

7. Лабораторная работа № 7. Изучение потоковой многозадачности

Номер варианта выбирается следующим образом: $\text{№} \% 5$, где № - номер студента в списке группы ЭУ, отсортированной по алфавиту, если результат равен нулю, то выбирается вариант 5.

Представить 2 варианта программы: использование шаблона *future* и функции *async* и использование класса *thread*.

Вариант 1

Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток печатает $n1$ первых значений последовательности Фибоначчи; второй поток печатает $n2$ значений первых натуральных чисел (1, 2, 3, ... $n2$), третий поток печатает $n3$ значений, выдаваемых ГПСЧ. Значения $n1$, $n2$, $n3$ задаются константами или вводятся с клавиатуры (примерный диапазон 20..30). Каждый поток имеет свое имя (например, `thread1`, `thread2`, `thread3`), перед печатью значения, поток должен напечатать с новой строки свое имя. После завершения дочерних потоков главный поток выдает сообщение об окончании работы. Имена потоков и значения $n1$, $n2$, $n3$ передаются в потоковую функцию через ее параметры.

1. Запустить программу несколько раз при одних и тех же исходных данных, посмотреть, как меняются результаты вывода. Сделать выводы.
2. Между печатью имени потока и значением установить небольшую задержку, например, 10 мс. Посмотреть, как меняются результаты вывода. Сделать выводы.

Вариант 2

Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток печатает $n1$ раз результат операций умножения двух вещественных чисел (печатаются 2 операнда и результат операции); второй поток печатает $n2$ раз результат операций сложения двух вещественных чисел (печатаются 2 операнда и результат операции),

третий поток печатает n_3 значений, выдаваемых ГПСЧ. Значения n_1 , n_2 , n_3 задаются константами или вводятся с клавиатуры (примерный диапазон 20..30). Каждый поток имеет свое имя (например, `thread1`, `thread2`, `thread3`), перед печатью результата операции или значением поток должен напечатать с новой строки свое имя. После завершения дочерних потоков главный поток выдает сообщение об окончании работы. Имена потоков и значения n_1 , n_2 , n_3 передаются в потоковую функцию через ее параметры. Значения вещественных чисел, над которыми выполняется операция в потоке, передаются в виде 2-х массив в потоковую функцию (операция выполняется над i -м элементом одного массива и i -м элементом другого массива). Массивы предварительно до запуска потоков заполняются числами с использованием ГПСЧ.

1. Запустить программу несколько раз при одних и тех же исходных данных, посмотреть, как меняются результаты вывода. Сделать выводы.
2. Между печатью имени потока и результатом операции (значением) установить небольшую задержку, например, 10 мс. Посмотреть, как меняются результаты вывода. Сделать выводы.

Вариант 3

Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток сортирует первую половину вещественного массива, второй поток сортирует вторую половину вещественного массива, третий поток запускается после завершения первых двух, он сортирует массив полностью (уже частично отсортированный). Каждый поток имеет свое имя (например, `thread1`, `thread2`, `thread3`), и печатает отсортированный массив, перед печатью каждого значения элемента массива поток должен напечатать с новой строки свое имя. После завершения дочерних потоков главный поток выдает сообщение об окончании работы. Имена потоков и массив передаются в потоковую функцию через ее параметры, использовать одну потоковую функцию для всех трех потоков. Массив предварительно до запуска потоков заполняется числами с использованием ГПСЧ. Для сортировки использовать метод прямого выбора.

1. Запустить программу несколько раз при одних и тех же исходных данных, посмотреть, как меняются результаты вывода. Сделать выводы.

2. Между печатью имени потока и значением установить небольшую задержку, например, 10 мс. Посмотреть, как меняются результаты вывода. Сделать выводы.

Вариант 4

Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток сортирует первую половину вещественного массива, второй поток сортирует вторую половину вещественного массива, третий поток запускается после завершения первых двух, он сортирует массив полностью (уже частично отсортированный). Каждый поток имеет свое имя (например, `thread1`, `thread2`, `thread3`), и печатает отсортированный массив, перед печатью каждого значения элемента массива поток должен напечатать с новой строки свое имя. После завершения дочерних потоков главный поток выдает сообщение об окончании работы. Имена потоков и массив передаются в потоковую функцию через ее параметры, использовать одну потоковую функцию для всех трех потоков. Массив предварительно до запуска потоков заполняется числами с использованием ГПСЧ. Для сортировки использовать метод прямого обмена (метод «пузырька»).

1. Запустить программу несколько раз при одних и тех же исходных данных, посмотреть, как меняются результаты вывода. Сделать выводы.
2. Между печатью имени потока и значением установить небольшую задержку, например, 10 мс. Посмотреть, как меняются результаты вывода. Сделать выводы.

Вариант 5

Реализовать программу, в которой кроме главного создается три отдельных потока: первый поток сортирует первую половину вещественного массива, второй поток сортирует вторую половину вещественного массива, третий поток запускается после завершения первых двух, он сортирует массив полностью (уже частично отсортированный). Каждый поток имеет свое имя (например, `thread1`, `thread2`, `thread3`), и печатает отсортированный массив, перед печатью каждого значения элемента массива поток должен напечатать с новой строки свое имя. После завершения дочерних потоков главный поток выдает сообщение об

окончании работы. Имена потоков и массив передаются в потоковую функцию через ее параметры, использовать одну потоковую функцию для всех трех потоков. Массив предварительно до запуска потоков заполняется числами с использованием ГПСЧ. Для сортировки использовать метод прямого включения.

1. Запустить программу несколько раз при одних и тех же исходных данных, посмотреть, как меняются результаты вывода. Сделать выводы.
2. Между печатью имени потока и значением установить небольшую задержку, например, 10 мс. Посмотреть, как меняются результаты вывода. Сделать выводы.