#define STOP "STOP" - определяется макрос для константы "STOP".

#define NAME L[\\\\Across\\pipe\\Tube](file:///\\\\Across\\pipe\\Tube) - определяется макрос для имени канала.

using namespace std;

int main()

{

setlocale(LC\_ALL, "rus");

HANDLE cH; - объявляется дескриптор канала.

DWORD lp; объявляется переменная для хранения количества переданных или полученных байтов.

char ibuf[50], obuf[50]; - объявляются буферы для передачи и приема данных по каналу.

try {

cout << "ClientNPсt\n\n"; - выводится информация о клиенте.

int countMessage; - объявляется переменная для хранения количества сообщений.

cout << "Number of messages: "; - выводится запрос на ввод количества сообщений.

cin >> countMessage; - считывается количество сообщений.

for (int i = 1; i <= countMessage; i++) { - цикл для отправки заданного количества сообщений.

string obufstr = "Hello from ClientNPct " + to\_string(i); - формируется строка-сообщение для отправки.

strcpy\_s(obuf, obufstr.c\_str()); - строка-сообщение преобразуется в массив символов и копируется в выходной буфер.

if (!CallNamedPipe(NAME, obuf, sizeof(obuf), ibuf, sizeof(ibuf), &lp, NMPWAIT\_WAIT\_FOREVER)) { - вызывается функция CallNamedPipe для отправки сообщения по именованному каналу и получения ответа. Функция блокирует выполнение до получения ответа или истечения заданного времени NMPWAIT\_WAIT\_FOREVER. В случае ошибки функции выбрасывается исключение.

throw SetPipeError("CallNamedPipe: ", GetLastError());

}

cout << ibuf << endl; - выводится полученный ответ из канала.

}

system("pause");

}

catch (string ErrorPipeText) {

cout << endl << ErrorPipeText;

}

}