NAME

server - multi-client server that listens for and executes client transactions on the specified port

SYNOPSIS

server [port]

DESCRIPTION

server is the consumer of the client-server application that consumes transactions sent by clients.

Messages sent to the server must be in the following format:
`machinename.pid \ t T<n> \ t`
Where **machinename** is the name of the client sending the message, **pid** is the process id of the client that acts as a unique identifier, **T<n>** is the transaction with n being the input to the transaction, and **\ t** is the delimiter.

The server runs as follows:

1. Creates a socket and binds to the port
2. Listens for client connections and connects to clients
3. Executes client transactions sent to the server
4. Returns Done receipt to client along with its transaction id
5. Logs all activity into ‘server.log‘
6. Timeouts out after 30 seconds of inactivity

NOTES

This multi-client server does not use threads

If no <port> is provided, then the server falls back to port 8888

EXAMPLES

./server*8000*

Start a server that listens on port 8000

OPTIONS

<port> The port number that the server listens to for communications from clients. The port must be in the range 5,000 to 64,000

FILES

server.cpp

main.h

tands.cpp

utils.cpp

DIAGNOSTICS

The activity performed by this server is logged in the ‘server.log‘ file

COPYRIGHT

CREDITS

https://www.binarytides.com/server-client-example-c-sockets-linux/
https://www.geeksforgeeks.org/socket-programming-in-cc-handling-multiple-clients-on-server-without-multi-threading/