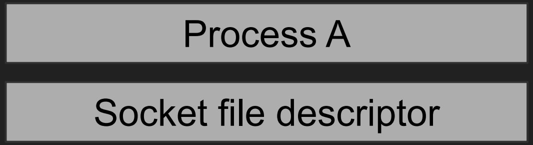
husseinnasser

Sockets, Connections and Queues

Kernel networking structures

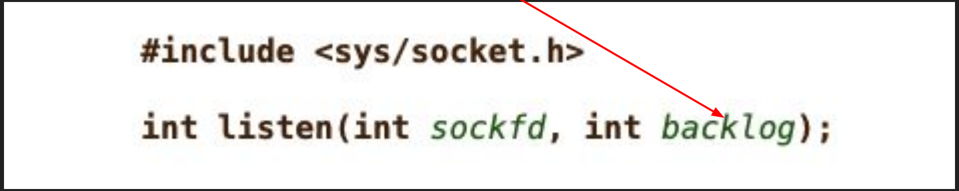
Socket

- When a process listens on an IP/Port it produces a socket
- Socket is a file (at least in linux)
- The process owns the socket
- Can be shared during fork



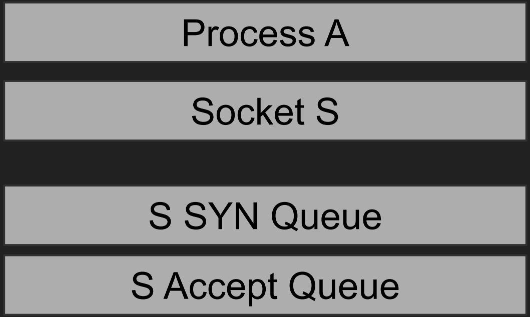
SYN Queue, Accept Queues

- When a socket is created we get two queues with it
- SYN Queue, stores incoming SYNs
- Accept Queue, stores completed connections
- The size of the queues is determined by the backlog
- Not really queues but hash tables



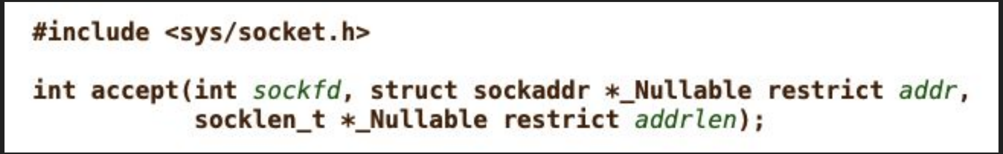
```
#include <sys/socket.h>

int listen(int sockfd, int backlog);
```



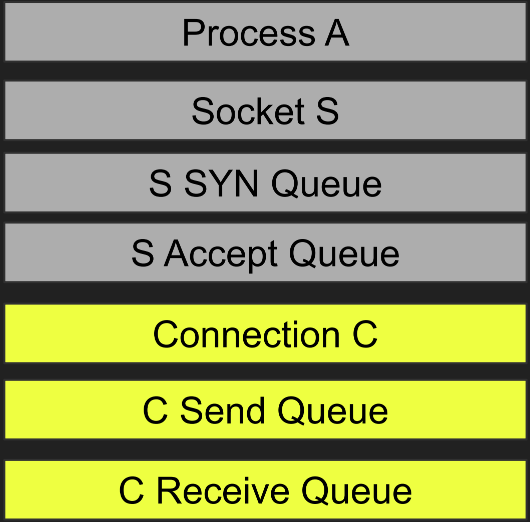
Connection, Receive and Send queue

- Completed connections are placed in the accept queue
- When a process “accepts” a connection is created
- Accept returns a file desc for the connection
- Two new queues created with the connection
- Send queue stores connection outgoing data
- Receive queue stores incoming connection data



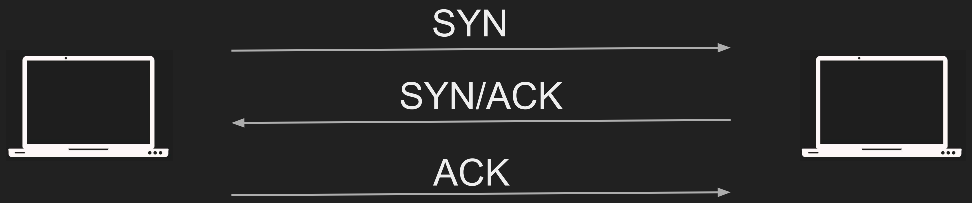
```
#include <sys/socket.h>

int accept(int sockfd, struct sockaddr *_Nullable restrict addr,
           socklen_t *_Nullable restrict addrlen);
```



Connection Establishment

- TCP Three way handshake
- SYN/SYN-ACK/ACK
- But what happens on the backend?



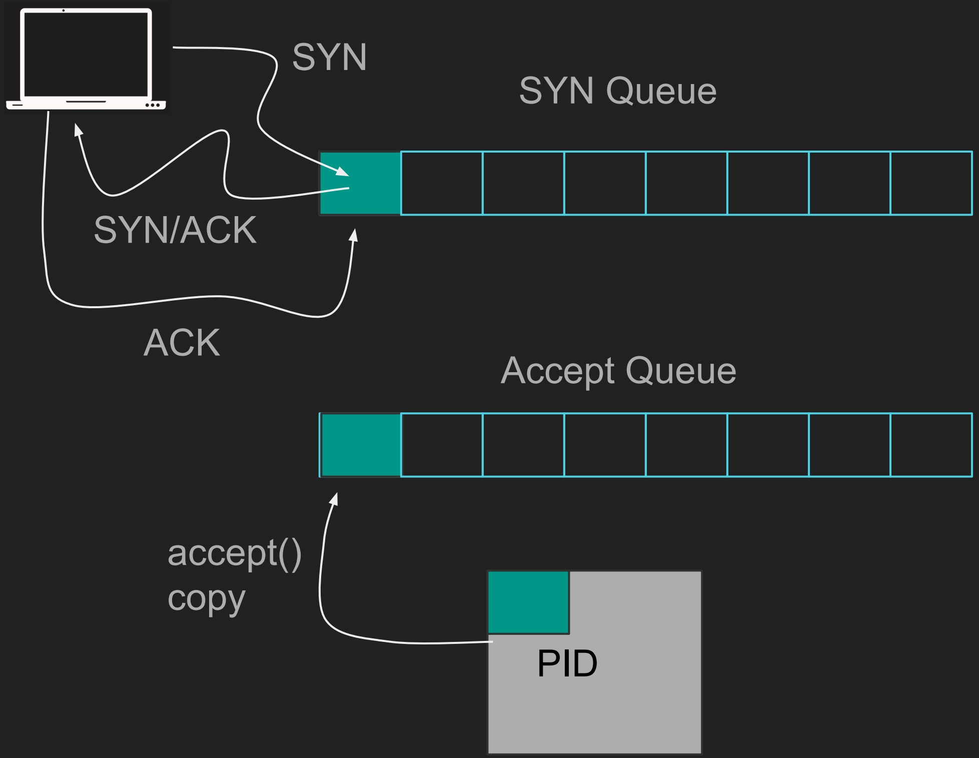
Connection Establishment

- Server Listens on an address:port
- Client connects
- Kernel does the handshake creating a connection
- Backend process “Accepts” the connection

Connection Establishment

- Kernel creates a socket & two queues SYN and Accept
- Client sends a SYN
- Kernels adds to SYN queue, replies with SYN/ACK
- Client replies with ACK
- Kernel finish the connection
- Kernel removes SYN from SYN queue
- Kernel adds full connection to Accept queue
- Backend accepts a connection, removed from accept queue
- A file descriptor is created for the connection

Connection Establishment

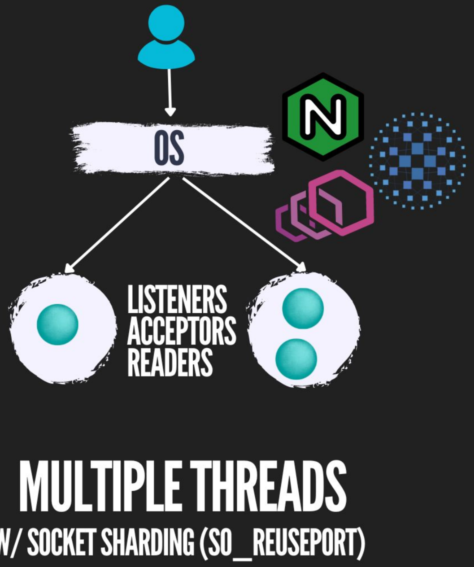


Problems with accepting connections

- Backend doesn’t accept fast enough
- Clients who don’t ACK
- Small backlog

Socket Sharding

- Normally listening on active port/ip fails
- But you can override it with SO_REUSEPORT
- Two distinct sockets different processes on the same ip/port pair



Summary

- Kernel manages networking
- Socket represents a port/ip
- Each connected client gets a connection
- Kernel managed data structures