МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«КРЫМСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ им. В. И. ВЕРНАДСКОГО»

ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ

Кафедра компьютерной инженерии и моделирования

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2**

«Приоритеты потоков»

Лабораторная работа

по дисциплине «Параллельные и распределительные вычисления» студента 3 курса группы ПИ-б-о-221(2)

Чертова Дениса Александровича

09.03.04 «Программная инженерия»

Симферополь, 2025

**Цель:** изучить все виды приоритетов у потоков и понять как они работают.

**Задача:** написать программу и создать в ней не меньше 100 потоков 3 видов: среднего, низкого, высокого. Запустить в них цикл, который выводит их тип. Выяснить какой из потоков был последним. Сделать выводы о приоритетах у потоков.

**Ход работы:**

Создадим массив размером 300, который будет хранить все создаваемые потоки, чтобы можно было позже на них ссылаться.

Далее создаем цикл для создания этих потоков. В нём объявим переменную для хранения текущего приоритета потока и с помощью условия распределим по ровно приоритеты для всех поток. После чего создадим новый поток, внутри которого будет выводится сообщение о типе приоритета потока в консоль, а также добавим задержку на 100 мс, чтобы имитировать выполнение работы потока. Это даст возможность другим потокам выполняться, так как создастся пауза. Установим приоритет для потока перед его запуском с помощью свойства Priority и запустим поток с помощью метода Start.

Запусти цикл foreach, который будет проходить по массиву всех созданных потоков, и используем Join, чтобы блокировать вызывающий поток до завершения потока, на который он ссылается. Это гарантирует, что все потоки завершатся до того, как мы перейдем к следующему шагу. В конце сделаем вывод сообщения об окончании программы

Код программы (рис. 1):

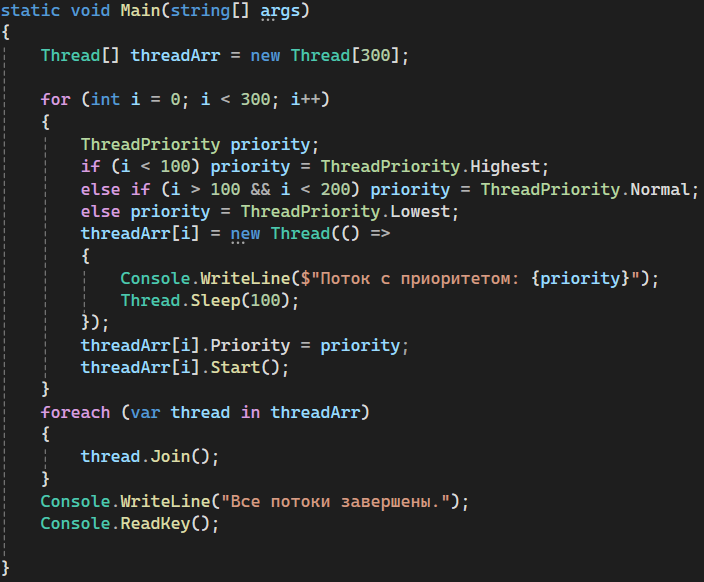


Рисунок 1 – код программы

Проведем пару тестов (рис. 2, 4).

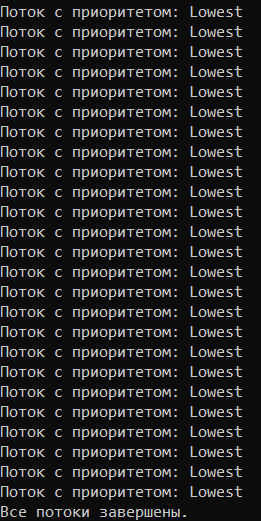
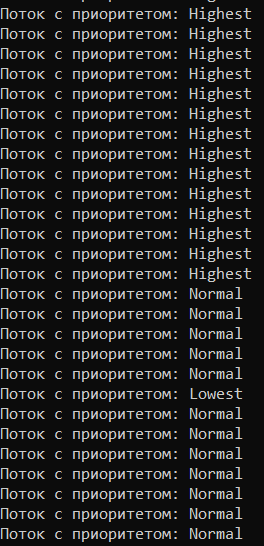


Рисунок 2 – тест №1

В результате первыми выводится потоки Highest, далее Normal и в конце Lowest. Попробуем поменять местами потоки с приоритетами Highest и Lowest внутри цикла (рис. 3)

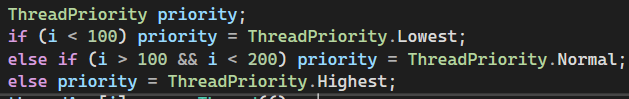


Рисунок 3 – замена Highest на Lowest и наоборот

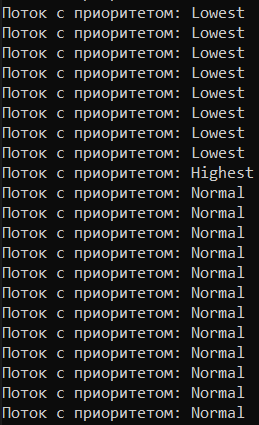
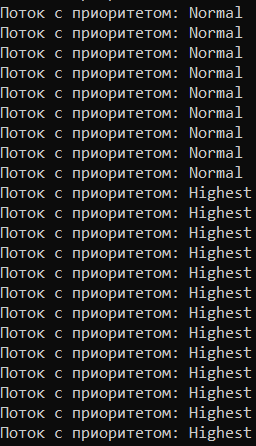
 

Рисунок 3 – тест №2

Теперь первыми выводится Lowest потоки, после чего вывелся один Highest поток, а все остальные высокоприоритетные потоки вывелись в самом конце после Normal.

Таким образом, несмотря на установку приоритета, это не гарантирует, что потоки с более высоким приоритетом завершатся первыми. Приоритеты потоков влияют на то, как операционная система распределяет процессорное время между потоками, но не обеспечивают строгую последовательность выполнения. Это означает, что даже потоки с высоким приоритетом могут быть отложены, если система занята другими задачами или если другие потоки с более низким приоритетом получают больше времени на выполнение.

**Вывод:** в ходе лабораторной работы изучены все виды приоритетов потоков, а также написана программа с 3-мя видами потоков, в ходе которой было выяснено, что более высокий приоритет может улучшить производительность задач, которые требуют быстрой обработки, но это не гарантирует, что высокоприоритетный поток завершится первым, если он не объявлен первым. В большинстве случаев, если задачи относительно равны по сложности, устанавливать приоритеты не обязательно. Их следует использовать только для задач, требующих серьезного анализа производительности.