

Manutenção de computadores II

Prof. Moisés Souto

Moisés Souto
professor.moissouto.com.br
moises.souto@ifrn.edu.br

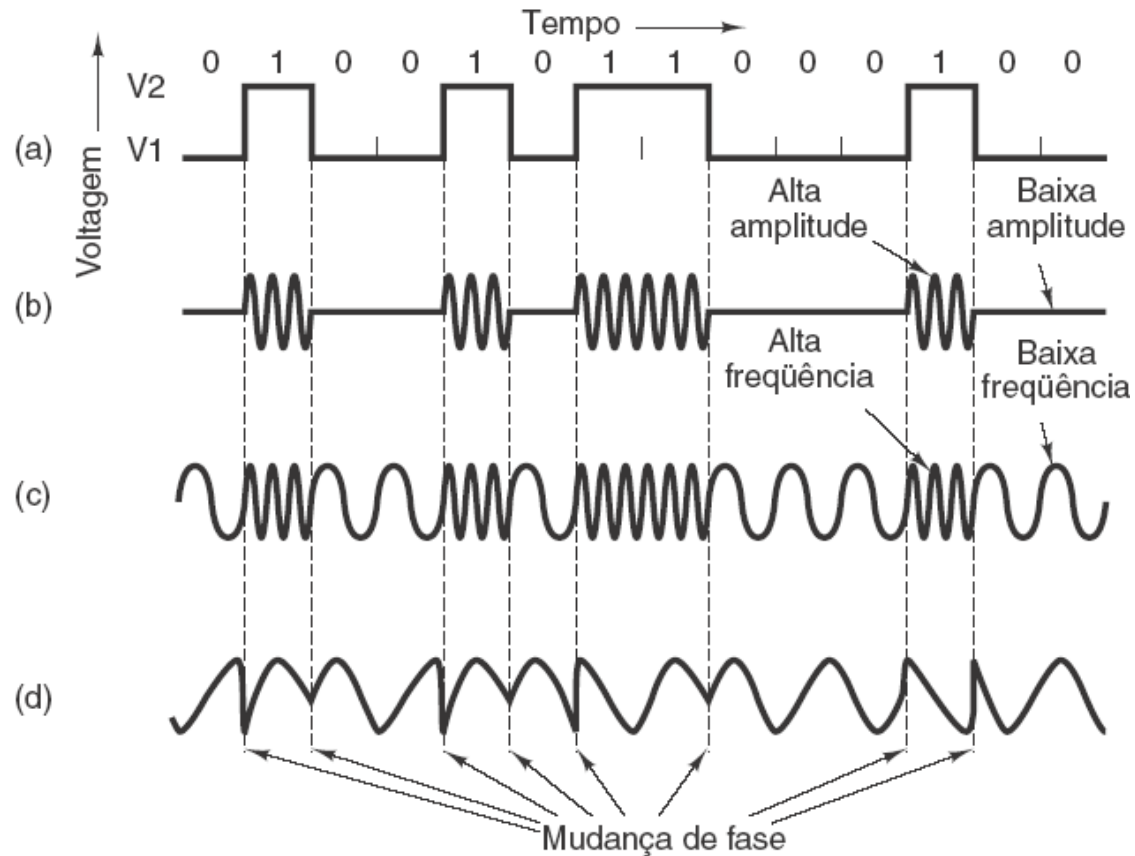
Aula 04

Revisão OC I

Comunicações



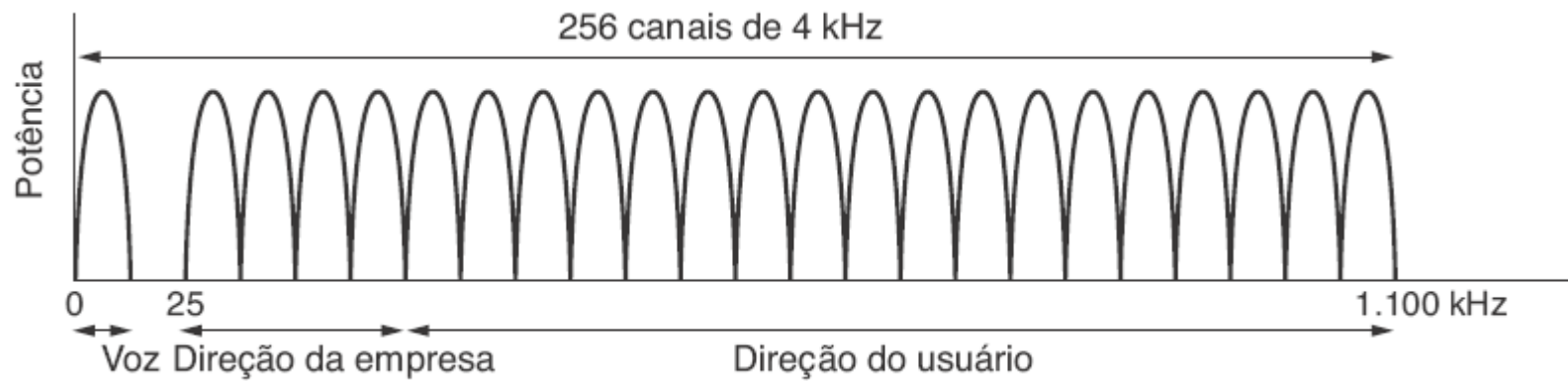
Telecomunicações



Transmissão, bit a bit, do número binário 01001010000100 por uma linha telefônica. (a) Sinal de dois níveis. (b) Modulação de amplitude. (c) Modulação de frequência. (d) Modulação de fase.



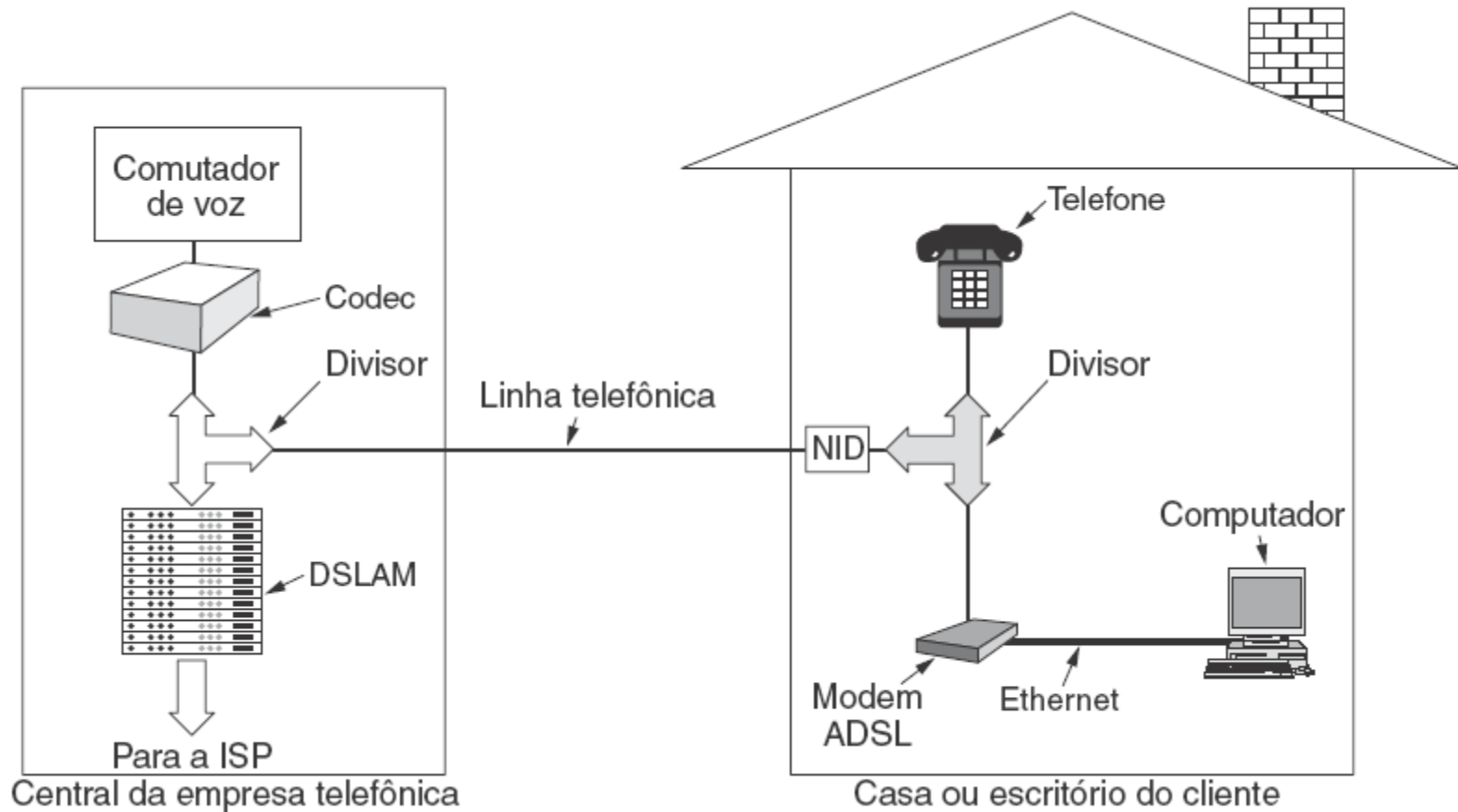
Linhas Digitais de Assinante (1)



Operação do ADSL.



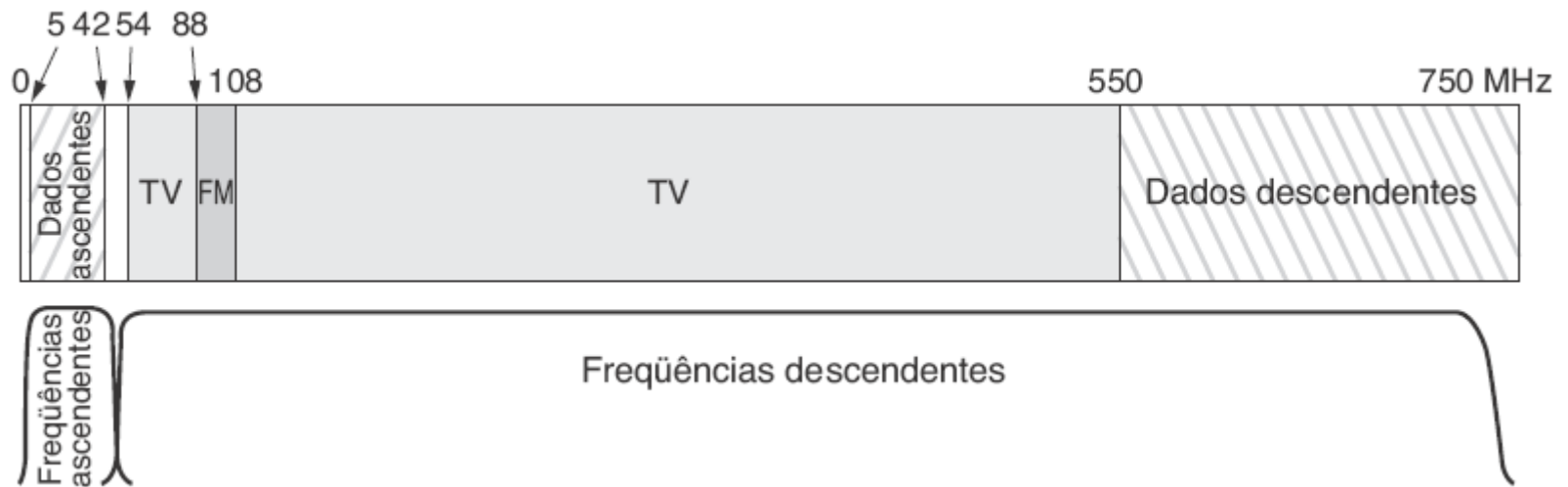
Linhas Digitais de Assinante (2)



Configuração típica de equipamento ADSL.



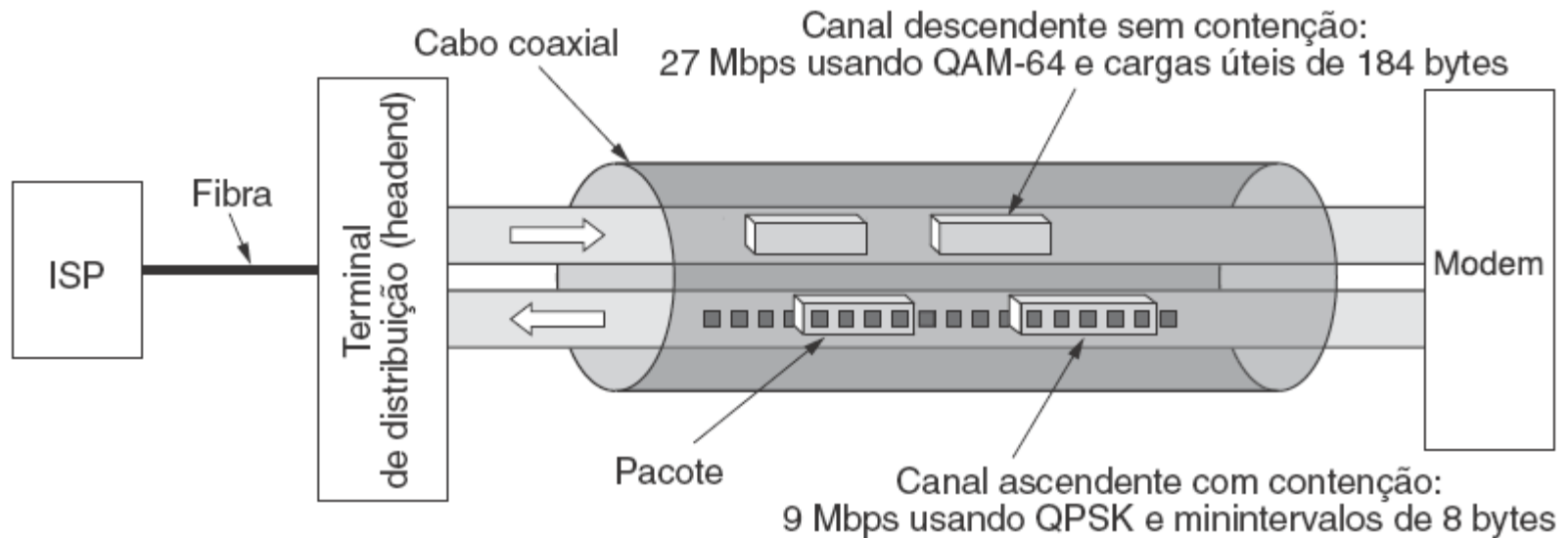
Internet por Cabo (1)



Alocação de frequência em um sistema de TV a cabo usado para acesso à Internet.



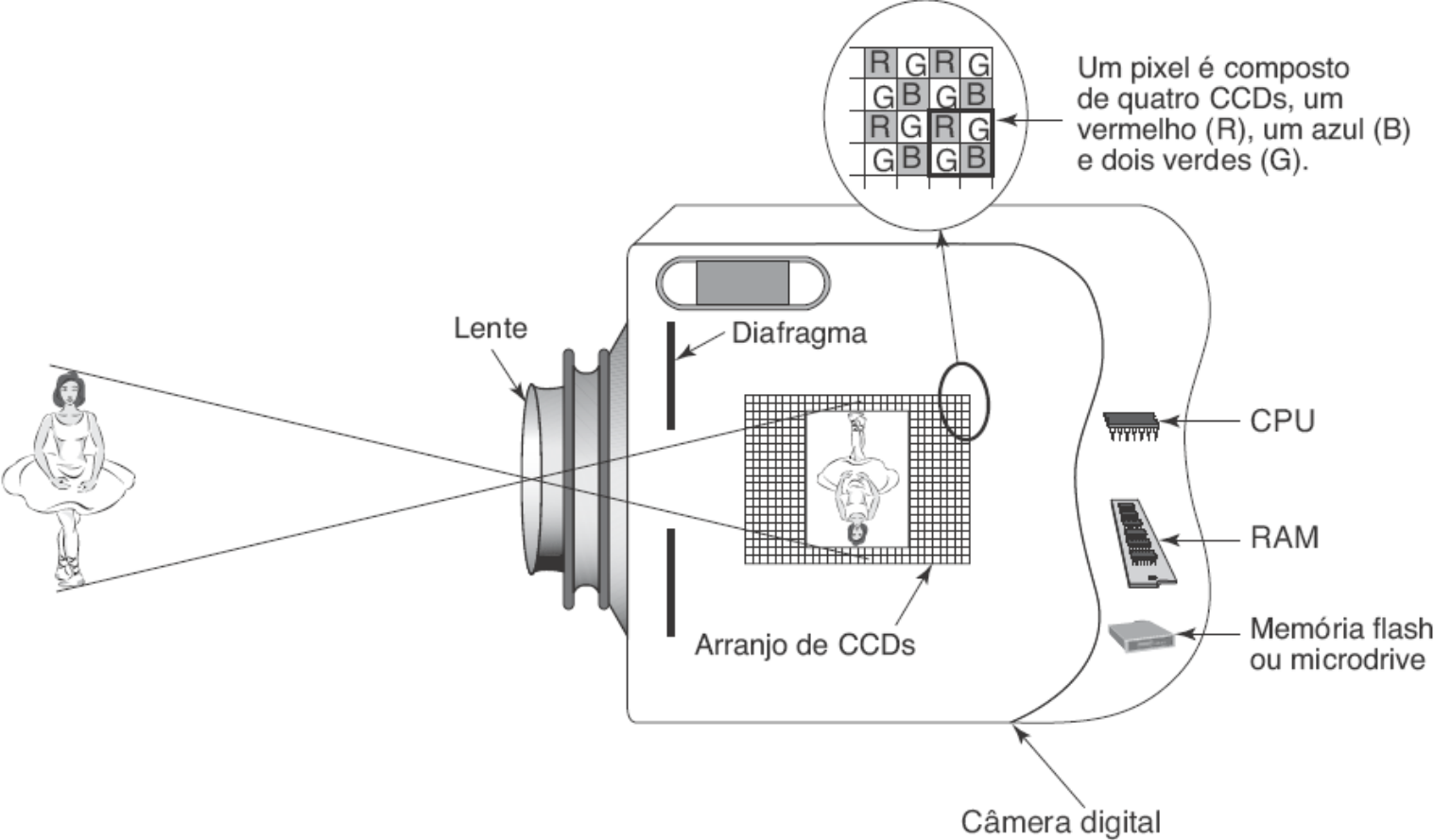
Internet por Cabo (2)



Detalhes típicos dos canais ascendentes e descendentes na América do Norte. QAM-64 (Modulação de amplitude em quadratura) permite 6 bits/Hz, mas funciona somente em altas frequências. QPSK (Modulação por chaveamento de fase em quadratura) funciona em baixas frequências, mas permite apenas 2 bits/Hz.



Câmeras Digitais



2.4 Suporte de hardware para sistemas operacionais

- **As arquiteturas de computador contêm:**
 - Recursos que executam funções de sistemas operacionais rapidamente em hardware para melhorar o desempenho.
 - Recursos que habilitam o sistema operacional a impor rígida proteção.



Integração com SO



2.4.1 Processador

- **O processador implementa mecanismos de proteção do sistema operacional.**
 - Evita que os processos acessem instruções privilegiadas ou memória.
 - Os sistemas de computador geralmente dispõem de diferentes modos de execução:
 - Modo usuário (estado usuário ou estado-problema)
 - O usuário pode executar apenas um subconjunto de instruções.
 - Modo núcleo (estado supervisor)
 - O processador pode acessar instruções privilegiadas e recursos em nome dos processos.



2.4.1 Processador

■ **Proteção e gerenciamento da memória**

- Impede que processos acessem memória que não lhes foi designada.
- É implementada por meio de registradores de processador que somente podem ser modificados com instruções privilegiadas.

■ **Interrupções e exceções**

- Quando ocorre um evento, a maioria dos dispositivos envia ao processador um sinal denominado interrupção.
- As exceções são interrupções geradas em resposta a erros.
- O sistema operacional pode responder a uma interrupção notificando os processos que estão à espera desses eventos.



2.4.2 Temporizadores e relógios

■ Temporizadores

- O temporizador de intervalo gera periodicamente uma interrupção.
- Os sistemas operacionais usam temporizadores de intervalo para impedir que processos monopolizem o processador.

■ Relógios

- Oferecem uma medida de continuidade.
- Um relógio de 24 horas habilita o sistema operacional a determinar a hora e a data atuais.



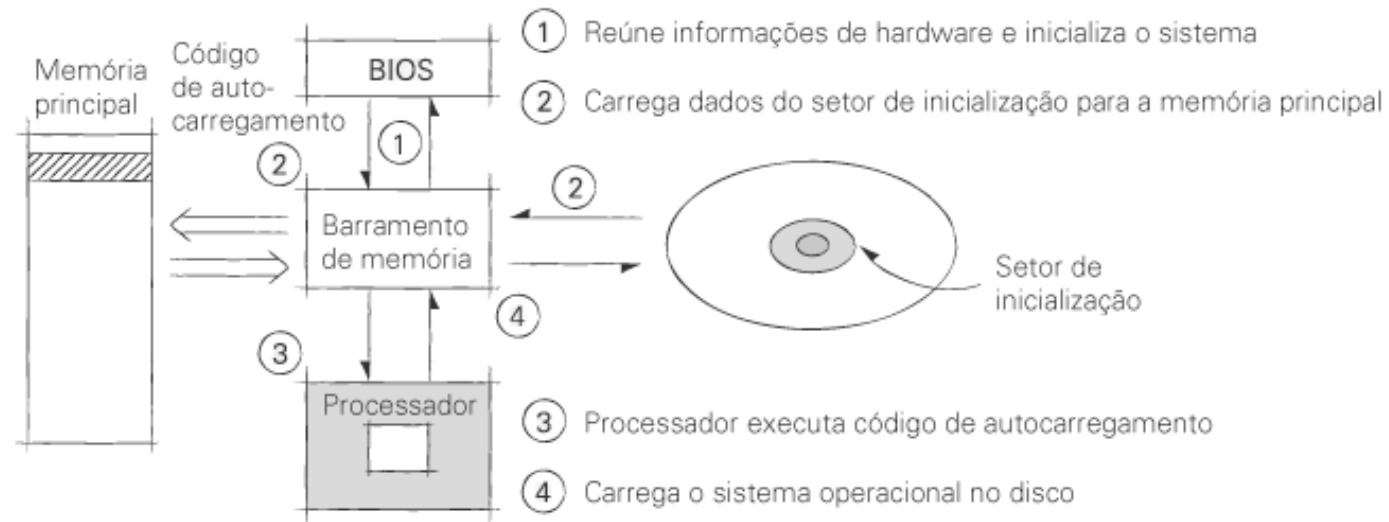
2.4.3 Autocarregamento (Bootstrapping)

- **Autocarregamento: carregamento inicial dos componentes do sistema operacional na memória**
 - É executado pelo BIOS (Basic Input/Output System) do computador.
 - Inicializa o hardware do sistema.
 - Carrega instruções na memória principal provenientes de uma área do armazenamento secundário denominada setor de inicialização (*boot sector*).
 - Se o sistema não for carregado, o usuário não poderá acessar nenhum hardware do computador.



2.4.3 Autocarregamento (Bootstrapping)

Figura 2.6 Autocarregamento.



2.4.4 Plug and Play

- **Tecnologia Plug-and-Play**
 - Permite que os sistemas operacionais configurem e usem um hardware recém-instalado, sem precisar interagir com o usuário.
 - Para suportar a tecnologia plug-and-play, um dispositivo de hardware deve:
 - Identificar-se exclusivamente no sistema operacional.
 - Comunicar-se com o sistema operacional para indicar os recursos e serviços que requer para funcionar adequadamente.
 - Identificar o driver correspondente que permite que o dispositivo seja configurado (por exemplo, que se atribua o dispositivo a um canal DMA).



2.5 Caching e buffer

■ **Caches**

- São memórias relativamente rápidas.
- Mantêm uma cópia dos dados que serão acessados logo em seguida.
- Aumentam a velocidade de execução do programa.
- Exemplos incluem:
 - Caches L1 e L2 do processador.
 - A memória principal pode ser considerada um cache para unidades de disco rígido e outros dispositivos de armazenamento secundário.



2.5 Caching e buffer

■ **Buffers**

- Área de armazenamento temporário que guarda dados durante transferências de E/S.
- São usados principalmente para:
 - Coordenar comunicações entre dispositivos que funcionam em diferentes velocidades.
 - Armazenar dados para processamento assíncrono.
 - Permitir que alguns sinais sejam emitidos assincronamente.

■ **Spooling**

- Técnica de buffer por meio da qual um dispositivo intermediário, como um disco, é interposto entre um processo e um dispositivo de E/S de baixa velocidade.
- Permite que os processos solicitem operações a um dispositivo periférico sem que esse dispositivo esteja preparado para atender a essa solicitação.



2.6 Visão geral do software

- **Linguagens de programação**
 - Algumas são compreendidas diretamente pelos computadores, outras exigem tradução.
 - Em geral, são classificadas como:
 - Linguagem de máquina
 - Linguagem de montagem ou
 - Linguagem de alto nível



Software



2.6.1 Linguagem de máquina e linguagem de montagem

■ Linguagem de máquina

- É definida pelo projeto de hardware do computador.
- Em geral consiste em cadeias de números (reduzidos a 1s e 0s) que instruem os computadores a executar suas operações mais elementares.
- Um computador só entende sua própria linguagem de máquina.

■ Linguagem de montagem

- Representa instruções em linguagem de máquina por meio de abreviaturas da língua inglesa.
- Os montadores convertem a linguagem de montagem em linguagem de máquina.
- Agiliza a programação e reduz a possibilidade de erros.



2.6.2 Interpretadores e compiladores

- **Linguagens de alto nível**
 - As instruções se assemelham ao inglês do dia-a-dia.
 - Executam tarefas mais substanciais com uma quantidade menor de comandos.
 - Requerem compiladores e interpretadores.
- **Compilador**
 - Programa de tradução que converte programas em linguagem de alto nível em linguagem de máquina.
- **Interpretador**
 - Programa que executa diretamente o código-fonte ou um código reduzido a uma linguagem de baixo nível que não é o código de máquina.



2.6.3 Linguagens de alto nível

■ Linguagens de alto nível mais conhecidas

- Em geral são processuais e orientadas a objeto.
- Fortran
 - Usada para aplicações científicas e de engenharia.
- COBOL
 - Usada para aplicações de negócios que processam grande volume de dados.
- C
 - Linguagem de desenvolvimento do sistema operacional UNIX.
- C++/Java
 - Linguagens populares orientadas a objeto.
- C#
 - Linguagem de desenvolvimento orientada a objeto para a plataforma .NET.



2.8 Compilação, ligação e carregamento

- **Para que um programa em linguagem de programação de alto nível possa ser executado, é necessário:**
 - Traduzi-lo para linguagem de máquina.
 - Ligá-lo a vários outros programas em linguagem de máquina dos quais ele dependa.
 - Carregá-lo na memória.



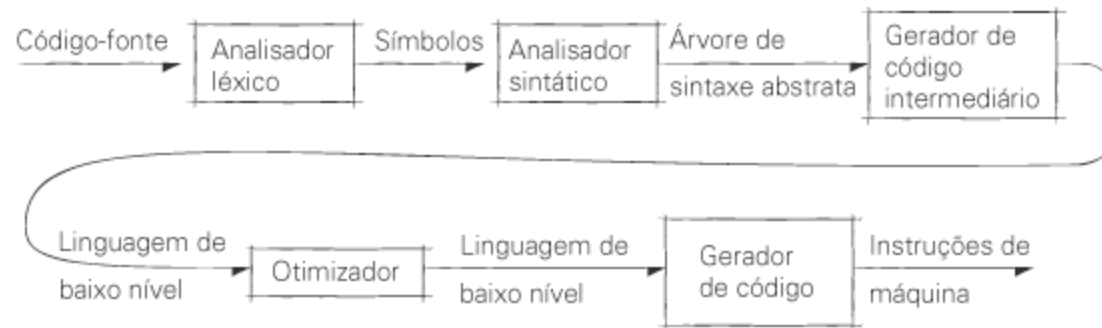
2.8.1 Compilação

- **Tradução de código de alto nível para código de máquina**
 - Aceita o código-fonte como entrada e retorna um código-objeto.
 - As fases da compilação incluem:
 - Analisador de léxico (lexer)
 - Separa os caracteres do fonte de um programa em símbolos (*tokens*).
 - Analisador sintático (parser)
 - Agrupa os símbolos em comandos sintaticamente corretos.
 - Gerador de código intermediário
 - Converte a estrutura sintática em uma cadeia de instruções simples.
 - Optimizador
 - Melhora a eficiência de execução do código e reduz os requisitos de memória do programa.
 - Gerador de código
 - Produz o arquivo-objeto contendo as instruções em linguagem de máquina.



2.8.1 Compilação

Figura 2.8 Fases da compilação.



2.8.2 Ligação

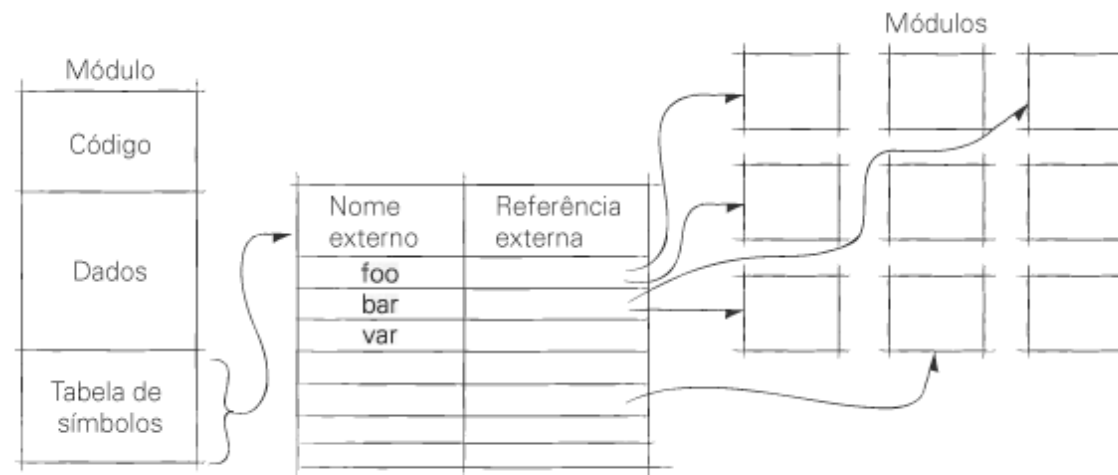
■ **Ligadores**

- Criam uma única unidade executável.
- Integram módulos previamente compilados, denominados bibliotecas, referidos por um programa.
- Atribuem endereços relativos a diferentes programas ou unidades de dados.
- Resolvem todas as referências externas entre subprogramas.
- Produzem um módulo integrado denominado módulo de carga.
- A ligação pode ser realizada em tempo de compilação, antes do carregamento, em tempo de carregamento ou em tempo de execução.



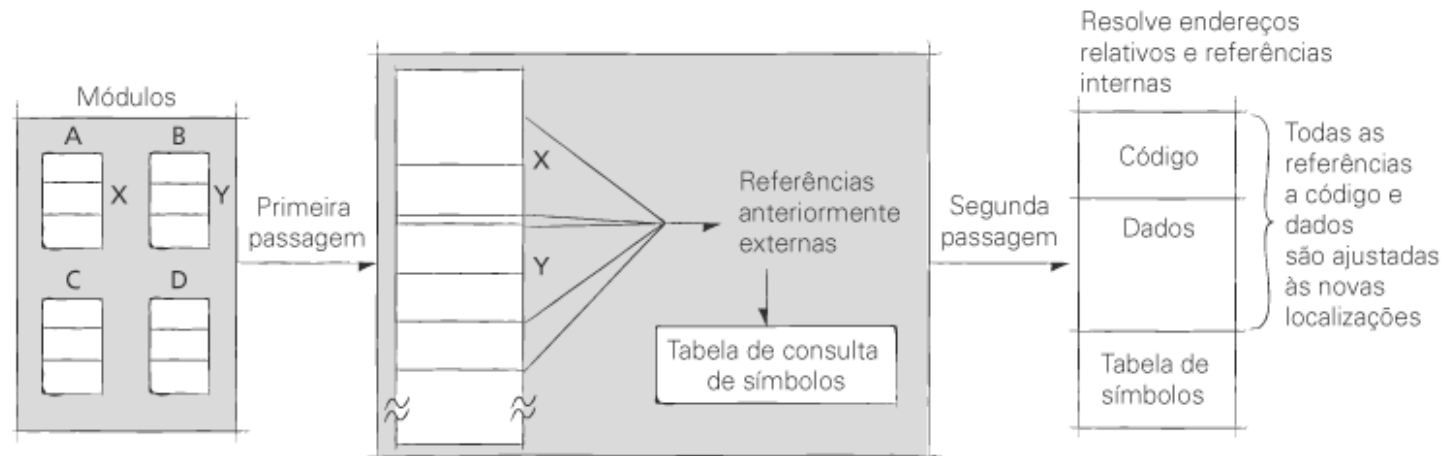
2.8.2 Ligação

Figura 2.9 Módulo-objeto.



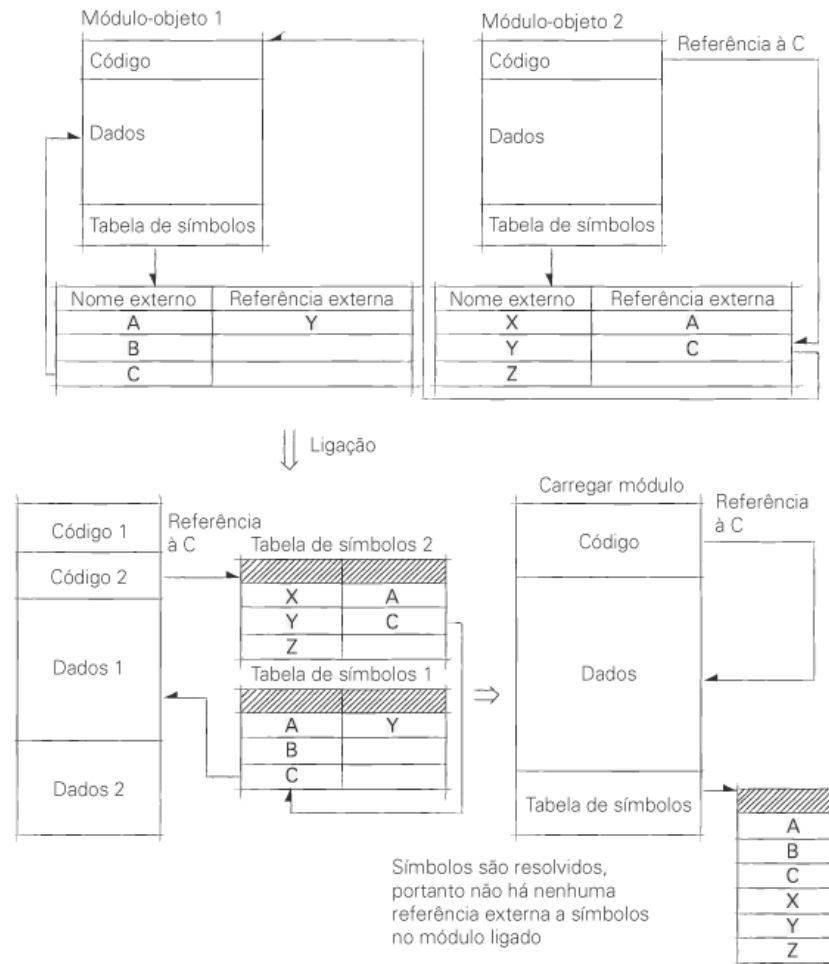
2.8.2 Ligação

Figura 2.10 Processo de ligação.



2.8.2 Ligação

Figura 2.11 Resolução de símbolos.



2.8.3 Carregamento

■ Carregadores

- Convertem endereços relativos em endereços físicos.
- Colocam cada instrução e dado na memória principal.

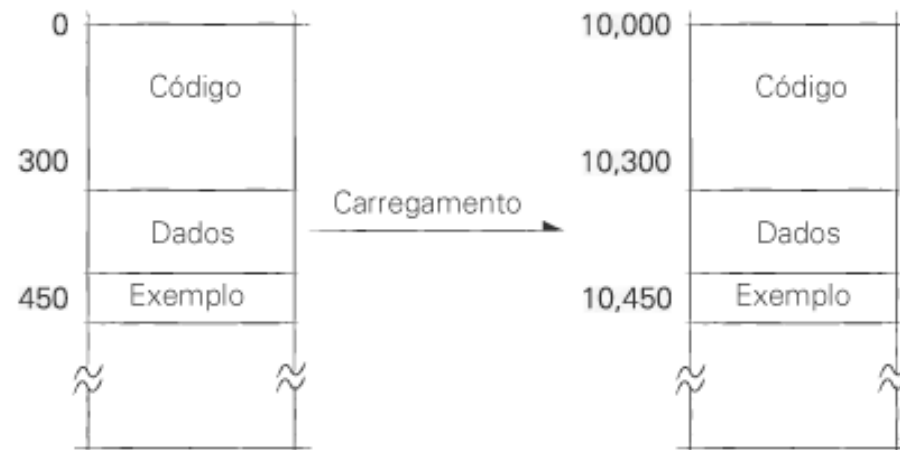
■ Técnicas de carregamento de um programa na memória

- Carregamento absoluto
 - Coloca o programa nos endereços especificados pelo programador ou compilador (supondo que esses endereços estejam disponíveis).
- Carregamento realocável
 - Realoca os endereços do programa de modo que correspondam à sua localização real na memória.
- Carregamento dinâmico
 - Carrega módulos de programa na primeira utilização.



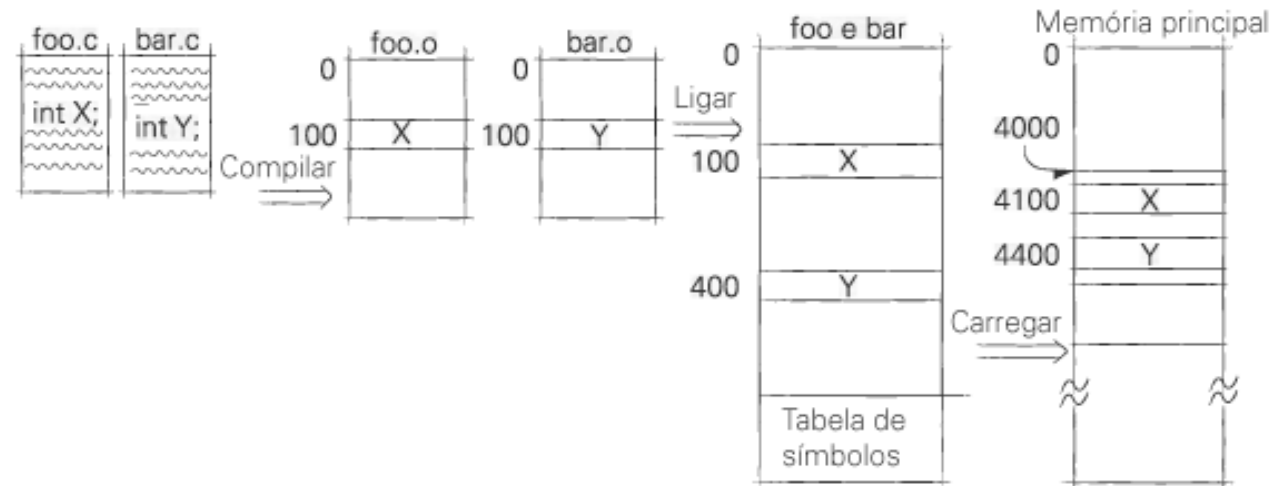
2.8.3 Carregamento

Figura 2.12 Carregamento.



2.8.3 Carregamento

Figura 2.13 Compilação, ligação e carregamento.



2.9 Firmware

- **O firmware contém instruções executáveis armazenadas na memória persistente ligada a um dispositivo.**
 - É programado com microprogramação.
 - Uma camada de programação abaixo da linguagem de máquina de um computador.
 - Microcódigo
 - Instrução simples, fundamental, necessária para implementar todas as operações em linguagem de máquina.



2.10 Middleware

- **Middleware é um software para sistemas distribuídos**
 - Permite interações entre vários processos executados em um ou mais computadores interligados na rede.
 - Facilita a operação em sistemas heterogêneos distribuídos.
 - Simplifica o desenvolvimento de aplicações.
 - Exemplo, Conectividade aberta para banco de dados (Open DataBase Connectivity — ODBC).
 - Permite que as aplicações acessem um banco de dados por meio de um middleware denominado unidade de ODBC.



Referências

Notas de aula. Arquitetura e organização de computadores. Glaucus Brelaz.

**Slides do livro Organização Estruturada de Computadores
Andrew S. Tanenbaum**

**Arquitetura e Organização de Computadores.
William Stallings**

**Organização Estruturada de Computadores
Andrew S. Tanenbaum**

Obrigado

Moisés Souto

professor.moisessouto.com.br

moisessouto@ifrn.edu.br