**Лабораторная работа №4**

**Управление потоками**

**Цель работы:** Получение практических навыков в работе с потоками в OC Windows и Linux.

!!!Задания являются накопительными, т.е. для выбранного уровня сложности нужно выполнить также все предыдущие.

**Темы для предварительного изучения:**

* Утилита Sysinternals Process Explorer ([ссылка](https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer));
* Работа с потоками в Windows (лекция и [документация](https://learn.microsoft.com/en-us/windows/win32/procthread/processes-and-threads)) и POSIX (лекция и [документация](https://pubs.opengroup.org/onlinepubs/9799919799/functions/pthread_create.html)).

**Постановка задачи для Windows**:

**Уровень 4-5**:

**Функция Lab\_04х:**

Функция выполняет N-ое количество итераций в цикле. На каждой итерации поток выводит следующее:

* Идентификатор процесса;
* Идентификатор потока;
* Номер итерации;
* Одну из букв вашего пользователя (см. в конце описания функции).

Количество итераций задаётся через аргументы функции.

Между итерациями должна быть задержка в 300-400 мс.

Предусмотреть обработку ошибок и корректную очистку всех ресурсов.

Вывод данной функции должен быть организован через семейство функций **printf**.Сам вывод должен удовлетворять шаблону:

PID – TID - №Итерации – Буква из имени вашего пользователя

Причём буквы из вашего имени выводятся циклически. (Прим. Для User-d3hc25h на 1-й итерации будет U, 2-й – s, 3-й – e, и так далее)

**Приложение Lab-04a:**

Программа создаёт два дочерних потока в которых выполняется функция Lab\_04x. При этом в первом и втором потоках выполняются 50 и 125 итераций соответственно.

В главном потоке также вызывается функция Lab\_04x так, чтобы выполнялось 100 итераций.

Главный поток должен ожидать завершения всех дочерних потоков перед тем, как завершать свою работу.

Пока процесс запущен, изучить информацию об имеющихся потоках через PowerShell и Process Explorer.

**Приложение Lab-04b:**

Программа в целом идентична **Lab-04a**, за исключением следующего:

* + Вызов функции Lab\_04x в главном потоке заменить на аналогичный цикл;
  + Добавить в цикл приостановку первого потока на 20 итерации и возобновление на 60 итерации;
  + Добавить приостановку второго потока на 40 итерации, а возобновление после выполнения этого цикла.

Главный поток должен ожидать завершения всех дочерних потоков перед тем, как завершать свою работу.

Пока процесс запущен, изучить информацию об имеющихся потоках через PowerShell и Process Explorer. В частности, изучить в каких состояниях находятся потоки в соответствующие этапы работы приложения!

**Приложение Lab-04c:**

Программа в целом идентична **Lab-04a**, за исключением следующего:

* + Вызов функции Lab\_04x в главном потоке заменить на аналогичный цикл;
  + Добавить в цикл завершение второго потока на 40 итерации.

Главный поток должен ожидать завершения всех дочерних потоков перед тем, как завершать свою работу.

Пока процесс запущен, изучить информацию об имеющихся потоках через PowerShell и Process Explorer. В частности, проверить в Process Explorer исчез ли дескриптор завершенного потока!

**Приложение Lab-04d:**

Программа запускает

Создать программу, которая выводит список запущенных в процессе потоков. Для каждого потока выводится его идентификатор. ([Tool Help Library](https://learn.microsoft.com/en-us/windows/win32/api/_toolhelp/)).

**Уровень 7-8**:

**Функция L4Primes:**

Функция выполняет поиск простых чисел в заданном диапазоне. Диапазон задаётся через аргументы потока. Найденные простые числа сохраняются в промежуточном буфере, который расположен в Thread Local Storage, доступ к которому обеспечивается через функции **WinAPI**.

После завершения цикла, поток начинает запись значений из **TLS** в некоторый глобальный массив.

Никакой синхронизации!

**Приложение Lab-04p:**

Программа запускает заданное количество потоков для нахождения простых чисел. Главный поток ожидает завершения всех созданных потоков. После этого полученный глобальный массив сортируется по возрастанию и выводится на консоль. После этого программа завершает своё выполнение.

Структура вызова данного приложения:

**Lab-04p** <Количество создаваемых потоков> <Нижний порог диапазона> <Верхний порог диапазона>

Указанный диапазон должен быть разделён на равные части в количестве равном количеству создаваемых потоков. Программа не может быть завершена пока не завершены все потоки.

Предусмотреть обработку ошибок и очистку ресурсов.

***Вопрос:*** *Вся ли**информация о простых числах в целости? Почему?*

**Приложение Lab-04p2:**

Скопировать предыдущее приложение и заменить использование **TLS** с функций **WinAPI** на использование переменных со специальными модификатором **\_\_declspec(thread)**.

Провести замеры производительности приложения **Lab-04p** ианалогичного задания на поиск простых чисел из 3 лабораторной работы. Провести серию запусков с увеличением количества процессов/потоков: от 1 до 2N, где N – количество физических ядер вашего процессора (с увеличением количества на каждом шаге в 2 раза, например, 1, 2, 4, 8, 12 или 16 в зависимости от количества). Диапазон при этом должен быть достаточно велик, желательно хотя бы от 1 до 220-221.

**Составить сравнительную таблицу результатов и сделать выводы.**

**Уровень 9-10**:

**Приложение Lab-04pi:**

Напишите программу, которая вычисляет число Пи при помощи ряда Лейбница. Расчёт ряда разделить на частичные суммы ряда. Каждая такая частичная сумма считается в отдельных волокнах (fibers), чьё количество определяется через параметр при запуске приложения.

Приложение работает «бесконечно». После нажатия Ctrl+C ([SetConsoleCtrlHandler](https://learn.microsoft.com/en-us/windows/console/setconsolectrlhandler)) программа должна как можно скорее завершаться, собирать частичные суммы ряда и выводить полученное приближение числа.

Опытным путём установить оптимальное количество итераций перед переключением на другое волокно! Условием переключения может быть не только заданное количество обработанных элементов, но и заданная точность частичной суммы и т.д.

Подсказка: Так как частичные суммы заранее не могут быть разделены на равные части, работа приложения бесконечна, то выдавайте следующую частичную сумму поэтапно, т.е. при завершении работы над одной, передайте волокну следующую (ряд ведь бесконечный). Размер одной частичной суммы может быть, например, 1000-5000 элементов ряда Лейбница.

**Постановка задачи для Linux**:

**Уровень 4-5**:

**Функция Lab\_04х:**

Функция полностью идентична таковой из заданий по Windows.

**Приложение Lab-04px:**

Программа создаёт два дочерних потока в которых выполняется функция Lab\_04x. При этом в первом и втором потоках выполняются 50 и 125 итераций соответственно.

В главном потоке также вызывается функция Lab\_04x так, чтобы выполнялось 100 итераций.

Главный поток должен ожидать завершения всех дочерних потоков перед тем, как завершать свою работу.

Изучить информацию о созданных потоках через файловую систему (каталог /proc) и утилиту ps. Предусмотреть обработку ошибок и очистку ресурсов.

**Уровень 7-8**:

Разработать приложения аналогичные приложениям из задания для Windows на уровень 7-8. С уточнением, что:

* Код для инициализации TLS реализовать через функцию **pthread\_once**.
* Модификатор **\_\_declspec(thread)** заменить на аналогичный для Linux.

Провести аналогичные исследования между приложением из этого задания и приложением из 3 лабораторной работы.

**Уровень 9-10**:

Написать приложение аналогичное приложению из Windows на 9-10 с учётом специфики операционной системы Linux.

**Вопросы для самоконтроля:**

1. Что такое поток? Что такое многопоточность?
2. Каково отличие потока от процесса?
3. Когда лучше использовать потоки вместо дополнительных процессов для многозадачности?
4. Какие функции WinAPI для работы с потоками вы знаете? POSIX?
5. Какие ресурсы доступны потоку?
6. Какие бывают потоки? В чём их преимущества и недостатки?
7. Что такое Thread Local Storage? **(7-8)**
8. Для чего был придуман TLS? **(7-8)**
9. Что такое потокобезопасность? Реентерабельность? **(7-8)**
10. Что такое волокно? **(9-10)**
11. Какова ситуация с волокнами в Linux? **(9-10)**
12. Какие функции WinAPI для работы с волокнами вы знаете? POSIX? **(9-10)**
13. Назовите структуры ядер ОС Windows и Linux описывающие поток. **(9-10)**