

Министерство науки и образования Российской Федерации
ФГБОУ ВО «Ульяновский государственный технический университет»
Кафедра «Вычислительная техника»

Лабораторная работа №4

Дисциплина: «Операционные системы»

«Реализация пула потоков»

Выполнил студент
группы ИВТАСбд-31
Долгов А. П.

Проверила:
преподаватель кафедры ВТ
Беляев К. С.

Ульяновск, 2023

Постановка задачи

При использовании пулов потоков задача передается в пул и выполняется потоком из пула. Работа передаётся в пул с помощью очереди, а доступный поток удаляет работу из очереди. Если доступных потоков нет, работа остается в очереди до тех пор, пока один из них не станет доступным. Если работы нет, потоки ожидают уведомления, пока задача не станет доступной.

Этот проект включает в себя создание пула потоков и управление им. Его нужно выполнить с использованием PThreads API и примитивов синхронизации POSIX (мьютексы и семафоры).

Пул потоков состоит из следующих функций:

- `pool_init()` – создаст потоки при запуске, а также инициализирует блокировки взаимного исключения(mutex) и семафоры.
- `pool_submit()` – помещает выполняемую функцию, а также ее данные, в структуру задачи. Структура задачи представляет работу, которая будет выполнена потоком в пуле. `pool_submit()` добавит эти задачи в очередь, вызывая функцию `enqueue()`, а рабочие потоки вызовут `dequeue()` для получения работы из очереди.
- `worker()` выполняется каждым потоком в пуле, где каждый поток будет ожидать доступной работы. Как только работа станет доступной, поток удалит ее из очереди и вызовет метод `execute()` для запуска указанной функции. Семафор можно использовать для уведомления ожидающего потока, когда работа передается в пул потоков. Блокировка мьютекса необходима во избежание состояний гонки при доступе или изменении очереди.
- `pool_shutdown()` отменит каждый рабочий поток, а затем будет ждать завершения каждого потока, вызывая `pthread_join()`. Также потребуется функция `sem_wait()`, которая позволяет отменить поток, ожидающий семафора.

Пример работы программы

Запустим программу несколько раз, чтобы убедиться в правильности работы пула потоков – ожидается, что вывод каждый раз будет разным.

```
2: enter sem
2: exit sem
1: enter sem
1: exit sem
3: enter sem
3: exit sem
I add two values 1 and 1 result = 2
3: enter sem
I add two values 2 and 2 result = 4
1: enter sem
1: exit sem
I add two values 3 and 3 result = 6
2: enter sem
2: exit sem
3: exit sem
I add two values 1 and 1 result = 2
3: enter sem
3: exit sem
I add two values 3 and 3 result = 6
1: enter sem
1: exit sem
I add two values 2 and 2 result = 4
2: enter sem
2: exit sem
I add two values 1 and 1 result = 2
2: enter sem
I add two values 3 and 3 result = 6
3: enter sem
3: exit sem
I add two values 2 and 2 result = 4
1: enter sem
2: exit sem
1: exit sem
```

```
2: enter sem
2: exit sem
1: enter sem
1: exit sem
3: enter sem
3: exit sem
I add two values 2 and 2 result = 4
1: enter sem
1: exit sem
I add two values 1 and 1 result = 2
3: enter sem
3: exit sem
I add two values 3 and 3 result = 6
2: enter sem
2: exit sem
I add two values 1 and 1 result = 2
2: enter sem
I add two values 3 and 3 result = 6
1: enter sem
1: exit sem
I add two values 2 and 2 result = 4
3: enter sem
3: exit sem
2: exit sem
I add two values 2 and 2 result = 4
3: enter sem
I add two values 3 and 3 result = 6
1: enter sem
1: exit sem
I add two values 1 and 1 result = 2
2: enter sem
2: exit sem
3: exit sem
```

```
1: enter sem
1: exit sem
3: enter sem
3: exit sem
2: enter sem
2: exit sem
I add two values 1 and 1 result = 2
2: enter sem
2: exit sem
I add two values 3 and 3 result = 6
1: enter sem
1: exit sem
I add two values 2 and 2 result = 4
3: enter sem
3: exit sem
I add two values 1 and 1 result = 2
3: enter sem
3: exit sem
I add two values 3 and 3 result = 6
2: enter sem
2: exit sem
I add two values 2 and 2 result = 4
1: enter sem
1: exit sem
I add two values 1 and 1 result = 2
1: enter sem
1: exit sem
I add two values 3 and 3 result = 6
3: enter sem
3: exit sem
I add two values 2 and 2 result = 4
2: enter sem
2: exit sem
```

Рис. 1 «Запуск №1»

Рис. 2 «Запуск №2»

Рис. 3 «Запуск №3»

Программа запускает потоки каждый раз по-разному, а значит работает правильно.

Вывод

В результате выполнения данной лабораторной работы был успешно разработан и реализован пул потоков с использованием PThreads API и примитивов синхронизации POSIX. В ходе работы были созданы функции для инициализации пула потоков (`pool_init()`), добавления задачи в очередь (`pool_submit()`), выполнения задачи потоком из пула (`worker()`), а также завершения работы пула потоков (`pool_shutdown()`).

Разработанный пул потоков предоставляет эффективный механизм для параллельного выполнения задач, обеспечивая при этом безопасность и синхронизацию при доступе к ресурсам. Реализация соответствует поставленным задачам и демонстрирует успешное использование концепций многопоточности и синхронизации в рамках POSIX-стандарта.