Computer Science 118

Computer Network Fundamentals

Learning Assistant Slides
Week 1

Many firsts

Learning Assistants

Omar Elamri
Discussion 1E, 1F
2nd year, Computer Science
omar@eado.me
Discord: @eado *or* eado.me/cs118discord

Paul Serafimescu
Discussion 1A, 1B
4th year, Computer Science & Engineering
pserafimescu9@g.ucla.edu
Discord: @nullptr_.

About Me

- Incoming MS CS @ UCLA (ESAP)
- CSE + Economics Double Major
- Been an LA for 2 years now
  - Until now, only for CS 35L
- Interned at Cisco IoT
- Feel free to reach out for advice about whatever

bit.ly/s24cs118

Activity

Let’s get to know each other!

Everyone, stand up and **order yourselves by first name**.

What works best for you?

Since we get to create new content, you get to decide!

- Weekly notes
- Slides (like this)
- Interactive activities (like the stand-up)
- Code exploration (for projects; in a GitHub Codespace)

Form: eado.me/cs118form

Socket Programming Basics

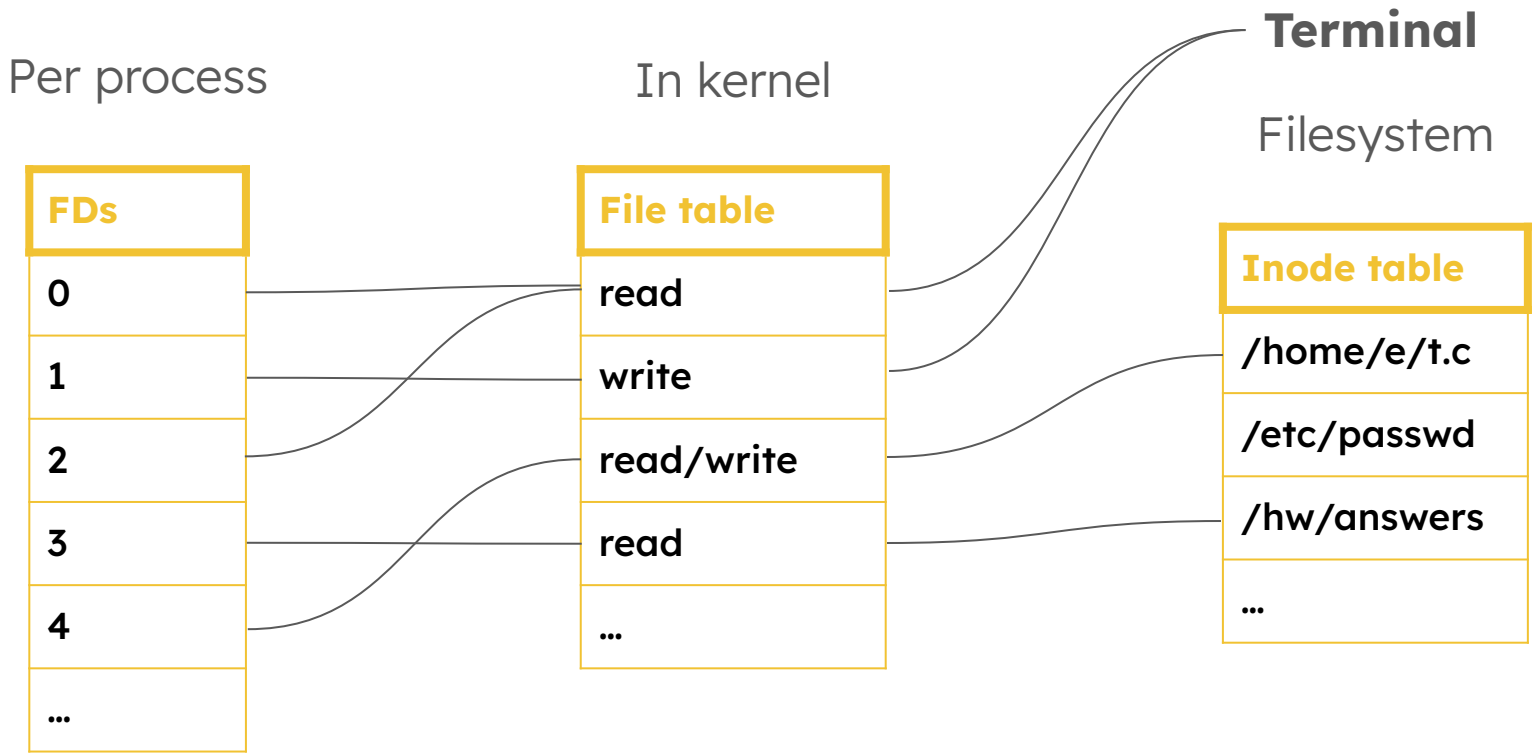
What is a socket?

Wikipedia:

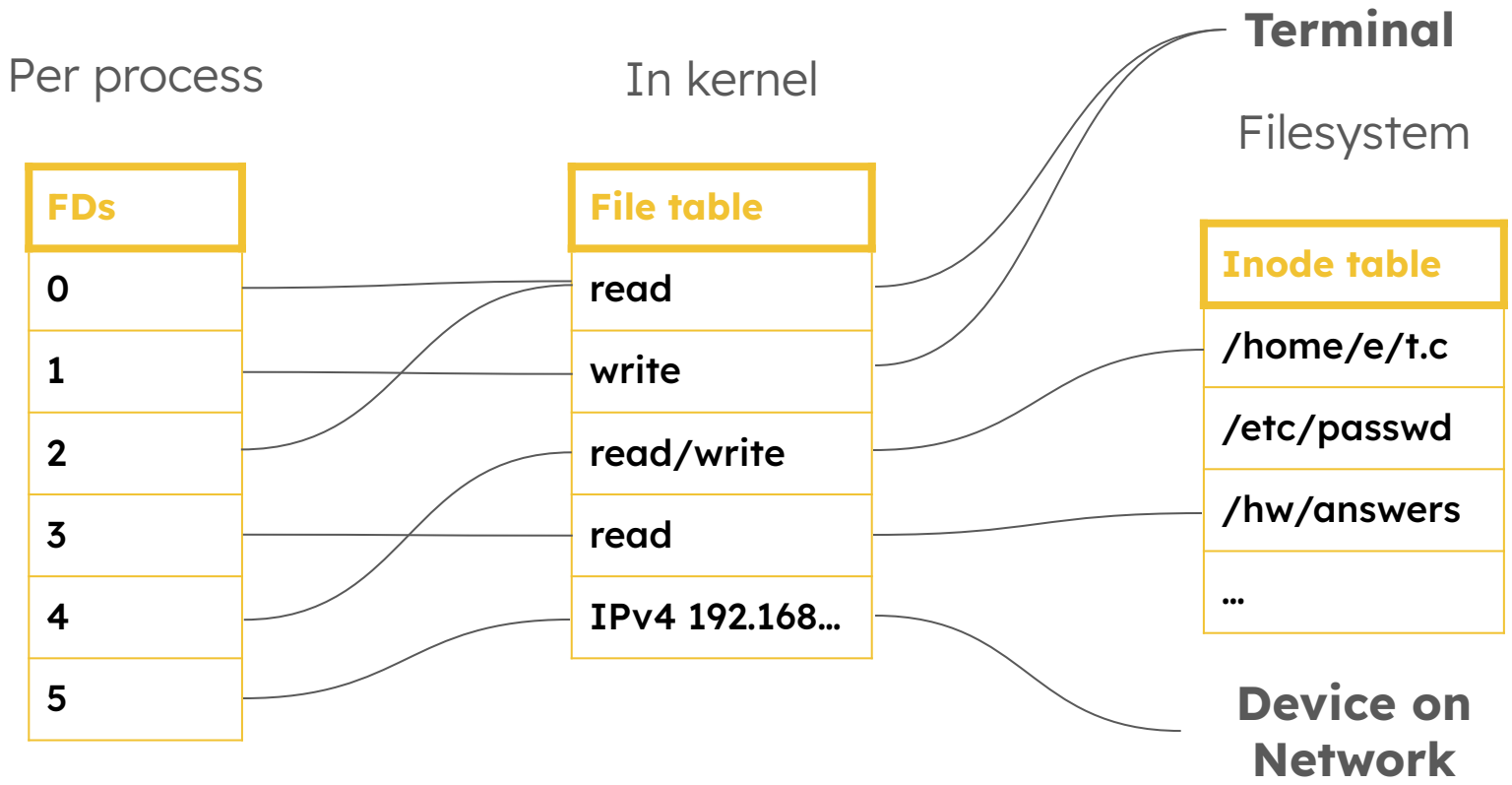
A **network socket** is a software structure within a network node of a computer network that serves as an endpoint for sending and receiving data across the network. The structure and properties of a socket are defined by an application programming interface (API) for the networking architecture. Sockets are created only during the lifetime of a process of an application running in the node.

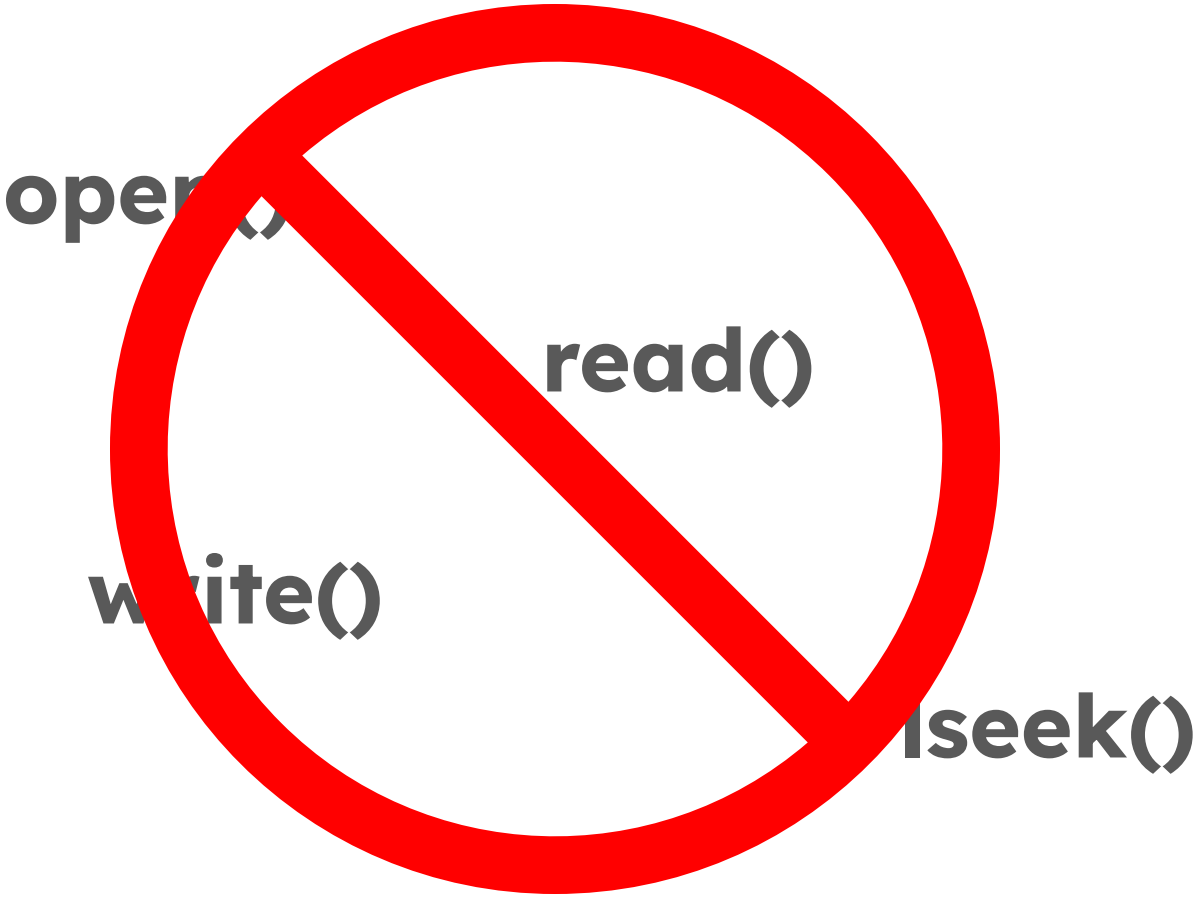
An abstraction that lets us interface with networked devices

File Descriptors (POSIX)

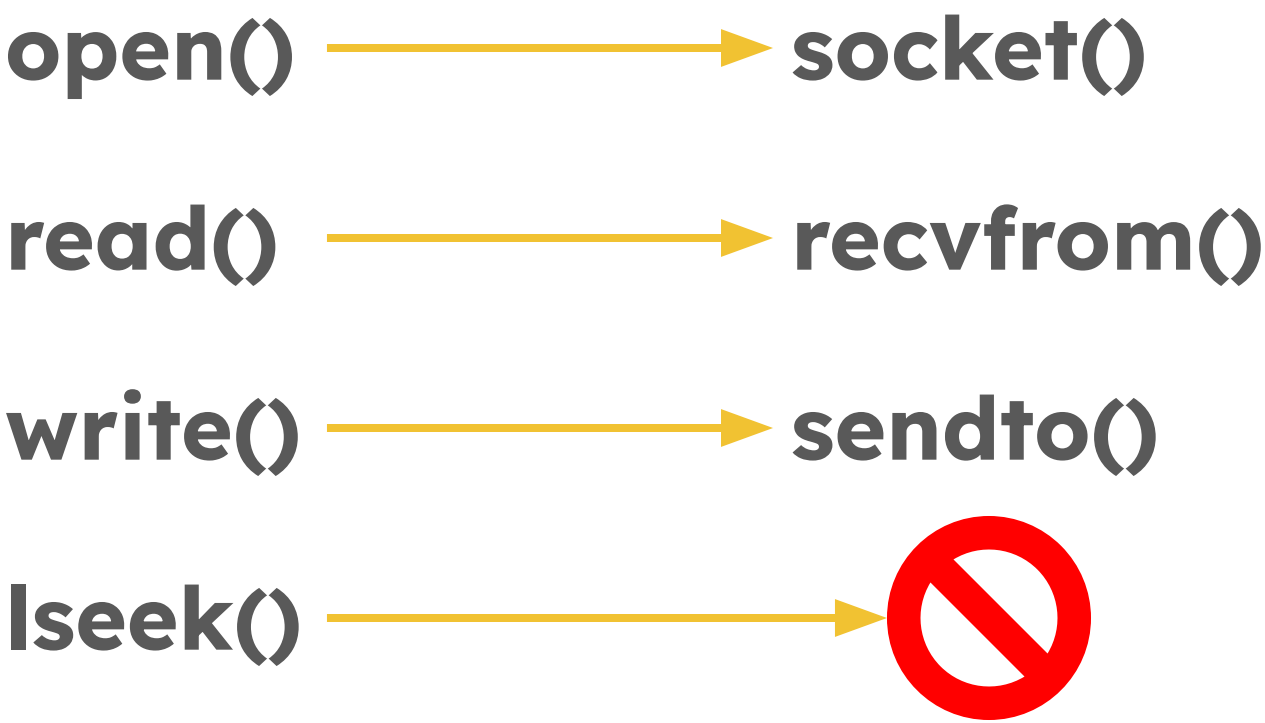


Now, with sockets




read()
lseek()

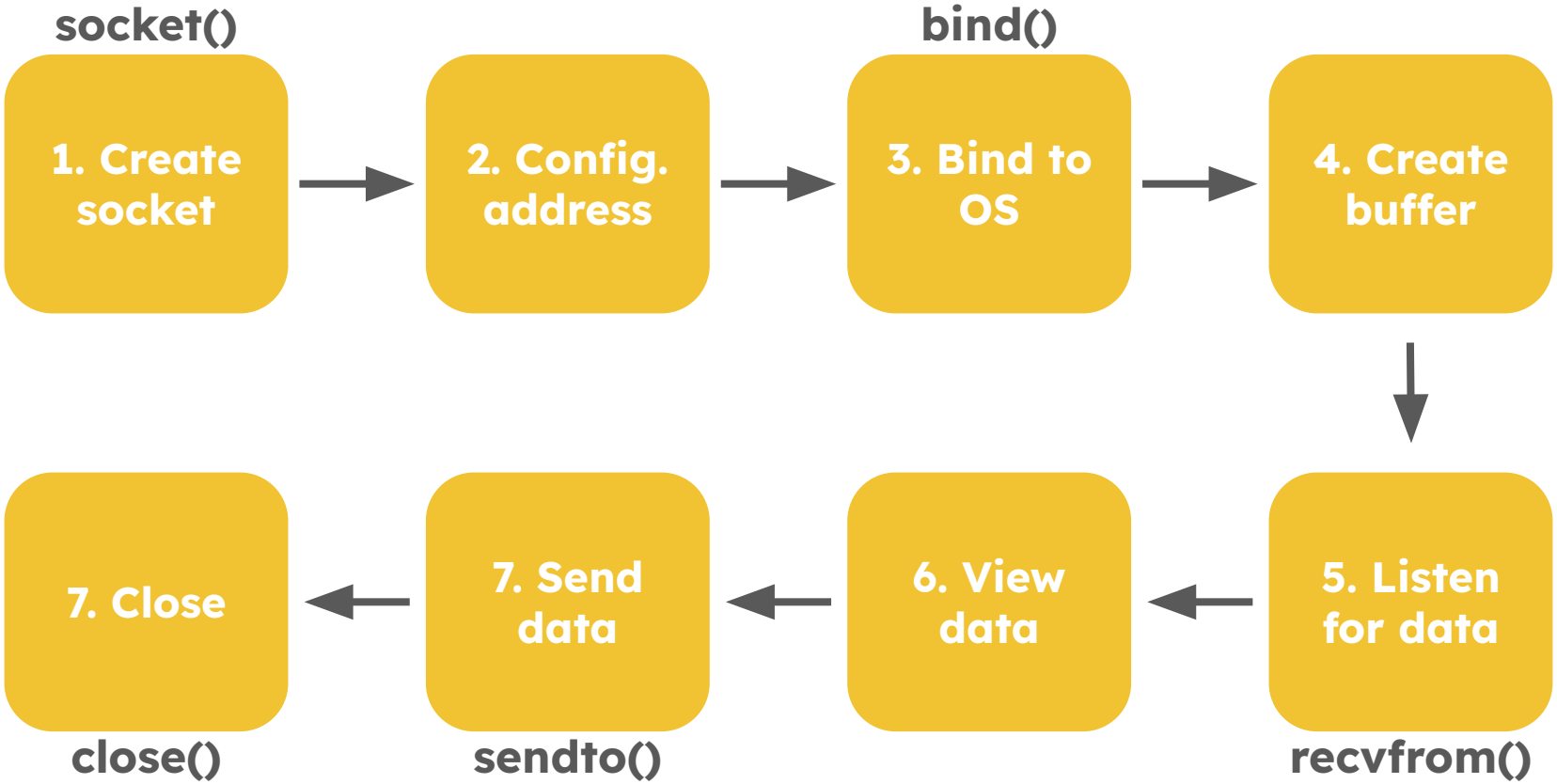
Sockets are special



Sockets are special

- Must specify
  - network layer **protocol**, **type**
- For Internet Protocol
  - source/destination **address** and **port**
- Network communication is always **BIG ENDIAN**

Datagram Server Steps



```c
// main.c
#include <sys/socket.h>
#include <arpa/inet.h>
#include <string.h>
#include <unistd.h>
int main() {
/* 1. Create socket */
int sockfd = socket(AF_INET, SOCK_DGRAM, 0);
                    // use IPv4  use UDP
```

```c
/* 2. Construct our address */
struct sockaddr_in servaddr;
servaddr.sin_family = AF_INET; // use IPv4
servaddr.sin_addr.s_addr = INADDR_ANY; // accept all
                           // same as inet_addr("0.0.0.0")
                           // "Address string to network bytes"

// Set receiving port
int PORT = 8080;
servaddr.sin_port = htons(PORT); // Big endian
```

```c
/* 3. Let operating system know about our config */
int did_bind = bind(sockfd, (struct sockaddr*) &servaddr,
                    sizeof(servaddr));
// Error if did_bind < 0 :(
```

```c
/* 4. Create buffer to store incoming data */
int BUF_SIZE = 1024;
int client_buf[BUF_SIZE];
struct sockaddr_in clientaddr; // Same information, but client
socklen_t clientsize= sizeof(clientaddr);
```

```c
/* 5. Listen for data from clients */
int bytes_recvd = recvfrom(sockfd, client_buf, BUF_SIZE,
                           // socket  store data  how much
                              0, (struct sockaddr*) &clientaddr,
                           // flags   buffer for client address
                              &clientsize);
// Execution will stop here until `BUF_SIZE` is read
// or termination/error
// Error if bytes_recvd < 0 :(
```

```c
/* 6. Inspect data from client */
// Remember, client data is in CLIENT_BUF
char* client_ip = inet_ntoa(clientaddr.sin_addr);
// "Network bytes to address string"
int client_port = ntohs(clientaddr.sin_port); // Little endian
```

```c
/* 7. Send data back to client */
char server_buf[] = "Hello world!";
int did_send = sendto(sockfd, server_buf, strlen(server_buf),
                      // socket  send data    how much to send
                         0, (struct sockaddr*) &clientaddr,
                      // flags       where to send
                         sizeof(clientaddr));
```

```
/* 8. You're done! Terminate the connection */
close(sockfd);
return 0;
}
```

…and you're done!

- More material for client and TCP coming soon
- Next week: exploration with sockets
- You can view this code in more detail on the LA webpage

eado.me/cs118 > Miscellaneous > Socket Programming Tips