

Programação Avançada: Guia de Exame Prático

Comunicações UDP, Manipulação de Bits e Estruturas de Dados

1. PROTOCOLOS BINÁRIOS COM ESTRUTURAS

Atenção ao Layout: Para que o **recvfrom** mapeie os dados corretamente, a ordem na **struct** deve ser IDÊNTICA à do enunciado.

Definição de Estrutura Empacotada

Usa **__attribute__((packed))** para evitar que o compilador adicione "espaços" (padding) entre os campos.

```
typedef struct __attribute__((packed)) {  
    uint8_t operacao;        // 1 byte - Byte 0  
    uint32_t n_estudante;    // 4 bytes - Bytes 1 a 4  
    uint32_t telefone;       // 4 bytes - Bytes 5 a 8  
} pedido_t;
```

Receber Estrutura (Servidor)

```
pedido_t pedido;  
// Recebe diretamente para a struct  
udp_read_bytes = recvfrom(socket, &  
  
// Converter Endianness de campos >
```

Enviar Estrutura (Cliente)

```
pedido_t msg;  
msg.operacao = 1;  
// Converte para Network Order antes  
msg.n_estudante = htonl(2222222);  
msg.telefone = htonl(912345678);
```

```
uint32_t aluno = ntohl(pedido.n_est  
uint32_t tel = ntohl(pedido.telefon
```

```
sendto(socket, &msg, sizeof(msg), 0
```

2. TIPOS DE DADOS E ENDIANNESS

uint8_t (1 Byte)

Uso: Códigos de operação (0-255), Flags.

Conversão: Nenhuma necessária.

uint32_t (4 Bytes)

Uso: Números de Estudante, IPs, Telefones.

Conversão: **htonl()** / **ntohl()**.

3. MANIPULAÇÃO DE BITS (BITWISE)

Operações e Visualização

```
// INVERTER o bit (FLIP)  
dado ^= (1 << pos);  
  
// Função para imprimir Binário  
void print_binary(uint8_t valor) {  
    for (int j = 7; j >= 0; j--) {  
        int bit = (valor >> j) & 1;  
        printf("%d", bit);  
    }  
    printf("\n");  
}
```

4. VALIDAÇÕES E ERROS COMUNS

Lógica de Fluxo: Se houver erro nos dados, envia a resposta de erro e usa **continue** para saltar o resto do ciclo **while(1)**.

Checklist de Validação

- **Tamanho:** `if (udp_read_bytes != sizeof(pedido_t))`.
- **Gama:** Estudante entre [2.000.000, 3.000.000[.
- **Telefone:** Começar por 9 ($\geq 900.000.000$ e $< 1.000.000.000$).
- **Array:** Não exceder 200 contactos totais.

5. TABELAS AUXILIARES

Formatação de Strings

%x	Hex minúsculo (a)
%X	Hex maiúsculo (A)
%02x	Hex de 2 dígitos (0a)

sscanf / sprintf

```
// Extrair de string "10,20"
sscanf(str, "%d,%d", &a, &b);

// Criar string formatada
sprintf(res, "Soma: %d", a+b);
```