808 CREATED MESSAGE:
TYPE: 0
HASH: 33E3
```

SIZE: 170
AMOUNT OF PUSHED MESSAGES: 3

PROCESS (TID): 140182510622400 CONSUMED MESSAGE:
TYPE: 0
HASH: 637F
SIZE: 66
AMOUNT OF EXTRACTED MESSAGES: 1

PROCESS (TID): 140182502229696 CONSUMED MESSAGE:
TYPE: 0
HASH: 0A3C
SIZE: 188
AMOUNT OF EXTRACTED MESSAGES: 2

PROCESS (TID): 140182519015104 CREATED MESSAGE:
TYPE: 0
HASH: 71CB
SIZE: 83
AMOUNT OF PUSHED MESSAGES: 4

l

```
PROCESSES(PID)
    MAIN
    140182531573568
    PRODUCERS
    140182527407808
    140182519015104
    CONSUMERS
    140182510622400
    140182502229696
```

SIZE OF QUEUE 5

PROCESS (TID): 140182510622400 CONSUMED MESSAGE:
TYPE: 0
HASH: 33E3
SIZE: 170
AMOUNT OF EXTRACTED MESSAGES: 3

PROCESS (TID): 140182502229696 CONSUMED MESSAGE:
TYPE: 0
HASH: 71CB
SIZE: 83
AMOUNT OF EXTRACTED MESSAGES: 4

INCREASED SIZE OF QUEUE. NOW IT IS 6

PROCESS (TID): 140182502229696 CONSUMED MESSAGE:

TYPE: 0
HASH: 1F17
SIZE: 222
AMOUNT OF EXTRACTED MESSAGES: 8

PROCESS (TID): 140182519015104 CREATED MESSAGE:
TYPE: 0
HASH: 9FCE
SIZE: 178
AMOUNT OF PUSHED MESSAGES: 10

d

DECREASED SIZE OF QUEUE. NOW IT IS 5

q

4 ВЫВОД

В ходе лабораторной работы была решена задача производители-потребители для процессов, были изучены механизмы работы общей памяти, семафоров, были предусмотрены тупиковые ситуации, если отсутствуют потребители или производители. Также были использованы такие функции как `sem_open()`, `sem_unlink()`, `sem_trywait()`, `sem_post()`, `pthread_mutex_lock()`, `pthread_mutex_unlock()`, `pthread_join()`, `pthread_self()`, `pthread_cancel()`, `pthread_create()`.