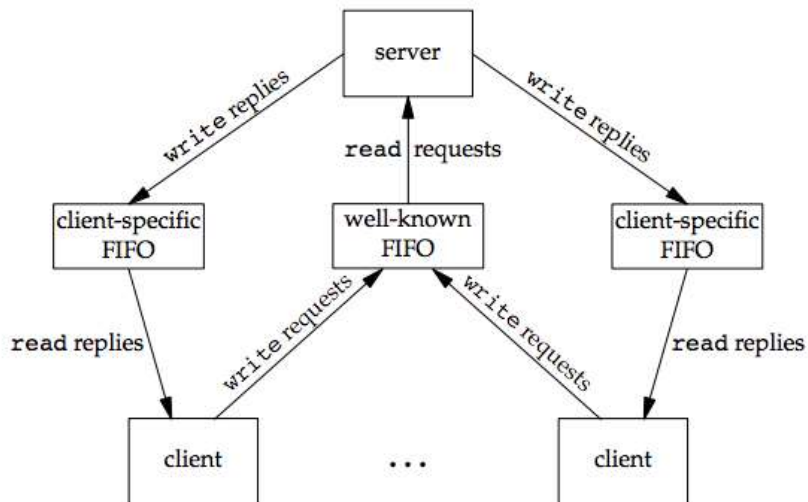


System Programming hw3 Project



Threads are used instead of `fork()` used in the first assignment. When the program runs, a thread is created until the `poolSize` entered and the thread that is empty from the pool processes the request for each incoming request. The missing part of the program is to run the client as much as update, download and `maxclient`. The program runs every incoming client. Mutex is used.

Server:

The provided code is a server program written in C. It creates a server that handles requests from multiple clients. Here's a brief overview of what the code does:

The code includes various header files and defines some constants. It declares global variables, including file descriptors, counters, and arrays. It defines two structures: `Client` and `Task`, which are used for storing client information and task details, respectively. The code includes function prototypes for various utility functions.

The main function reads the command-line arguments, creates a directory if it doesn't exist, and sets up the main FIFO for communication with clients. Inside the main function, a thread pool is initialized, and the server starts waiting for client requests. When a client request is received, a thread from the pool is assigned to handle the request. The `handle_request` function is the entry point for each thread and is responsible for processing the client's request. The `do_request` function handles the specific request made by the client, such as reading a file, writing to a file, or performing other operations. The code also includes several helper functions for file operations and logging.

writeStringToEnd: This function appends the given content (string) to the end of the file specified by the file path.

do_request: This function processes the client's request. It performs different actions based on the prompt received from the client. The actions include displaying help messages, listing files in a directory, reading file contents, writing to a file, and handling other specific requests.

writeStringToLine: This function writes the given string to a specific line in the file specified by the file name and line number. If the line number is -1, it writes the string to the end of the file.

readFile_full: This function reads the entire content of a file and returns it as a string. It takes the file name as input and uses file I/O operations to read the file's contents.

read_file: This function reads a specific line from a file. It takes the file path and the line number as inputs. It opens the file, reads the desired line, and returns it as a string.

handle_request: This function is executed by each thread in the thread pool. It receives a task (client request) and passes it to the do_request function for processing.

err_sys: This function is used for error handling. It prints an error message and exits the program.

append_to_log: This function appends client request information (client PID and request) to a log file.

quit_killserver: This function is called when the server needs to be terminated. It closes file descriptors, unlinks FIFOs, cancels active threads, and exits the program.

ignore_sigint: This function is a signal handler for SIGINT (interrupt signal, usually triggered by pressing Ctrl+C). It calls the quit_killserver function to gracefully terminate the server.

Client:

INPUT_MAX: Specifies the maximum input size.

PID_MAX: Defines the maximum size of a process ID.

PROMPT_MAX: Defines the maximum size of a prompt.

Structures:

Client: Represents client information, including process ID, prompt, line number, filename, and write string.

Function Definitions:

err_sys: Prints error information to the screen and exits the program.

usage: Prints the program usage information.

prompt: Displays a prompt message on the screen.

printClient: Prints the details of a Client structure.

append_to_log: Appends a request to the server log file.

parse_command: Parses a command to extract the file name and line number.

split_string_by_spaces: Splits a string into multiple substrings using spaces and populates a Client structure.

handler_signal: Handles the signal for the program.

Program Flow:

The program establishes a connection with the server by opening a main FIFO (named pipe).

It waits for user commands and parses them to extract the required information.

The parsed information is sent to the server through the main FIFO.

The program creates a result FIFO (named pipe) with a unique name based on the process ID.

It opens the result FIFO and waits to receive the server's response.

The response is displayed on the screen, appended to the log file, and the result FIFO is closed.

The process continues to wait for new user commands until the user requests to quit.

Upon quitting, the program sends a quit command to the server, closes the main and result FIFOs, and exits.

```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./server dirnames 3
Usage: ./server <dirname> <max.#ofClients> <poolSize>
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./server dirnames 3 5
>>Server Started Pid: 21469
>>Waiting for Clients..
>>Client PID:21472 connected as client1
```

```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./client
>biboClient Connect Server..
>>Waiting for Que..Connection established:

>>Enter command: help
    Available comments are:
    help, list,readF,writer, upload, download, quit, killServer

>>Enter command: help list
    help list
    Sends a request to display the list of files in Servers directory

>>Enter command: help readf
    readf <file> <line #>
    display the #th line of the <file>, returns with an error if <file> does not exists

>>Enter command: help writer
    writer <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: help upload
    upload <file>
    uploads the file from the current working directory of client to the Servers director. (beware of the cases no file
in clients current working directory and name on Servers side)

>>Enter command: help download
```

```
>>Enter command: help download
    download <file>
    request to receive <file> from Servers directory to client side

>>Enter command: help quit
    quit
    Send write request to server side log file and quits

>>Enter command: help killServer
    killServer <file>
    Sends a kill request to the Server

>>Enter command: help list
    help list
    Sends a request to display the list of files in Servers directory

>>Enter command: list
    log.txt
    b.txt
    Report.pdf
    161044116_hw2.c

>>Enter command:
```

Start Again :

```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./server dirnames 3 5
>>Server Started Pid: 22081
>>Waiting for Clients..
>>Cliend PID:22084 connected as client1
```

```
>>Enter command: readf flag.txt 11
        medeniyeti gibi çeşitli eski Anadolu medeniyetlerinin ardından, Make
>>Enter command: █
```

New Client:

```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./server dirnames 3 5
>>Server Started Pid: 22081
>>Waiting for Clients..
>>Cliend PID:22084 connected as client1
>>Cliend PID:22157 connected as client2
```

```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./client
>biboClient Connect Server..
>>Waiting for Que..Connection established:

>>Enter command: 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: writeT flag.txt 6 burdayım
        Write request has been completed.

>>Enter command: █
```

before request flag.txt file:

```
4 Irak ve Suriye ile komşudur. Güneyini Kıbrıs Adası ve Akdeniz, batısını Ege Denizi
5 çevreler. Marmara Denizi ise İstanbul Boğazı ve Çanakkale Boğazı ile birlikte Anac
6
7 rupa'dan ayırır.
8 Türkiye toprakları üzerindeki ilk yerleşmeler Yontma Taş Devri'nde başlar. Doğu Tr
9 Hititler Erişler Lidyalılar ve Dor istilası sonucu Yunanistan'dan kaçan Akalar
```

After request flag.txt file:

```
4 Irak ve Suriye ile komşudur. Güneyini Kıbrıs Adası ve Akdeniz
5 çevreler. Marmara Denizi ise İstanbul Boğazı ve Çanakkale Boğ
6 burdayım
7
8 rupa'dan ayırır.
9 Türkiye toprakları üzerindeki ilk yerleşmeler Yontma Taş Devri
```

```
>>Enter command: writeT flag.txt 6 burdayım
        Write request has been completed.

>>Enter command: writeT flag.txt sondayım
        Write request has been completed.

>>Enter command: █
```


After request flag.txt file:

```
carından Türkiye Büyük Millet Meclisi tarafından seçilen Kâzım Karabekir  
İmparatorluğu tarihe karıştı. 29 Ekim 1923'te cumhuriyetin ilan edilme  
kaldırılıp Osmanlı Hanedanı'nın yurt dışına sürgün edilmesinden sonra,  
yer tutacak olan bir dizi devrim gerçekleştirildi.  
31 sondayım
```

Plain Text

Start again:

```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./server dirnames 2 10  
>>Server Started Pid: 23125  
>>Waiting for Clients..  
>>Cliend PID:23136 connected as client1  
>>kill signal from 23136 terminating..  
>>bye  
toprak@toprak-virtual-machine:~/Desktop/midterm$
```

```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./client  
>biboClient Connect Server..  
>>Waiting for Que..Connection established:  
  
>>Enter command: help  
Available comments are:  
help, list,readF,writeT, upload, download, quit, killServer  
  
>>Enter command: killserver
```

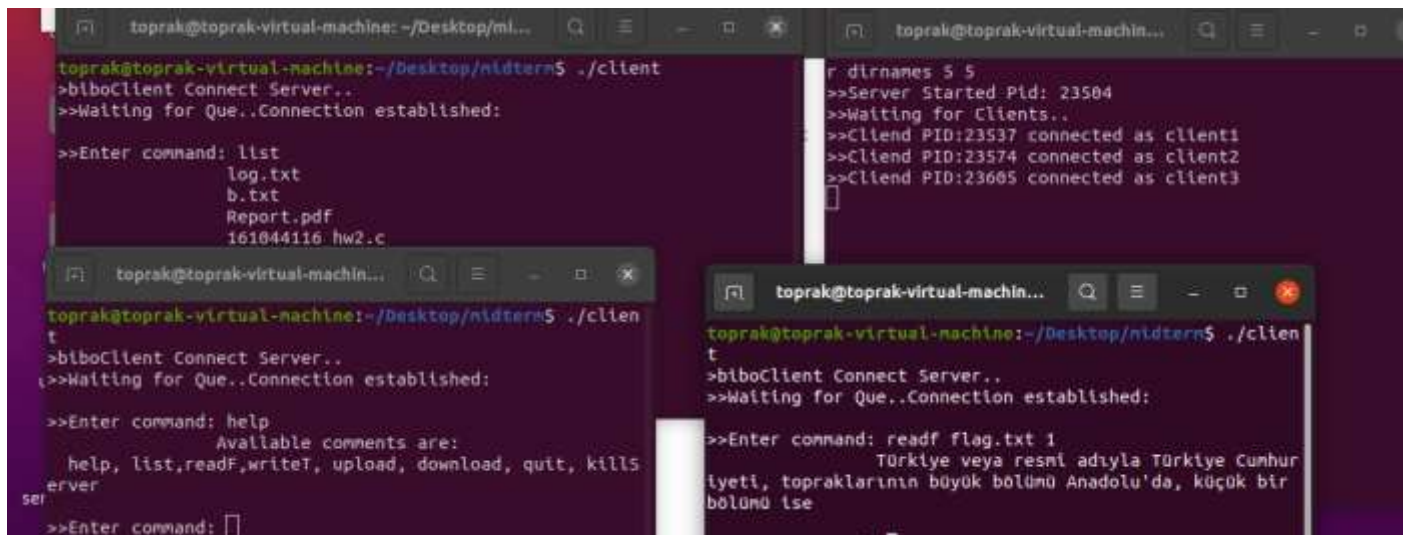
```
toprak@toprak-virtual-machine:~/Desktop/midterm$ ./client  
>biboClient Connect Server..  
>>Waiting for Que..Connection established:  
  
>>Enter command: help  
Available comments are:  
help, list,readF,writeT, upload, download, quit, killServer  
  
>>Enter command: help list  
help list  
Sends a request to display the list of files in Servers directory  
  
>>Enter command: quit  
Sending write request to server log file waiting forlogfile.  
logfile write request granted.  
>>byetoprak@toprak-virtual-machine:~/Desktop/midterm$
```

Log file <clientpid>.txt

```
1  
2 -----  
3 Submitted Request: help  
4  
5 -----  
6 Submitted Request: help list  
7  
8 -----  
9 Submitted Request: quit
```

in log.txt for server

```
330 -----
331 Client PID: 22157
332 Request: help writeT
333
334 -----
335 Client PID: 22157
336 Request: writeT
337
338 -----
339 Client PID: 22157
340 Request: writeT
341
342 -----
343 Client PID: 22157
344 Request: readF
345
346 -----
347 Client PID: 22157
348 Request: readF
```



The screenshot displays four terminal windows from a virtual machine named 'toprak'.

- Top Left Window:** Shows the execution of `./client`. It connects to the server and lists files: `log.txt`, `b.txt`, `Report.pdf`, and `161044116_hw2.c`.
- Top Right Window:** Shows the execution of the server program. It starts with PID 23504 and lists connected clients: `Client PID:23537 connected as client1`, `Client PID:23574 connected as client2`, and `Client PID:23605 connected as client3`.
- Bottom Left Window:** Shows the execution of `./client` again. It connects to the server and displays the help menu: `Available comments are: help, list, readF, writeT, upload, download, quit, kills server`.
- Bottom Right Window:** Shows the execution of `./client` with the command `readf flag.txt 1`. It displays the content of the file: `Türkiye veya resmi adıyla Türkiye Cumhuriyeti, topraklarının büyük bölümü Anadolu'da, küçük bir bölümü ise`.