

Benjamin Alcorn (628002952)

Vinay Balamurali (936002032)

ECEN 602 Machine Problem 1

Description

The architecture has been designed to handle multiple client connections to a single socket. We are achieving this using the 'fork()' system call. The block diagram explains the basic data flow.

1. The parent process is responsible for handling new incoming connections.
2. The child is responsible for communicating with a client.

The parent forks the current process when a new connection is accepted and passes over control to the child. This is the key idea to handle multiple client requests.

Contributions

Benjamin Alcorn:

- Development of the client module
- Development of the 'readline' method to read characters from a socket descriptor
- Documentation
- Testing

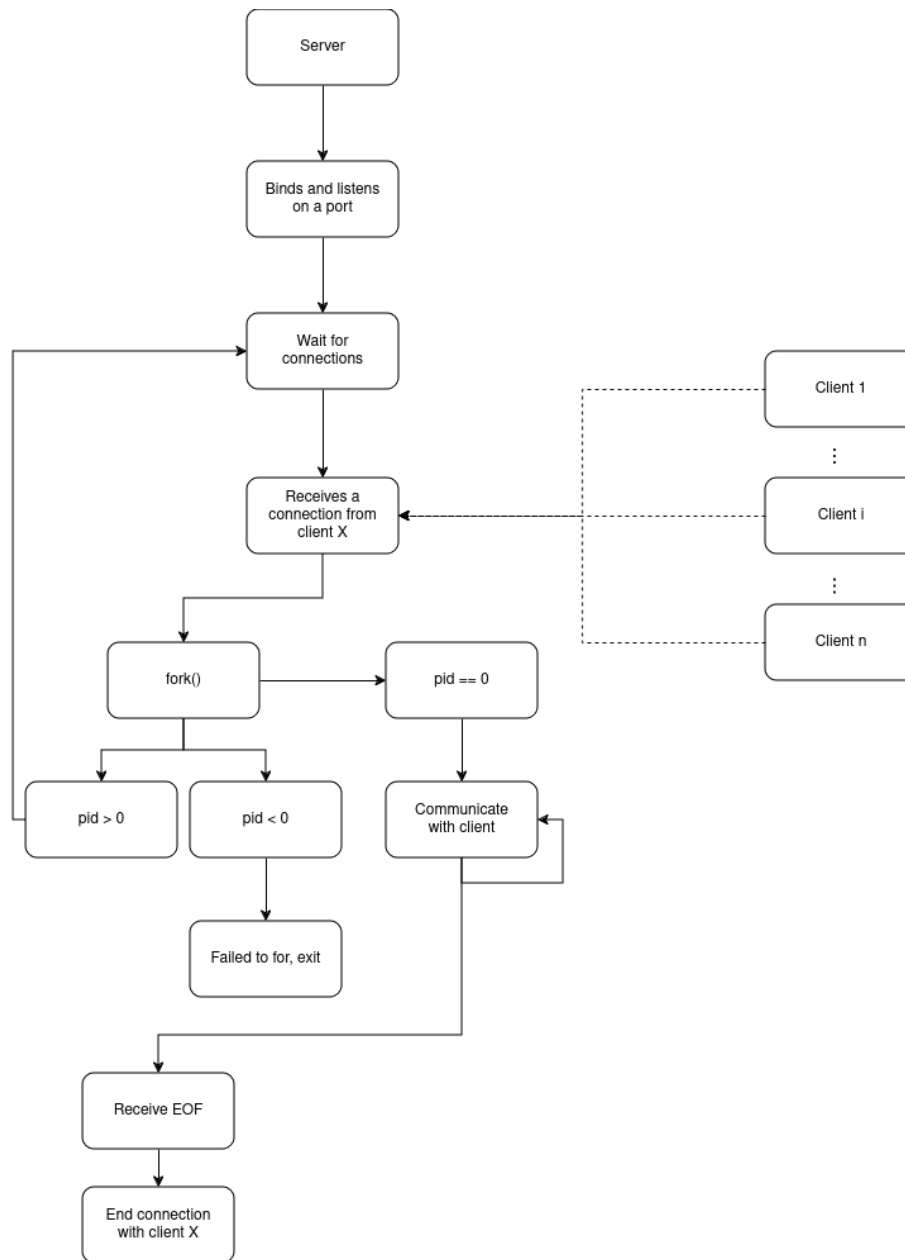
Vinay Balamurali:

- Development of the server module
- Development of the 'writen' method to write characters to a socket descriptor
- Development of Makefile
- Documentation

Benjamin Alcorn (628002952)

Vinay Balamurali (936002032)

Workflow



Benjamin Alcorn (628002952)

Vinay Balamurali (936002032)

Test Cases

1. Line of text terminated by a newline

Client:

```
Starting client.  
Socket created successfully  
Successfully connected to server.  
Enter text to send to server: (Press Control-D to stop)  
test case  
Message from server: test case
```

Server:

```
Socket created successfully.  
Socket binding successfull.  
Listening for incoming connections...  
Successfully established a connection with the client!  
Waiting for data...  
Message from client: test case
```

2. Line of text the maximum line length without a newline

Benjamin Alcorn (628002952)
Vinay Balamurali (936002032)

Client (buffer size of 20 bytes):

```
Enter text to send to server: (Press Control-D to stop)
aaaaaaaaaaaaaaaaaaaaa
Message from server: aaaaaaaaaaaaaaaaaaa
Enter text to send to server: (Press Control-D to stop)
Message from server: a

Enter text to send to server: (Press Control-D to stop)
aaaaaaaaaaaaaaaaaaaaa
Message from server: aaaaaaaaaaaaaaaaaaa
Enter text to send to server: (Press Control-D to stop)
█
```

Server:

```
Message from client: aaaaaaaaaaaaaaaaaaa
Message from client: a

Message from client: aaaaaaaaaaaaaaaaaaa
█
```

3. Line with no characters and EOF

Client:

```
Starting client.
Socket created successfully
Successfully connected to server.
Enter text to send to server: (Press Control-D to stop)

Disconnecting from server
```

Server:

```
Successfully established a connection with the client!
Waiting for data...
Received EOF from client. Terminating connection!
█
```

4. Client terminated after entering text

Benjamin Alcorn (628002952)
Vinay Balamurali (936002032)

Client:

```
Starting client.  
Socket created successfully  
Successfully connected to server.  
Enter text to send to server: (Press Control-D to stop)  
test  
Disconnecting from server
```

Server:

```
Successfully established a connection with the client!  
Waiting for data...  
Received EOF from client. Terminating connection!
```

5. Three clients connected to the server

Benjamin Alcorn (628002952)
Vinay Balamurali (936002032)

Client 1:

```
Starting client.  
Socket created successfully  
Successfully connected to server.  
Enter text to send to server: (Press Control-D to stop)  
hello from client 1  
Message from server: hello from client 1  
Enter text to send to server: (Press Control-D to stop)
```

Client 2:

```
Starting client.  
Socket created successfully  
Successfully connected to server.  
Enter text to send to server: (Press Control-D to stop)  
hello from client 2  
Message from server: hello from client 2  
Enter text to send to server: (Press Control-D to stop)
```

Client 3:

```
Starting client.  
Socket created successfully  
Successfully connected to server.  
Enter text to send to server: (Press Control-D to stop)  
hello from client 3  
Message from server: hello from client 3  
Enter text to send to server: (Press Control-D to stop)
```

Server:

```
Successfully established a connection with the client!  
Waiting for data...  
Successfully established a connection with the client!  
Waiting for data...  
Successfully established a connection with the client!  
Waiting for data...  
Message from client: hello from client 1  
Message from client: hello from client 2  
Message from client: hello from client 3
```