MC833 - Tarefa 6

Erik de Godoy Perillo - RA: 135582

1 de junho de 2016

1. –

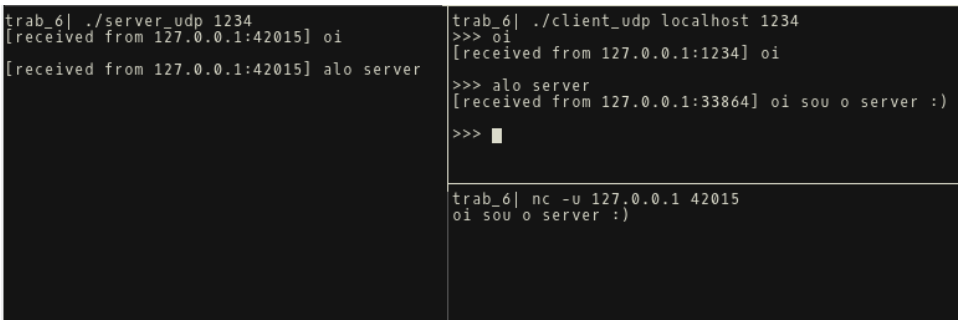
   - `ssize_t recvfrom(int sockfd, void* buf, size_t len, int flags, struct sockaddr* src_addr, socklen_t *addrlen)`
     Essa função pode ser usada para receber mensagens pela rede por protocolos sem conexão. Por `sockfd` especifica-se o *socket* do sistema operacional a se "ouvir" na espera de uma nova mensagem, que é guardada em `buf` e tem o tamanho máximo especificado por `len`. `flags` pode ser usado para especificar opções sobre o comportamento da função, como setá-la para *blocking/nonblocking*, por exemplo. Se não forem `NULL`, `src_addr` e `addrlen` são preenchidos com o endereço de quem enviou a mensagem e o tamanho da estrutura que representa o endereço, respectivamente. O valor retornado vai ser o número de *bytes* lidos em caso de sucesso ou -1 caso contrário.
   - `ssize_t sendto(int sockfd, const void* buf, size_t len, int flags, const struct sockaddr* dest_addr, socklen_t addrlen)`
     A contraparte de `recvfrom`. Pode ser usada para mandar mensagens pela rede por protocolos sem conexão. Por `sockfd` especifica-se o *socket* do sistema operacional a se usar para mandar a mensagem, que é especificada por `buf` e tem tamanho em *bytes* dado por `len`. Por `flags` pode-se configurar algumas partes do comportamento da função, como setar para *blocking/nonblocking*. O endereço destino da mensagem é especificado por `dest_addr` e seu tamanho é especificado em `addrlen`. Em sucesso, retorna o número de bytes enviados. Se falhar, retorna -1.

2. O servidor e cliente foram implementados. Uma demonstração é ilustrada abaixo, onde o cliente feito e o programa `nc` são usados ao mesmo

tempo para mandar mensagens ao servidor, que as manda de volta para os respectivos clientes:

3. A principal diferença entre o protocolo UDP e o TCP é que o primeiro não é conectado a conexão, enquanto o outro é. Assim, UDP é efêmero no sentido que só existe a troca da mensagem entre um *host* e outro, enquanto que no TCP uma conexão é mantida até que não se queira/possa mais trocar mensagens. O UDP usa o método conhecido como *best effort*, isto é: tenta-se enviar a mensagem ao destinatário, mas nada é garantido. Em TCP, há diversas medidas que asseguram que a mensagem foi enviada ao destinatário, como mensagens de confirmação etc. Em TCP também garante-se que os pacotes chegam em ordem, em UDP, não. Por esses motivos e outros, o TCP é naturalmente mais complexo que o UDP. Tende-se a usar o TCP em casos em que a) pode-se pagar pelo *overhead* e b) é importante que todas as mensagens cheguem e o façam em ordem. UDP é mais usado em casos onde não é terrível a perda de alguns pacotes e um custo computacional baixo é apreciado/necessário.

4. O servidor foi modificado para mostrar a porta de quem o envia as mensagens. Assim, por meio do uso do `nc`, pôde-se enviar uma mensagem para o cliente, a qual ele aceitou sem problema algum.



5. O cliente foi modificado para checar o endereço/porta de quem o man-

dou a mensagem. Se eles forem de algum modo diferentes dos especificados como sendo do servidor, a mensagem é ignorada e um aviso é mostrado, como ilustrado abaixo: