Минобрнауки России  
Федеральное государственное автономное образовательное  
Учреждение высшего образования  
«Санкт-Петербургский государственный электротехнический  
Университет им. В.И. Ульянова (Ленина)»  
(СПГЭТУ «ЛЭТИ»)  
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники  
  
  
  
**Отчет по лабораторной работе №4**

**на тему: «Межпроцессное взаимодействие»**

**по дисциплине «Операционные системы»**

Выполнил студент группы 9308: Паникаровская Д. А.

Принял: к.т.н., доцент Тимофеев А. В.

Санкт-Петербург  
2021 г.

Содержание

[Цель работы 3](#_Toc91632837)

[Реализация решения задачи о читателях-писателях 3](#_Toc91632838)

[Указания к выполнению 3](#_Toc91632839)

[Результаты выполнения программы 5](#_Toc91632840)

[Вывод по заданию 9](#_Toc91632841)

[Использование именованных каналов для реализации сетевого межпроцессного взаимодействия 10](#_Toc91632842)

[Указания к выполнению 10](#_Toc91632843)

[Результаты выполнения программы 11](#_Toc91632844)

[Вывод по заданию 12](#_Toc91632845)

[Вывод 13](#_Toc91632846)

# Цель работы

Исследовать инструменты и механизмы взаимодействия процессов в Windows.

## Реализация решения задачи о читателях-писателях

## Указания к выполнению

1. Выполнить решение задачи о читателях-писателях, для чего необходимо разработать консольные приложения «Читатель» и «Писатель»:

* одновременно запущенные экземпляры процессов-читателей и процессов-писателей должны совместно работать с буферной памятью в виде проецируемого файла:
  + размер страницы буферной памяти равен размеру физической страницы оперативной памяти;
  + число страниц буферной памяти равно сумме цифр в номере студенческого билета без учета первой цифры.
* страницы буферной памяти должны быть заблокированы в оперативной памяти (функция **VirtualLock**);
* длительность выполнения процессами операций «чтения» и «записи» задается случайным образом в диапазоне от 0,5 до 1,5 сек;
* для синхронизации работы процессов необходимо использовать объекты синхронизации типа «семафор» и «мьютекс»;
* процессы-читатели и процессы-писатели ведут свои журнальные файлы, в которые регистрируют переходы из одного «состояния» в другое (начало ожидания, запись или чтение, переход к освобождению) с указанием кода времени (функция **TimeGetTime**). Для состояний «запись» и «чтение» необходимо также запротоколировать номер рабочей страницы.

2. Запустите приложения читателей и писателей, суммарное количество одновременно работающих читателей и писателей должно быть не менее числа страниц буферной памяти. Проверьте функционирование приложений, проанализируйте журнальные файлы процессов, постройте сводные графики смены «состояний» для не менее 5 процессов-читателей и 5 процессов-писателей, дайте свои комментарии относительно переходов процессов из одного состояния в другое. Постройте графики занятости страниц буферной памяти (проецируемого файла) во времени, дайте свои комментарии.

## Результаты выполнения программы

Было написано два консольных приложения – писатель и читатель, их запуск производится отдельным консольным приложением, которое запускает 14 читателей и писателей (сумма цифр студенческого билета без первой + 1). Каждый читатель и писатель работает 3 раза (случайное число).

Вывод данного приложения представлен на рисунке 1.

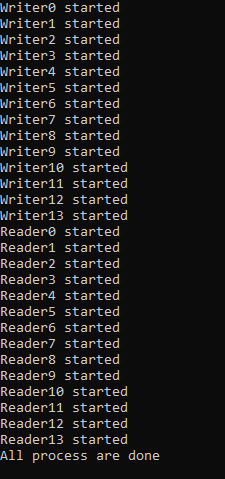


Рисунок 1 – Вывод приложения, запускающего читателей и писателей

Для обоих приложений (писателя и читателя) были созданы журнальные файлы процессов. Примеры содержания этих файлов на рисунках 2 и 3 соответственно.

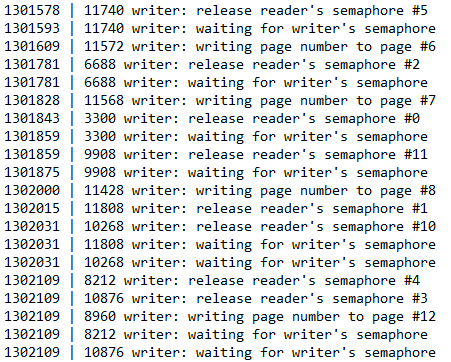


Рисунок 2 – Содержание журнального файла для приложения-писателя

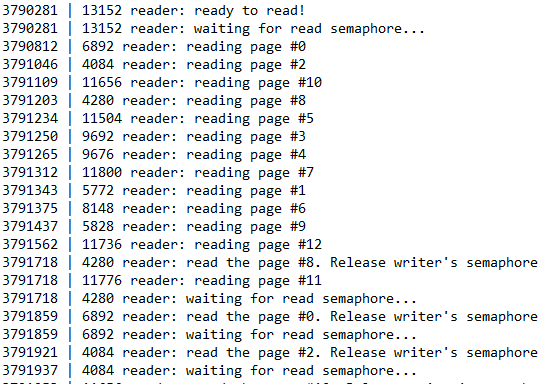


Рисунок 3 – Содержание журнального файла для приложения-читателя

По данным, полученным в журнальных файлах, в программе Octave был построен график смены состояний процесса, который приведен на рисунке 4. Снизу – данные приложения-писателя, сверху – приложения-читателя. Красным цветом выделено состояние работы процесса, а зеленым – ожидание. Наглядно видно, что читатели приступают к работе, когда писатели завершают свою работу, и наоборот.

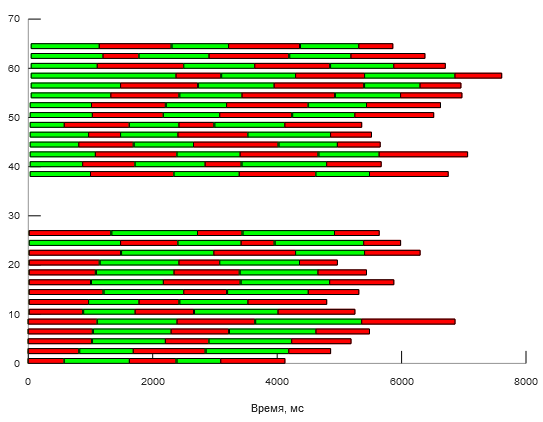


Рисунок 4 – График смены состояний

Так же в программе Octave был простроен график занятости страниц, приведенный на рисунке 5. Синим цветом показаны активные писатели, красным – читатели, а зеленым – свободные страницы. На графике видно, что в основном количество свободных страниц меняется между 0 и 1, так как писатели и читатели постоянно занимают эти страницы, сменяя друг друга.

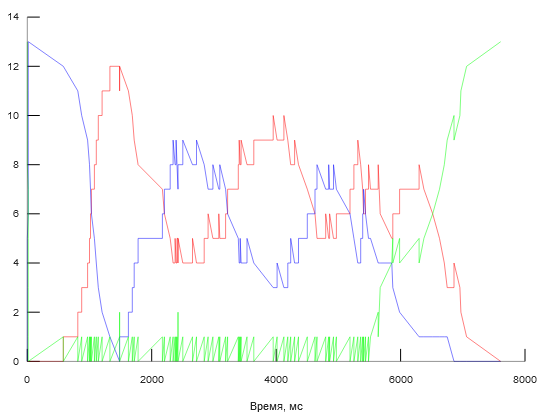


Рисунок 5 – График занятости страниц

## Вывод по заданию

В ходе выполнения задания были созданы приложения -читатель и -писатель, которые совершали обмен информацией, используя общую память между процессами в виде проецируемого файла, число буферных страниц в котором было равно 13. Для синхронизации работы процессов были использованы объекты синхронизации, такие как Mutex и Semaphore.

## Использование именованных каналов для реализации сетевого межпроцессного взаимодействия

## Указания к выполнению

1. Создайте два консольных приложения с меню (каждая выполняемая

функция и/или операция должна быть доступна по отдельному пункту меню), которые выполняют:

* приложение-сервер создает именованный канал (функция Win32

API – **CreateNamedPipe**), выполняет установление и отключение соединения (функции Win32 API – **ConnectNamedPipe**, **DisconnectNamedPipe**), создает объект «событие» (функция Win32 API – **CreateEvent**) осуществляет ввод данных с клавиатуры и их асинхронную запись в именованный канал (функция Win32 API – **WriteFile**), выполняет ожидание завершения операции ввода вывода (функция Win32 API – **WaitForSingleObject**);

2. Запустите приложения и проверьте обмен данных между процессами. Запротоколируйте результаты в отчет. Дайте свои комментарии в отчете относительно выполнения функций Win32 API.

## Результаты выполнения программы

Было написано 2 консольных приложения: server.exe и client.exe. Сначала нужно запустить server.exe, а затем – client.exe. При корректном запуске первого приложения на экран выводится сообщение, показанное на рисунке 6.

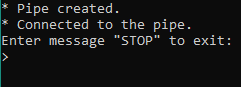


Рисунок – Вывод приложения server,exe

При запуске второго приложения будет выведено на экран сообщение, приведенное на рисунке 7

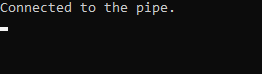


Рисунок – Вывод приложения client.exe

При вводе данных в приложении server.exe, приложение client.exe выводит их, как показано на рисунке 8.

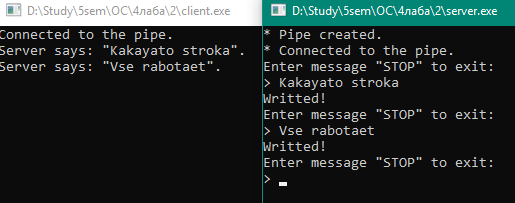


Рисунок – Ввод данных и их отображение

При вводе стоп-слова оба приложения будут автоматически закрыты.

## Вывод по заданию

В ходе выполнения задания были изучены функции Win32 API, с помощью которых производится обмен между процессами, используя именованный канал.

Одно из приложений – сервер, который создает именованный канал, подключает клиента и передает ему сообщение по каналу. Второе приложение – клиент, который подключается к именованному каналу и считывает из него сообщение.

# Вывод

В ходе выполнения лабораторной работы были исследованы инструменты и механизмы взаимодействия процессов в Windows на примере реализации решения задачи о читателях/писателях и с использованием именованных каналов для создания сетевого межпроцессного взаимодействия.