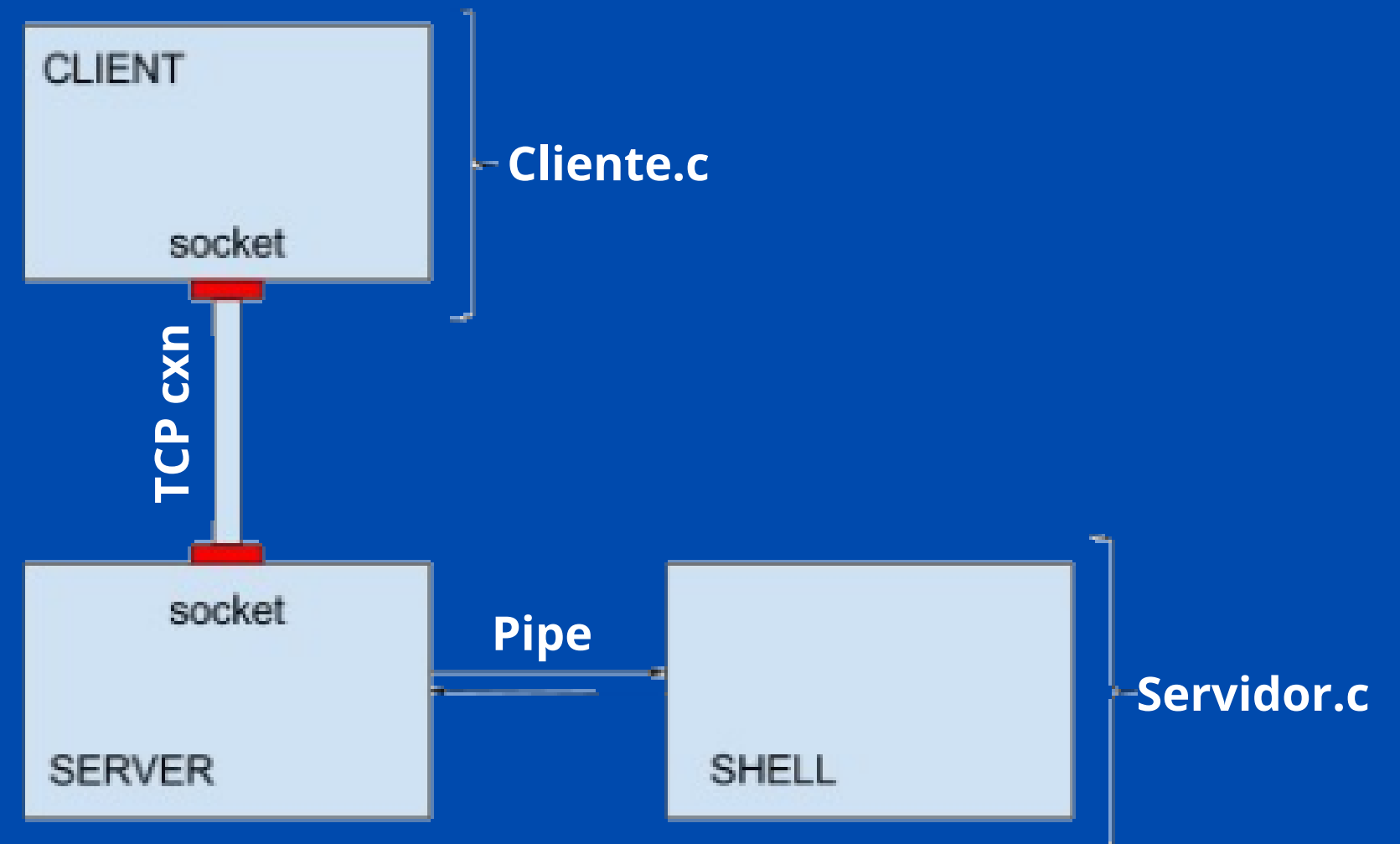


Compressed Network Communication

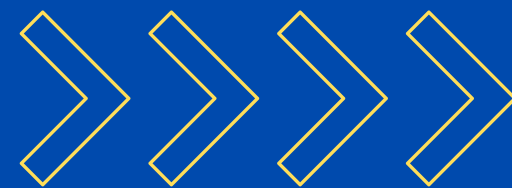
Comunicação de rede compactada

ENTENDA COMO FUNCIONA

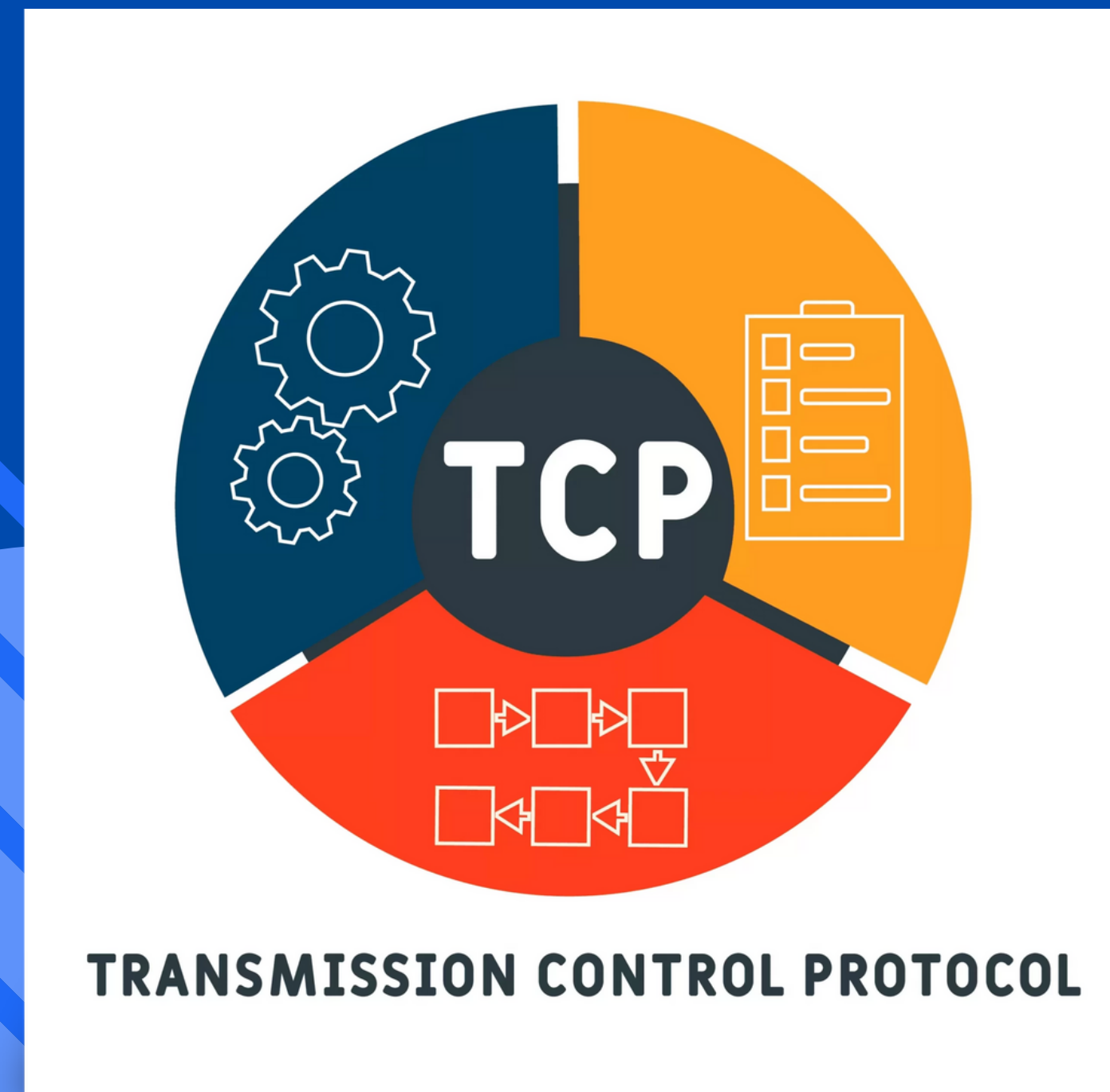


CNC

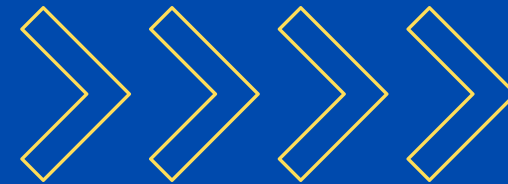
TCP



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



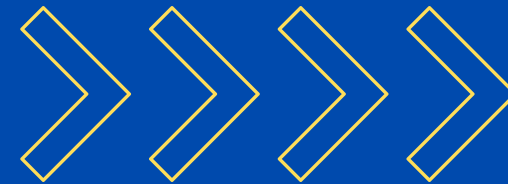
Sockets



- 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



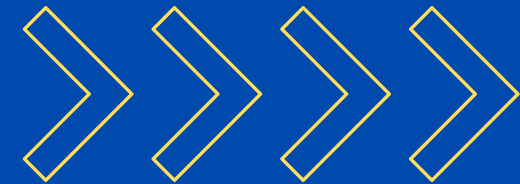
- Duplicação de Processos
- Processo "pai" vai indicar o processo do "filho" e vice versa
- O filho herda assim as informações contidas no processo-pai.

```
android-sdk 26.1.1-1 (Mon Feb 1)
runtime dependencies...
buildtime dependencies...
loading sources...
loading sdk-tools-linux-4333796.zip...
% Received % Xferd Average Speed Time Ti
Dload Upload To
100 147M 0 0 4682k 0 0:00
android-sdk.sh
android-sdk.csh
android-sdk.conf
license.html
Verifying source files with sha1sums... Passed
sdk-tools-linux-4333796.zip ... Passed
sh ... Passed
```

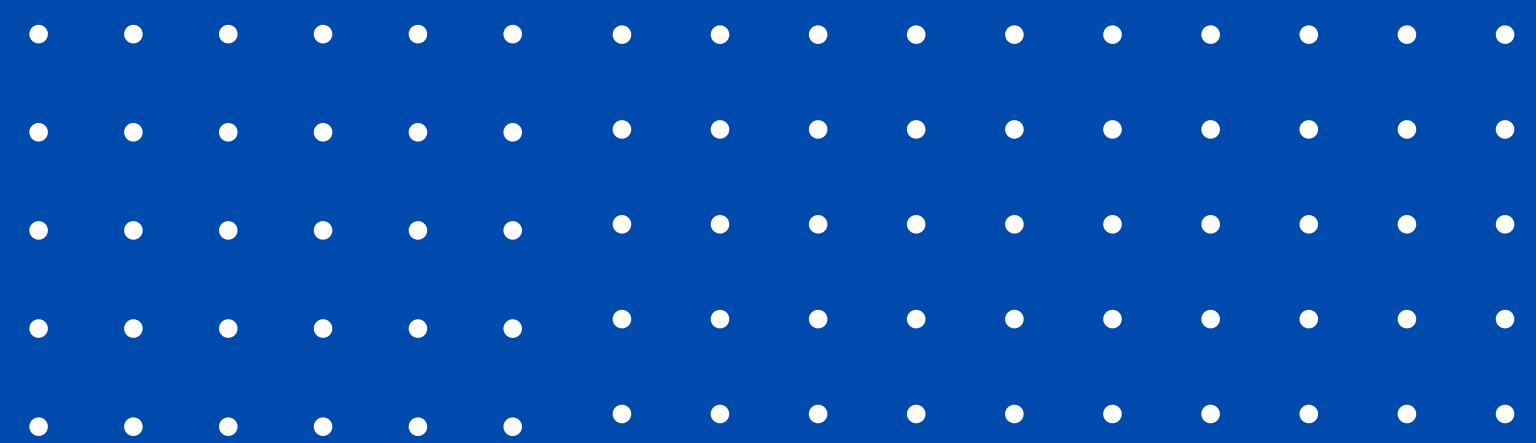
PID	USER	PR	NI	PPID	PPRI	NI	PPRI	NI	PPRI
51288	saikiran	20	0	429	20	0	20	0	20
22651	saikiran	20	0	429	20	0	20	0	20
902	saikiran	20	0	429	20	0	20	0	20
472	root	20	0	429	20	0	20	0	20
21853	saikiran	20	0	429	20	0	20	0	20
380	root	20	0	429	20	0	20	0	20
1	root	20	0	429	20	0	20	0	20
1	root	20	0	429	20	0	20	0	20



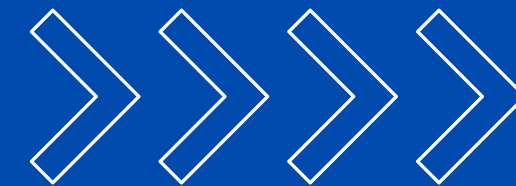
Pipes



- 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



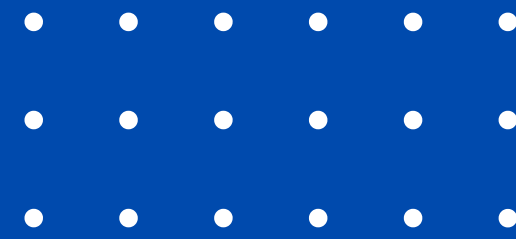
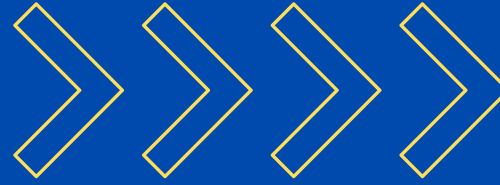
TCP X UDP

INTEGRIDADE DE DADOS


```

1  #include <sys/types.h>
2  #include <sys/socket.h>
3  #include <netdb.h>
4  #include <stdio.h>
5  #include <string.h>
6  #include <time.h>
7  #include <netinet/in.h>
8  #include <netinet/ip.h>
9  #include <unistd.h>
10 #include <arpa/inet.h>
11
12 void main(void){
13     char cadena [100];
14     int sockfd;
15     struct sockaddr_in servaddr;
16     FILE *myf = fopen("Conversação_Cliente.txt", "a");
17     time_t t;
18     struct tm *tm;
19     char hora [100];
20     char *tmp;
21     char sendline [100] = "Usando a porta 22000\n";
22
23     sockfd = socket(AF_INET, SOCK_STREAM, 0);
24     bzero (&servaddr, sizeof(servaddr));
25
26     servaddr.sin_family = AF_INET;
27     servaddr.sin_port = htons (22000);
28
29     inet_pton(AF_INET,"172.17.0.1", &(servaddr.sin_addr));
30     connect(sockfd, (struct sockaddr*)&servaddr, sizeof(servaddr));
31
32     printf("\n\n\t\t-----Chat Inicializado-----\n\n");
33     fputs("\n\n\t\t-----Chat Inicializado-----\n\n", myf);
34
35     while (!strstr(cadena, "Até") && !strstr(sendline, "Até")){
36         bzero(cadena, 100);
37         t = time(NULL);
38         tm = localtime (&t);
39         strftime (hora, 100, "\n Eu (%H:%M) -> ", tm);
40         printf("%s", hora);
41         fgets (sendline, 100, stdin);
42         tmp = strcat(hora, sendline);
43         fputs (tmp, myf);
44         write(sockfd, sendline, strlen(sendline)+1);
45         if(!strstr(cadena, "Até")){
46             strftime (hora, 100, "\n Servidor (%H:%M) -> ", tm);
47             read (sockfd, cadena, 100);
48             tmp = strcat (hora, cadena);
49             printf("%s", tmp);
50             fputs (tmp, myf);
51         }
52     }
53     printf("\n\n Bate Papo finalizado\n");
54     printf("\n O arquivo foi gerado-> Conversa_Cliente.txt.txt");
55     fclose(myf);
56 }

```



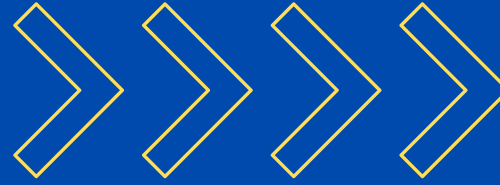
Programa Servidor com Chat

SERVER.C

```

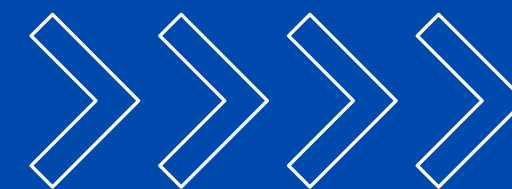
1  #include <sys/types.h>
2  #include <sys/socket.h>
3  #include <netdb.h>
4  #include <stdio.h>
5  #include <string.h>
6  #include <time.h>
7  #include <netinet/in.h>
8  #include <netinet/ip.h>
9  #include <unistd.h>
10
11 void main() {
12     char caractere[100];
13     int listen_fd, comm_fd;
14     struct sockaddr_in servaddr;
15     FILE *myf = fopen("Conversação_servidor.txt", "a");
16     time_t t;
17     struct tm *tm;
18     char hora[100];
19     char *tmp;
20     char sendline[ 100] = "Usando a porta 22000\n";
21
22     listen_fd = socket(AF_INET, SOCK_STREAM, 0);
23     bzero(&servaddr, sizeof(servaddr));
24
25     servaddr.sin_family = AF_INET;
26     servaddr.sin_addr.s_addr = htonl(INADDR_ANY);
27     servaddr.sin_port = htons(22000);
28
29     bind(listen_fd, (struct sockaddr *)&servaddr, sizeof(servaddr));
30     listen(listen_fd, 10);
31     comm_fd = accept(listen_fd, (struct sockaddr*) NULL, NULL);
32
33     printf("\n\n\t\t----Chat Inicializado----\n\n");
34     fputs("n\n\t\t----Chat Inicializado----- - \n\n", myf);
35
36     while(!strstr(cadena, "Até") && !strstr(sendline, "Até ")){
37         bzero(caractere, 100);
38         t = time(NULL);
39         tm = localtime(&t);
40         strftime(hora, 100, "\n Cliente (%H:%M) ->", tm);
41
42         read(comm_fd, caractere, 100);
43         tmp = strcat(hora, caractere);
44         printf("%s", tmp);
45         fputs (tmp, myf);
46         if(!strstr(caractere, "Até")){
47             strftime (hora, 100, "\n Eu (%H:%M) -> ", tm);
48             printf("%s", hora);
49             fgets(sendline, 100, stdin);
50             tmp = strcat(hora, sendline);
51             write(comm_fd, sendline, strlen(sendline)+1);
52             fputs (tmp, myf);
53         }
54     }
55     printf("\n\n Conversa finalizada \n");
56     printf("\n O arquivo foi gerado-> Conversa_servidor.txt.txt");
57     fclose(myf);
58 }

```



Programa Cliente com Chat

CLIENTE.C



Obrigado pela atenção

**NUNCA DESISTA DOS
SEUS SONHOS**

