

6. Assignment

Network IPC: Sockets

Issue: June 2—Due: June 8

Exercise 8: Message of the day server

10 Points

Implement a simple server listening on port 2224 accepting multiple connections. If a client connects it gets an individual port and the server is listening for further incoming connections on port 2224, which will be queued and processed after the active client is served.¹

After the client/server connection is established, the server sends the message of the day (motd) stored in `/etc/motd` as an ascii-stream. After sending the message the server closes the connection to the client.

A typical session:

<pre>\$./motds [motds] server is up and listening on port 2224. [motds] Client has connected. [motds] Sending 'message of the day'. [motds] Closing connection. Bye. [Awaits new incoming connections]</pre>	<pre>\$ telnet 127.0.0.1 2224 Trying 127.0.0.1... Connected to 127.0.0.1. Escape character is '^]'. Have a lot of fun... <--- motd Connection closed by foreign host.</pre>
---	--

Some related manual pages:

- `socket(2)` - create an endpoint for communication
- `byteorder(3)` - convert values between host and network byte order
- `inet(3)` - internet address manipulation routines
- `bind(2)` - bind a name to a socket
- `listen(2)` - listen for connection on a socket
- `accept(2)` - accept a connection on a socket

Some related literature:

- Glatz, Chapter 5.6 - Rechnerübergreifende Interprozesskommunikation
- Helmut Herold, Chapter 19.7 - Netzwerkprogrammierung mit TCP/IP
- Stevens, Rago, Chapter 16 - Network IPC

¹You may, however, implement concurrent processing using, for instance, `fork(2)` at the right places.