

CS 392 - Server and client

```

1  #include <sys/socket.h>
2  #include <sys/type.h>
3  #include <sys/socket.h>
4  // queue is how many connections can be
5  int listen(int sockfd, int queue_size);
6
7  // returns file descriptor once connection
   established
8  // used for I/O with client
9  // NOT the one returned by socket() - for
   binding with IP
10 // need to run accept 3 times for 3 clients.
11 // closing this closes connection with
   client.
12 int accept(int sockfd, struct sockaddr*
   addr, socklen_t* addrlen);

```

maybe

```

1  int clients[];
2  for(int i=0; i < queue_size; i++) {
3      clients[i] = accept(...);
4  }

```

For client side

```

1  // client_fd from socket()

```

```

2  // used for I/O
3  connect(client_fd, (struct sockaddr*),
    &server_addr, addr_size);

```

Other

- use `read()` and `write()` instead of `recv` and `send`.
- remember to `memset` buffers.

BUT - This is sequential, which doesn't work well for multiple clients. Thus



- Server cannot `read()` a client fd if nothing has been written yet, so how do you listen to one when it is readable given you don't know which one it will be?

```

1  /* The while loop for "actively" monitoring */
2  while(1) {
3
4      /* Monitor file descriptors */
5      select(...);
6
7      /* Perform I/O when a descriptor is ready */
8      read(...);
9
10 }

```

```

1 #include <sys/select.h>
2 /* initializes the set pointed to by fdset to be empty. */
3 void FD_ZERO(fd_set* fdset);
4
5 /* adds the file descriptor fd to the set
6    pointed to by fdset. */
7 void FD_SET(int fd, fd_set* fdset);
8
9 /* removes the file descriptor fd from the set
10    pointed to by fdset. */
11 void FD_CLR(int fd, fd_set* fdset);
12
13 /* returns true if the file descriptor fd is
14    a member of the set pointed to by fdset. */
15 int FD_ISSET(int fd, fd_set* fdset);

```

```

1 #include <sys/select.h>
2 int select(int nfds, fd_set* readfds, fd_set* writefds,
3           fd_set* exceptfds, struct timeval* timeout);

```

- Using `FD_SET` - used to add fds into selector
- How do you know which is ready? Use `FD_ISSET`
- Calling `select()` will leave only the fd that is ready, the rest are thrown out
 - Need to save the rest elsewhere
- At the start of the loop, reconstruct fd set.

MORE LINUX STUFF

- CS 631 - Advanced coding in UNIX
- CS 615 - System administration