

# Roteiro de Estudos: Endereçamento IP e Subnetting em C

---

## Módulo 1: Sistemas Numéricos e Conversão Base

1. **Objetivo do Módulo:** Dominar a representação interna e a conversão de valores entre as bases **Decimal** e **Binária**, simulando um octeto (8 bits) ou um endereço IPv4 (32 bits).

### Exercício Prático (Progressivo):

- **Foco:** Funções de conversão Decimal  $\leftrightarrow$  Binário e Binário  $\leftrightarrow$  Decimal para um `unsigned char` (octeto).
- **Tarefa:** Crie um pequeno programa que defina uma função, digamos `void dec_para_bin(unsigned char octeto)`, que imprima o valor binário (exatamente 8 bits) desse octeto. Crie outra função, `unsigned char bin_para_dec(const char* bin_str)`, que faça o inverso.

### Provocação de Raciocínio (Crucial):

- Quando você converte Decimal  $\leftrightarrow$  Binário, você usa divisões ou operações bitwise? Qual é mais eficiente para um computador e por quê?
- Onde você deve colocar os zeros à esquerda para garantir que a saída binária tenha sempre 8 bits? O que acontece se você simplesmente usar `%d` ou `%x` do `printf` para imprimir um valor inteiro em C?

3. **Ponto de Consolidação:** Se você consegue converter o valor **203** (decimal) para **11001011** (binário) e vice-versa, e entende que o computador armazena isso como **bits** em um tipo inteiro, você dominou o básico.

---

## Módulo 2: Manipulação de Bits e a Lógica da Máscara

1. **Objetivo do Módulo:** Implementar a lógica da Máscara de Sub-rede através da operação **AND bitwise** (`&`) para isolar a porção de **Rede** de um endereço IP.

### Exercício Prático (Progressivo):

- **Foco:** O Operador **AND bitwise** (&) e o conceito de **Máscara**.

**Tarefa:** Crie um programa que faça o seguinte:

1. Defina dois `unsigned int`: `endereco_ip` (ex: \$192.168.10.15\$) e `mask_subrede` (ex: \$255.255.255.0\$). (Dica: Pense em como representar esses 4 octetos em um único `unsigned int`).
2. Crie uma função, `unsigned int calcular_endereco_rede(unsigned int ip, unsigned int mask)`, que use a operação AND para retornar o endereço da Rede.
3. Imprima o IP, a Máscara e o Endereço de Rede resultante no formato **Decimal Pontuado** (ex: \$192.168.10.0\$).

### Provocação de Raciocínio (Crucial):

- Como você consegue empacotar os quatro octetos (4 bytes) do IP em um único `unsigned int` (4 bytes)? Quais operadores bitwise (<<, |) são essenciais para isso?
- Se a Máscara é \$255.255.255.0\$, o que acontece com a porção Host quando o AND é aplicado? Por que essa porção é **zerada** e o que isso significa conceitualmente em termos de rede?

3. **Ponto de Consolidação:** Se você consegue isolar corretamente a porção **Rede** (onde os bits da máscara são 1) da porção **Host** (onde os bits da máscara são 0), você está pronto para o Subnetting.

---

## Módulo 3: Notação CIDR e Cálculo do Endereço de Broadcast

1. **Objetivo do Módulo:** Entender a **Notação CIDR** (/n) e usá-la para calcular a Máscara de Sub-rede e, em seguida, o **Endereço de Broadcast**.

### Exercício Prático (Progressivo):

- **Foco:** Geração de Máscara a partir do prefixo /n e o Operador **OR bitwise** (|).

**Tarefa:**

1. Crie uma função `unsigned int prefixo_para_mask(int prefixo_cidr)` que receba um valor `n` (ex: 24) e retorne o `unsigned int` representando a máscara (ex: \$255.255.255.0\$). (Dica: ~0 é um bom ponto de partida).

2. Crie uma função para calcular o **Endereço de Broadcast**. Lembre-se que o Broadcast é o endereço da Rede com todos os bits da porção **Host ligados** (setados para 1).
3. Calcule o **Wildcard Mask** (o inverso da Máscara de Sub-rede) e utilize-o para calcular o Broadcast.

### Provocação de Raciocínio (Crucial):

- O que a operação **NOT bitwise** (`~`) faz com a máscara gerada a partir do prefixo CIDR? Como você pode usar isso para criar um "coringa" (Wildcard Mask) que te dá *apenas* os bits da porção Host?
- Por que a operação **OR bitwise** (`|`) do Endereço de Rede com o Wildcard Mask resulta no Endereço de Broadcast? Qual a lógica por trás de usar o OR com os bits da porção Host?

3. **Ponto de Consolidação:** Se para o IP `$10.1.1.100$` com prefixo `$/16$` você consegue calcular o Endereço de Rede (`$10.1.0.0$`) e o Endereço de Broadcast (`$10.1.255.255$`), você domina o conceito central.

---

## Módulo 4: Subnetting - Cálculo de Hosts Válidos e Casos Especiais

1. **Objetivo do Módulo:** Entender como a mudança no prefixo CIDR afeta o número de Hosts Válidos e focar em sub-redes pequenas (como `$/30$`).

### Exercício Prático (Progressivo):

- **Foco:** Cálculo  $2^n$  e manipulação de faixas.

### Tarefa:

1. Crie uma função `int calcular_hosts_validos(int prefixo_cidr)` que retorne o número de hosts utilizáveis (total de endereços **menos 2** - Rede e Broadcast).
2. Simule o cálculo de uma sub-rede `$/30$`. Receba um endereço IP de rede (ex: `$10.0.0.0$`) e um prefixo `$/30$`. Imprima: o Endereço da Rede, o Primeiro Host Válido, o Último Host Válido e o Endereço de Broadcast.

### Provocação de Raciocínio (Crucial):

- Na função `calcular_hosts_validos`, por que é crucial subtrair 2? O que você faz se o prefixo for  $2^{32}$  ou  $2^{31}$ ? O que a RFC 3021 permite para  $2^{31}$  e qual é a implicação prática disso (dica: *links ponto-a-ponto*).
- Se você tem o endereço de Rede e o Broadcast, como você obtém o **Primeiro Host Válido** e o **Último Host Válido** usando apenas a aritmética de ponteiros e o incremento (`++`) e decremento (`--`) no `unsigned int` que armazena o endereço?

3. **Ponto de Consolidação:** Se você entende que o prefixo  $2^{30}$  é o mínimo para ter dois hosts utilizáveis em links ponto-a-ponto tradicionais e consegue calcular essa faixa perfeitamente, o domínio sobre o Subnetting está consolidado.

---

## Módulo 5: O Salto Lógico: Roteamento e Interfaces

---

1. **Objetivo do Módulo:** Entender como a lógica do Subnetting (Módulos 1-4) é aplicada para decidir se um pacote deve ser entregue diretamente na rede local ou se deve ser enviado ao **Roteador (Default Gateway)**.

### Exercício Prático (Progressivo):

- **Foco:** Simulação da Tabela de Rotas e o papel do Gateway.

**Tarefa:** Crie uma função, `int decidir_rota(unsigned int meu_ip, unsigned int meu_mask, unsigned int destino_ip, unsigned int gateway_ip)`. Esta função deve simular o processo de roteamento em um host:

1. Calcular o Endereço de Rede (`Rede_A`) do *seu* IP usando a sua máscara.
2. Calcular o Endereço de Rede (`Rede_B`) do *IP de destino* usando a sua máscara.
3. Comparar `Rede_A` com `Rede_B`.
4. Se forem iguais, retornar um código indicando "Entrega Direta". Se forem diferentes, retornar um código indicando "Enviar para Gateway".

### Provocação de Raciocínio (Crucial):

- Por que o Roteador precisa de **duas** ou mais interfaces, cada uma com seu próprio IP e Máscara de Sub-rede? Qual é a implicação prática de um pacote destinado a uma rede remota (`Rede_A != Rede_B`) ser roteado, em vez de entregue diretamente?
- Em um host, o que acontece se o **Default Gateway** for inacessível ou se pertencer a uma sub-rede diferente da interface que está tentando usá-lo?

3. **Ponto de Consolidação:** Se você consegue explicar por que um pacote para `$192.168.1.10$` deve ir direto, mas um para `$8.8.8.8$` deve ir para o Gateway `$192.168.1.1$` (dado o IP/Máscara `$192.168.1.100/24$`), você compreendeu o papel do roteamento em nível de host.

---

## Módulo 6: Diagnóstico de Rede: Interpretação de Logs e Saída

---

1. **Objetivo do Módulo:** Desenvolver a capacidade de diagnóstico, conectando a lógica de Subnetting (os cálculos) com a saída de ferramentas de rede padrão no ambiente Unix/Linux.

### Exercício Prático (Progressivo):

- **Foco:** Entender as ferramentas de Diagnóstico (`ping`, `ip addr`, `ip route`) e a interpretação de mensagens de erro.

**Tarefa:** Crie um *guia de cenários* em formato de texto (não em código C, mas como anotações em seu projeto) que correlacione a saída de comandos de diagnóstico com os erros de Subnetting que você implementou:

1. **Cenário A: Máscara Incorreta.** O que a saída do comando `ip addr show` revelaria se sua máscara estivesse errada (ex: `/23` em vez de `/24`)?
2. **Cenário B: Falha de Gateway.** Qual mensagem de erro (`ping` ou `ip route get`) você veria se o **Gateway Padrão** estivesse configurado, mas inalcançável?
3. **Cenário C: Network Unreachable.** O que essa mensagem de erro significa em termos de Tabela de Rotas? Onde você deveria buscar o problema (no seu host ou no roteador)?

### Provocação de Raciocínio (Crucial):

- Ao receber a mensagem "Destination Host Unreachable" após um `ping`, como você usaria a lógica AND bitwise (Módulo 2) para determinar se a falha ocorreu **dentro** da sua sub-rede ou se a falha veio **após** o seu Gateway?
- Em um log de sistema (hipotético), você vê um erro de *ARP Request Timeout*. Qual falha de configuração de Endereçamento IP poderia causar uma falha na resolução de endereço de camada 2 (MAC)?

3. **Ponto de Consolidação:** Se você consegue olhar para a saída de `ping: Network is unreachable` e dizer imediatamente que o problema é a **ausência ou incorreção de uma rota na tabela**, e não apenas um host desligado, você tem a mentalidade de depuração.