

Client-server architectures

- Have *server* process, which waits for client requests and processes them once they arrive
- Multiple clients may connect and request service
- Standard paradigm for services on the internet
- Have in addition to IP-addresses *port numbers* assigned to particular services
- Examples:
- Port 80 for http
- Port 25 for sending mail (smtp)
- Port 143 for reading mail via imap
- Port assignments listed in `/etc/services`
- Client-Server architecture implemented via *sockets*

Sockets

How to setup a client/server connection via sockets:

Initialisation phase for the server:

- Server creates endpoint via `socket-system` call
- Server specifies port number and protocol in structure `sockaddr_in`
- Server assigns information in `sockaddr_in`-structure to socket via `bind-system` call

Now server waits for incoming connection via `accept-system` call

When connection received, server reads data via `read-system` call and writes data back via `write-system` call

When server is finished with current connection, server closes connection via `close-system` call