

MC833AB - Programação de Redes de Computadores
2S 2015

Exercício 5: Implementação de Aplicação Cliente/Servidor com Sockets TCP
Professor Nelson Fonseca

Critérios para entrega do exercício

- Este exercício é em duplas.
- A resolução do exercício deve ser enviada em um arquivo .zip, .rar ou .tar.gz para carlos@ic.unicamp.br até o dia 12/10/2015. O nome do arquivo deve ser "raXXXXX-raXXXXX_ex5", onde XXXXX deve ser substituído pelo número do RA dos membros da dupla;
- O arquivo comprimido deverá conter:
 1. um único arquivo .pdf contendo o relatório comprovando a execução de todos os passos acima (Devem ser incluídas as saídas das execuções dos programas, devem ser comentados os nomes/endereços das máquinas utilizadas e devem ser respondidas as questões).
 2. Os códigos servidor.c e cliente.c modificados conforme solicitado nos passos acima e devidamente comentados.
 3. um único arquivo .pdf contendo os códigos do servidor.c e cliente.c (Podem ser utilizados os programas a2ps e ps2pdf para gerar o arquivo -- **Leia a manpage do a2ps!!! Por padrão a saída é enviada direto para a impressora!!!**).
- O assunto da mensagem deve ser "[MC833] Exercício 5".

Atividade

Passos:

1. Analise os códigos dos programas a seguir e procure nas páginas de manual do Linux, a descrição das funções que não conhecer. Procure também nos códigos a natureza dos parâmetros que cada programa deve receber, se for o caso. Explique o funcionamento das funções inet_pton, htons e htonl.
2. Compile e execute os programas [cliente.c](#) e [servidor.c](#) em uma mesma máquina. A função bind reportou algum erro? Em caso afirmativo, qual a sua causa? Se necessário, modifique os programas de forma que este erro seja corrigido e informe quais modificações foram realizadas.
3. Com as modificações feitas, re-execute os programas em uma mesma máquina e em máquinas diferentes. Quais as saídas dos programas?
4. Através de ferramentas existentes no sistema operacional, como você comprova, durante a execução em máquinas diferentes, que os códigos estão realizando uma comunicação via rede?
5. Adicione comentários ao código.
6. Modifique o programa cliente.c para que ele obtenha as informações do socket local (# IP, # porta local) através da função getsockname(). Modifique o programa servidor.c para que este obtenha as informações do socket remoto

do cliente (`# IP remoto`, `# porta remota`), utilizando a função `getpeername()`. Imprima esses valores na saída padrão.

7. Mantenha o binário do `servidor.c` executando em uma máquina A e execute três vezes seguidas o binário do `cliente.c` em uma máquina B. Observando a saída do comando `netstat`, qual das duas máquinas (A ou B) ficam no estado `TIME_WAIT`? Explique porque a outra máquina não fica nesse estado.
8. O programa `telnet` pode ser usado no lugar do binário do `client.c`? Por que? Caso o `telnet` possa ser usado, copie a saída do servidor e da execução do `telnet`. Cite uma modificação no `servidor.c` que impediria a utilização do `telnet`