

COMP 7005

Assignment 1

Design

Gursidh Sandhu
A01319563
Sept 21, 2024

Purpose	4
Data Types	4
Arguments	4
Settings	4
Context	5
Functions	5
server.cpp	5
client.cpp	6
Pseudocode	6
Server.cpp	6
create_socket	6
Parameters	6
Return	6
Pseudo Code	6
bind_socket	7
Parameters	7
Return	7
Pseudo Code	7
listen_on_socket	7
Parameters	7
Return	7
Pseudo Code	7
accept_connection	8
Parameters	8
Return	8
Pseudo Code	8
handle_request	8
Parameters	8
Return	8
Pseudo Code	9
locate_file_contents	9
Parameters	9
Return	9
Pseudo Code	9
send_response	10
Parameters	10
Return	10
Pseudo Code	10
close_socket	10
Parameters	10

Return	10
Pseudo Code	10
sigint_handler	11
Parameters	11
Return	11
Pseudo Code	11
setup_signal_handler	11
Parameters	11
Return	11
Pseudo Code	11
Client.cpp	12
create_socket	12
Parameters	12
Return	12
Pseudo Code	12
attempt_socket_connection	12
Parameters	12
Return	12
Pseudo Code	12
send_request	13
Parameters	13
Return	13
Pseudo Code	13
handle_response	13
Parameters	13
Return	13
Pseudo Code	13
print_file_contents	14
Parameters	14
Return	14
Pseudo Code	14
close_socket	14
Parameters	14
Return	14
Pseudo Code	14

Purpose

- This assignment is split up into two programs
 - Server.cpp
 - This program accepts 1 argument from the command line:
 - Socket path
 - Client.cpp
 - This program accepts 2 arguments from the command line:
 - Socket path
 - filename
- Server.cpp will return the contents of the specified filename provided back to the client
 - But **file must exist in the 'source/include' directory in project structure**
- Client.cpp will print out the file contents to screen

Data Types

Arguments

Purpose: To hold the unparsed command-line argument information

Field	Type	Description
argc	integer	The number of arguments
argv	string[]	The arguments
program_name	string	The name of the program
socket_path	string	The socket path where binding/connecting will be attempted
filename	string	The file that is to be looked for by the server

Settings

Purpose: To hold the settings the program needs to run.

Field	Type	Description
DEFAULTFD VALUE	integer	The default file descriptor value, should be changed during program execution when creating socket
NUMBOFAR GS	integer	The number of arguments expected for program, 2 for server and 3 for client

BUFFERSIZE	integer	The buffer size to hold incoming data during read function
EOFINDICATOR	string	Used to indicate end of file when reading data

Context

Purpose: To hold the arguments, settings, and exit information

Field	Type	Description
arguments	Arguments	The command line arguments
settings	Settings	The parsed command line arguments
exit_flag	sig_atomic	The signal flag to exit the program

Functions

server.cpp

Function	Description
create_socket	Creates the socket and returns file descriptor
bind_socket	Binds the socket to passed in socket path
listen_on_socket	Listen for any incoming requests to connect on socket
accept_connection	Accept any valid incoming connection and return new socket
handle_request	Process the incoming request from client
locate_file_contents	Find the file on the machine with the passed in value
send_response	Send back the file contents to the client that made request
close_socket	Close the socket and free up any resources
sigint_handler	Called by handler function, changes value of exit flag
setup_signal_handler	Sets up the functionality of when the exit flag should be set

client.cpp

Function	Description
create_socket	Creates the socket and returns file descriptor
attempt_socket_connection	Attempt to connect to the server associated with socket path
send_request	Send request to server including the filename argument
handle_response	Handle the response back from server
print_file_contents	Print out the response from server which is the file contents
close_socket	Close the socket and free up any resources

Pseudocode

Server.cpp

create_socket

Parameters

None.

Return

Value	Reason
socketFD	The socket was created successfully

Pseudo Code

```
set socketFD = new socket

if socketFD shows error
    Print "error in creating socket"
    Exit program
return socketFD
```

bind_socket

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket
socketPath	Const string	Command line argument that represents where to bind the socket

Return

No returns.

Pseudo Code

```
if bind on socketPath shows error
    Print "error in binding socket"
    Exit program
```

```
Print "socket successfully binded"
```

listen_on_socket

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket
maxConnections	integer	Maximum amount of client connections that can be held in queue at a time

Return

No returns.

Pseudo Code

```
if listening on socket shows error
    Print "error in listening on socket"
    Exit program
```

```
Print "listening for connections"
```

accept_connection

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket

Return

Value	Reason
clientFD	This client was successfully connected to the socket

Pseudo Code

```
Set clientFD = accept connection on socketFD
If clientFD shows error
    Print "error connecting to client"
    Return -1
Print "client successfully connected"
Return clientFD
```

handle_request

Parameters

Parameter	Type	Description
clientFD	integer	The client file descriptor that is connected to socketFD

Return

Value	Reason
-------	--------

fileName	The file that client is requesting contents for
----------	---

Pseudo Code

```

Set request = read request using clientFD
If request shows error
    Print "error reading client request"
    Exit program
Else
    Set fileName = extract fileName from request
Return fileName

```

locate_file_contents

Parameters

Parameter	Type	Description
fileName	Const string	The file to be looked for and return its contents

Return

Value	Reason
fileContents	The contents of the file were successfully found

Pseudo Code

```

Set fileContents to empty list
Search current program directory for fileName
If fileName not found
    Print "could not find file"
    Exit program
Else
    Loop through file and copy data to fileContents
Return fileContents

```

send_response

Parameters

Parameter	Type	Description
clientFD	integer	The client file descriptor that is connected to socketFD
fileContents	String vector	The array that contains file contents to be sent back in response to client

Return

No returns.

Pseudo Code

```
send fileContents to client using clientFD
If send shows error
    Print "error sending response to client"
    Exit program
Else
    Print "response sent successfully"
```

close_socket

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket
socketPath	Const string	The socket path on which binding of socket occurred

Return

No returns.

Pseudo Code

```
If closing socket shows error
    Print "error closing socket"
    Exit program
```

```
Else
    Print "socket closed successfully"
```

sigint_handler

Parameters

Parameter	Type	Description
signum	integer	The signal number that was received by program

Return

No returns.

Pseudo Code

```
Set flag to true
```

setup_signal_handler

Parameters

None.

Return

No returns.

Pseudo Code

```
Setup all the configurations to handle interrupts
Catch CTRL-C by user
```

Client.cpp

create_socket

Parameters

None.

Return

Value	Reason
socketFD	The socket was created successfully

Pseudo Code

```
set socketFD = new socket

if socketFD shows error
    Print "error in creating socket"
    Exit program

return socketFD
```

attempt_socket_connection

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket
socketPath	string	Command line argument that represents where socket is located

Return

No returns.

Pseudo Code

```
Attempt connection on socketPath using socketFD
If socketFD shows error
    Print "error connecting to server socket"
```

```
        Exit program
Else
    Print "successfully connected to server socket"
```

send_request

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket
fileName	string	The fileName that server needs to return contents of, passed in as a command line argument

Return

No returns.

Pseudo Code

```
send request containing fileName to server using socketFD
If request shows error
    Print "error sending request to server"
    Exit program
Else
    Print "successfully sent request to server"
```

handle_response

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket

Return

Value	Reason
fileContents	The fileContents were successfully found from server response

Pseudo Code

```
Wait for response from server using socketFD
```

```
If response shows error
    Print "error handling response from server"
    Exit program
Else
    Set fileContents = extract fileContents from response
Return fileContents
```

print_file_contents

Parameters

Parameter	Type	Description
fileContents	String vector	The array that contains file contents sent from server

Return

No returns.

Pseudo Code

```
Loop through fileContents and print each line
```

close_socket

Parameters

Parameter	Type	Description
socketFD	integer	The socket file descriptor that is a reference to a socket

Return

No returns.

Pseudo Code

```
If closing socket shows error
    Print "error closing socket"
    Exit program
Else
    Print "socket closed successfully"
```