A Concurrent File Access System Report

Emre Yavuz 200104004003

Introduction of server.c

The **server.c** program represents a robust server implementation designed to handle multiple client connections concurrently. This report provides an in-depth analysis of the program's architecture, functionality, key features, and implementation details.

Architecture and Components

The server.c program is a robust server implementation designed to handle multiple client connections concurrently. It's structured around several key components and mechanisms that manage client-server interactions, file operations, signal handling, and process management.**Initialization (initialize_server)**:

1. **Initialization (initialize_server):** The server begins its operation by initializing certain components. It creates a specified directory and opens a log file within that directory. The server then prepares to accept client connections. It also sets up a semaphore (sem) for resource synchronization, ensuring that multiple processes can access shared resources without conflicts.
2. **Client Connection Handling (accept_client, handle_client_connection):** The server waits for incoming client connections via a named pipe (FIFO). Once a client connects, the server forks a new process to handle that client's requests independently. This allows the server to handle multiple clients concurrently. Each client is assigned a unique client name and manages bidirectional communication with the server using a dedicated FIFO.
3. **Command Handling (handle_client_request):** The server is capable of parsing and executing commands received from clients. Supported commands include:
    - list: Lists files in the server's directory.
    - readF: Reads specific lines or entire files from the server.
    - writeT: Writes content to specific lines in server files.
    - upload: Transfers files from clients to the server.
    - download: Sends files from the server to clients.
    - archServer: Archives server files into a tarball.
    - killServer: Terminates the server.
    - quit: Disconnects a client from the server.

- Detailed help messages (help <command>) are available to guide clients on command usage.

4. **File Operations (handle_readF_command, handle_writeT_command, handle_upload_command, handle_download_command):** The server performs various file operations based on client requests. These operations include reading, writing, uploading, and downloading files. File transfers are managed securely using FIFOs and appropriate file handling techniques to ensure data integrity and security.
5. **Signal Handling (handle_kill_signal, handle_child_termination, handle_sigint):** The server has implemented signal handlers to manage different signals (SIGTERM, SIGCHLD, SIGINT) gracefully. This ensures proper cleanup and termination of processes and resources when the server is interrupted or terminated.

Implementation Details

- **Concurrency and Synchronization: The server uses a semaphore (sem) to manage concurrency and ensure that critical sections (e.g., file operations) are executed safely. This prevents race conditions and ensures data consistency.**
- **Error Handling: System calls and file operations are checked for errors, and appropriate error messages are logged or communicated to clients. This helps in debugging and ensures that clients are informed of any issues.**
- **Process Management: Child processes are forked to handle individual client connections, maintaining responsiveness and scalability. This allows the server to handle multiple clients without slowing down.**
- **Communication Protocol: Clients communicate with the server using named pipes (FIFOs), enabling bidirectional data transfer and command execution. This allows for efficient and reliable communication between the server and its clients.**

For testing and validation, the server program should be thoroughly tested to ensure robustness, reliability, and security. Testing scenarios should include concurrent client connections, various file operations (reading, writing, uploading, downloading), handling of edge cases (e.g., empty directories, invalid commands), and signal handling and process termination scenarios.

In terms of future enhancements, several recommendations can be made:

- **Security Enhancements: Implement authentication mechanisms to secure client-server interactions. Enforce access controls for file operations based on user permissions.**

- **Performance Optimization: Optimize file transfer mechanisms for efficiency, especially when dealing with large files or multiple concurrent clients.**
- **Logging and Monitoring: Enhance logging capabilities to capture detailed information for debugging and auditing purposes. Implement monitoring tools to track server performance and resource usage.**
- **User Interface: Develop a user-friendly interface or command-line interface (CLI) for interacting with the server, simplifying command execution and management.**

In conclusion, the server.c program provides a solid foundation for a multi-client server application capable of handling diverse file-related operations and client requests. By following best practices in error handling, concurrency management, and signal processing, the server ensures reliability and responsiveness in real-world deployment scenarios.

Introduction of client.c

The client.c program is a comprehensive client-side application designed to interact with a server program via named pipes (FIFOs). It is part of a larger system that allows multiple clients to connect to a server and perform various file operations concurrently. The client program is responsible for establishing a connection with the server, sending requests, handling server responses, and managing file transfers. It also implements signal handling to ensure graceful termination of the program.

Architecture and Components

The client program is structured around various functions and mechanisms responsible for establishing communication with the server, sending requests, handling server responses, managing file transfers, and implementing signal handling. Here's an overview of its main components:

1. **Connecting to the Server (connect_to_server): The client establishes a connection with the server by writing its process ID (PID) to the server's named pipe and creating a unique client-specific FIFO for bidirectional communication. Depending on the specified option (Connect or tryConnect), the client either waits for a spot in the server queue or attempts an immediate connection.**
2. **Sending Requests (send_request_to_server): The client sends various requests to the server via the client-specific FIFO. These requests include file upload, download, server commands, and others. The client also handles the "quit" and "killServer" commands, which allow the client to disconnect from the server or terminate the server, respectively.**

3. **Handling Server Responses (handle_server_response): After sending a request, the client waits for a response from the server, which it reads from its designated FIFO. The response is then displayed to the user.**
4. **Signal Handling (handle_sigint): The client implements a signal handler to intercept Ctrl+C signals. Upon receiving this signal, the client sends a "quit" message to the server, performs cleanup tasks, and exits gracefully.**

Implementation Details

The client program is implemented in C and uses several system calls and libraries to perform its functions:

1. **File Operations: The client supports file upload and download operations with the server. Files are read from or written to the local filesystem and transferred to/from the server using FIFOs.**
2. **Concurrency and Synchronization: The client uses a semaphore to manage concurrency and control access to critical sections of the code, ensuring safe execution of file transfer operations.**
3. **Error Handling: The client checks system calls and file operations for errors and displays or logs appropriate error messages.**
4. **Inter-Process Communication (IPC): IPC between the client and server is facilitated through named pipes (FIFOs), providing a reliable means of communication for command execution and data transfer.**
5. **Command Parsing and Execution: The client parses and validates requests before sending them to the server for execution. Responses from the server are processed and presented to the user in a readable format.**

Testing and Validation

The client program should undergo comprehensive testing to ensure robustness, reliability, and proper functionality under various scenarios. This includes testing different command inputs, verifying file upload/download operations with files of varying sizes and types, and testing signal handling and cleanup procedures upon termination.Testing different command inputs (e.g., valid commands, invalid commands).

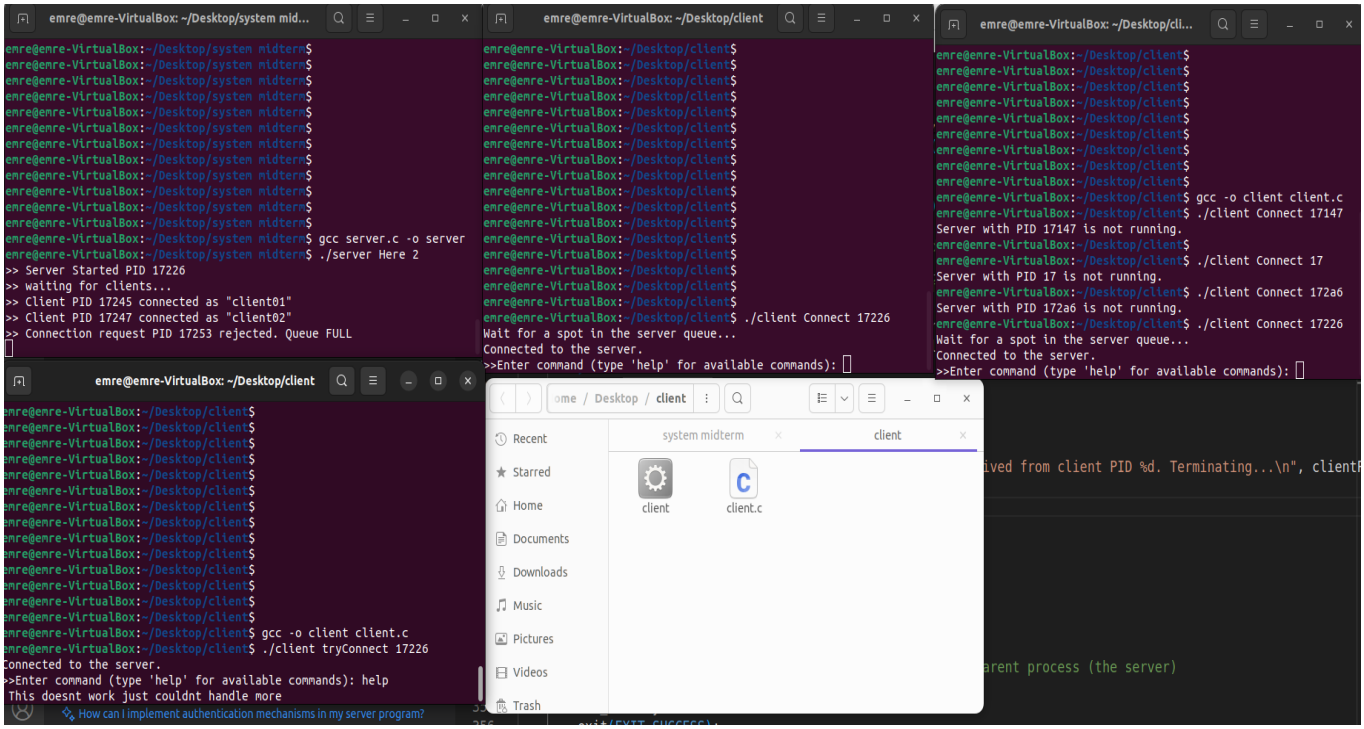
Recommendations and Future Enhancements

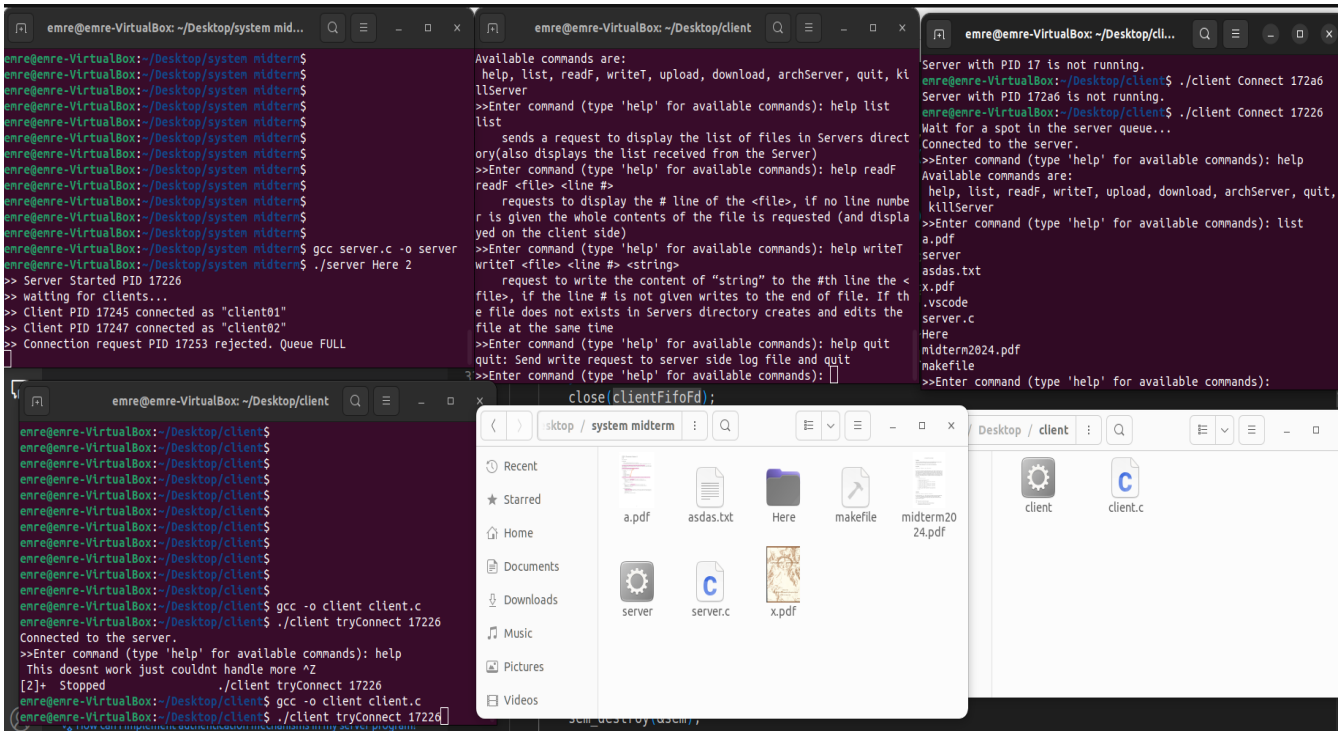
1. **Enhanced Command Set: The range of supported commands could be expanded to include additional functionalities for server interaction, such as user authentication and directory management.**

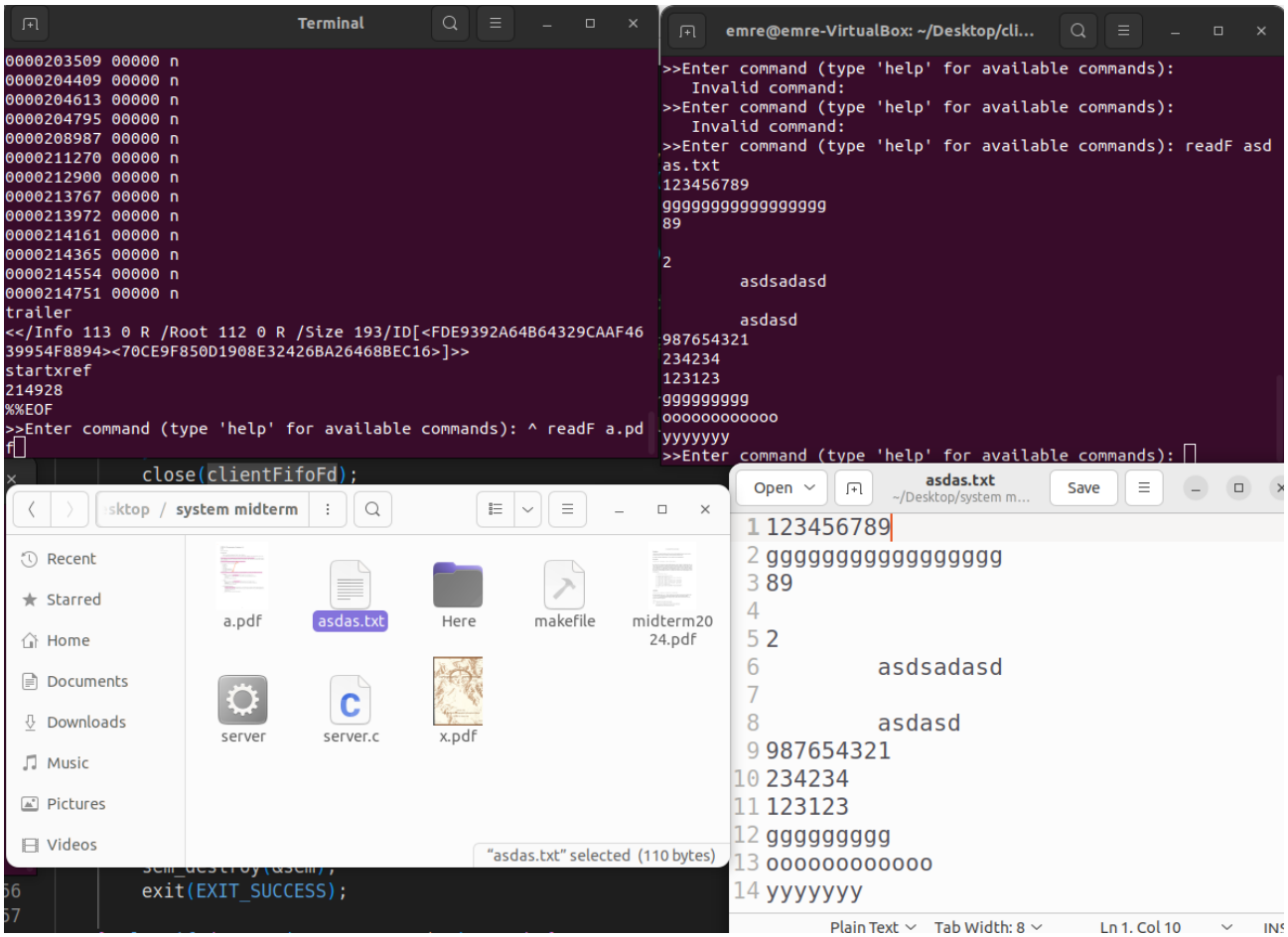
2. **User Interface: A more intuitive and user-friendly interface could be implemented for command input and response presentation, possibly incorporating menu-driven interactions.**
3. **Security Measures: Encryption mechanisms could be integrated for secure data transmission between the client and server. User authentication and access control measures could also be implemented to enhance system security.**
4. **Performance Optimization: File transfer algorithms and FIFO handling techniques could be optimized for improved efficiency and scalability, especially when dealing with large files or concurrent client connections.**

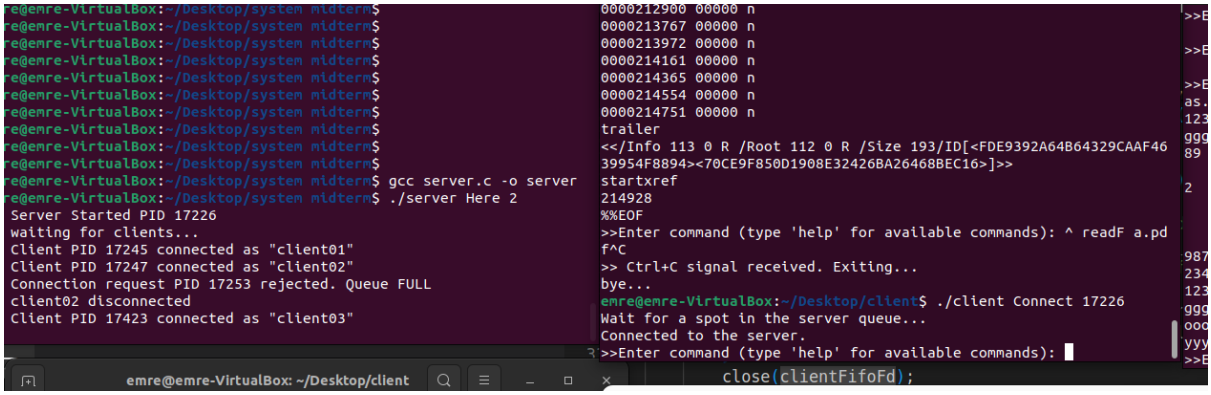
Conclusion

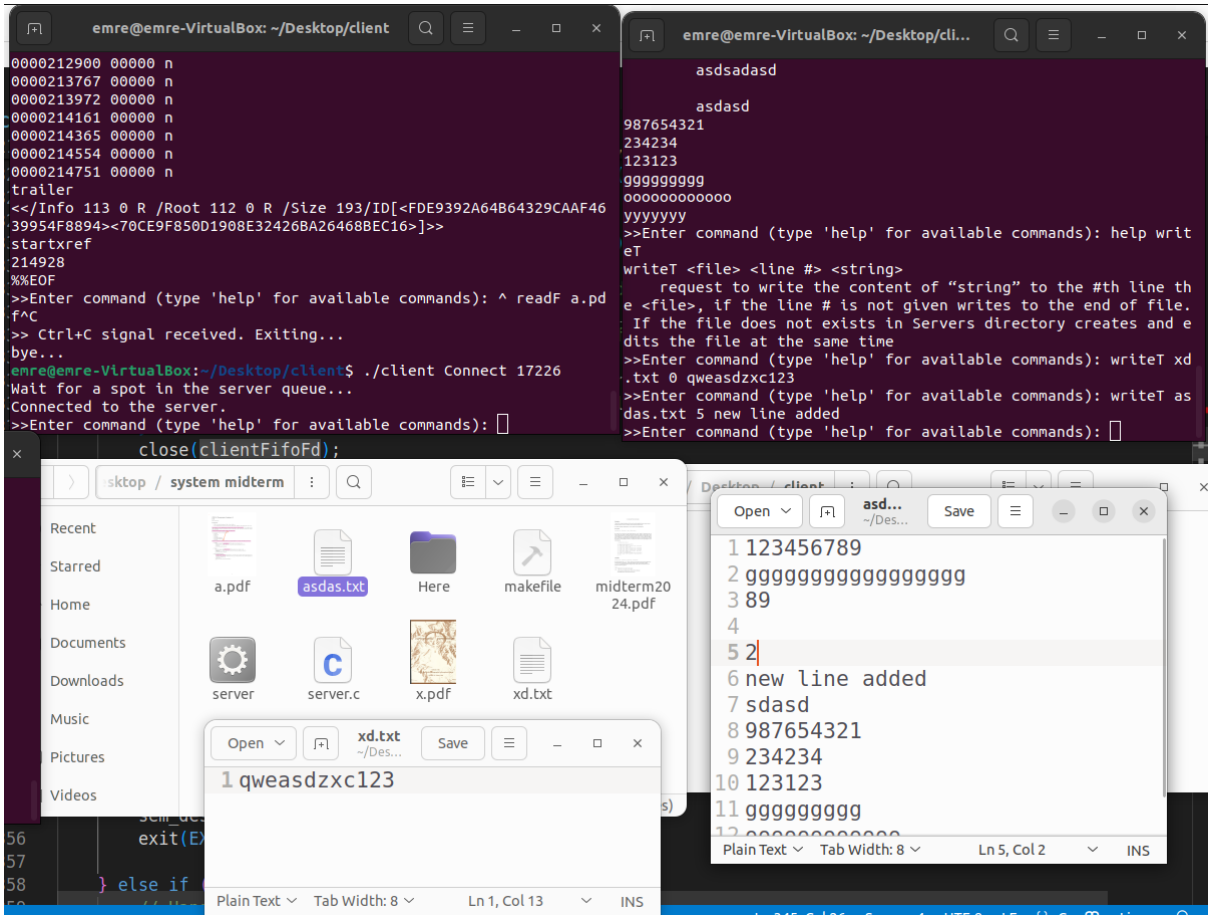
The **client.c** program serves as an essential component of a client-server application, offering a flexible interface for users to interact with server functionalities. By adhering to best practices in IPC, error handling, and command processing, the client program ensures seamless communication and efficient data transfer with the server. Future enhancements can further enrich its capabilities and usability, making it suitable for diverse client-server applications across different domains.

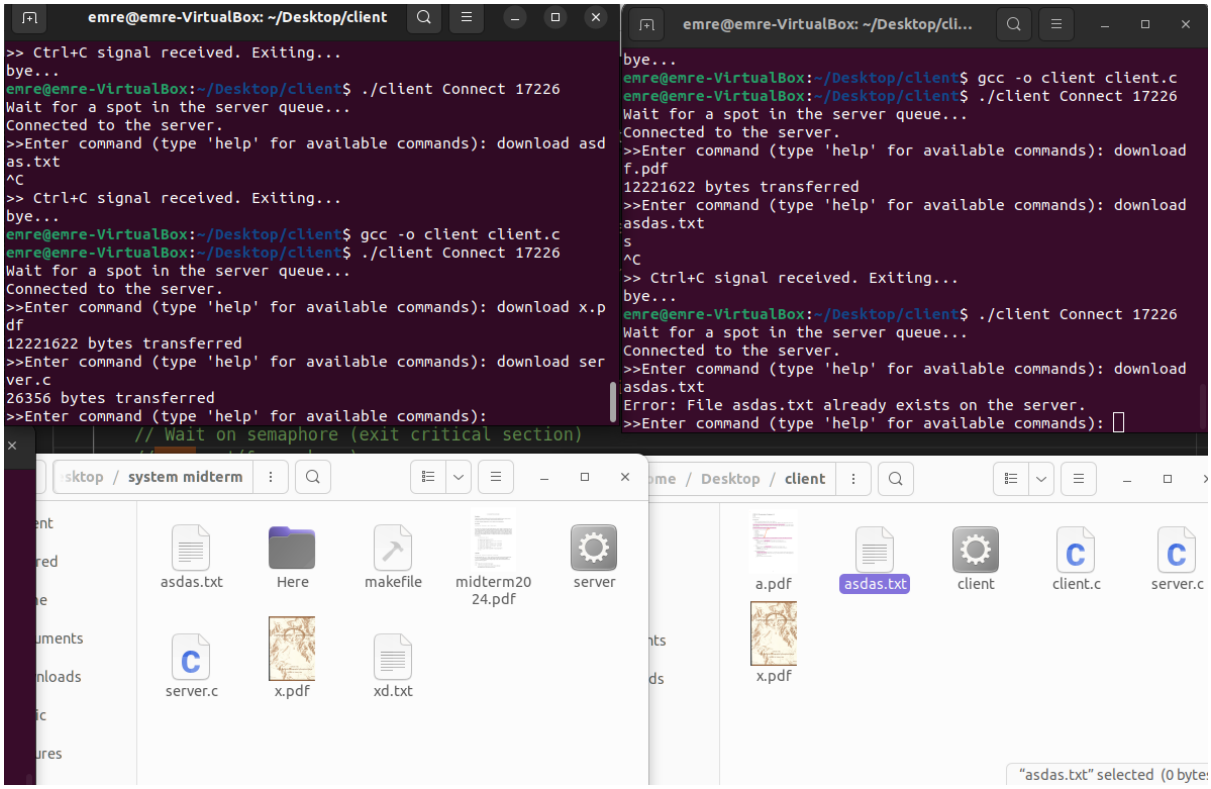


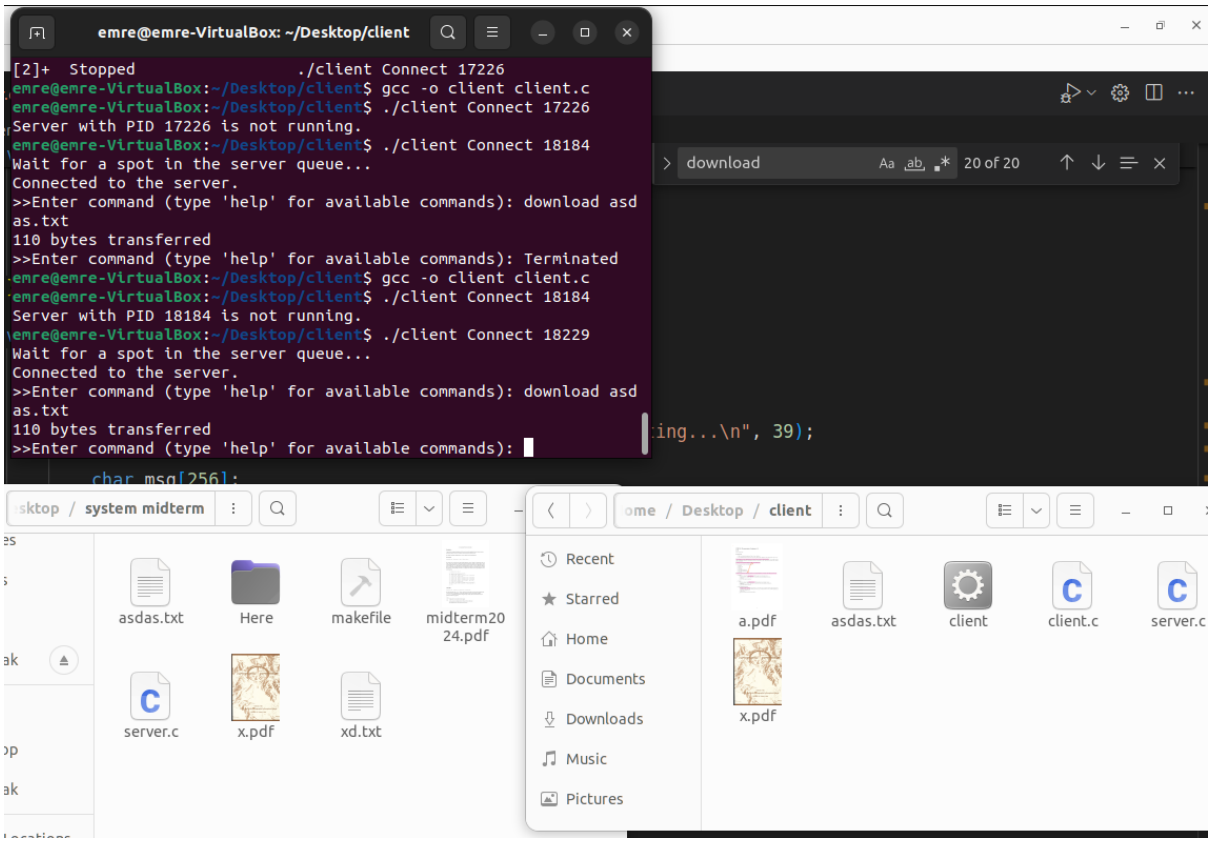

emre@emre-VirtualBox: ~/Desktop/system mid...
emre@emre-VirtualBox:~/Desktop/system midterm$ gcc server.c -o server
emre@emre-VirtualBox:~/Desktop/system midterm$ ./server Here 2
>> Server Started PID 17226
>> waiting for clients...
>> Client PID 17245 connected as "client01"
>> Client PID 17247 connected as "client02"
>> Connection request PID 17253 rejected. Queue FULL
emre@emre-VirtualBox: ~/Desktop/client
Available commands are:
help, list, readF, writeT, upload, download, archServer, quit, killServer
>>Enter command (type 'help' for available commands): help list
list
sends a request to display the list of files in Servers directory(also displays the list received from the Server)
>>Enter command (type 'help' for available commands): help readF
readF <file> <line #>
requests to display the # line of the <file>, if no line number is given the whole contents of the file is requested (and displayed on the client side)
>>Enter command (type 'help' for available commands): help writeT
writeT <file> <line #> <string>
request to write the content of "string" to the #th line the <file>, if the line # is not given writes to the end of file. If the file does not exists in Servers directory creates and edits the file at the same time
>>Enter command (type 'help' for available commands): help quit
quit: Send write request to server side log file and quit
>>Enter command (type 'help' for available commands):
emre@emre-VirtualBox: ~/Desktop/cli...
Server with PID 17 is not running.
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 172a6
Server with PID 172a6 is not running.
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): help
Available commands are:
help, list, readF, writeT, upload, download, archServer, quit, killServer
>>Enter command (type 'help' for available commands): list
a.pdf
server
asdas.txt
x.pdf
.vscode
server.c
Here
midterm2024.pdf
makefile
>>Enter command (type 'help' for available commands):
emre@emre-VirtualBox: ~/Desktop/client
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client tryConnect 17226
Connected to the server.
>>Enter command (type 'help' for available commands): help
This doesnt work just couldnt handle more ^Z
[2]+ Stopped ./client tryConnect 17226
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client tryConnect 17226
close(clientFifoFd);
sktop / system midterm
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
a.pdf
asdas.txt
Here
makefile
midterm2024.pdf
server
server.c
x.pdf
Desktop / client
client
client.c

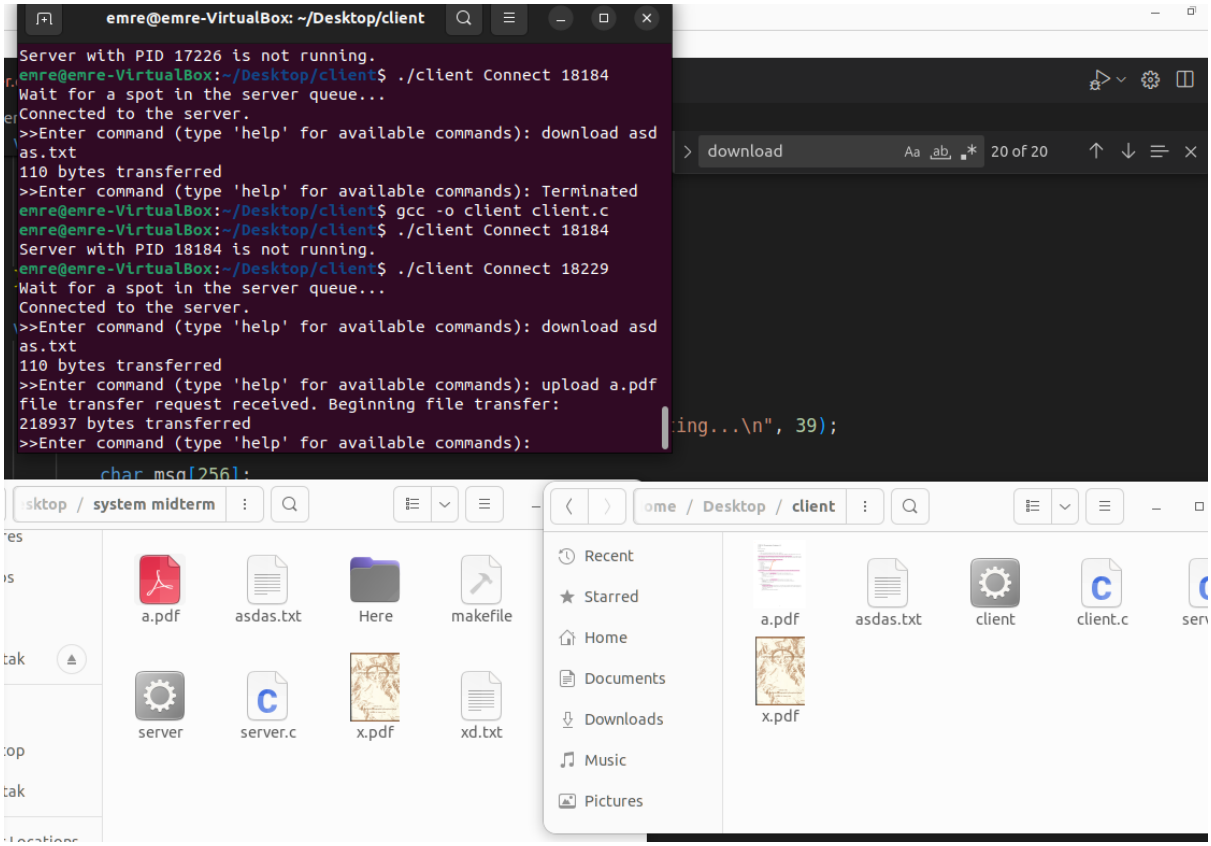

Terminal
0000203509 00000 n
0000204409 00000 n
0000204613 00000 n
0000204795 00000 n
0000208987 00000 n
0000211270 00000 n
0000212900 00000 n
0000213767 00000 n
0000213972 00000 n
0000214161 00000 n
0000214365 00000 n
0000214554 00000 n
0000214751 00000 n
trailer
<</Info 113 0 R /Root 112 0 R /Size 193/ID[<FDE9392A64B64329CAAF4639954F8894><70CE9F850D1908E32426BA26468BEC16>]>>
startxref
214928
%%EOF
>>Enter command (type 'help' for available commands): ^ readF a.pdf
emre@emre-VirtualBox: ~/Desktop/cli...
>>Enter command (type 'help' for available commands):
Invalid command:
>>Enter command (type 'help' for available commands):
Invalid command:
>>Enter command (type 'help' for available commands): readF asdas.txt
123456789
ggggggggggggggggg
89
2
asdsadasd
asdasd
987654321
234234
123123
ggggggggg
ooooooooooo
yyyyyyy
>>Enter command (type 'help' for available commands):
close(clientFifoFd);
sktop / system midterm
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
a.pdf
asdas.txt
Here
makefile
midterm2024.pdf
server
server.c
x.pdf
"asdas.txt" selected (110 bytes)
exit(EXIT_SUCCESS);
Open
asdas.txt
~/Desktop/system m...
Save
123456789
ggggggggggggggggg
89
2
asdsadasd
asdasd
987654321
234234
123123
ggggggggg
ooooooooooo
yyyyyyy
Plain Text
Tab Width: 8
Ln 1, Col 10

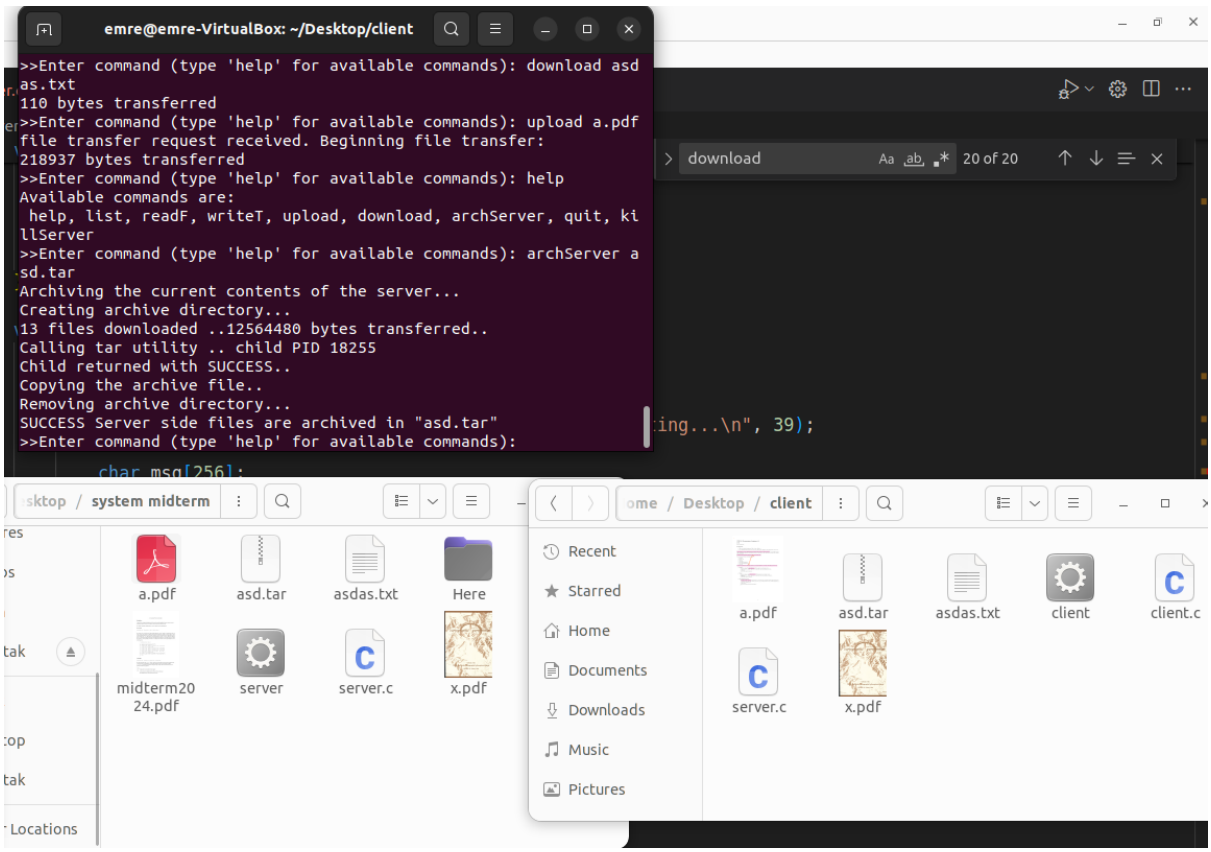

emre@emre-VirtualBox:~/Desktop/system midterm$ gcc server.c -o server
emre@emre-VirtualBox:~/Desktop/system midterm$ ./server Here 2
Server Started PID 17226
waiting for clients...
Client PID 17245 connected as "client01"
Client PID 17247 connected as "client02"
Connection request PID 17253 rejected. Queue FULL
client02 disconnected
Client PID 17423 connected as "client03"
0000212900 00000 n
0000213767 00000 n
0000213972 00000 n
0000214161 00000 n
0000214365 00000 n
0000214554 00000 n
0000214751 00000 n
trailer
<</Info 113 0 R /Root 112 0 R /Size 193/ID[<FDE9392A64B64329CAAF46
39954F8894><70CE9F850D1908E32426BA26468BEC16>]>>
startxref
214928
%%EOF
>>Enter command (type 'help' for available commands): ^ readF a.pd
f^C
>> Ctrl+C signal received. Exiting...
bye...
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands):
emre@emre-VirtualBox: ~/Desktop/client
close(clientFifoFd);

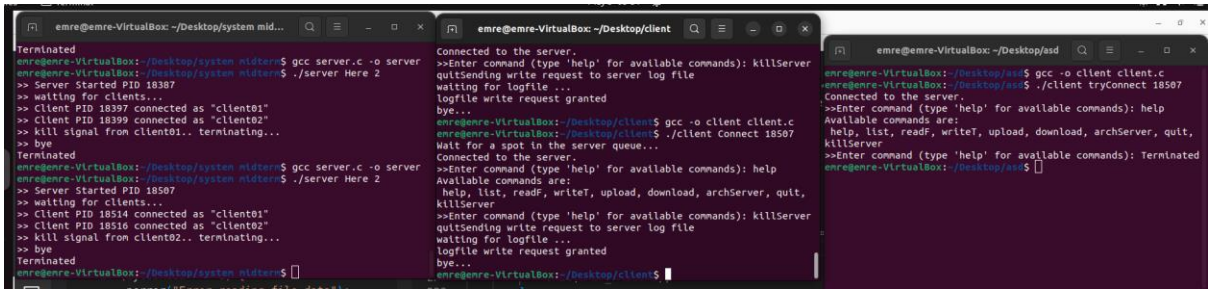

emre@emre-VirtualBox: ~/Desktop/client
0000212900 00000 n
0000213767 00000 n
0000213972 00000 n
0000214161 00000 n
0000214365 00000 n
0000214554 00000 n
0000214751 00000 n
trailer
<</Info 113 0 R /Root 112 0 R /Size 193/ID[<FDE9392A64B64329CAAF46
39954F8894><70CE9F850D1908E32426BA26468BEC16>]>>
startxref
214928
%%EOF
>>Enter command (type 'help' for available commands): ^ readF a.pd
f^C
>> Ctrl+C signal received. Exiting...
bye...
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands):
emre@emre-VirtualBox: ~/Desktop/cli...
asdsadasd
asdasd
987654321
234234
123123
ggggggggg
ooooooooooo
yyyyyyy
>>Enter command (type 'help' for available commands): help writ
eT
writeT <file> <line #> <string>
request to write the content of “string” to the #th line th
e <file>, if the line # is not given writes to the end of file.
If the file does not exists in Servers directory creates and e
dits the file at the same time
>>Enter command (type 'help' for available commands): writeT xd
.txt 0 qweasdzxc123
>>Enter command (type 'help' for available commands): writeT as
das.txt 5 new line added
>>Enter command (type 'help' for available commands):
close(clientFifoFd);
system midterm
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Videos
a.pdf
asdas.txt
Here
makefile
midterm2024.pdf
server
server.c
x.pdf
xd.txt
Open
xd.txt
~/Des...
Save
1 qweasdzxc123
Plain Text
Tab Width: 8
Ln 1, Col 13
INS
Open
asd...
~/Des...
Save
1 123456789
2 ggggggggggggggggg
3 89
4
5 2
6 new line added
7 sdasd
8 987654321
9 234234
10 123123
11 ggggggggg
Plain Text
Tab Width: 8
Ln 5, Col 2
INS
exit(E
} else if

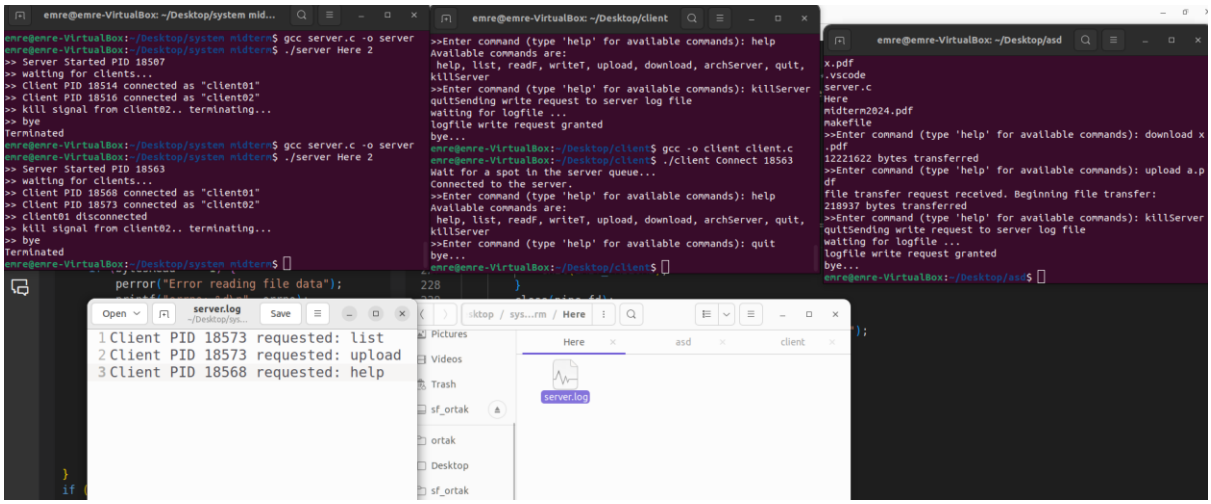

emre@emre-VirtualBox: ~/Desktop/client
>> Ctrl+C signal received. Exiting...
bye...
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download asd
as.txt
^C
>> Ctrl+C signal received. Exiting...
bye...
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download x.p
df
12221622 bytes transferred
>>Enter command (type 'help' for available commands): download ser
ver.c
26356 bytes transferred
>>Enter command (type 'help' for available commands):
emre@emre-VirtualBox: ~/Desktop/cli...
bye...
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download
f.pdf
12221622 bytes transferred
>>Enter command (type 'help' for available commands): download
asdas.txt
s
^C
>> Ctrl+C signal received. Exiting...
bye...
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download
asdas.txt
Error: File asdas.txt already exists on the server.
>>Enter command (type 'help' for available commands):
// Wait on semaphore (exit critical section)
system midterm
asdas.txt
Here
makefile
midterm20
24.pdf
server
server.c
x.pdf
xd.txt
client
a.pdf
asdas.txt
client
client.c
server.c
x.pdf
"asdas.txt" selected (0 byte


emre@emre-VirtualBox: ~/Desktop/client
[2]+ Stopped ./client Connect 17226
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 17226
Server with PID 17226 is not running.
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18184
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download asd
as.txt
110 bytes transferred
>>Enter command (type 'help' for available commands): Terminated
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18184
Server with PID 18184 is not running.
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18229
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download asd
as.txt
110 bytes transferred
>>Enter command (type 'help' for available commands):
download
20 of 20
ing...\n", 39);
char msg[256];
system midterm
asdas.txt
Here
makefile
midterm20
24.pdf
server.c
x.pdf
xd.txt
Recent
Starred
Home
Documents
Downloads
Music
Pictures
a.pdf
asdas.txt
client
client.c
server.c
x.pdf


emre@emre-VirtualBox: ~/Desktop/client
Server with PID 17226 is not running.
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18184
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download asd
as.txt
110 bytes transferred
>>Enter command (type 'help' for available commands): Terminated
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18184
Server with PID 18184 is not running.
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18229
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): download asd
as.txt
110 bytes transferred
>>Enter command (type 'help' for available commands): upload a.pdf
file transfer request received. Beginning file transfer:
218937 bytes transferred
>>Enter command (type 'help' for available commands):
download
20 of 20
system midterm
a.pdf
asdas.txt
Here
makefile
server
server.c
x.pdf
xd.txt
client
client.c
Recent
Starred
Home
Documents
Downloads
Music
Pictures


emre@emre-VirtualBox: ~/Desktop/client
>>Enter command (type 'help' for available commands): download asd
as.txt
110 bytes transferred
>>Enter command (type 'help' for available commands): upload a.pdf
file transfer request received. Beginning file transfer:
218937 bytes transferred
>>Enter command (type 'help' for available commands): help
Available commands are:
help, list, readF, writeT, upload, download, archServer, quit, ki
llServer
>>Enter command (type 'help' for available commands): archServer a
sd.tar
Archiving the current contents of the server...
Creating archive directory...
13 files downloaded ..12564480 bytes transferred..
Calling tar utility .. child PID 18255
Child returned with SUCCESS..
Copying the archive file..
Removing archive directory...
SUCCESS Server side files are archived in "asd.tar"
>>Enter command (type 'help' for available commands):
download
20 of 20
system midterm
a.pdf
asd.tar
asdas.txt
Here
midterm2024.pdf
server
server.c
x.pdf
client
client.c
Recent
Starred
Home
Documents
Downloads
Music
Pictures
Locations


emre@emre-VirtualBox: ~/Desktop/system mid...
Terminated
emre@emre-VirtualBox:~/Desktop/system midterm$ gcc server.c -o server
emre@emre-VirtualBox:~/Desktop/system midterm$ ./server Here 2
>> Server Started PID 18387
>> waiting for clients...
>> Client PID 18397 connected as "client01"
>> Client PID 18399 connected as "client02"
>> kill signal from client01.. terminating...
>> bye
Terminated
emre@emre-VirtualBox:~/Desktop/system midterm$ gcc server.c -o server
emre@emre-VirtualBox:~/Desktop/system midterm$ ./server Here 2
>> Server Started PID 18507
>> waiting for clients...
>> Client PID 18514 connected as "client01"
>> Client PID 18516 connected as "client02"
>> kill signal from client02.. terminating...
>> bye
Terminated
emre@emre-VirtualBox:~/Desktop/system midterm$
emre@emre-VirtualBox: ~/Desktop/client
Connected to the server.
>>Enter command (type 'help' for available commands): killServer
quitSending write request to server log file
waiting for logfile ...
logfile write request granted
bye...
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18507
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): help
Available commands are:
help, list, readF, writeT, upload, download, archServer, quit,
killServer
>>Enter command (type 'help' for available commands): killServer
quitSending write request to server log file
waiting for logfile ...
logfile write request granted
bye...
emre@emre-VirtualBox:~/Desktop/client$
emre@emre-VirtualBox: ~/Desktop/asd
emre@emre-VirtualBox:~/Desktop/asd$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/asd$ ./client tryConnect 18507
Connected to the server.
>>Enter command (type 'help' for available commands): help
Available commands are:
help, list, readF, writeT, upload, download, archServer, quit,
killServer
>>Enter command (type 'help' for available commands): Terminated
emre@emre-VirtualBox:~/Desktop/asd$


emre@emre-VirtualBox: ~/Desktop/system mid...
emre@emre-VirtualBox:~/Desktop/system midterm$ gcc server.c -o server
emre@emre-VirtualBox:~/Desktop/system midterm$ ./server Here 2
>> Server Started PID 18507
>> waiting for clients...
>> Client PID 18514 connected as "client01"
>> Client PID 18516 connected as "client02"
>> kill signal from client02.. terminating...
>> bye
Terminated
emre@emre-VirtualBox:~/Desktop/system midterm$ gcc server.c -o server
emre@emre-VirtualBox:~/Desktop/system midterm$ ./server Here 2
>> Server Started PID 18563
>> waiting for clients...
>> Client PID 18568 connected as "client01"
>> Client PID 18573 connected as "client02"
>> client01 disconnected
>> kill signal from client02.. terminating...
>> bye
Terminated
emre@emre-VirtualBox:~/Desktop/system midterm$
emre@emre-VirtualBox: ~/Desktop/client
>>Enter command (type 'help' for available commands): help
Available commands are:
help, list, readF, writeT, upload, download, archServer, quit,
killServer
>>Enter command (type 'help' for available commands): killServer
quitSending write request to server log file
waiting for logfile ...
logfile write request granted
bye...
emre@emre-VirtualBox:~/Desktop/client$ gcc -o client client.c
emre@emre-VirtualBox:~/Desktop/client$ ./client Connect 18563
Wait for a spot in the server queue...
Connected to the server.
>>Enter command (type 'help' for available commands): help
Available commands are:
help, list, readF, writeT, upload, download, archServer, quit,
killServer
>>Enter command (type 'help' for available commands): quit
bye...
emre@emre-VirtualBox:~/Desktop/client$
emre@emre-VirtualBox: ~/Desktop/asd
x.pdf
.vscode
server.c
Here
midterm2024.pdf
makefile
>>Enter command (type 'help' for available commands): download x
.pdf
12221622 bytes transferred
>>Enter command (type 'help' for available commands): upload a.p
df
file transfer request received. Beginning file transfer:
218937 bytes transferred
>>Enter command (type 'help' for available commands): killServer
quitSending write request to server log file
waiting for logfile ...
logfile write request granted
bye...
emre@emre-VirtualBox:~/Desktop/asd$
perror("Error reading file data");
Open
server.log
Save
1 Client PID 18573 requested: list
2 Client PID 18573 requested: upload
3 Client PID 18568 requested: help
Pictures
Videos
Trash
sf_ortak
ortak
Desktop
sf_ortak
Here
asd
client
server.log