**Задание 01.Windows**

1. С помощью Notepad (Notepad+) создайте на дисковом устройстве текстовый файл **OS09\_01.txt.** Заполните его 10 строками из списка студентов вашей подгруппы. Буквы кириллические.



1. Разработайте приложение **OS09\_01**.
2. Приложение **OS09\_01** вызывает функцию **printFileInfo**, имеющую следующий прототип.



1. Функция **printFileInfo** выводит в стандартный поток вывода следующую информацию:

- имя файла;

- тип файла;

- размер файла;

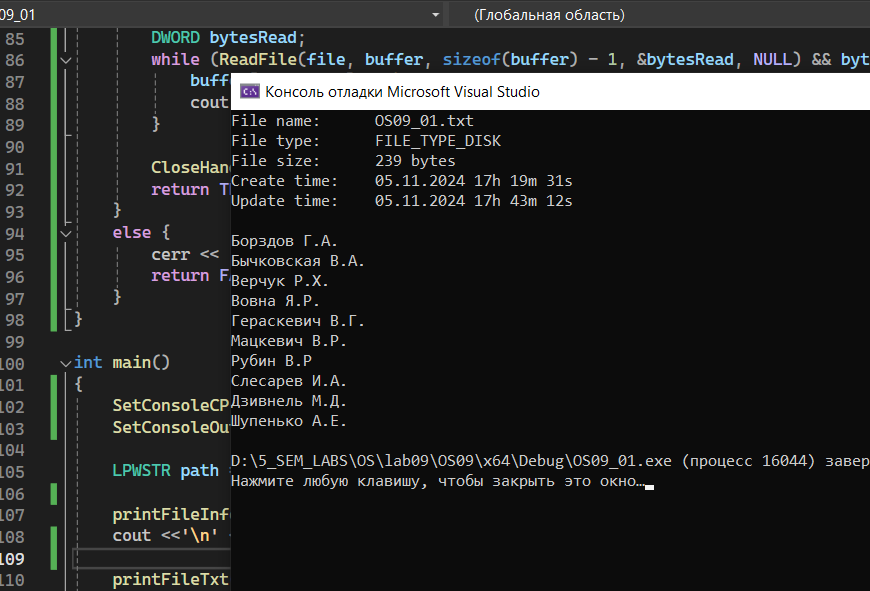
- дата и время создания файла;

- дата и время последнего обновления.

1. Приложение **OS09\_01** вызывает функцию **printFileTxt**, имеющую следующий прототип.



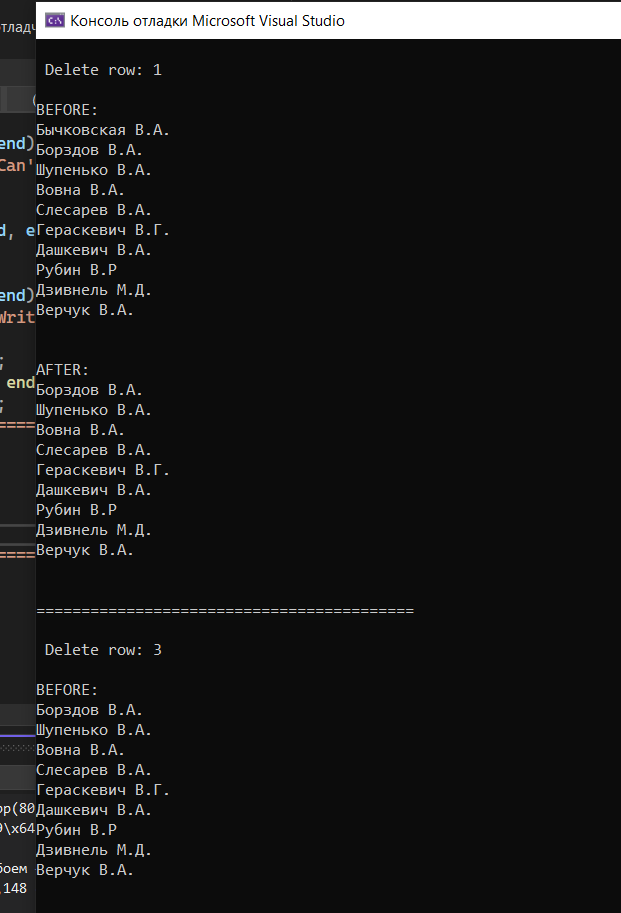
1. Функция **printFileTxt** выводит в стандартный поток вывода содержимое файла.
2. При вызове функции укажите в качестве параметра имя текстового файла, созданного в п.2.
3. Продемонстрируйте работоспособность приложения **OS09\_01**.

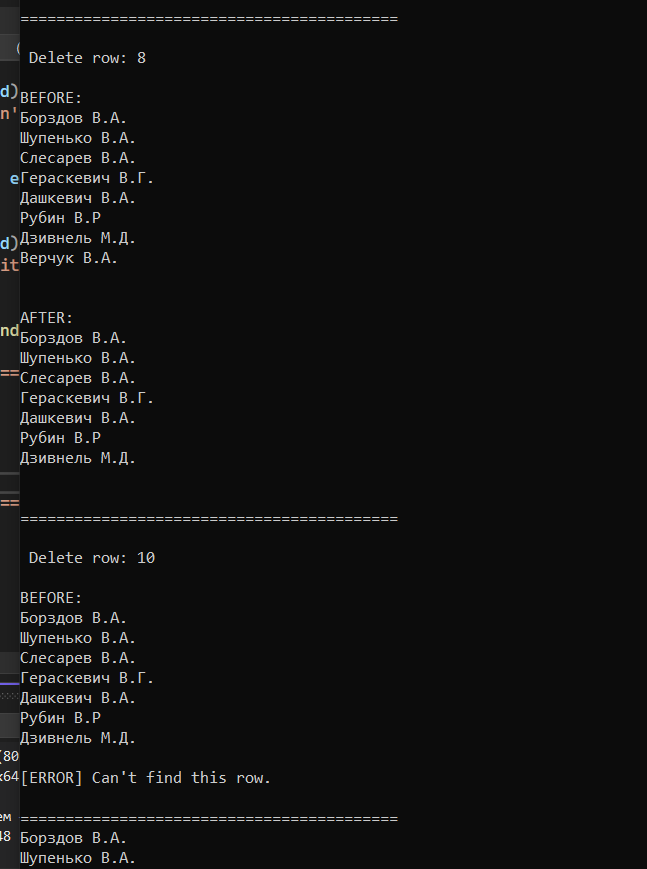


**Задание 02.Windows**

1. Разработайте приложение **OS09\_02.**
2. Приложение **OS09\_02** вызывает функцию **delRowFileTxt**, имеющую следующий прототип.



1. Функция применяется к файлу **OS09\_01.txt** (п.2) и вызывается последовательно 4 раза, с row = 1,3,8,10. Результат выполнения продемонстрируйте с помощью функции **printFileTxt** (п.6).
2. Продемонстрируйте работоспособность приложения **OS09\_02**.
3. 

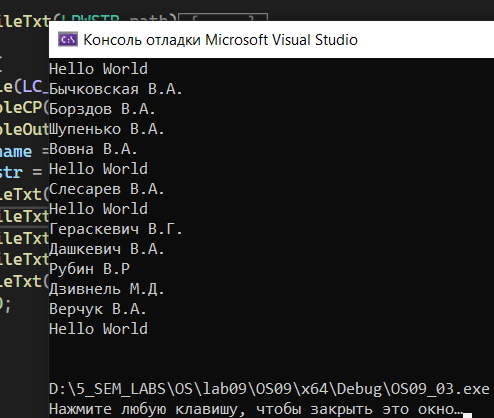


**Задание 03.Windows**

1. Разработайте приложение **OS09\_03.**
2. Приложение **OS09\_03** вызывает функцию **insRowFileTxt**, имеющую следующий прототип.



1. Функция применяется к файлу **OS09\_01.txt** (п.2) и вызывается последовательно 4 раза, с row = 0,-1,5,7. Результат выполнения продемонстрируйте с помощью функции **printFileTxt** (п.6).
2. Продемонстрируйте работоспособность приложения **OS09\_03**.

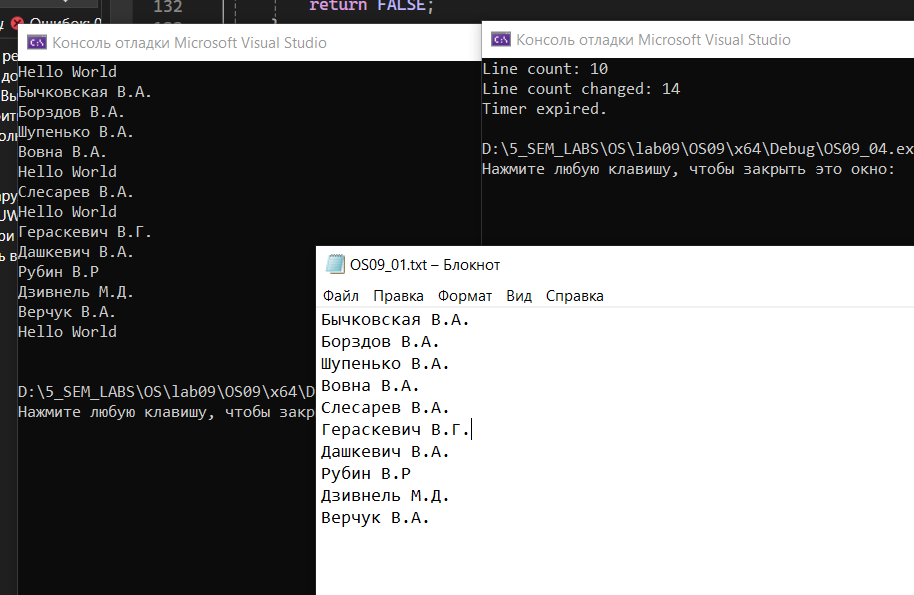


**Задание 04.Windows**

1. Разработайте приложение **OS09\_04.**
2. Приложение **OS09\_04** вызывает функцию **printWathRowFileTxt**, имеющую следующий прототип.

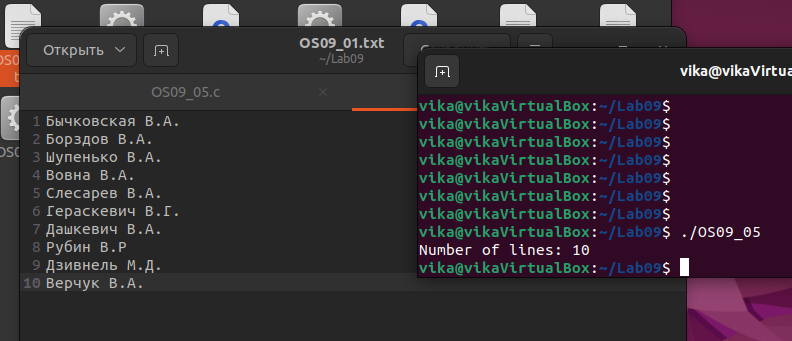


1. Функция применяется к файлу **OS09\_01.txt** (п.2), следит (***используйте функцию наблюдения за файлами в каталоге***) за изменением количества строк в файле в течении mlsec и выводит информацию об изменениях в стандартный поток вывода.
2. Продемонстрируйте работоспособность приложения **OS09\_04** совместно с приложениями **OS09\_03** и **OS09\_04**.



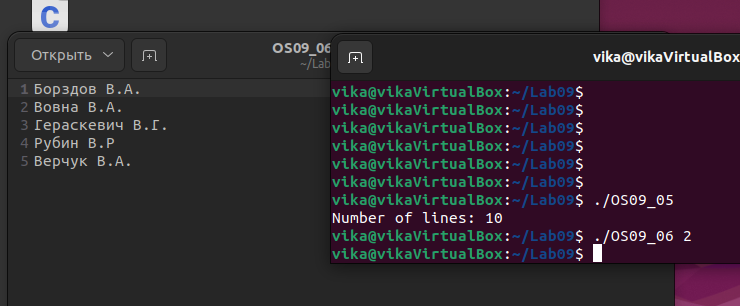
**Задание 05.Linux**

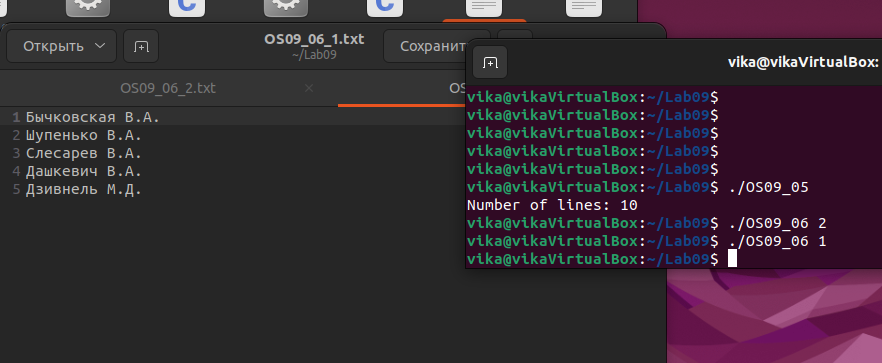
1. Создайте текстовый файл **OS09\_05.txt,** аналогичный файлу **OS09\_01.txt** (п.2).
2. Разработайте приложение **OS09\_05,** подсчитывающее количество строк и выводящее это значение в стандартный поток.
3. Продемонстрируйте работоспособность приложения **OS09\_05**.



**Задание 06.Linux**

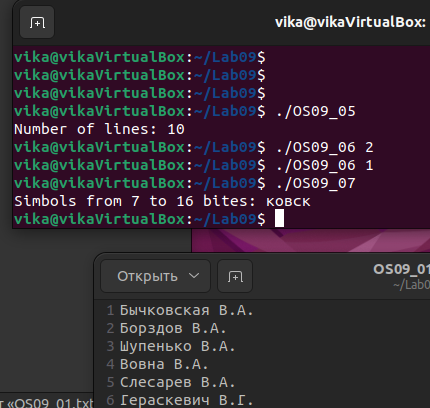
1. Разработайте приложение **OS09\_06,** принимающее 1 параметр, принимающее целочисленное числовое значение.
2. Если числовое значение принимает нечетное значение, то приложение создает новый файл **OS09\_06\_1.txt**, содержащий только нечетные строки из файла **OS09\_05.txt**.
3. Если числовое значение принимает четное значение, то приложение создает новый файл **OS09\_06\_2.txt**, содержащий только четные строки из файла **OS09\_05.txt**.
4. Продемонстрируйте работоспособность приложения **OS09\_06**.





**Задание 07.Linux**

1. Разработайте приложение **OS09\_07,** демонстрирующее возможности функции **lseek.**
2. Поясните назначение и принцип работы функции **lseek**.

****

**31. Что такое файл?**

Файл — это логическая структура данных, используемая для хранения информации на носителе (диск, флешка и т.д.). Файлы могут содержать текст, изображения, аудио, видео или любые другие данные.

**32. Перечислите основные характеристики (атрибуты) файла.**

* **Имя файла**: уникальное название файла в пределах каталога.
* **Размер файла**: объем данных, хранящихся в файле.
* **Тип файла**: определяет формат или содержание файла (например, текст, изображение).
* **Дата и время создания**: время, когда файл был создан.
* **Дата и время последнего изменения**: время, когда файл был последний раз изменен.
* **Атрибуты**: свойства файла, такие как только для чтения, скрытый, системный и т.д.

**33. Что такое файловая система?**

Файловая система — это способ организации, хранения и управления файлами на носителе данных. Она определяет, как данные записываются, читаются и организуются на диске.

**34. Перечислите основные функции файловой системы.**

* Создание и удаление файлов и каталогов.
* Чтение и запись данных в файлы.
* Управление атрибутами файлов.
* Организация доступа к данным.
* Поддержка иерархической структуры каталогов.

**35. Перечислите 3 названия файловой системы.**

* NTFS (New Technology File System)
* FAT32 (File Allocation Table 32)
* ext4 (Fourth Extended Filesystem)

**36. Какая файловая система установлена на вашем компьютере под Windows? под Linux?**

* **Windows**: Обычно NTFS.
* **Linux**: Обычно ext4, но могут использоваться и другие (например, XFS, Btrfs).

**37. Что такое каталог файловой системы? Перечислите наименования специальных каталогов.**

Каталог — это структура, которая содержит ссылки на файлы и другие подкаталоги. Специальные каталоги:

* **Корневой каталог**: основной каталог файловой системы.
* **Текущий каталог**: каталог, в котором в данный момент находится программа.
* **Домашний каталог**: каталог, связанный с конкретным пользователем.

**38. Поясните понятие «текущий каталог приложения».**

Текущий каталог приложения — это каталог, в котором выполняется программа или приложение. Он является базой для относительных путей к файлам.

**39. Что такое специальные имена файлов? Перечислите их, для чего они нужны.**

Специальные имена файлов — это зарезервированные имена, которые имеют особое значение. Примеры:

* . (текущий каталог)
* .. (родительский каталог)
* CON, PRN, AUX, NUL в Windows (особые устройства).  
  Они нужны для упрощения навигации и доступа к системным ресурсам.

**40. Для чего используются буферы ввода-вывода?**

Буферы ввода-вывода используются для временного хранения данных, передаваемых между устройствами ввода-вывода и приложениями. Это позволяет оптимизировать производительность, уменьшая количество операций чтения и записи.

**41. Поясните понятие «кэширование».**

Кэширование — это процесс временного хранения данных для ускорения доступа к ним. Когда данные запрашиваются, система сначала проверяет кэш; если данные там находятся, они считываются быстрее, чем если бы их загружали из основного хранилища.

**42. Поясните понятие «указатель позиции файла».**

Указатель позиции файла — это смещение, которое указывает, где в файле происходит чтение или запись. Он определяет текущее положение в открытом файле.

**43. Поясните понятие «маркер конца файла».**

Маркер конца файла — это специальный индикатор, который указывает на конец содержимого файла. Он используется для обозначения места, где чтение или запись данных должны прекращаться.

**44. Поясните понятие «блокировка файла».**

Блокировка файла — это механизм, который предотвращает одновременный доступ к файлу несколькими процессами. Это необходимо для защиты данных от повреждений или неконсистентности.

**45. Windows. Функция OS API для создания файла.**

Функция CreateFile используется для создания нового файла или открытия существующего.

**46. Windows. Функция OS API для открытия файла.**

Функция CreateFile также используется для открытия файла с различными параметрами доступа.

**47. Windows. Функция OS API для удаления файла.**

Функция DeleteFile используется для удаления указанного файла из файловой системы.

**48. Windows. Функция OS API для записи в файл.**

Функция WriteFile используется для записи данных в открытый файл.

**49. Windows. Функция OS API для чтения файла.**

Функция ReadFile используется для чтения данных из открытого файла.

**50. Windows. Назначение и отличие функций OS API: CopyFile, MoveFile, ReplaceFile.**

* **CopyFile**: копирует файл из одного места в другое.
* **MoveFile**: перемещает файл, удаляя его из исходного местоположения.
* **ReplaceFile**: заменяет файл другим, с возможностью резервного копирования.

**51. Windows. Перечислите функции OS API, которые изменяют текущее значение указателя позиции файла.**

* SetFilePointer
* SetFilePointerEx
* ReadFile
* WriteFile

**52. Windows. Перечислите функции OS API для блокировки и разблокировки файлов.**

* LockFile
* UnlockFile
* LockFileEx
* UnlockFileEx

**53. Windows. Поясните механизм «наблюдение за каталогом», перечислите набор функций OS API, позволяющий реализовать этот механизм.**

Механизм «наблюдение за каталогом» позволяет отслеживать изменения в файловой системе (например, создание, изменение или удаление файлов). Функции для реализации:

* FindFirstChangeNotification
* FindNextChangeNotification
* FindCloseChangeNotification

**54. Windows. Перечислите функции OS API для работы с каталогами, поясните их назначения.**

* CreateDirectory: создает новый каталог.
* RemoveDirectory: удаляет пустой каталог.
* GetFileAttributes: получает атрибуты файла или каталога.
* SetFileAttributes: устанавливает атрибуты файла или каталога.

**55. Linux. Что такое FHS?**

FHS (Filesystem Hierarchy Standard) — это стандарт, описывающий структуру каталогов в операционных системах на базе Linux. Он определяет, какие каталоги должны существовать и что в них должно находиться.

**56. Linux. Перечислите типы файловых систем.**

* ext4
* XFS
* Btrfs
* JFS
* FAT32
* NTFS

**57. Linux. Что такое inode?**

inode — это структура данных в файловой системе, содержащая информацию о файле (например, его размер, права доступа, временные метки и расположение данных на диске). Каждый файл и каталог имеют уникальный inode.

**58. Linux. Поясните назначение функций open, read, write, close, ioctl, stat, flush, lseek, lstat, fstat.**

* **open**: открывает файл и возвращает дескриптор.
* **read**: считывает данные из файла.
* **write**: записывает данные в файл.
* **close**: закрывает открытый файл.
* **ioctl**: управляет устройствами через файловые дескрипторы.
* **stat**: получает информацию о файле (например, размер и атрибуты).
* **flush**: очищает буфер вывода, записывая данные в файл.
* **lseek**: изменяет текущее положение указателя в файле.
* **lstat**: получает информацию о файле, включая символические ссылки.
* **fstat**: получает информацию о файле по дескриптору.