Socket API
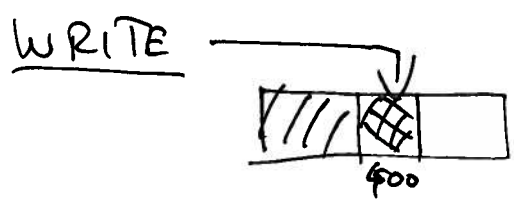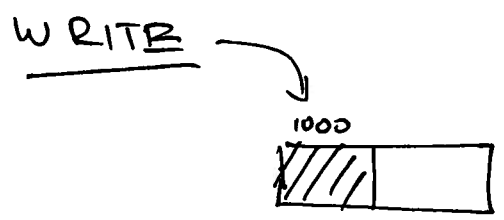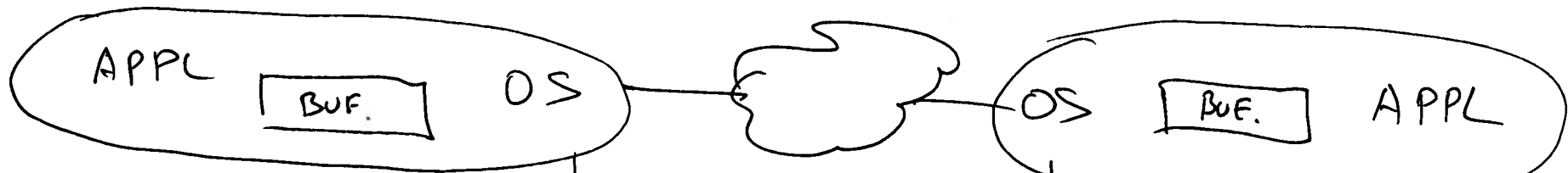
▸ Berkeley socket API (4.2 BSD Unix, 1983)

▸ POSIX socket API (reentrant)

▸ Designed to support any protocol - not just TCP/UDP/IP

▸ Defined in C, but adopted by essentially all programming languages

Main operations

- ▸ Address resolution

- ▸ Binding to a port number

- ▸ Opening connection to a server

- ▸ Accepting connections from clients

- ▸ Sending and receiving data

- ▸ Getting and setting connection parameters

- ▸ Closing connection
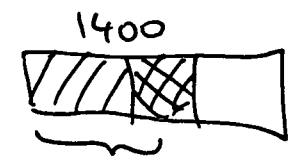
- ▸ Handling simultaneous communications

Perspectives

▸ Reliable, stream-oriented service (TCP)

  – Connection-oriented client-side

  – Connection-oriented server-side

▸ Unreliable, datagram service (UDP)

APPL BUF. OS

OS BUF. APPL

WRITE

1000

WRITE

400

READ

1200

APPL.
BLOCKED

1400

1400

SOCKETS:

DATA

FLOW

WAKE.YP

DATA
1200

200

STANDARD SERVER ARCHITECTURE

```
while (1)
    accept ( ... )
    if fork ( )           // child
        handle conn. {
        close conn.
        exit()
    else                  // par.

        close conn
```

SERVER PROCESS

SERVICE PROCESS
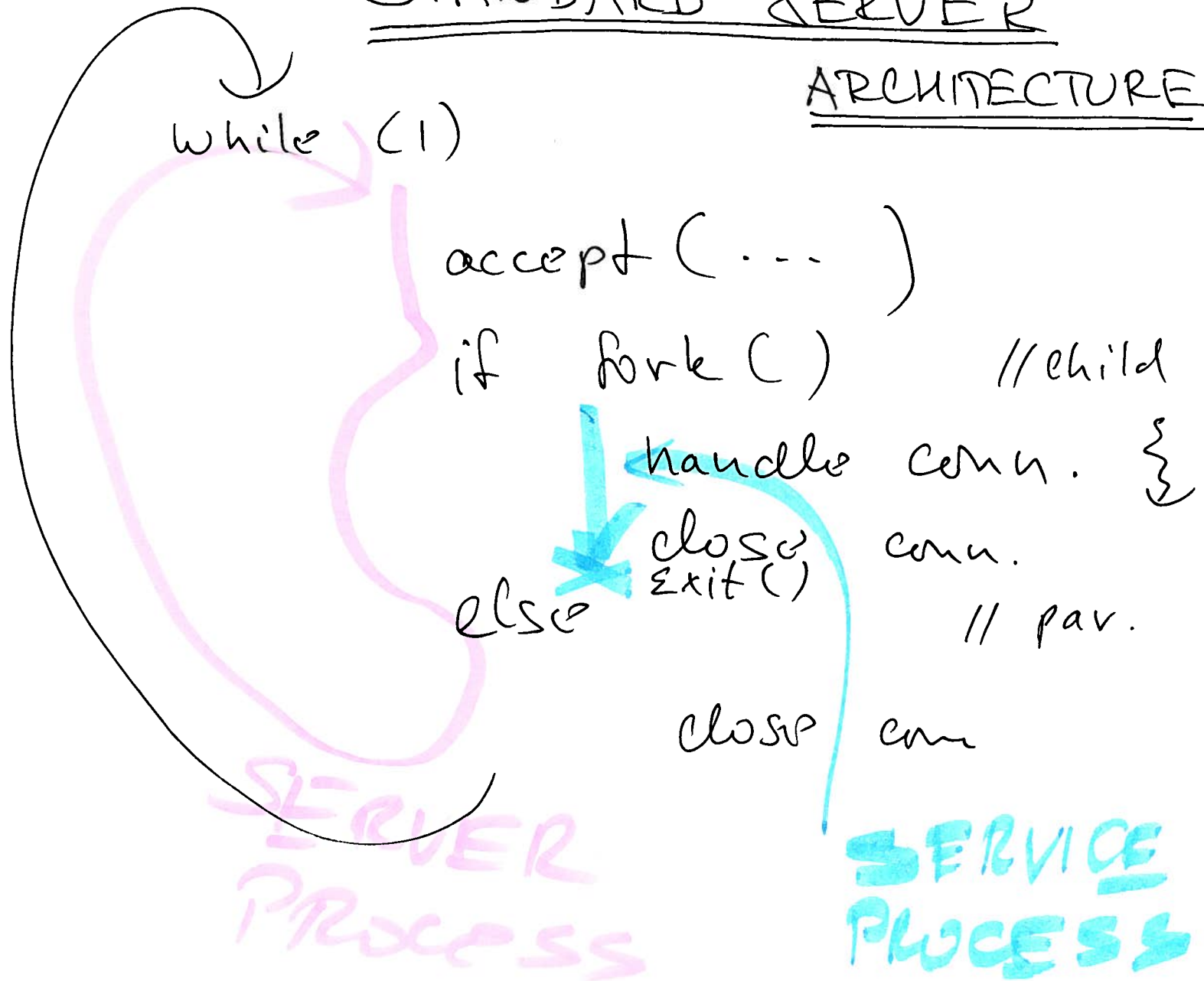
Event-driven approach

Node.js

```javascript
var http = require('http');

http.createServer(
  function (request, response) {
    response.writeHead(200, {'Content-Type': 'text/plain'});
    response.end('Hello World\n');
  }
).listen(8000);

console.log('Server running at http://localhost:8000/');
```

Source: Wikipedia article on Node.js