**Лабораторная работа №3**

**Управление процессами II**

**Цель работы:** Получение практических навыков в работе с процессами, в частности с механизмами передачи информации между ними, а также синхронизации

Перед выполнением задания прочитайте его полностью. Схематично представьте себе, что от вас требуется и только после этого выполняйте.

**Постановка задачи для Windows:**

**Приложение Lab-03a-client:**

Программа выполняет поиск простых чисел в заданном диапазоне. Диапазон задаётся через аргументы командной строки. Найденные простые числа сохраняются в промежуточном буфере, а затем после завершения работы цикла отправляются по анонимному каналу родительскому процессу. Доступ к каналу, синхронизируется при помощи механизма «Мьютекс».

Структура вызова данного приложения:

**Lab-03a-client** <Нижний порог диапазона> <Верхний порог диапазона> [Имя мьютекса]

Имя «мьютекса» является опциональным. В программе предусмотреть значение по умолчанию (здесь и в дальнейшем использовать глобальные переменные окружения ОС).

Предусмотреть обработку ошибок и корректную очистку всех ресурсов.

**Приложение Lab-03a-server:**

Программа запускает заданное количество дочерних процессов для нахождения простых чисел. Также программа создаёт анонимный канал для работы с дочерними процессами. (Передача дескриптора через STARTUPINFO). При получении информации от дочернего процесса программа должна выводит её на консоль. Также программа ответственна за создание объекта «Мьютекс», который будет использоваться дочерними процессами.

Структура вызова данного приложения:

**Lab-03a-server** <Количество создаваемых процессов> <Нижний порог диапазона> <Верхний порог диапазона> [Имя мьютекса]

Имя «мьютекса» является опциональным. В программе предусмотреть значение по умолчанию.

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

Предусмотреть обработку ошибок и очистку ресурсов.

Использовать Process Explorer и найти дескрипторы канала и «мьютекса» в момент работы приложения.

***Вопрос:*** *Какой тип дескриптора указан в Process Explorer для канала? Почему?*

**Приложение Lab-03b-client:**

Программа отправляет на сервер строку в виде массива символов (принимаются как кириллические, так и латинские) с задержкой в 5 секунд. После отправки, клиент получает обратно некоторый результат (размер ответа не будет превосходить размера исходной строки) и выводить его на консоль и завершать свою работу.

Для отправки строк на сервер требуется использовать именованный канал, при чём клиент пытается подключиться к уже существующему каналу и не должен даже пытаться его создавать.

Для синхронизации отправки и получения строки использовать бинарный вариант механизма «Семафор» (Бинарный «Семафор» может предоставлять доступ только к одному ресурсу, т. е. имеет максимальное значение счётчика = 1). Т. е. разделяемым ресурсом в данном случае будет «внимание» от сервера и пока клиент не завершит работу с ним, другие клиенты ждут. (Атомарное действие = Отправка + Получение)

Структура вызова данного приложения:

**Lab-03b-client** <Строка> [Имя канала] [Имя Семафора]

Имена «Семафора» и канала являются опциональными. В программе предусмотреть использование значений по умолчанию.

Предусмотреть обработку ошибок и очистку ресурсов.

**Приложение Lab-03b-server:**

Программа создаёт именованный канал, по которому в последующем получает строки, при этом размер получаемой строки ограничен размером указанного при создании буфера канала. Сервер приводит всю строку к верхнему регистру и отправляет клиенту назад. Сервер производит логирование на консоль исходной и получившейся строк. После этого сервер входит в режим ожидания новой строки.

Сервер работает бесконечно, пока не закроется приложение.

Структура вызова данного приложения:

**Lab-03b-server** [Имя канала]

Имя канала является опциональным. В программе предусмотреть использование значения по умолчанию.

Предусмотреть обработку ошибок.

Использовать Process Explorer и найти дескрипторы канала и «семафора» в момент работы приложения.

**Постановка задачи для Linux:**

Разработать приложения аналогичные приложениям **Lab-03b-\*** из заданий для Windows. С уточнением, что вместо именованных каналов используются FIFO-очереди. Для синхронизации используются «семафоры».

Изучить информацию о FIFO-очереди и «семафоре» через файловую систему (каталог /proc).

**Требования к выполненной работе:**

* Разработанные приложения должны быть скомпилированы и протестированы как на Windows, так и на ОС из семейства Linux.
* Для генерации и сборки проекта использовать средство CMake.
* Для компиляции файлов при сборке проекта использовать компилятор Clang.

**Дополнительно (совершенно не обязательно, но огромный плюс):**

Написать аналог приложений **Lab-03a-\*** для ОС Linux.

В задании с **Lab-03b-server** написать код таким образом, чтобы корректно обрабатывались строки, превышающие размер буфера именованного канала.

**Вопросы для контроля:**

1. Что такое межпроцессное взаимодействие (IPC)?
2. Какие группы IPC-механизмов вам известны? Какие механизмы входят в эти группы?
3. Какой механизм взаимодействия выбрать, когда требуется максимальная скорость передачи данных?
4. Какой механизм взаимодействия выбрать, когда требуется повышенная устойчивость к ошибкам?
5. Что такое синхронизация?
6. Какие механизмы синхронизации вы знаете? Объясните любые три механизма.
7. Какое главное отличие CriticalSection в Windows от всех остальных механизмов? Какой основной плюс и минус этого отличия?