

Лабораторная работа №2

Сдать до 07.10

Тема: «Взаимодействие процессов в Windows».

Общее задание:

Задача: Написать консольную программу, которая запускает саму себя в качестве дочернего процесса для обработки данных. Взаимодействие между родительским и дочерним процессами должно осуществляться через неименованные каналы (anonymous pipes).

а) Логика работы программы:

Программа должна уметь работать в двух режимах в зависимости от аргументов командной строки:

- Режим Родителя (запуск без аргументов):

Создает массив целых чисел (размер и элементы вводятся с клавиатуры или генерируются случайно).

Создает два неименованных канала: один для отправки данных дочернему процессу (Pipe1), второй для получения результата (Pipe2).

Подготавливает структуры STARTUPINFO и PROCESS_INFORMATION. В STARTUPINFO настраивает перенаправление стандартного ввода (stdin) дочернего процесса на чтение из Pipe1, а стандартного вывода (stdout) — на запись в Pipe2.

Запускает копию своего же исполняемого файла с помощью CreateProcess, передав ему специальный аргумент командной строки (например, "child"), чтобы тот запустился в режиме "Потомка".

Записывает размер массива, а затем и сам массив в Pipe1.

Важно: после записи всех данных закрывает свой дескриптор на запись в Pipe1, чтобы дочерний процесс смог определить конец входных данных (EOF).

Читает из Pipe2 результат вычислений, отправленный дочерним процессом.

Дождидается завершения дочернего процесса с помощью WaitForSingleObject.

Выводит полученный результат на консоль.

Корректно закрывает все открытые дескрипторы (CloseHandle).

- Режим Потомка (запуск с аргументом "child"):

Читает из своего стандартного ввода (stdin, который на самом деле является Pipe1) сначала размер массива, а затем его элементы, пока не достигнет конца потока.

Выполняет над массивом операцию согласно своему индивидуальному варианту.

Записывает полученный результат в свой стандартный вывод (stdout, который на самом деле является Pipe2).

Завершает свою работу.

б) Обязательные условия:

Для создания процесса использовать функцию CreateProcess.

Для межпроцессного взаимодействия — CreatePipe, WriteFile, ReadFile.

Для синхронизации — WaitForSingleObject.

Обязательно выполнять корректное закрытие всех дескрипторов с помощью CloseHandle во избежание "зависаний" (deadlocks).

Глобальные переменные для передачи данных между процессами не использовать.

Проверять коды возврата всех функций Win32 API на наличие ошибок.

с) Индивидуальные варианты:

1. Дочерний процесс должен прочитать массив и вычислить:
2. Сумму всех положительных элементов массива.
3. Минимальный элемент массива.
4. Максимальный элемент массива.
5. Количество чётных элементов в массиве.
6. Количество элементов, кратных 3.
7. Произведение всех ненулевых элементов массива.
8. Среднее арифметическое всех элементов.
9. Количество элементов в заданном диапазоне [A, B] (A и B передаются от родителя после массива).
10. Индекс первого вхождения заданного числа X (X передается от родителя).
11. Сумму квадратов всех элементов массива.