

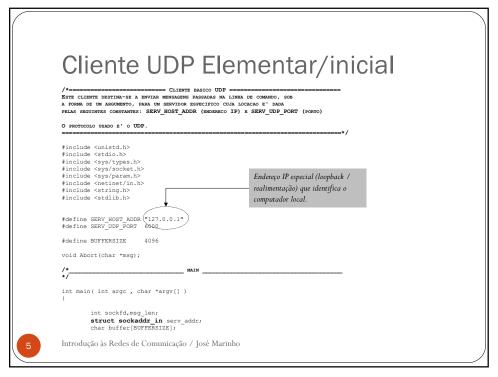
Servidor UDP Elementar/inicial

3

```
Servidor UDP Elementar/
                                                                                       poder ser preenchida com as coordenadas
do remetente do datagrama.
                       == PASSA A ATENDER CLIENTES INTERACTIVAMENTE =
         while (1) {
                 fprintf(stderr, "<SER1>Esperando datagram...\n");
                 length_addr = sizeof(cli_addr);
nbytes = recvfrom(sockfd , buffer , sizeof(buffer) , 0 , (struct sockaddr *(fcli_addr , flength_addr);
                         Abort("Erro na recepcao de datagrams");
                 buffer[nbytes]='\0'; /*Termina a cadeia de caracteres recebidos com '\0'*/

Buffer para onde é copiado o
                                                                                             conteúdo do datagrama recebido.
                 source_port = ntohs( cli_addr.sin_port); /*Network TO Host SHORT*/
strcpy( source_ip , (char *)inet_ntoa( cli_addr.sin_addr ) ); /*Network TO Ascir*/
                 printf( "\n<SER1>Mensagem recebida (%s) de (IP: %s; porto: %d)\n" , buffer, source_ip , source_port );
  * Abort

Mostra a mensagem de erro associada ao ultimo erro no SO e termina com "exit status" a 1 (EXIT_FAILURE)
void Abort (char *msg)
         exit(EXIT_FAILURE);
Introdução às Redes de Comunicação / José Marinho
```



5

Cliente UDP Elementar/inicial == TESTA A SINTAXE = if(argc != 2) { :- 2) (fprintf(stderr, "Sintaxe: %s frase_a_enviar\n",argv[0]); exit(EXIT_FAILURE); /*====== CRIA SOCKET PARA ENVIO/RECEPCAO DE DATAGRAMAS =======*/ sockfd = socket(PF_INET , SOCK_DGRAM , 0); socktd = suchet(--_ if(sockfd < 0) Abort("Impossibilidade de criar socket");</pre> ====== PREENCHE ENDERECO DO SERVIDOR ===== bzero((char*)&serv_addr , sizeof(serv_addr)); /*Coloca a zero Todos os bytes*/ serv_addr.sin_family = AF_INET; /*Address Family: Internet*/ serv_addr.sin_addr.s_addr = inet_addr(SERV_HOST_ADDR); /*IP NO FORMATO "DOTTED DECIMAL" => 32 BITS*/ serv_addr.sin_port = htons(SERV_UDP_PORT); /*Host TO Nerowork Short*/ ======= ENVIA MENSAGEM AO SERVIDOR ======= msg_len = strlen(argv[1]); $Como\ n\~{a}o\ foi\ associado\ qualquer\ porto\ local\ ao\ socket, o\ sistema\ operativo\ atribui\ um\ de\ forma\ autom\'atica.$ $\label{eq:matternative} \textit{Em alternative}, \textit{pode ser usada a rotina bind (ver servidor) e o porto zero (addr.sin_port = htons(0);)}.$ Introdução às Redes de Comunicação / José Marinho

Cliente UDP Elementar/inicial

7

Introdução às Redes de Comunicação / José Marinho

7

Compilação e Execução

• Compilar

```
gcc -o servidor servidorUDP_v2.c
gcc -o cliente clienteUDP.c
```

• Obter ajuda

```
man inet_ntoa
```

Executar

```
./servidor
./cliente Hello!
./cliente "Hello servidor!"
```

 Caso o servidor e os clientes não sejam lançados no mesmo computador, o valor da constante SERV_HOST_ADDR deve ser ajustado nos clientes e estes recompilados

8

Introdução às Redes de Comunicação / José Marinho

Aulas práticas

- Encontram-se disponíveis no Moodle, desde o primeiro dia de aulas, todos os elementos de estudo necessários, incluindo:
 - O plano das aulas práticas;
 - A descrição dos comandos elementares Unix;
 - A forma de compilar programas em C em ambientes Unix;
 - Uma introdução aos sockets BSD;
 - Os exemplos abordados nas aulas práticas.
- Para estudar programação é necessário programar e fazer experiências
- Ao longo das aulas práticas, os códigos fonte do cliente e do servidor UDP, apresentados nos acetatos anteriores, vão evoluir de modo a incluir e explorar funcionalidades adicionais



Introdução às Redes de Comunicação / José Marinho

9

Aulas práticas

- Estratégia a adoptar nas aulas práticas
 - O docente identifica a(s) funcionalidade(s) pretendida(s) e/ou o(s) problema(s) a resolver durante a aulas;
 - O docente descreve qualquer matéria nova que seja relevante no âmbito da aula;
 - Os alunos tentam atingir os objectivos propostos de uma forma autónoma;
 - No final da aula, o docente indica um trabalho que deverá ser realizado até à próxima aula.
- Evolução do Servidor UDP
 - Reenviar o datagrama recebido ao cliente
 - Reenviar, ao cliente, o tamanho da mensagem recebida
 - Em formato ascii (é reenviada uma string);
 - Em formato binário (é reenviado um inteiro).
 - Obter o porto de escuta pretendido através da linha de comando



Introdução às Redes de Comunicação / José Marinho

Aulas práticas

- Evolução do cliente UDP
 - Aguardar pela recepção da resposta e visualizá-la
 - Versão 1: mensagem original devolvida;
 - Versão 2: tamanho da mensagem em formato ascii;
 - Versão 3: tamanho da mensagem em formato binário.
 - Determinar o porto automático atribuído pelo sistema operativo
 - Verificar se a origem do datagrama recebido coincide com o servidor
 - Verificar se a resposta é a esperada
 - Obter o endereço IP e o porto do servidor através da linha de comando
 - Aguardar pela resposta apenas durante um determinado tempo máximo (timeout)
 - Com SIGALRM / alarm() / signal();
 - Com setsockopt() / errno == EAGAIN;
 - Com select().

11

Introdução às Redes de Comunicação / José Marinho