# цель работы

Получить навыки работы с механизмом удаленного вызова процедур (RPC).

Реализовать программу-сервер и программу-клиент, работающие под Windows 7-10. Сервер должен предоставлять доступ локальным и удаленным клиентам к файлам в своей файловой системе.

Требования:

- Statefull сервер;

- Сервер не должен быть интерактивным (интерфейс командной строки);

- Взаимодействие с клиентами должно осуществляться с помощью механизма RPC;

- При обслуживании клиента должна осуществляться его имперсонация;

- Пользователю должны предоставляться следующие операции: копирование указанного файла с клиента на сервер, загрузка указанного файла с сервера на клиента, удаление указанного файла на сервере;

- Имя файла передается в формате UNC.

# ход работы

## Интерфейс

Для использования интерфейса RPC, первым делом было необходимо реализовать интерфейс, с описанием функций и их аргументов (рисунок 1)

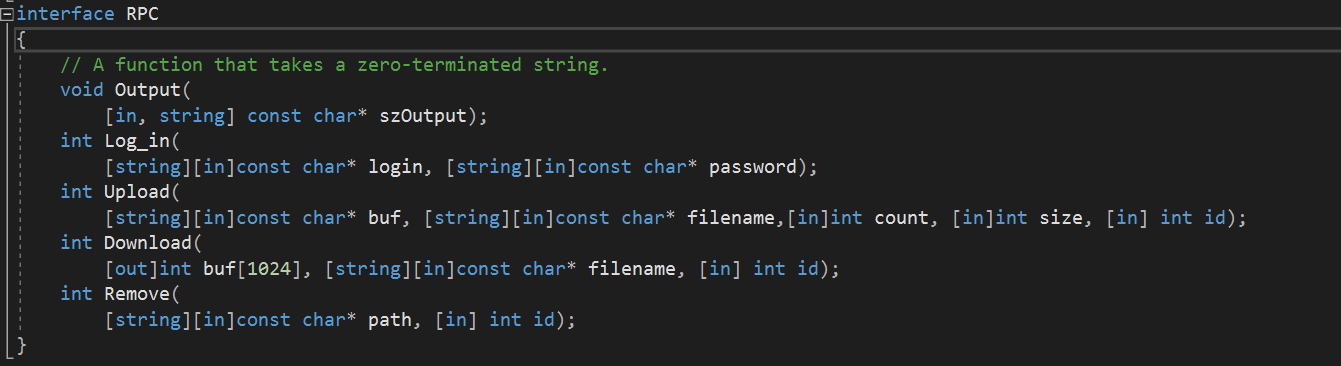


Рисунок 1 – Реализация интерфейса

В данном интерфейсе описаны 5 функций а именно: Вывод строки символов, функция захода на сервер, загрузка, скачивание и удаление какого-либо файла.

Отличительной особенностью описания функций в интерфейсе является то что для каждого аргумента еще необходимо прописать является ли он входным или выходным (in, out).

## Клиент

В реализации клиента были использованы 2 rpc функции: RpcStringBindingComposeA и RpcBindingFromStringBindingA. Первая нужна для создает дескриптор привязки строки, а вторая для возвращения дескриптора привязки из строкового представления дескриптора привязки. После чего уже можно использовать RPC запросы на сервер. После удачного создания дескриптора связывания, пользователю необходимо ввести логин и пароль локального пользователя, чтобы иметь его права доступа (Рисунки 2-3)

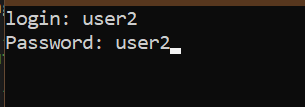


Рисунок 2 – Авторизация пользователя

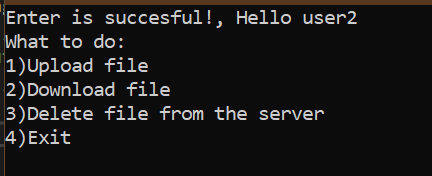


Рисунок 3 – Имеющиеся функции

После успешной авторизации пользователь может использовать следующие функции: Загрузка файла на сервер, скачивание файла с сервера и удаление файла с сервера. Если у пользователя недостаточно прав для какой-либо функции, сервер ответит ему отказом и выведет соответствующее сообщение (рисунок 4)

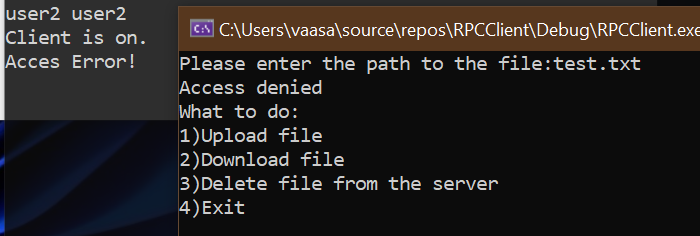


Рисунок 4 – Пример отказа сервера

## Сервер

В реализации сервера использовались следующие RPC функции:

* RpcServerRegisterAuthInfoA - Функция регистрирует информацию об аутентификации в библиотеке времени выполнения RPC.
* RpcServerRegisterIf2 - Регистрирует интерфейс с библиотекой времени выполнения RPC.
* RpcServerListen - Сигнализирует библиотеке времени выполнения RPC прослушивать удаленные вызовы процедур

После удачной регистрации интерфейса сервер готов выполнять удаленные функции.

После получения запроса сервер первым делом вызывает функцию “ImpersonateLoggedOnUser”, для того чтобы выполнять следующие действия от лица соответствующего пользователя. Если у пользователя нет прав для какой-либо операции, сервер отправит ошибку клиенту (Рисунок 4)