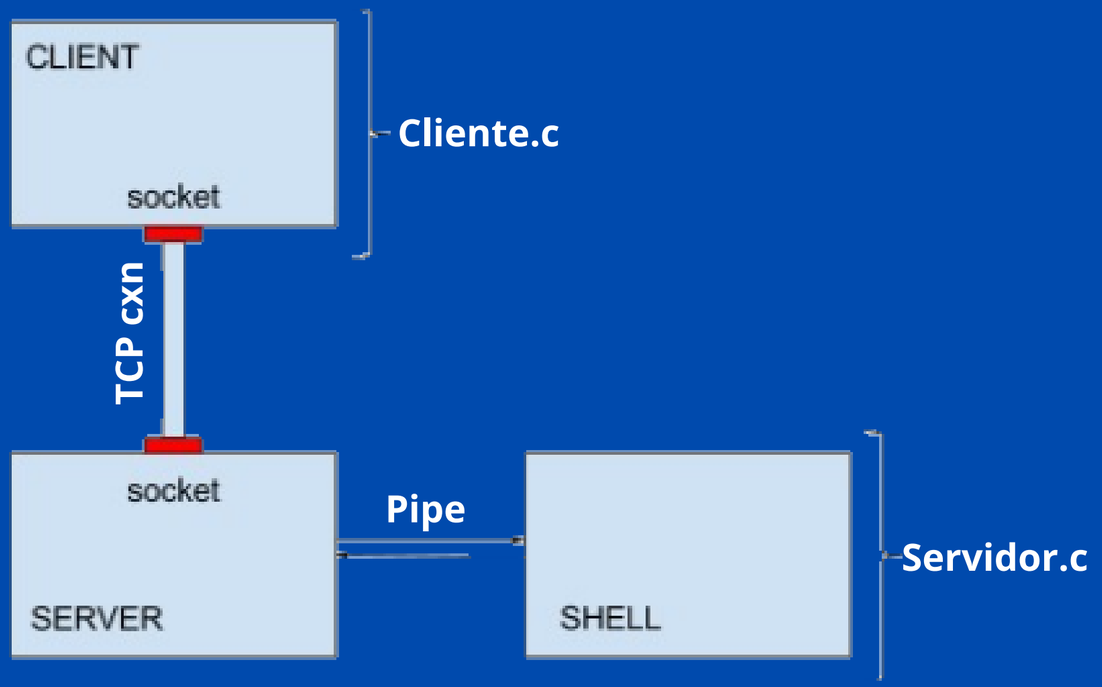
Compressed Network Communication

Comunicação de rede compactada

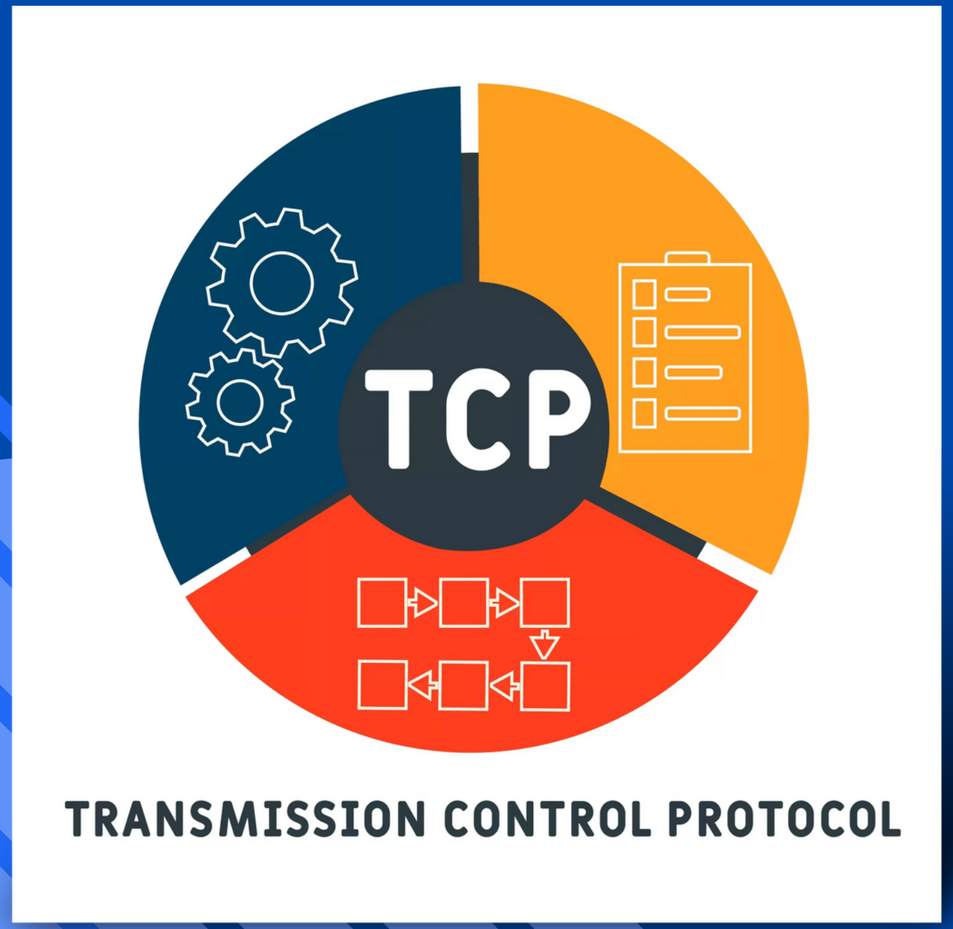
ENTENDA COMO FUNCIONA



CNC

TCP

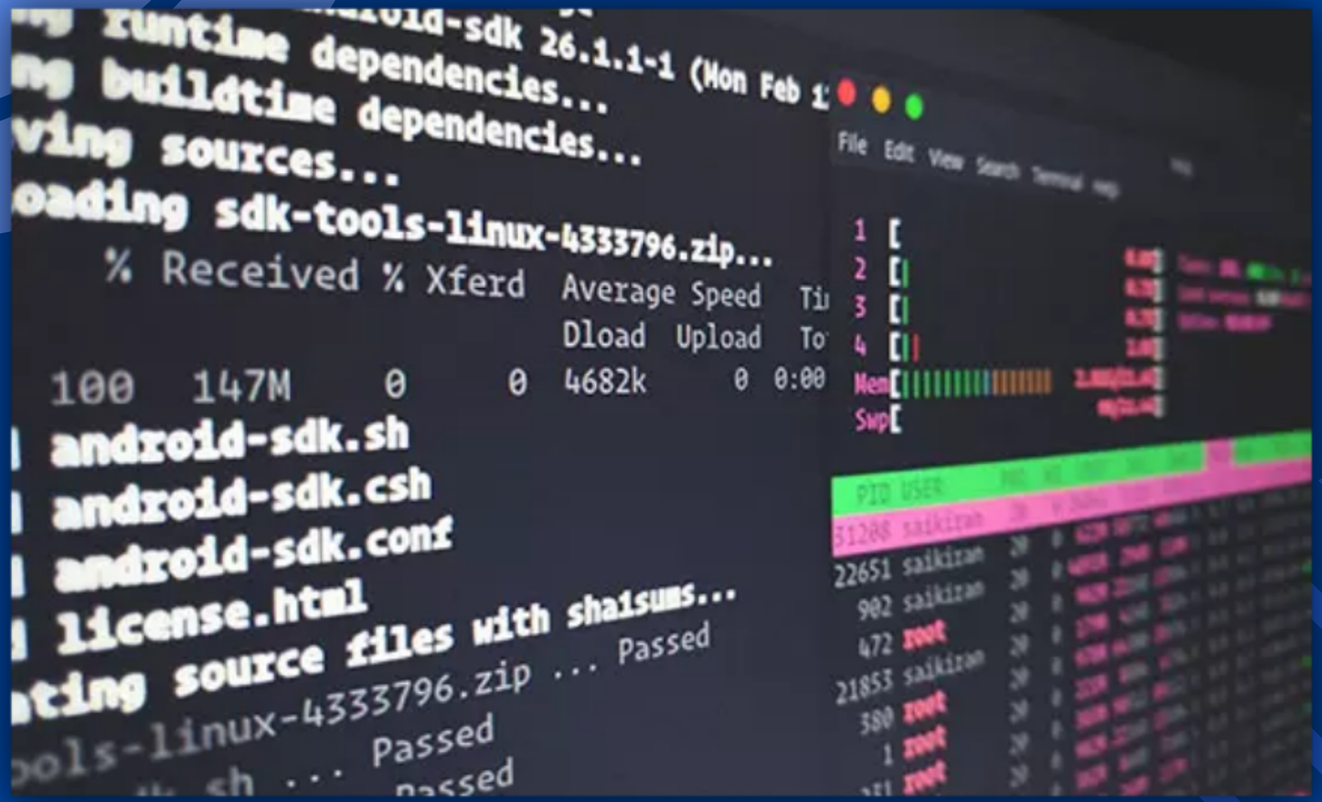
- **Protocolo de Transmissão**
- **4 Camadas** (Appli./Transp./Rede/Acesso a rede)
- **Verificação dos dados**
- **Three-Way Handshake** (Syn, Ack-Syn, Ack)



Sockets

- Comunicação entre Processos
- Envio e recebimento de mensagens de um processo
- Podem ou não estar na mesma máquina
- camada de aplicação e a de transporte dentro de um hospedeiro
- funciona como uma porta
- Ex: Requisitar paginas WEB

Fork



- Duplicação de Processos
- Processo "pai" vai indicar o processo do "filho" e vice versa
- O filho herda assim as informações contidas no processo-pai.

Pipes

- Comunicação unidirecional
- Par de arquivos que é criado num "processo pai-filho”.
- É dito como anônimo.
- Lado de escrita e o lado de leitura.

A comunicação

TCP X UDP

INTEGRIDADE DE DADOS

```c
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <unistd.h>
#include <arpa/inet.h>

void main(void){
    char cadena [100];
    int sockfd;
    struct sockaddr_in servaddr;
    FILE *myf = fopen("Conversação_Cliente.txt", "a");
    time_t t;
    struct tm *tm;
    char hora [100];
    char *tmp;
    char sendline [100] = "Usando a porta 22000\n";

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    bzero (&servaddr, sizeof(servaddr));

    servaddr.sin_family = AF_INET;
    servaddr.sin_port = htons (22000);

    inet_pton(AF_INET,"172.17.0.1", &(servaddr.sin_addr));
    connect(sockfd, (struct sockaddr*)&servaddr, sizeof(servaddr));

    printf("\n\n\t\t------Chat Inicializado------\n\n");
    fputs("\n\n\t\t------Chat Inicializado------\n\n", myf);

    while (!strstr(cadena, "Até") && !strstr(sendline, "Até")){
        bzero(cadena, 100);
        t = time(NULL);
        tm = localtime (&t);
        strftime (hora, 100, "\n Eu (%H:%M) -> ", tm);
        printf("%s", hora);
        fgets (sendline, 100, stdin);
        tmp = strcat(hora, sendline);
        fputs (tmp, myf);
        write(sockfd, sendline, strlen(sendline)+1);
        if(!strstr(cadena, "Até")){
            strftime (hora, 100, "\n Servidor (%H:%M) -> ", tm);
            read (sockfd, cadena, 100);
            tmp = strcat (hora, cadena);
            printf("%s", tmp);
            fputs (tmp, myf);
    }
}
printf("\n\n Bate Papo finalizado\n");
printf("\n O arquivo foi gerado-> Conversa_Cliente.txt.txt");
fclose(myf);
}
```

Programa Servidor com Chat

SERVER.C

```c
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <netinet/in.h>
#include <netinet/ip.h>
#include <unistd.h>

void main() {
    char caractere[100];
    int listen_fd, comm_fd;
    struct sockaddr_in servaddr;
    FILE *myf = fopen("Conversação_servidor.txt", "a");
    time_t t;
    struct tm *tm;
    char hora[100];
    char *tmp;
    char sendline[ 100] = "Usando a porta 22000\n";

    listen_fd = socket(AF_INET, SOCK_STREAM, 0);
    bzero(&servaddr,sizeof(servaddr));

    servaddr.sin_family = AF_INET;
    servaddr.sin_addr.s_addr = htons(INADDR_ANY);
    servaddr.sin_port = htons(22000);

    bind(listen_fd,(struct sockaddr *)&servaddr, sizeof(servaddr));
    listen(listen_fd, 10);
    comm_fd = accept(listen_fd, (struct sockaddr*) NULL, NULL);

    printf("\n\n\t\t----Chat Inicializado----\n\n");
    fputs("n\n\t\t----Chat Inicializado------ - \n\n", myf);

    while(!strstr(cadena, "Até") && !strstr(sendline, "Até ")){
        bzero(caractere, 100);
        t = time(NULL);
        tm = localtime(&t);
        strftime(hora, 100, "\n Cliente (%H:%M) ->", tm);

        read(comm_fd, caractere, 100);
        tmp = strcat(hora, caractere);
        printf("%s", tmp);
        fputs (tmp, myf);
        if(!strstr(caractere,"Até")){
            strftime (hora, 100, "\n Eu (%H:%M) -> ", tm);
            printf("%s", hora);
            fgets(sendline, 100, stdin);
            tmp = strcat(hora, sendline);
            write(comm_fd, sendline, strlen(sendline)+1);
            fputs (tmp, myf);
    }
}
printf("\n\n Conversa finalizada \n");
printf("\n O arquivo foi gerado-> Conversa_servidor.txt.txt");
fclose(myf);
}
```

Programa Cliente com Chat

CLIENTE.C

Obrigado pela atenção

NUNCA DESISTA DOS SEUS SONHOS