

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра электронных вычислительных машин

Лабораторная работа №5
по курсу
«Операционные системы и системное программирование»
на тему
«Потоки исполнения, взаимодействие и синхронизация»

Выполнил:

студент группы 350501

Слепица О.Н.

Проверил:

старший преподаватель каф. ЭВМ

Поденок Л.П.

Минск 2025

1 ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Данная лабораторная работа содержит в себе две лабораторных:

1) Аналогична лабораторной № 4, но только с потоками, POSIX-семафорами и мьютексом в рамках одного процесса. Дополнительно обрабатывается еще две клавиши - увеличение и уменьшение размера очереди. Следует предусмотреть обработку запроса на уменьшение очереди таким образом, чтобы при появлении пустого места уменьшался размер очереди, а не очередной производитель размещал там свое сообщение;

2) Аналогична лабораторной № 1, но с использованием условных переменных (см. лекции СПОВМ/ОСиСП).

Требования к сборке аналогичны требованиям из лабораторной № 2.

Программы компилируются с ключами

`-W -Wall -Wextra -std=c11 -pedantic`

Допускается использование ключей

`-Wno-unused-parameter -Wno-unused-variable`

Для компиляции, сборки и очистки используется `make`.

2 ОПИСАНИЕ АЛГОРИТМОВ И РЕШЕНИЙ

Программа реализует классическую модель взаимодействия производителей и потребителей с использованием разделяемой очереди сообщений на основе кольцевого буфера. Вся работа выполняется в рамках одного процесса с использованием многопоточности. Основной поток управляет созданием рабочих потоков-производителей и потребителей, а также обеспечивает их синхронизацию.

Архитектура системы построена вокруг кольцевого буфера, который содержит указатели на сообщения, индексы головы и хвоста, текущее количество сообщений и максимальную емкость. Для синхронизации доступа используются мьютексы, семафоры POSIX или условные переменные. Мьютекс защищает критическую секцию при работе с разделяемыми данными, а условные переменные обеспечивают эффективное ожидание изменения состояния буфера.

Потоки-производители генерируют сообщения случайного размера, используя функцию `rand()`. Каждое сообщение содержит тип, контрольную сумму CRC16, размер данных и сами данные. Перед добавлением в очередь производитель блокирует мьютекс и ожидает появления свободного места через условную переменную `not_full`. После добавления сообщения производитель сигнализирует потребителям через переменную `not_empty` о появлении новых данных.

Потоки-потребители аналогичным образом получают доступ к очереди, ожидая появления сообщений через условную переменную `not_empty`. После извлечения сообщения потребитель проверяет его целостность путем повторного вычисления CRC16 и сравнения с хранящимся значением. В случае несоответствия выводится предупреждение. Освободившееся место сигнализируется производителям через переменную `not_full`.

Основной поток предоставляет интерфейс управления, позволяющий динамически создавать и завершать потоки производителей и потребителей, изменять размер буфера и выводить статистику работы системы. Завершение работы выполняется корректно, с оповещением всех потоков и освобождением ресурсов.

3 ФУНКЦИОНАЛЬНАЯ СТРУКТУРА ПРОЕКТА

Проект реализует модель взаимодействия производителей (producers) и потребителей (consumers) через общую очередь сообщений с использованием многопоточного взаимодействия и синхронизации. Система состоит из нескольких ключевых модулей, каждый из которых выполняет определенную функцию.

Основной модуль программы реализован в файле `main.c` и выполняет следующие функции:

- 1) Обработка сигнала SIGINT для graceful shutdown с помощью функции `handle_sigint()`;
- 2) Управление программой через консольное меню с помощью функции `display_main_menu()`;
- 3) Инициализация и очистка примитивов синхронизации с помощью функций `initialize_sync_primitives()` и `cleanup_sync_primitives()`;
- 4) Корректное завершение работы программы с помощью функции `graceful_shutdown()`;
- 5) Управление потоками производителей и потребителей;
- 6) Изменение размера буфера и вывод статистики.

Модуль работы с очередью реализован в файле `queue.c` и выполняет следующие функции:

- 1) Добавление элемента в очередь с помощью функции `push()`;
- 2) Удаление элемента из очереди с помощью функции `pop()`;
- 3) Вычисление контрольной суммы данных с помощью функции `crc16()`;
- 4) Инициализация сообщения с помощью функции `init_mes()`;
- 5) Вывод информации о сообщении с помощью функции `print_mes()`;
- 6) Очистка всей очереди с помощью функции `queue_clear()`;
- 7) Очистка отдельного сообщения с помощью функции `mes_clear()`.

Модуль работы с потоками и синхронизацией реализован в файле `utils.c` и выполняет следующие функции:

- 1) Обработка сигналов для остановки потоков с помощью функции `thread_stop_handler()`;
- 2) Реализация потребителей с использованием семафоров с помощью функции `consumer_routine()`;
- 3) Реализация производителей с использованием семафоров с помощью функции `producer_routine()`;
- 4) Реализация потребителей с использованием условных переменных с помощью функции `consumer_routine_cond()`;
- 5) Реализация производителей с использованием условных переменных с помощью функции `producer_routine_cond()`.

4 ПОРЯДОК СБОРКИ И ЗАПУСКА ПРОЕКТА

Порядок сборки и запуска состоит в следующем:

- 1) Разархивировать каталог с проектом;
- 2) Перейти в каталог с проектом
`$cd "Слепица О.Н./lab05";`
- 3) Собрать проект используя `make`;
- 4) После сборки проекта можно использовать, прописав
`$. /bin/main.`

5 РЕЗУЛЬТАТЫ РАБОТЫ ПРОГРАММЫ

1)

```
helga@fedora:~/tar_working_dir/СлепицаО.Н./lab05$ ./bin/main
```

```
Select synchronization method:
```

```
1 - POSIX semaphores and mutex
```

```
2 - Conditional variables
```

```
Your choice: 1
```

```
Using POSIX semaphores
```

МЕНЮ УПРАВЛЕНИЯ
1 - Добавить производителя
2 - Удалить производителя
3 - Добавить потребителя
4 - Удалить потребителя
l - Показать статистику
+ - Увеличить размер буфера
- - Уменьшить размер буфера
q - Выход
m - Показать меню

2)

+

МЕНЮ УПРАВЛЕНИЯ
1 - Добавить производителя
2 - Удалить производителя
3 - Добавить потребителя
4 - Удалить потребителя
l - Показать статистику
+ - Увеличить размер буфера
- - Уменьшить размер буфера
q - Выход
m - Показать меню

```
Выберите действие:
```

```
Current queue size: 11
```

3)

q

```
Producer 1 has finished
```

```
Consumer 1 has finished
```

```
Graceful shutdown complete.
```