Networking:

Network -> connect computers

Governed by protocols

UDP -> datagram based (SOCK_DGRAM)

↳ connectionless

↳ unreliable

TCP -> stream based (SOCK_STREAM)

↳ connection based (SYN, SYN/ACK, ACK ... each ... the FIN)

↳ reliable

Addressing (IP)

(host, port)

$ mytalk port

Server

-> socket(2)

-> bind (2)

-> listen (2)

-> accept (2)

-> send (2)

:

-> close (2)

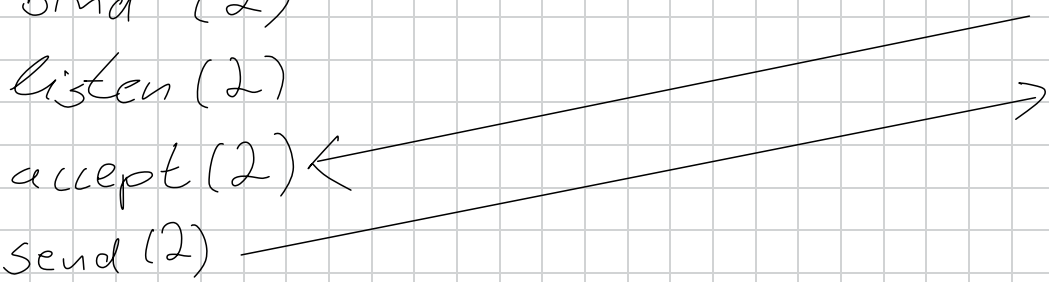
$ mytalk host port

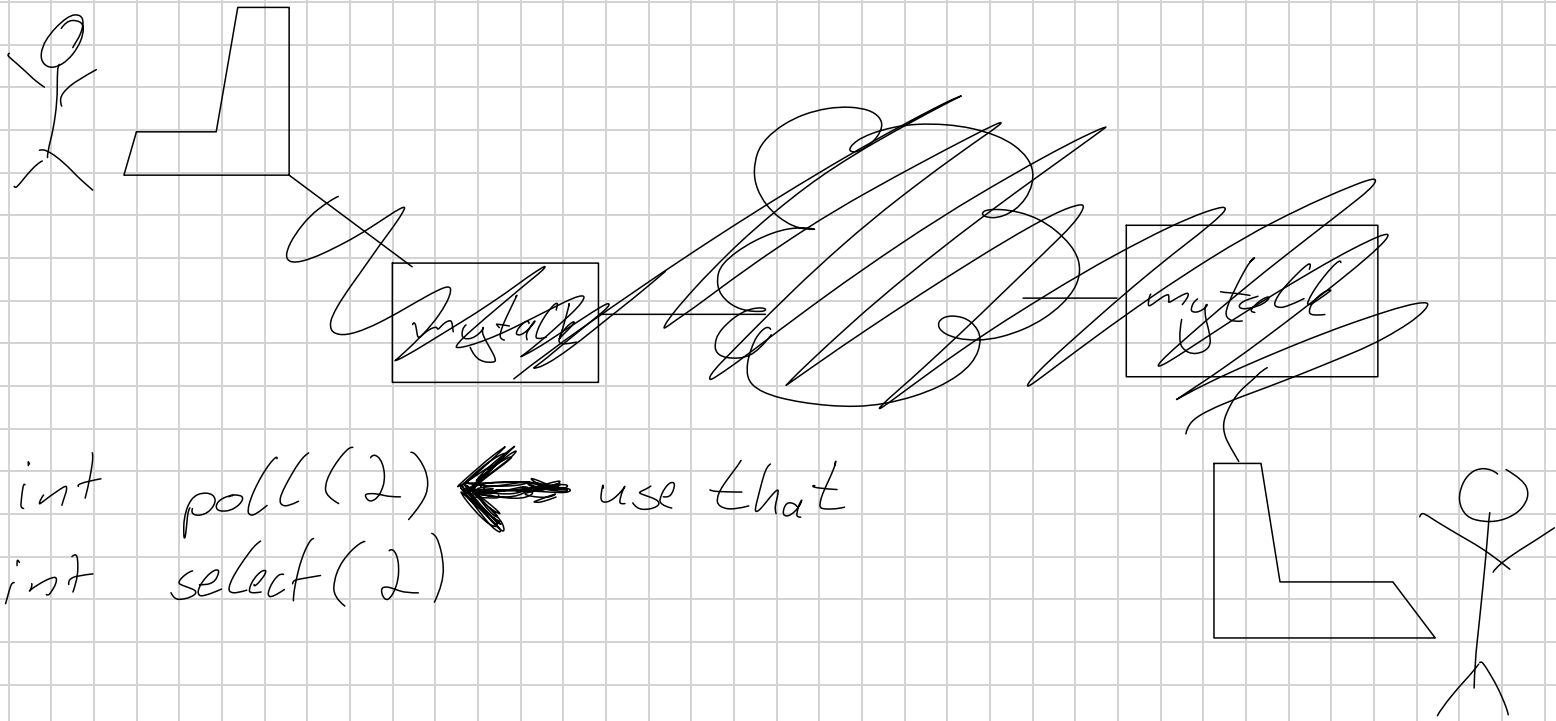
Client

socket(2)

connect (2)

-> recv (2)





int poll(2) ← use that

int select(2)

```
Node *merge (Node *a, Node *b){
    Node *res;
    if (!a)
        res = b;
    else if (!b)
        res = a;
    else if (a->data < b->data){
        res = a;
        a->next = merge(a->next, b);
    }else{
        res = b;
        b->next = merge(a, b->next);
    }
    return res;
}
```

```
Node * merge(Node *a, Node *b){
    Node dummy;
    Node *tail = &dummy;
    while(a && b){
        if(a->data < b->data){
            tail->next = a;
            a = a->next;
            tail = tail->next;
        }else{
            tail->next = b;
            b = b->next;
            tail = tail->next;
        }
    }
    /* hit end */
    if(!a)
        tail->next = b;
    else tail->next = a;
    /* done (yay!) */
    return dummy.next;
}
```