Министерство науки и образования Российской Федерации

ФГБОУ ВО «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Лабораторная работа №4**

Дисциплина: «Операционные системы»

«Проектирование пула потоков»

Выполнил студент  
группы ИВТАCбд-32  
Давыдов Д. Д.

Проверил:  
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 и Posix синхронизации.

Этот проект будет включать в себя создание ряда потоков с использованием Pthreads API, а также использование мьютексов POSIX и семафоров для синхронизации.

Функция pool\_init() создаст потоки при запуске, а также инициализирует блокировки взаимного исключения(mutex) и семафоры.

Функция 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. Функция pool\_init() имеет возвращаемое значение int, которое используется для указания того, была ли задача успешно отправлена в пул (0 указывает на успех, 1 указывает на неудачу). Если очередь реализована с использованием массивов, pool\_init() вернет 1, если будет попытка отправить работу и очередь заполнена. Если очередь реализована как связанный список, pool\_init() пула всегда должен возвращать 0, если только не произойдет ошибка выделения памяти.

Функция 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 использовать для уведомления ожидающего потока, когда работа передается в пул потоков. Могут использоваться как именованные, так и безымянные семафоры.

Блокировка мьютекса необходима во избежание состояний гонки при доступе или изменении очереди.

Функция pool\_shutdown() отменит каждый рабочий поток, а затем будет ждать завершения каждого потока, вызывая pthread join().

**Пример работы программы**

Ввод клиента:

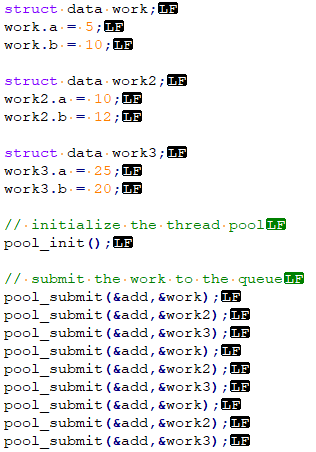


Рисунок 1. Ввод клиента

Вывод программы:

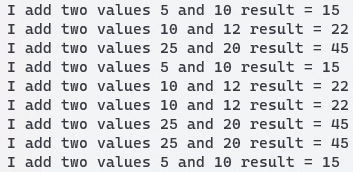


Рисунок 2. Вывод программа

**Вывод**

Программа успешно осуществляет создание и управление пулом потоков. В процессе были использованы семафоры и мьютексы для блокировки ресурсов. А также успешна реализована очередь. Программа успешно работает.