Programação Avançada: Sebenta de Exame

Comunicações UDP, Manipulação de Bits, Strings e Memória

1. Comunicações UDP e Estruturas

> **Regra de Ouro:** Sempre que enviares ou receberes estruturas via rede, usa `__attribute__((packed))` para evitar falhas de alinhamento de memória.

Definição da Mensagem

```
typedef struct __attribute__((__packed__)) {
    uint8_t  op;           // 1 byte
    uint32_t id_estudante; // 4 bytes (necessita htonl/ntohl)
    uint16_t porto;        // 2 bytes (necessita htons/ntohs)
} msg_t;
```

Enviar (Cliente)

```
msg_t m;
m.op = 1;
m.id_estudante = htonl(2190000);
m.porto = htons(9999);

sendto(sock, &m, sizeof(m), 0,
      (struct sockaddr*)&serv_addr,
       sizeof(serv_addr));
```

Receber (Servidor)

```
msg_t m;
struct sockaddr_in cli;
socklen_t len = sizeof(cli);

recvfrom(sock, &m, sizeof(m), 0,
         (struct sockaddr*)&cli, &len);

uint32_t id = ntohl(m.id_estudante);
```

2. Manipulação de Bits (Bitwise)

Essencial para verificar estados de sensores ou flags de configuração.

Operações Rápidas (Bit na posição `n`)

Operação	Código C	Explicação
Verificar	`(valor & (1 << n))`	Resultado é > 0 se o bit estiver a 1.
Ativar (Set)	`valor |= (1 << n)`	Força o bit `n` a ser 1.
Desativar (Clear)	`valor &= ~(1 << n)`	Força o bit `n` a ser 0.
Inverter (Flip)	`valor ^= (1 << n)`	Troca 0 por 1 e vice-versa.

Visualização Binária (Impressão)

```
void print_bits(uint8_t v) {
    for(int i = 7; i >= 0; i--) {
        printf("%d", (v >> i) & 1);
    }
    printf("\n");
}
```

3. Manipulação de Strings e Comandos

sscanf - Extrair dados

Lê de uma string em vez do teclado.

```
char input[] = "IP:192.168.1.1,PORT:8080"
char ip[20];
int port;
sscanf(input, "IP:%[^,],PORT:%d", ip, &p
```

sprintf - Montar Strings

Útil para criar comandos para a função `system()`.

```
char cmd[256];
char cor[] = "#FF5733";
sprintf(cmd, "convert -size 100x100 xc:%
system(cmd); // Executa no terminal
```

strncat e strncpy - Segurança

Sempre preferir versões com `n` para evitar Buffer Overflow.

```
char dest[50] = "Olá ";
char src[] = "Mundo da Programação";
strncat(dest, src, sizeof(dest) - strlen(dest) - 1);
// Garante que não escreve fora do array
```

4. Validações de Rede e Erros

Validação de IP (Presentation to Network)

```
#include <arpa/inet.h>
struct in_addr addr;
if (inet_pton(AF_INET, args.ip_arg, &addr) != 1) {
    // IP Inválido!
}
```

Lógica de Servidor UDP: Ao validar pacotes, se um dado estiver fora de gama (ex: nota > 20), deves enviar uma resposta de erro ao cliente e usar o comando `continue;` para voltar ao início do `while(1)` sem processar o resto.

5. Tabela de Formatação (printf)

Especificador	Tipo	Exemplo / Uso
`%02x`	Hexadecimal	`0a` (Sempre 2 dígitos, minúsculo). Ideal para cores.
`%X`	Hexadecimal	`1A2B` (Maiúsculas). Ideal para debug de endereços.
`%u`	Unsigned Int	`uint32_t` (Sempre positivo).
`%zu`	size_t	Usado para o retorno de `sizeof()` ou `strlen()`.

Guia Prático ESTG - Programação Avançada. Boa sorte no exame!