**Лабораторная работа # 3 по программированию**

**Студент: Стаценко А.О**

**Группа: ИКС-432**

**Преподаватель: Вейлер Андрей Игоревич**

**На оценку 3**

Для начала создадим два потока, которые будут являться дочерним и родительским, относительно приоритета работы.

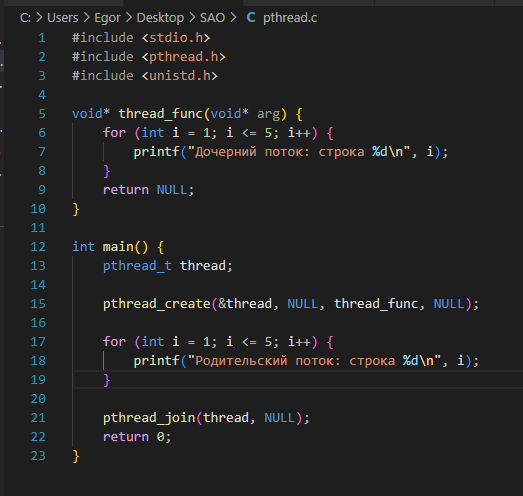


Рисунок 1 – программа родительского и дочернего потока

В данном случае, вывод будет представлять собой подобный текст:

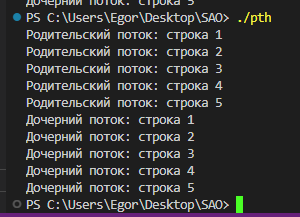


Рисунок 2 – Вывод двупоточной программы без приоритетов

Изменим приоритет и обозначим, что дочерний поток должен выполнить свою программу, раньше основной “родительской программы”, для этого изменим pthread\_join(thread, NULL) на более раннюю позицию, перед основной программой, что даст ей приоритет в выполнении.

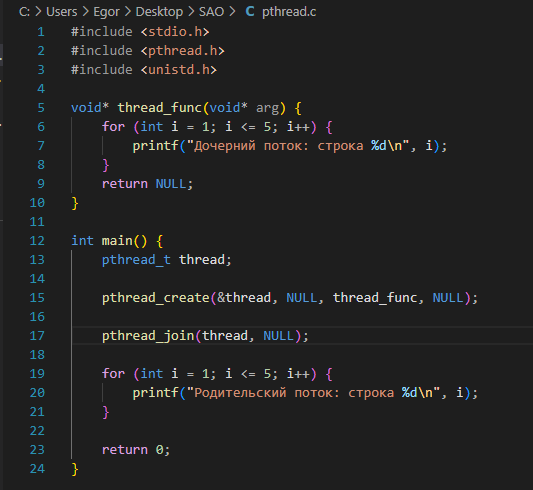


Рисунок 3 - программа родительского и дочернего потока с приоритетом выполнения

В данном случае, вывод программы измениться на подобный:

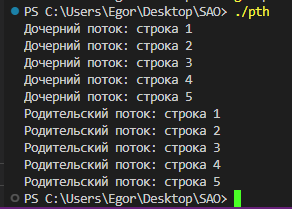


Рисунок 4 – Вывод двупоточной программы с приоритетом

Изменим программу, чтобы она могла сама создавать необходимо количество потоков, каждый из которых будет в свою очередь выводить свой номер и сообщение своего потока.

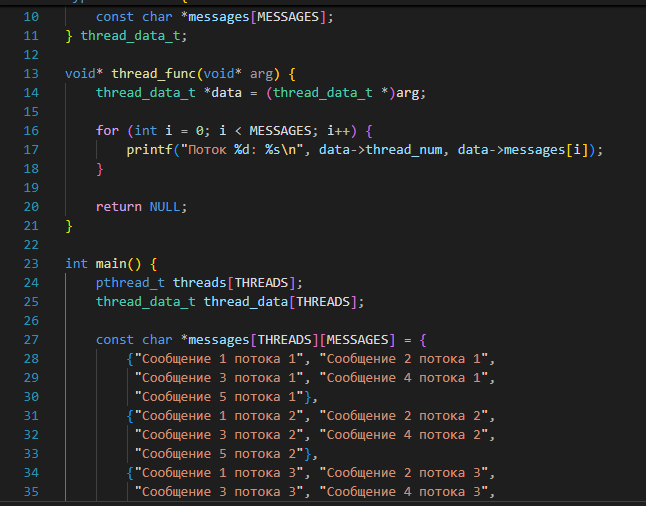


Рисунок 5 – многопоточный вывод.

В данном случае, вывод строк будет представляться подобным образом:

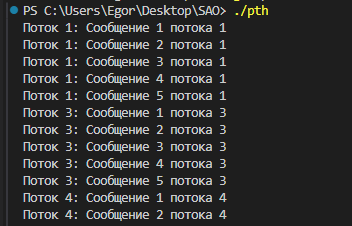


Рисунок 6 – Вывод многопоточной программы.

Изменим вывод функции, добавив параметр sleep, определяющий, что программа должна прождать 1 секунду между каждым новым выводом потока, а так же внесём параметр sleep, для основной программы, который будет отвечать за то, что через 2 секунды, основной поток прервёт работу всех существующих потоков.

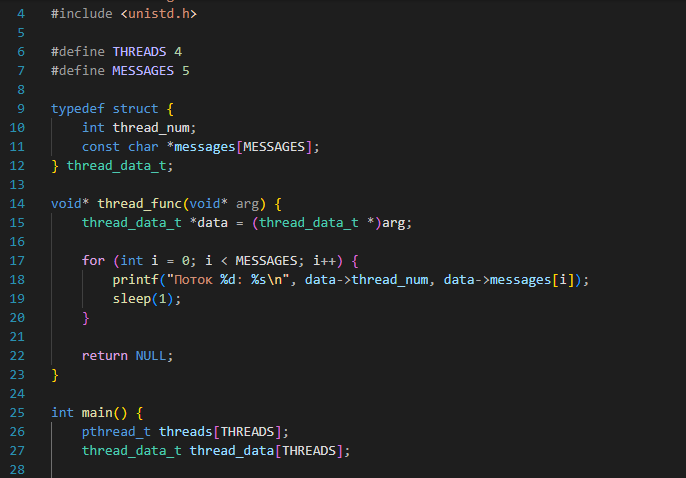


Рисунок 7 – Ожидающие потоки

С параметрами ожидания, вывод будет представлять собой подобный пример:

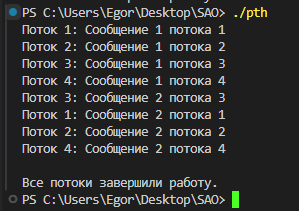


Рисунок 8 – вывод с параметром ожидания

С помощью pthread\_cleanup\_push(), мы инициализируем завершение работы потоков, выводя об этом сообщения в терминал.



Рисунок 9 – Программа с выводом завершения работы потоков.

Выводом данной программы будет:

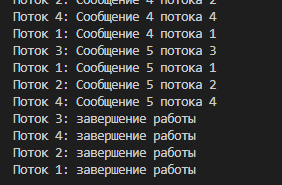


Рисунок 10 – вывод оповещения о завершении работы потоков

Создадим сортировку Sleep\_Sort, основанную на принципе вывода значения своего аргумента спустя время, равное аргументу элемента массива.

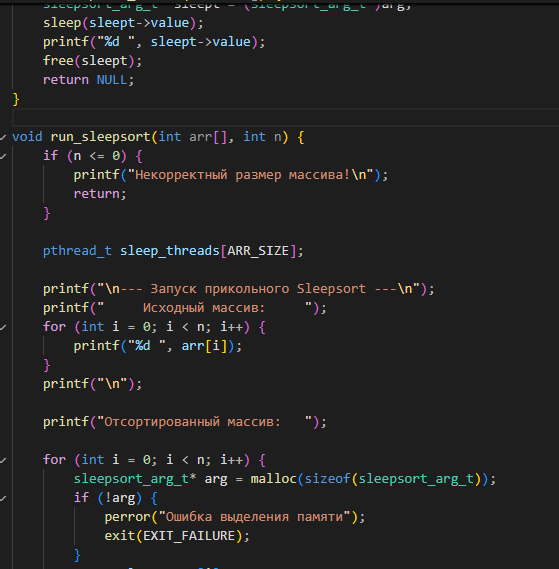


Рисунок 11 – Программа SleepSort

В данном случае, работа такой сортировки будет давать подобный результат:

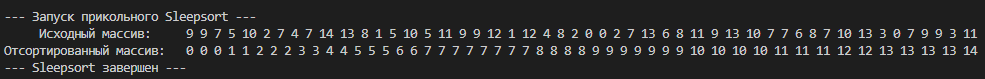


Рисунок 12 – результат SleepSort