

BANKINGSERVER.C

bankingServer.c is a program that creates a banking server that will accept commands from connected clients to create and modify accounts.

DESCRIPTION

LISTENCONNECTIONS:

void* listenConnections(void* ptrListenSock);

This function listens for incoming connections from clients.

CLIENTSESSIONS:

void* clientSession(void* args);

After connecting to a client, this function is called and handles the parsing and execution of the commands that the client gives the server.

SIGNAL_HANDLER:

void* signal_handler(void* args)

This function handles any signals that may be sent by the server.

OPTIONS

The program takes as its second parameter a port number. This is a necessary parameter.

EXAMPLES

./bankingServer 19231

SEE ALSO

banking.h

bankingClient.c

BUGS

None currently known.

AUTHORS:

Eric Lin and Mohammad Nadeem

BANKINGCLIENT.C

bankingClient.c creates a client that can connect to the server and send the server all input and get all input from the user, parses it and sends relevant commands to the server. It then gets server responses and prints them out to STDIN.

DESCRIPTION

ISNUMERIC:

```
int isNumeric(char * string)
```

PARSEINPUT:

```
char* parseInput(char * input)
```

GET_AND_PRINT:

```
void* get_and_print(void *sf_p)
```

GET_AND_SEND:

```
void* get_and_send(void *sf_p)
```

OPTIONS

The program has two necessary parameters. The first one is the URL to connect to while the second parameter is the port number.

EXAMPLES

```
./bankingClient less.cs.rutgers.edu 19231
```

SEE ALSO

banking.h

bankingServer.c

BUGS

None currently known.

AUTHORS:

Eric Lin and Mohammad Nadeem

BANKING.H

banking.h is a header file that describes necessary structs for the server.

DESCRIPTION

```
struct _Account
```

This is a struct that holds relevant information for accounts.

```
struct _Account{
    char* name;
    double balance;
    int inService;
    pthread_mutex_t* serviceLock;
}
```

char * name describes the account name. The balance field describes the balance of the account. The serviceLock variable allows for the data in the Account to be locked with a mutex to prevent errors. The inService field denotes if the Account is currently being serviced. It is 1 if it is and 0 otherwise.

```
struct _sigArgs
```

This is a struct that holds relevant information for connecting to clients.

```
struct _sigArgs {  
    int *listenSockFD;  
    sigset_t * sigSet;  
    pthread_t * listenThread;  
}
```

listenSockFD is the file descriptor for the socket. sigSet allows us to define a custom signal handler. listenThread is the thread that listens to the socket.

SEE ALSO

bankingClient.c

bankingServer.c

BUGS

None currently known.

AUTHORS:

Eric Lin and Mohammad Nadeem

ISSUES:

We had some issues with parsing the input and allowing for an account names with spaces. We also had some issues with getting the client to shutdown when the server was terminated. The problem we were having was that fgets() was preventing shutdown until input was entered. In the end, we had to remotely kill the thread to get the client to shutdown.

TESTING METHODOLOGY:

We tested for connectivity by trying multiple different CS machines. Then, we tested for creating account names. We tested account names with numbers, letters, punctuation, spaces and even tested an account name with an empty string "". Then, we checked to make sure all of these accounts could be served. Then, we tried depositing money in both integer and decimal amount. We then tested withdrawals of different kinds of amounts. We also tested the quit, end and query commands. We also tested multiple client connections.