

COMP 7005

Computer Networks & Protocols

Assignment-01

Design

Anmol Mittal

A01397754

January 26, 2025

Course Reference Number (CRN): 92522

Purpose.....	4
Data Types	4
Arguments	4
Settings	4
Functions	5
Client.....	5
Server	6
States.....	6
Client.....	6
Server	6
State Table.....	7
Client.....	7
Server	7
State Transition Diagram	8
Pseudocode	9
Client.....	9
validate_arguments:	9
prepare_filename:.....	9
open_file:.....	10
get_file_size:	10
read_file_content:.....	10
close_file:	11
create_client_socket:.....	11
connect_to_server:	12
send_message_to_server:.....	13
receive_server_response:	13
close_socket:	14
Server	15
validate_arguments:	15

setup_server_socket:	15
accept_client_connections:	16
process_client_message:	17
caesar_encrypt:	18
handle_signal:	19
cleanup:	19

Purpose

The programs implement inter-process communication using UNIX domain sockets. The server encrypts messages using the Caesar cipher with a shift provided at runtime, while the client reads a file, sends its content to the server, and displays the encrypted message received in response.

Both server and client program accept one command line argument as follows:

- ./server <Shift Value in Integer>
- ./client <file location>

Data Types

Arguments

Client

Field	Type	Description
argv	string[]	arguments
argc	integer	number of arguments
program_name	string	name of the program
file name	string	name of the file to encrypt

Server

Field	Type	Description
argv	string[]	arguments
argc	integer	number of arguments
program_name	string	name of the program
shift value	integer	value to shift alphabets by

Settings

Client

Field	Value	Description
S_PATH	string	Path for the UNIX domain socket file.
FILE_NAME_SIZE	integer	Maximum length of file name.
BUFFER_SIZE	integer	Size of the buffer for communication.

Server

Field	Value	Description
S_PATH	string	Path for the UNIX domain socket file.
BACKLOG	integer	Maximum number of pending connections.
BUFFER_SIZE	integer	Size of the buffer for communication.

Context

Client

Field	Value	Description
File	string	Reads the file content and manages communication with the server.
EXIT		Programs exist automatically after a successful or failed connection.

Server

Field	Value	Description
Shift Value	integer	Reads the shift value and shifts the context received by this value.
Handle Signal	integer	Handle SIGINT for graceful shutdown.

Functions

Client

Function	Description
main	Validates arguments, reads the file, and communicates with the server.
validate_arguments	Checks command-line arguments for validity.
prepare_filename	Copies and sanitizes the provided file name.
open_file	Opens the file safely for reading.
get_file_size	Determines the size of the file.
read_file_content	Reads file content into memory.
close_file	Closes the file safely.
create_client_socket	Creates a UNIX domain socket for the client.
connect_to_server	Establishes a connection to the server.
send_message_to_server	Sends file content to the server.
receive_server_response	Receives and displays the server's encrypted response.
close_socket	Closes the client socket.

Server

Function	Description
main	Validates arguments, initializes the server, and handles connections.
validate_arguments	Parses and validates command-line arguments for shift value.
setup_server_socket	Sets up the server's UNIX domain socket and starts listening.
accept_client_connections	Accepts client connections and processes messages.
process_client_message	Receives, encrypts, and sends back client messages.
caesar_encrypt	Encrypts a given message using the Caesar cipher.
handle_signal	Handles CTRL+C signal to perform a graceful shutdown.
cleanup	Closes the socket and removes the socket file.

States

Client

State	Description
START	Start client program and read the input file.
VALIDATE	Check file name and ensure all arguments are correct.
OPEN FILE	Get the file ready to write its content to the buffer.
READ	Read file contents to the Buffer.
INIT_SOCKET	Set up the UNIX domain socket.
CONNECT	Establish a connection to the server.
SEND	Transmit file content to the server.
RECEIVE	Receive and display the encrypted response.
CLEANUP/ CLOSE SOCKET	Close the connection and free memory.

Server

State	Description
START	Start server program and read the shift value.
VALIDATE	Check shift value and ensure all arguments are correct.
INIT_SOCKET	Set up the UNIX domain socket.
LISTEN	Wait for client connections.
ACCEPT	Accept incoming client connection.
PROCESS	Encrypt incoming messages and send back the response.
CLEANUP	Release resources and exit.

State Table

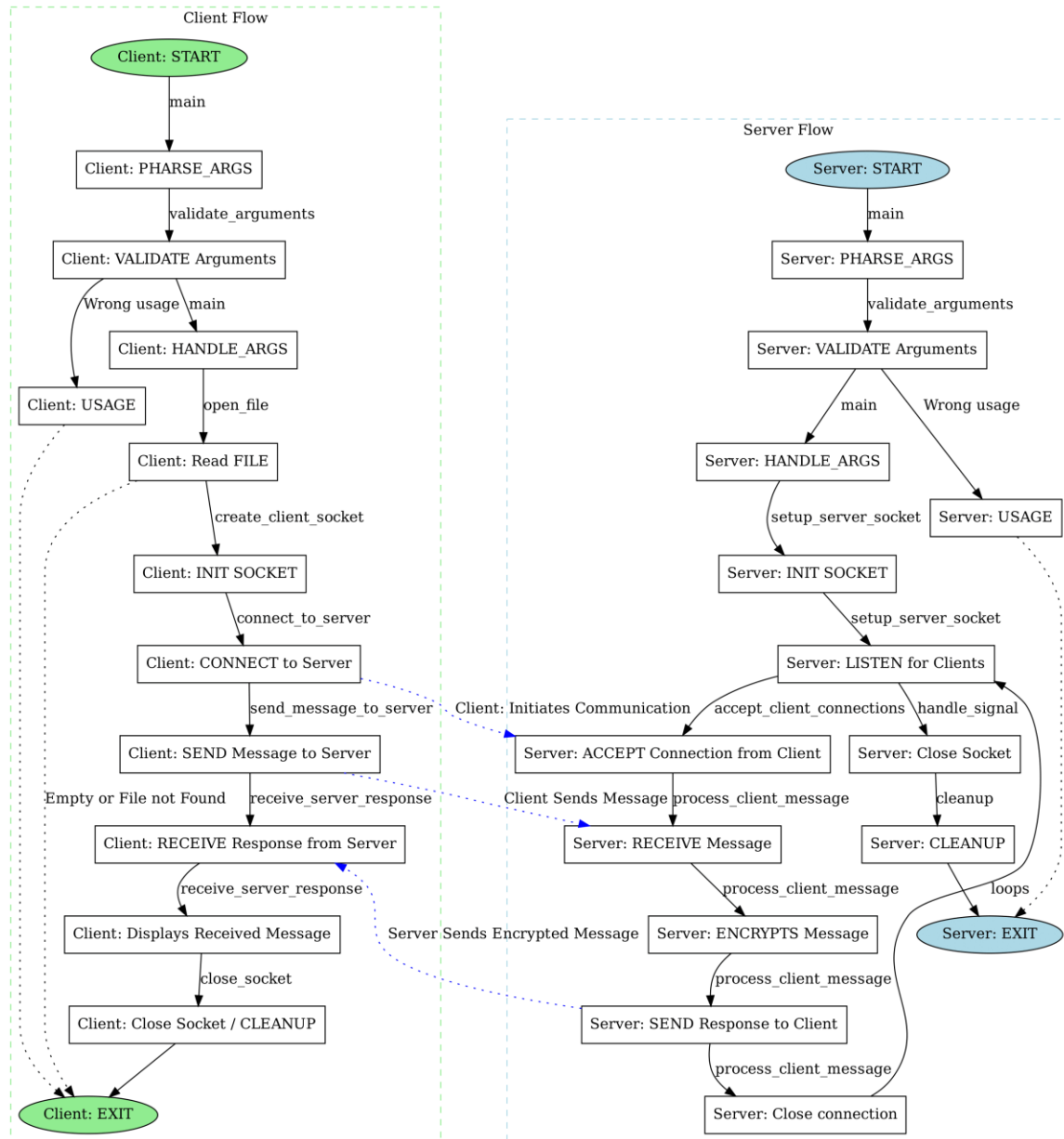
Client

From State	To State	Function
START	VALIDATE	validate_arguments
VALIDATE	OPEN FILE	prepare_filename
OPEN FILE	READ	open_file
READ	INIT_SOCKET	create_client_socket
INIT_SOCKET	CONNECT	connect_to_server
CONNECT	SEND	send_message_to_server
SEND	RECEIVE	receive_server_response
RECEIVE	CLEANUP/ CLOSE SOCKET	close_socket

Server

From State	To State	Description
START	VALIDATE	validate_arguments
VALIDATE	INIT_SOCKET	setup_server_socket
INIT_SOCKET	LISTEN	setup_server_socket
LISTEN	ACCEPT	accept_client_connections
ACCEPT	PROCESS	process_client_message
PROCESS	LISTEN	process_client_message
PROCESS	CLEANUP	handle_signal
CLEANUP	EXIT	cleanup

State Transition Diagram



Pseudocode

Client

validate_arguments:

Parameters

Parameter	Type	Description
argument_count	integer	The passed Arguments

Return

Value	Reason
None	

Pseudo Code

validate_arguments:

if the number of arguments is not 2:

print error message

terminate with failure status

prepare_filename:

Parameters

Parameter	Type	Description
dest	character string	A string to store the resulting filename
src	character string	A string containing the original filename

Return

Value	Reason
None	

Pseudo Code

prepare_filename:

copy up to (FILE_NAME_SIZE - 1) characters from source to destination

set the last character of destination to null-terminator

open_file:

Parameters

Parameter	Type	Description
filename	character string	A string containing the name of the file to be opened

Return

Value	Reason
filename	A reference or handle to the opened file (used for interacting with the file)

Pseudo Code

open_file:

```
try to open the file in read mode using the provided filename
if the file cannot be opened:
    print an error message to output
    terminate with failure status
return the file handle or reference
```

get_file_size:

Parameters

Parameter	Type	Description
file	A reference or handle	Determine file size

Return

Value	Reason
integer or long	Representing the size of the file in bytes.

Pseudo Code

get_file_size:

```
find file size
return the file size
```

read_file_content:

Parameters

Parameter	Type	Description
file	A reference or handle	To read the file.
file_size	unsigned integer or long	To know how many bytes to read.

Return

Value	Reason
string	File's content, null-terminated

Pseudo Code

read_file_content:

- create a container (e.g., string or buffer) to hold the file content, sized to fit the file's size
- if the container cannot be created:
 - print an error message
 - terminate with failure status

- read the entire file into the container
- ensure the container has an end-of-content marker (if required by the system)
- return the container containing the file's content

close_file:

Parameters

Parameter	Type	Description
file	A reference or handle	Need to be closed.

Return

Value	Reason
None	

Pseudo Code

close_file:

- if the file is valid:
 - close the file

create_client_socket:

Parameters

Parameter	Type	Description
None		

Return

Value	Reason
whole number	Represents the created socket, used for communication

Pseudo Code

`create_client_socket`:

- attempt to create a socket with appropriate settings (e.g., communication type and protocol)

- if socket creation fails:

- print an error message

- terminate with failure status

- return the socket descriptor or handle

`connect_to_server`:

Parameters

Parameter	Type	Description
<code>client_socket</code>	A reference or handle	Handle to the client-side communication channel

Return

Value	Reason
NA	

Pseudo Code

`connect_to_server`:

- create an address structure to specify server details

- initialize the address structure with default values

- set the address family to match the communication protocol

- set the server path or identifier

- attempt to connect to the server using the provided socket and address

- if the connection fails:

- print an error message

- close the client socket to release resources

- terminate with failure status

- print a message indicating the connection was successful

send_message_to_server:

Parameters

Parameter	Type	Description
client_socket	A reference or handle	Handle to the client-side communication channel
message	string	Message to be sent
size	whole number	Size of the message

Return

Value	Reason
None	

Pseudo Code

send_message_to_server:

 attempt to send the specified message through the client socket

 if the message cannot be sent:

 print an error message

 close the client socket to release resources

 terminate with failure status

 print a message indicating the message was successfully sent

receive_server_response:

Parameters

Parameter	Type	Description
client_socket	A reference or handle	handle representing the client socket used to receive data

Return

Value	Reason
None	

Pseudo Code

receive_server_response:

- create a buffer to hold incoming data
- initialize a flag to indicate if receiving is complete

print a message indicating the start of response reception

while not done receiving:

- receive data from the server into the buffer
- if data is received:
 - add an end-of-content marker to the buffer
 - display the received data
 - if the received data is less than the buffer size:
 - mark receiving as complete
- if no data is received:
 - mark receiving as complete and print
- if an error occurs:
 - print an error message
 - mark receiving as complete

print a message indicating disconnection from the server

close_socket:

Parameters

Parameter	Type	Description
client_socket	A reference or handle	Identifier for the communication channel to be terminated

Return

Value	Reason
None	

Pseudo Code

close_socket:

- close the communication channel represented by the socket

Server

validate_arguments:

Parameters

Parameter	Type	Description
client_socket	A reference or handle	Identifier for the communication channel to be terminated

Return

Value	Reason
None	

Pseudo Code

validate_arguments:

- if the number of arguments is not 2:
 - print an error message
 - terminate with failure status

- attempt to convert the second argument to a number (shift value)
- if the conversion is invalid (non-integer input):
 - print an error message "Error: Shift must be an integer"
 - terminate with failure status

setup_server_socket:

Parameters

Parameter	Type	Description
None		

Return

Value	Reason
whole number	Represents the server socket, used for accepting connections.

Pseudo Code

setup_server_socket:

- create a socket for server communication with appropriate settings (e.g., communication type and protocol)

- if socket creation fails:

 - print an error message

 - terminate with failure status

- create an address structure to store server details

- initialize the address structure with default values

- set the address family to match the communication protocol

- set the server path or identifier

- if the server path already exists:

 - remove the existing server path

- attempt to bind the server socket to the address

- if binding fails:

 - print an error message

 - clean up resources and terminate with failure status

- attempt to listen for incoming client connections

- if listening fails:

 - print an error message

 - clean up resources and terminate with failure status

- print a message indicating the server is listening

- return the server socket descriptor or handle

accept_client_connections:

Parameters

Parameter	Type	Description
server_socket	A reference or handle	Handle for the server's communication channel that listens for incoming connections.
shift	integer	A value to be applied to the client's message during processing.

Return

Value	Reason
None	

Pseudo Code

accept_client_connections:

- enter an infinite loop to continuously wait for client connections

- print a message indicating the server is waiting for a client

- attempt to accept a client connection

- if the client connection fails:

 - print an error message

 - continue to the next iteration to wait for another client

- print a message indicating the client is connected

- process the client's message using the provided shift value

- close the client socket to terminate the connection

- print a message indicating the client has disconnected

process_client_message:

Parameters

Parameter	Type	Description
server_socket	A reference or handle	Handle for the server's communication channel that listens for incoming connections.
shift	integer	A value to be applied to the client's message during processing.

Return

Value	Reason
None	

Pseudo Code

process_client_message:

- create a buffer to hold incoming data

- initialize a flag to indicate when receiving is complete

enter a loop to continuously receive data from the client

while not done receiving:

- clear the buffer

- receive data from the client into the buffer

- if data is received:

 - add an end-of-content marker to the buffer

 - apply a transformation (e.g., encryption) to the data using the provided shift value

 - send the transformed data back to the client

- if the received data is less than the buffer size:

 - mark receiving as complete

- if no data is received:

 - mark receiving as complete

- if an error occurs during receiving:

 - print an error message

 - mark receiving as complete

print a message indicating that the process is complete and data has been sent

caesar_encrypt:

Parameters

Parameter	Type	Description
message	character string	Characters representing the message to be encrypted.
shift	integer	A value to be applied to the client's message during processing.

Return

Value	Reason
None	

Pseudo Code

function caesar_encrypt:

- loop through each character of the message until the end of the string
 - if the character is a lowercase letter:
 - apply the shift to the character within the range of lowercase letters
 - if the character is an uppercase letter:
 - apply the shift to the character within the range of uppercase letters
- return the encrypted message

handle_signal:

Parameters

Parameter	Type	Description
signal	integer	Interrupt the server.

Return

Value	Reason
None	

Pseudo Code

handle_signal:

- if the received signal is SIGINT (Ctrl+C):
 - print a message indicating the signal was caught and the program is shutting down
 - clean up resources
 - terminate the program successfully

cleanup:

Parameters

Parameter	Type	Description
None		

Return

Value	Reason
None	

Pseudo Code

cleanup:

- if the server socket is valid:

 - close the server socket

 - print a message indicating the socket was closed

attempt to remove the socket file:

- if the removal is successful:

 - print a message indicating the socket file was unlinked

- else:

 - print an error message indicating failure to remove the socket file