

Exercício 3

Grupo:

Daniel Helú - 166215

Gustavo Fernandez - 169296

1.

inet_pton: converte uma string de caracteres dada em uma estrutura de endereço da família desejada, que pode ser AF_INET ou AF_INET6.

htons: converte a ordem de bytes de um short integer do host para a ordem de bytes da rede. A ordem de bytes está relacionado com o arquitetura da máquina interpretar os bytes como o byte mais significativo primeiro e o menos significativo depois ou o contrário.

htonl: converte a ordem de bytes de um long integer do host para a ordem de bytes da rede.

A diferença entre elas é que a função `inet_pton` transforma uma string de caracteres em uma estrutura binária de endereço, tanto para IPv4 quanto IPv6. Já as funções `hton` exigem um inteiro a ser transformado e se importam apenas quanto à ordenação deste.

2.

Erro 1: linha 33 do cliente.c

```
servaddr.sin_port = 8000;
```

A linha está errada, pois o código está estabelecendo na máquina um valor para um atributo da rede, sem se preocupar com uma discordância entre elas em relação à ordenação de bytes. Para corrigi-lo deve-se utilizar a função `htons` como abaixo.

```
servaddr.sin_port = htons(8000);
```

Erro 2: linha 36 do cliente.c

```
connect(sockfd, (struct sockaddr *)&servaddr, 1);
```

O último parâmetro da função *connect* deve ser o tamanho da estrutura de endereço enviada. 1 não é um valor válido, pois a estrutura não possui este tamanho. Para corrigi-lo basta fazer uso da função *sizeof*, como abaixo.

```
connect(sockfd, (struct sockaddr *) &servaddr, sizeof(servaddr));
```

Feitas as alterações acima, basta compilar e executar ambos programas. Sendo que primeiro deve ser o servidor.c e depois o cliente.c, além disso o programa do cliente deve ser escutado com um parâmetro de endereço de IP, o IP do loopback é suficiente.

3.

Utilizando o comando *netstat* como exibido abaixo, podemos analisar o tráfego de redes na interface de loopback.

```
netstat -I=lo -c
```

A saída obtida foi:

Iface	MTU	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
lo	65536	16	0	0	0	16	0	0	0	LRU
lo	65536	16	0	0	0	16	0	0	0	LRU
lo	65536	16	0	0	0	16	0	0	0	LRU
lo	65536	16	0	0	0	16	0	0	0	LRU
lo	65536	24	0	0	0	24	0	0	0	LRU
lo	65536	24	0	0	0	24	0	0	0	LRU
lo	65536	32	0	0	0	32	0	0	0	LRU
lo	65536	32	0	0	0	32	0	0	0	LRU

O programa cliente foi executado após a quarta linha acima e novamente após a sexta. E percebe-se a alteração nas colunas RX-OK (pacotes recebidos sem erro) e TX-OK (pacotes enviados sem erro), o que comprova a comunicação via esta interface.

6.

Na primeira vez que executamos três vezes o cliente, não houve nenhum estado de TIME_WAIT. Então para verificar quando esse estado ocorreria modificamos o código do servidor para encerrar a conexão sem o *read* do *close* do cliente. Com essa alteração podemos observar o estado TIME_WAIT para a conexão do cliente, como visto na última linha da imagem abaixo.

```
Active Internet connections (w/o servers)
```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	legiao.lab.ic.unic:swat	cebolinha.lab.ic.un:nfs	ESTABLISHED
tcp	0	0	legiao.lab.ic.uni:56920	149.154.175.50:https	ESTABLISHED
tcp	0	0	legiao.lab.ic.uni:49040	gru06s34-in-f3.1e:https	ESTABLISHED
tcp	0	0	legiao.lab.ic.uni:irdmi	kraft.lab.ic.unic:59372	TIME_WAIT

7.

É possível utilizar o telnet no lugar do cliente.c, pois o servidor está configurado para criar uma conexão do tipo TCP com qualquer aplicação que fizer uma requisição a sua porta de listen. Abaixo há um print do netstat da conexão.

```
ra169296:~ $ netstat -c -t | grep "kraft"
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
tcp        0      0 legiao.lab.ic.uni:irdmi kraft.lab.ic.unic:59444 ESTABLISHED
```

Não seria possível utilizar o telnet se o servidor fosse alterado para uma conexão de tipo UDP, já que o telnet depende de uma conexão contínua, o que não é garantido por UDP, pois uma vez que os pacotes são enviados a conexão é encerrada.