

БГУИР

Кафедра ЭВМ

Отчет по лабораторной работе № 4
Тема: «Задача производителя-потребителя для процессов»

Выполнил:
студент группы 150502 Альхимович Н.Г.

Проверил:
старший преподаватель Поденок Л.П.

Минск
2023

1 ЦЕЛЬ РАБОТЫ

Изучить принципы использования семафоров и мьютексов для обеспечения бесконфликтного взаимодействия процессов.

2 ПОСТАНОВКА ЗАДАЧИ

Основной процесс создает очередь сообщений, после чего ожидает и обрабатывает нажатия клавиш, порождая и завершая процессы 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 или одноместный семафор для монопольного доступа к очереди.

Формат сообщений представлен на рисунке 2.1.

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

Рисунок 2.1 – Формат сообщений

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 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 новое значение счетчика извлеченных сообщений.

При получении сигнала о завершении процесс должен завершить свой цикл и только после этого завершиться, не входя в новый.

Следует предусмотреть задержки, чтобы вывод можно было успеть прочитать в процессе работы программы.

Следует предусмотреть защиту от тупиковых ситуаций из-за отсутствия производителей или потребителей.

3 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

Результат работы программы с опцией c:

```
nina@nina-MacBook-Air:~/Документы/Альхимович Н.Г./Lab04:main
-$ make
gcc -W -Wall -Wextra -Wno-unused-parameter -Wno-unused-
variable -std=c11 -pedantic main.c -o main
gcc -W -Wall -Wextra -Wno-unused-parameter -Wno-unused-
variable -std=c11 -pedantic producer.c -o producer
gcc -W -Wall -Wextra -Wno-unused-parameter -Wno-unused-
variable -std=c11 -pedantic consumer.c -o consumer
nina@nina-MacBook-Air:~/Документы/Альхимович Н.Г./Lab04:main
-$ ./main
```

Выберите действие:

- c - создать процесс-производитель;
- + - создать процесс-потребитель;
- d - удалить процесс-производитель;
- - удалить процесс-потребитель
- q - завершить работу.

```
c
Процесс-производитель 4260 создан
```

Результат работы программы с опцией +:

```
+
Процесс-потребитель 4261 создан
Результат работы программы с опцией -:
```

Результат работы программы с опцией d:

```
d
Дочерний процесс 4260 удален
Осталось 0 дочерних процессов
```

Результат работы программы с опцией -:

-
Дочерний процесс 4261 удален
Осталось 0 дочерних процессов

Результат работы программы с опцией q:

q
Дочерний процесс 4262 удален
Все дочерние процессы удалены
Дочерний процесс 4265 удален
Все дочерние процессы удалены

Вывод процесса-производителя:

12-ое сообщение добавлено в очередь
Контрольные данные: 2500
Длина данных: 17
Данные сообщения: 214 114 101 216 115 225 127 146 102 208 102
118 19 131 14 47 193 90 39 179

Вывод процесса-потребителя:

3 сообщений извлечено из очереди

4 ВЫВОД

В ходе лабораторной работы были разработаны три программы: основной процесс (main), порождающий дочерние процессы-производители и процессы-потребители, в зависимости от выбранной опции управляющий их работой и создающий очередь сообщений заданной структуры; дочерний процесс-потребитель (producer), формирующий сообщение и добавляющий его в очередь; и дочерний процесс-потребитель, извлекающий сообщение из очереди.