**Лабораторная работа №3**

**Управление процессами**

**Цель работы:** Получение практических навыков в работе с процессами 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 процессами 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/fork.html));
* Работа с IPC (лекция, [Linux](https://support.tools/linux-pipes-ipc-mastery/), [Windows](https://learn.microsoft.com/en-us/windows/win32/ipc/interprocess-communications))

**Постановка задачи для Windows**:

**Уровень 4-5**:

**Приложение Lab-03х:**

Программа выполняет простейший цикл с заданным количеством итераций, при чём каждая итерация выполняется с задержкой 500 мс. Программа должна выводить PID на каждой итерации и перед стартом выводить количество итераций. Количество итераций передается в программу через аргументы командной строки. Если не задан аргумент командной строки, то количество итераций приложение пытается получить через переменную окружения ITER\_NUM. Если не заданы ни переменная окружения, ни аргумент командной строки приложение завершает свою работу с ошибкой (Использовать **ExitProcess**).

**Приложение Lab-03a:**

Программа создаёт три дочерних процесса с помощью **CreateProcess**. Все три процесса должны создаваться для программы **Lab-02x**, при чём для передачи имени исполняемого файла (в данном случае используется [полное имя файла](https://ru.wikipedia.org/wiki/%D0%98%D0%BC%D1%8F_%D1%84%D0%B0%D0%B9%D0%BB%D0%B0)) и параметра количества итераций должны быть выполнены следующие условия:

* При создании первого процесса используется только первый параметр функции **CreateProcess**. Количество итераций также требуется указать в первом параметре, так задумано (прим. «**Lab-03х.exe 50**»).
* При создании второго процесса используется только второй параметр функции **CreateProcess**. По сути взять строку из предыдущего пункта и передать во второй параметр.
* При создании третьего процесса для передачи имени программы используется – первый параметр, для количества итераций – второй. (при этом во втором параметре вам может понадобиться пробел в начале строки)

Родительский процесс должен дождаться завершения всех дочерних процессов (выполняются одновременно), перед тем как завершиться самому.

***Вопрос****: Все ли процессы созданы успешно? Если нет, почему?*

Также предусмотреть обработку ошибок и очистку ресурсов. Использовать Process Explorer, cmd и TaskManager для просмотра информации о созданных процессах. В Process Explorer проверить наличие дескрипторов дочерних процессов у родительского процесса.

Установить глобальную переменную окружения ITER\_NUM для вашего пользователя.

**Приложение Lab-03b:**

Создать программу идентичную программе Lab-03a, но:

* В первом вызове **CreateProcess** оставить только имя файла (т.е. убрать количество итераций, которое указывалось в Lab-03a).
* В третьем вызове передать NULL вместо второго параметра, и перед этим установить локальную переменную окружения ITER\_NUM. Значение должно отличаться от глобальной переменной.

***Вопрос:*** *Все ли процессы созданы успешно? Сколько итераций выполнил каждый процесс и почему?*

**Приложение Lab-03с:**

Создать программу, которая выводит список запущенных в системе процессов. Для каждого процесса выводится его имя, идентификатор и родительский идентификатор. ([Tool Help Library](https://learn.microsoft.com/en-us/windows/win32/api/_toolhelp/))

**Уровень 7-8**:

**Приложение Lab-03d-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.

Структура вызова данного приложения:

**Lab-03d-client** <Нижний порог диапазона> <Верхний порог диапазона>

Предусмотреть обработку ошибок и корректную очистку всех ресурсов.

**Приложение Lab-03d-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.

Структура вызова данного приложения:

**Lab-03d-server** <Количество создаваемых процессов> <Нижний порог диапазона> <Верхний порог диапазона>

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 дескриптор канала в момент работы приложения.

***Примечание! Никакой синхронизации предусматривать на данном этапе не требуется!***

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

**Уровень 9-10**:

Разработать два приложения: родительский и дочерний процессы, реализующие механизм передачи данных через структуру STARTUPINFO. Родительский процесс при создании дочернего передаёт через STARTUPINFO фразу «Hello, my dear child!», которую должен получить создаваемый процесс и выводить на консоль!

**Примечание!** Запрещено использовать переменные окружения, перенаправлять потоки ввода/вывода, передавать данные через аргументы командной строки дочернему процессу, использовать какие-либо стандартные механизмы IPC!

**Постановка задачи для Linux**:

**Уровень 4-5**:

**Приложение Lab-03x:**

Программа является полным аналогом программы из заданий для Windows.

**Приложение Lab-03a:**

Программа создаёт два дочерних процесса и загружает в них исполняемый файл **Lab-03x**. При создании процессов должны быть выполнены следующие условия:

* В ходе запуска первого дочернего процесса параметр с количеством итераций должен быть передан как аргумент командной строки
* В ходе запуска второго дочернего процесса параметр с количеством итераций должен быть передан через переменную окружения ITER\_NUM. Переменная создается непосредственно в родительском процессе перед созданием второго дочернего процесса

Родительский процесс должен дождаться завершения всех дочерних процессов (выполняются одновременно).

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

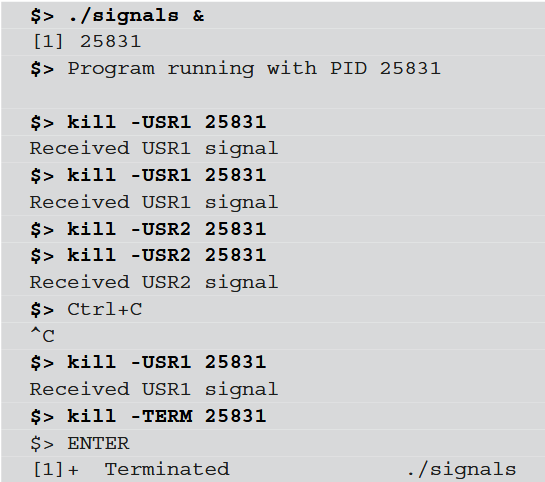
***Вопрос****: Все ли процессы созданы успешно? Если нет, почему?*

**Уровень 7-8**:

Написать аналог приложений **Lab-03d-\*** для ОС Linux. Предусмотреть обработку ошибок и очистку ресурсов.

**Уровень 9-10**:

Написать приложение, которое ведёт себя также, как представлено на следующем скриншоте:



Обращаю внимание, что Ctrl+C не способно остановить работу такого приложения!

**Вопросы для самоконтроля:**

1. Что такое процесс?
2. Какие ресурсы доступны процессу?
3. Что такое дочерний и родительский процесс?
4. Что такое системный вызов?
5. Какие системные вызовы предназначены для создания процессов в Windows? Linux?
6. Какие функции из WinAPI и POSIX связаны с ранее названными системными вызовами?
7. Сколько новых процессов появится в результате выполнения программой следующей последовательности вызовов **fork**() (учитывая, что все они выполнятся успешно)?

**fork**();

**fork**();

**fork**();

1. Для чего нужен **exec**?
2. Что такое межпроцессное взаимодействие (IPC)? **(7-8)**
3. Какие группы IPC-механизмов вам известны? Какие механизмы входят в эти группы? **(7-8)**
4. Какой механизм взаимодействия выбрать, когда требуется максимальная скорость передачи данных? **(7-8)**
5. Какой механизм взаимодействия выбрать, когда требуется повышенная устойчивость к ошибкам? **(7-8)**
6. Какие сигналы существуют в Linux? **(9-10)**
7. Как обеспечить обработку сигнала в процессе? **(9-10)**
8. Назовите структуры ядер ОС Windows и Linux описывающие процесс. **(9-10)**
9. Назовите способы просмотреть их структуру. **(9-10)**