

Работа 3. Управление файловой системой

Цель работы: исследовать управление файловой системой с помощью Win32 API.

Задание 3.1. Управление дисками, каталогами и файлами.

Указания к выполнению.

1. Создайте консольное приложение с меню (каждая выполняемая функция и/или операция должна быть доступна по отдельному пункту меню), которое выполняет:

- вывод списка дисков (функции Win32 API – **GetLogicalDrives**, **GetLogicalDriveStrings**);
- для одного из выбранных дисков вывод информации о диске и размер свободного пространства (функции Win32 API – **GetDriveType**, **GetVolumeInformation**, **GetDiskFreeSpace**);
- создание и удаление заданных каталогов (функции Win32 API – **CreateDirectory**, **RemoveDirectory**);
- создание файлов в новых каталогах (функция Win32 API – **CreateFile**);
- копирование и перемещение файлов между каталогами с возможностью выявления попытки работы с файлами, имеющими совпадающие имена (функции Win32 API – **CopyFile**, **MoveFile**, **MoveFileEx**);
- анализ и изменение атрибутов файлов (функции Win32 API – **GetFileAttributes**, **SetFileAttributes**, **GetFileInformationByHandle**, **GetFileTime**, **SetFileTime**).

2. Запустите приложение и проверьте его работоспособность на нескольких наборах вводимых данных. Запротоколируйте результаты в отчет. Дайте свои комментарии в отчете относительно выполнения функций Win32 API.

3. Перезапустите приложение.

4. Запустите командную строку от имени администратора, перейдите в каталог **c:\Tools\LiveKD** и запустите утилиту *LiveKd.exe*.

5. Определите в *LiveKd* с помощью команды *!process* идентификатор процесса приложения. Запротоколируйте результаты в отчет.

6. В дальнейшем после выполнения каждого пункта меню делайте обновление снимка для *LiveKd* (нажимайте *Ctrl+Break* и затем 'y') и выполняйте с помощью команды *!handle* анализ используемых приложением объектов типа «файл». Протоколируйте результаты в отчет с комментариями изменений и раскрытием их взаимосвязи с выполненными инструкциями приложения.

7. С помощью утилиты *Handle* выполните закрытие открытого приложением объекта типа «файл», затем проверьте реакцию приложения. Запротоколируйте результаты в отчет.

8. Подготовьте итоговый отчет с развернутыми выводами по заданию.

Задание 3.2. Копирование файла с помощью операций перекрывающегося ввода-вывода.

Приложение должно копировать существующий файл в новый файл, «одновременно» выполняя *n* перекрывающихся операций ввода-вывода (механизм APC) блоками данных кратными размеру кластера.

Указания к выполнению.

1. Создайте консольное приложение, которое выполняет:

- открытие/создание файлов (функция Win32 API – **CreateFile**, обязательно использовать флаги FILE_FLAG_NO_BUFFERING и FILE_FLAG_OVERLAPPED);
- файловый ввод-вывод (функции Win32 API – **ReadFileEx**, **WriteFileEx**) блоками кратными размеру кластера;
- ожидание срабатывания вызова функции завершения (функция Win32 API – **SleepEx**);
- измерение продолжительности выполнения операции копирования файла (функция Win32 API – **TimeGetTime**).

2. Запустите приложение и проверьте его работоспособность на копировании файлов разного размера для ситуации с

перекрывающимся выполнением одной операции ввода и одной операции вывода (для сравнения файлов используйте консольную команду **FC**). Выполните эксперимент для разного размера копируемых блоков, постройте график зависимости скорости копирования от размера блока данных. Определите оптимальный размер блока данных, при котором скорость копирования наибольшая. Запротоколируйте результаты в отчет. Дайте свои комментарии в отчете относительно выполнения функций Win32 API.

3. Произведите замеры времени выполнения приложения для разного числа перекрывающихся операций ввода и вывода (1, 2, 4, 8, 12, 16), не забывая проверять работоспособность приложения (консольная команда **FC**). По результатам измерений постройте график зависимости и определите число перекрывающихся операций ввода и вывода, при котором достигается наибольшая скорость копирования файла. Запротоколируйте результаты в отчет.

4. Перезапустите приложение.

5. Запустите командную строку от имени администратора, перейдите в каталог **c:\Tools\LiveKD** и запустите утилиту *LiveKd.exe*.

6. Выполните анализ процесса приложения и его объектов (команды *!process, !handle*). Запротоколируйте результаты в отчет.

7. Подготовьте итоговый отчет с развернутыми выводами по заданию.