

Exercício 3 de MC833 — Programação em Redes de Computadores

Raul Rabelo Carvalho, 105607, turma A

14 de Março de 2014

1 htons

A função `htons` faz parte de um conjunto de funções usadas para converter um inteiro, tanto de 16 bits quanto de 32 bits, entre a *byte order* de rede (*big endian*) e a *byte order* do *host* local, que pode ser *big endian* ou *little endian* a depender da arquitetura. É óbvio que caso a arquitetura do *host* seja *big endian*, nada é feito.

Especificamente, a função `htons` converte inteiros de 16 bits da *byte order* do *host* para da rede.

2 Execução com código-fonte não-modificado

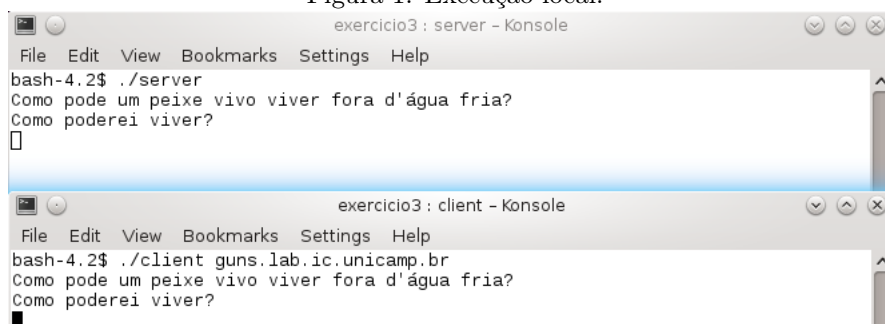
Não é possível executar o servidor sem alterações no *host* do laboratório do IC (`guns.ic.unicamp.br`), pois a função `bind` não tem permissão para utilizar a porta 10. Esta porta é uma das *well-known ports* e estão bloqueadas para usuários sem privilégios nas máquinas do IC.

3 Resolvendo o problema do bind

Alterando-se a porta empregada para 1234, tanto em `client.c` quanto em `server.c`, resolve-se o problema do servidor não conseguir reservar para si uma das *well-known ports*. No entanto, o cliente não consegue se conectar a esta porta, provavelmente devido à alguma política de bloqueio de portas no roteador dos laboratórios.

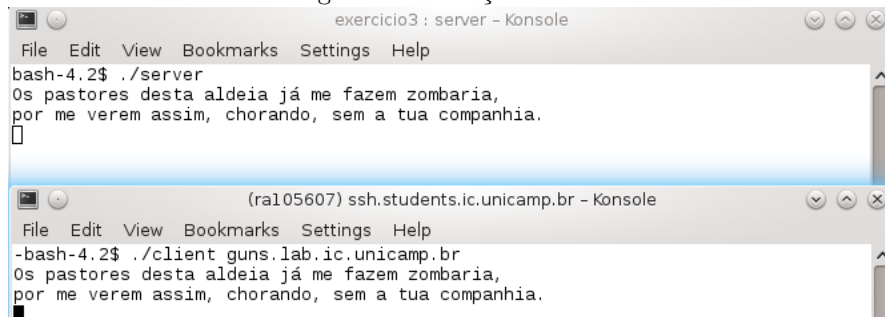
Fazendo uma segunda alteração para a porta 10000, o servidor consegue reservar a porta e executar normalmente, e o cliente consegue conectar-se a esta porta e comunicar-se com o servidor, como na figura 1.

Figura 1: Execução local.



Também foi possível executar o servidor no *host* local e o cliente na máquina *xaveco* via `ssh`, executado com o comando `./client guns.lab.ic.unicamp.br`.

Figura 2: Execução remota.



4 Múltiplos clientes

5 Verificando o uso da rede

Pode-se configurar a porta usada no servidor para uma porta disponível no *host* em que ele vá ser executado. Com conhecimento desta porta, se estabelece a conexão entre cliente e servidor normalmente. Em um terceiro **terminal**, executa-se o comando **netstat -tu** e, se a comunicação entre cliente e servidor se faz por rede, uma conexão estabelecida com endereço IP do *host* do servidor e a porta TCP empregada estará listada na saída desta ferramenta.

Como mostrado na figura 3, existe uma conexão estabelecida entre o servidor em **xaveco.lab.ic.unicamp.br**, usando a porta 10101. Foi usado o comando **netstat -tun**, pois a porta 10101, apesar de estar acima de 1024, tem uma aplicação conhecida associada a ela. A conexão em questão está listada na terceira linha da saída do comando **netstat** e um *reverse nslookup* foi feito para se confirmar que o endereço IP 143.106.16.163 corresponde ao nome **xaveco.lab.ic.unicamp.br**.

6 telnet

Figura 3: Netstat -tun.

```
(ra105607) ssh.students.ic.unicamp.br - Konsole
File Edit View Bookmarks Settings Help
-bash-4.2$ ./server
Tinha uma pedra no meio do caminho
No meio do caminho tinha uma pedra

(ra105607) ssh2.students.ic.unicamp.br - Konsole
File Edit View Bookmarks Settings Help
-bash-3.2$ ./client xaveco.lab.ic.unicamp.br
Tinha uma pedra no meio do caminho
No meio do caminho tinha uma pedra

(ra105607) ssh2.students.ic.unicamp.br - Konsole <2>
File Edit View Bookmarks Settings Help
-bash-3.2$ netstat -tun
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 143.106.16.165:57043    173.194.118.110:443    ESTABLISHED
tcp        0      0 127.0.0.1:25           127.0.0.1:44001        TIME_WAIT
tcp        0      0 143.106.16.165:37523    143.106.16.163:10101    ESTABLISHED
tcp        0 256 143.106.16.165:22       201.82.117.6:57820      ESTABLISHED
tcp        0      0 143.106.16.165:22       189.20.19.66:47167      ESTABLISHED
tcp        0      0 143.106.16.165:22       201.82.117.6:57819      ESTABLISHED
tcp        0      0 143.106.16.165:60138    141.101.115.213:80      ESTABLISHED
tcp        0      0 143.106.16.165:22       201.82.117.6:57817      ESTABLISHED
tcp        0      0 143.106.16.165:39777    173.194.118.97:443      ESTABLISHED
tcp        0      0 143.106.16.165:52262    143.106.16.167:389      ESTABLISHED
tcp        0      0 143.106.16.165:834      143.106.16.147:2049     ESTABLISHED
tcp        0      0 143.106.16.165:53077    143.106.10.155:80       CLOSE_WAIT
tcp        0      0 143.106.16.165:53060    143.106.10.155:80       ESTABLISHED
-bash-3.2$ nslookup 143.106.16.163
Server:      143.106.16.144
Address:     143.106.16.144#53

163.16.106.143.in-addr.arpa    name = xaveco.lab.ic.unicamp.br.
-bash-3.2$
```

Figura 4: Usando telnet como cliente.

```
(ra105607) ssh.students.ic.unicamp.br - Konsole
File Edit View Bookmarks Settings Help
-bash-4.2$ ./server
E agora, Jose?
a festa acabou,
a luz apagou,

(ra105607) ssh2.students.ic.unicamp.br - Konsole
File Edit View Bookmarks Settings Help
-bash-3.2$ telnet xaveco.lab.ic.unicamp.br 10101
Trying 143.106.16.163...
Connected to xaveco.lab.ic.unicamp.br (143.106.16.163).
Escape character is '^['.
E agora, Jose?
a festa acabou,
a luz apagou,
```