Apresentação da Disciplina

Disciplina Linguagens de Programação I
Bacharelado em Ciência da Computação da Uerj
Professores Guilherme Mota & Leandro Marzulo

ANSI C

```c
#include <stdio.h>
int main  ()
{
   printf("Hello World!");
   return 0;
}
```

O que vamos aprender?

- Lógica de Programação
- Solução de Problemas
- Programação em C (em ambiente Linux!)
  - Alocação dinâmica
  - Apontadores
  - Compilação com múltiplos arquivos
  - Criação de bibliotecas

Avaliação

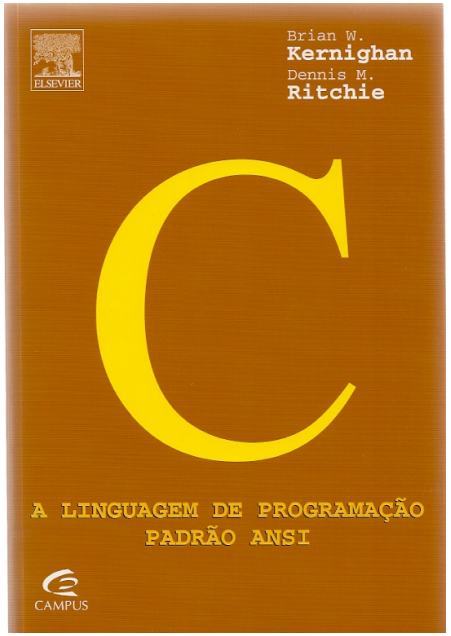
- Curso presencial
  - Permitido até 25% de faltas = 7 dias de aula
  - Não existe abono de falta
- 2 Provas
  - Questão teórica discursiva (2 pontos)
  - Solução de um problema no computador (8 pontos)
- Reposição somente com requerimento
  - Com atestado médico
  - Comprovante de viagem no trabalho

Recomendações

- Trabalhos práticos não valem nota diretamente
- Indiretamente, contudo, valem 100% da nota
- Mudar os trabalhos todos os períodos provou ter baixo custo benefício
  - Baixo comprometimento
- Programação = teoria + MUITA prática
- Quem não pratica dificilmente tem sucesso em aprender a programar
- Leitura prévia dos assuntos a serem abordados facilita o aprendizado

Material Didático

- Livro
  - C - a Linguagem de Programação Padrão Ansi
    - Autor: Kernighan, Brian / Ritchie, Dennis M
    - Editora: CAMPUS



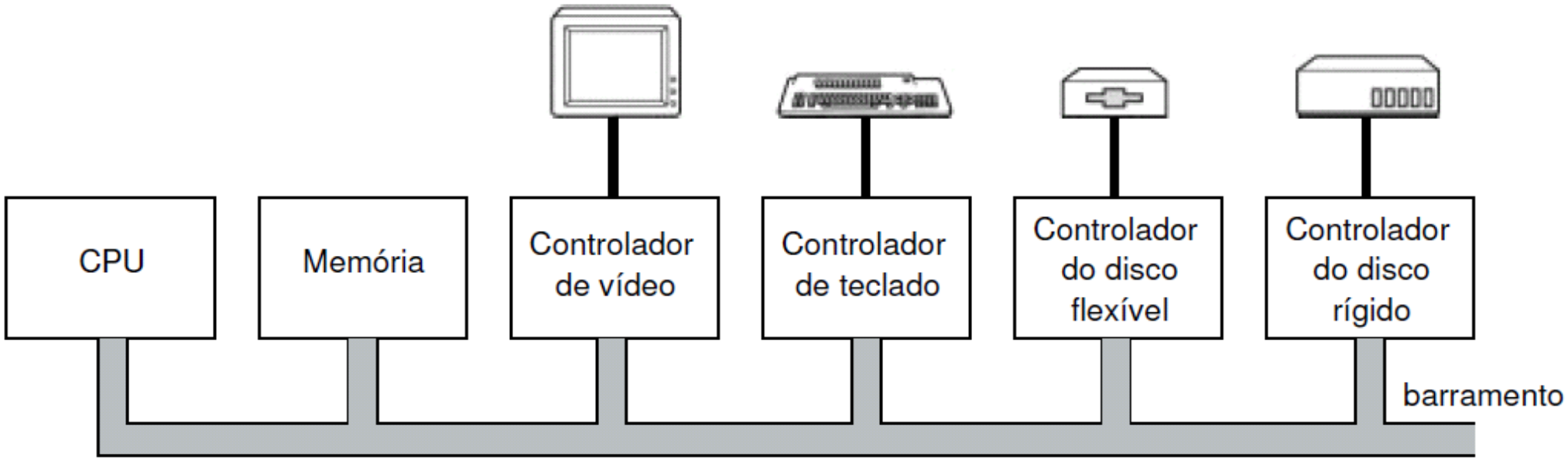
- Ambiente moodle da Uerj
  - Cadastro usuário e Inscrição em Linguagem de Programação I:
    - `www.ead.uerj.br/ava`
  - Chave de inscrição:
    - `lp1_uerj`

Instalação do Linux (Ubuntu)

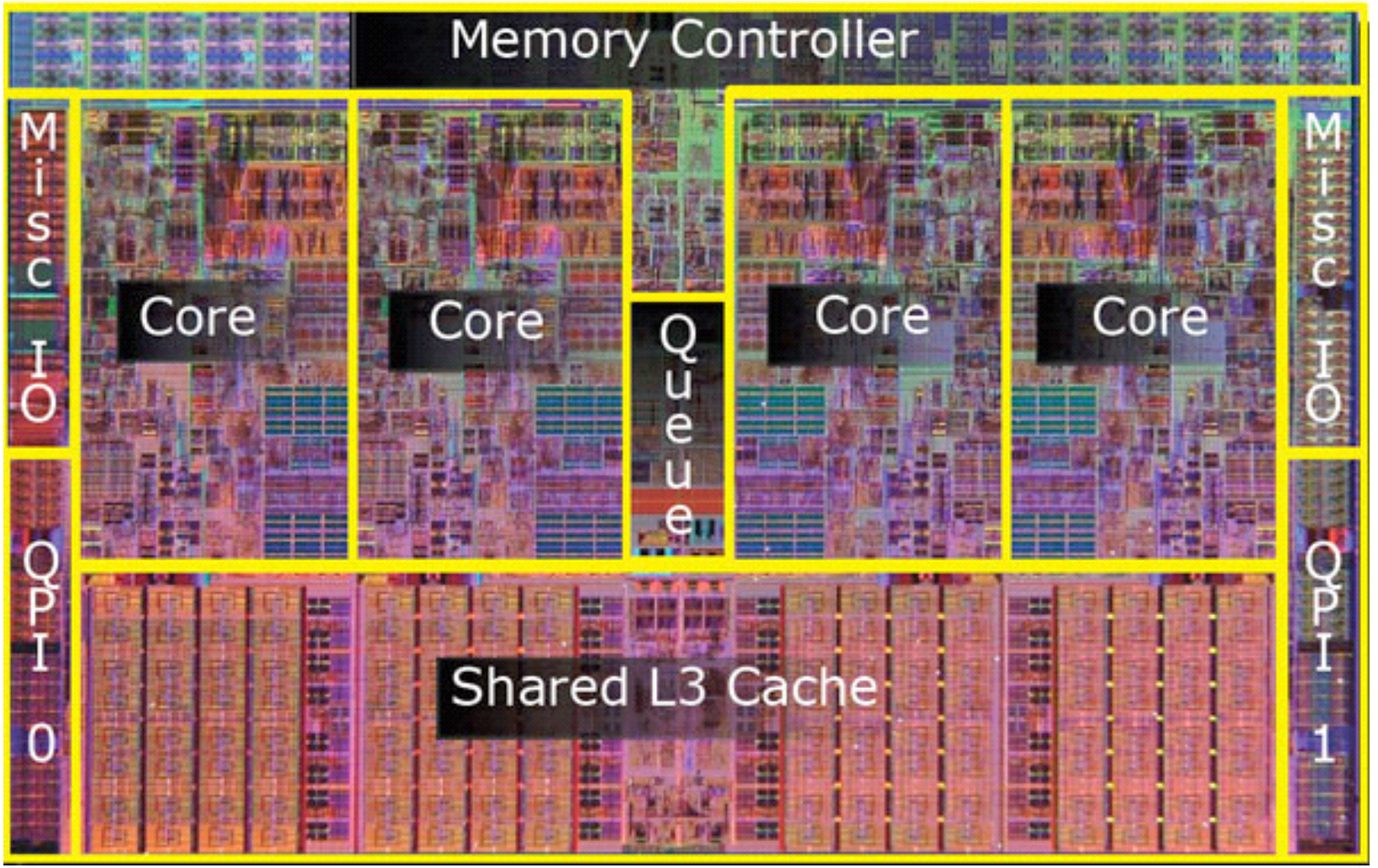
- Opções:
  - Dual boot
    - Redimensionar partição
    - Seguir os passos da instalação com o CD
  - Máquina Virtual
    - Virtualbox ou Vmware
    - É necessário ter uma máquina melhor
    - Habilitar virtualização na BIOS
  - Live CD (sem instalação)
  - Pendrive
  - Alternativa mais leve - lubuntu

Conceitos básicos Hardware

Um sistema de computação moderno

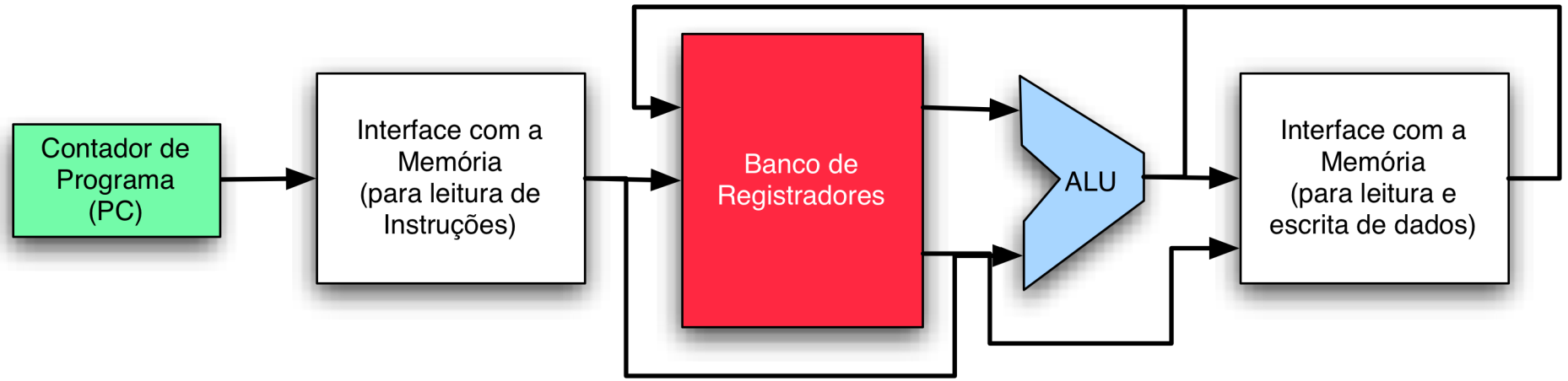


O processador



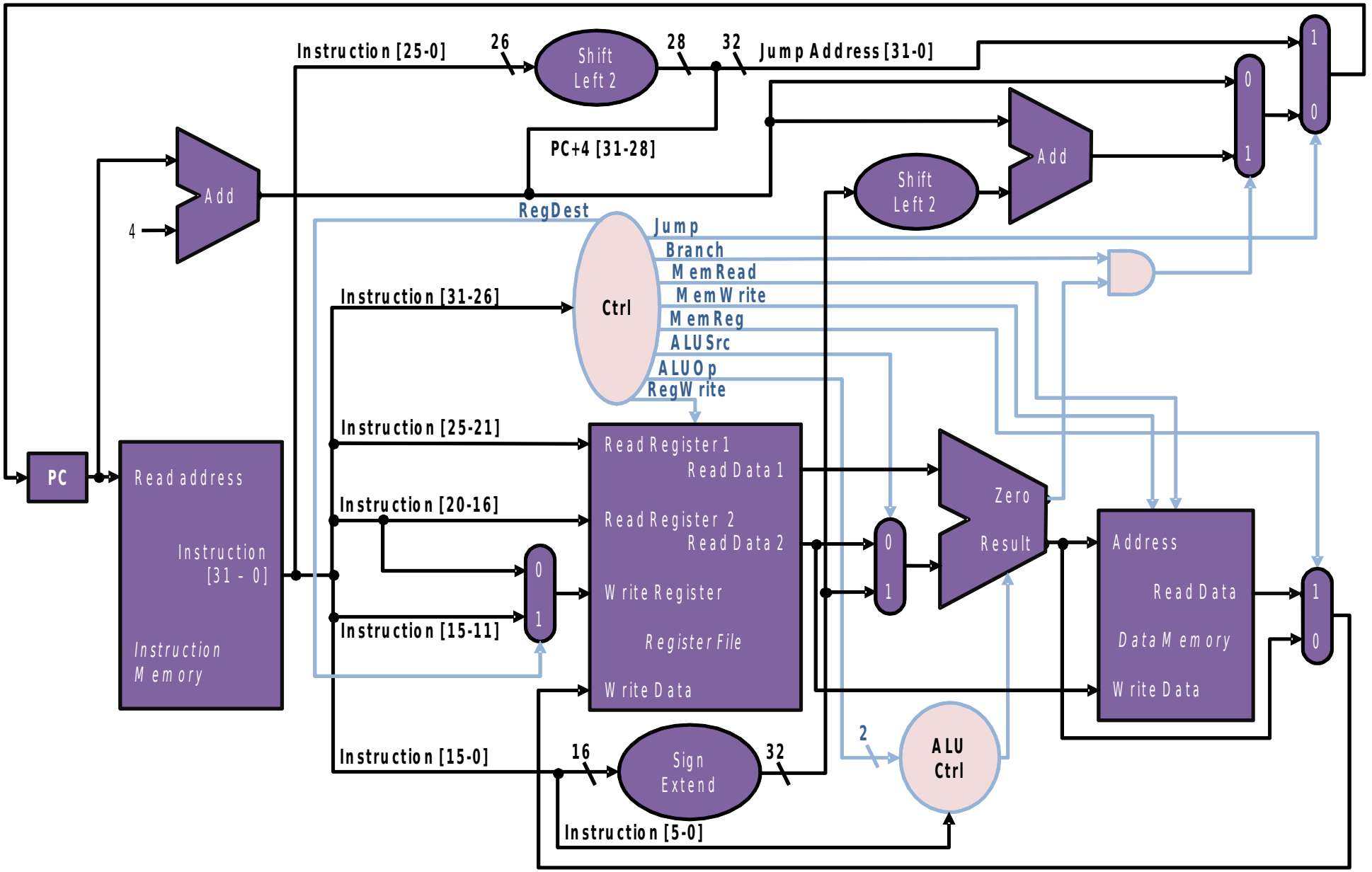
Caminho de Dados do Processador

- Visão abstrata MIPS monociclo



Mais detalhes em Arquitetura de Computadores II

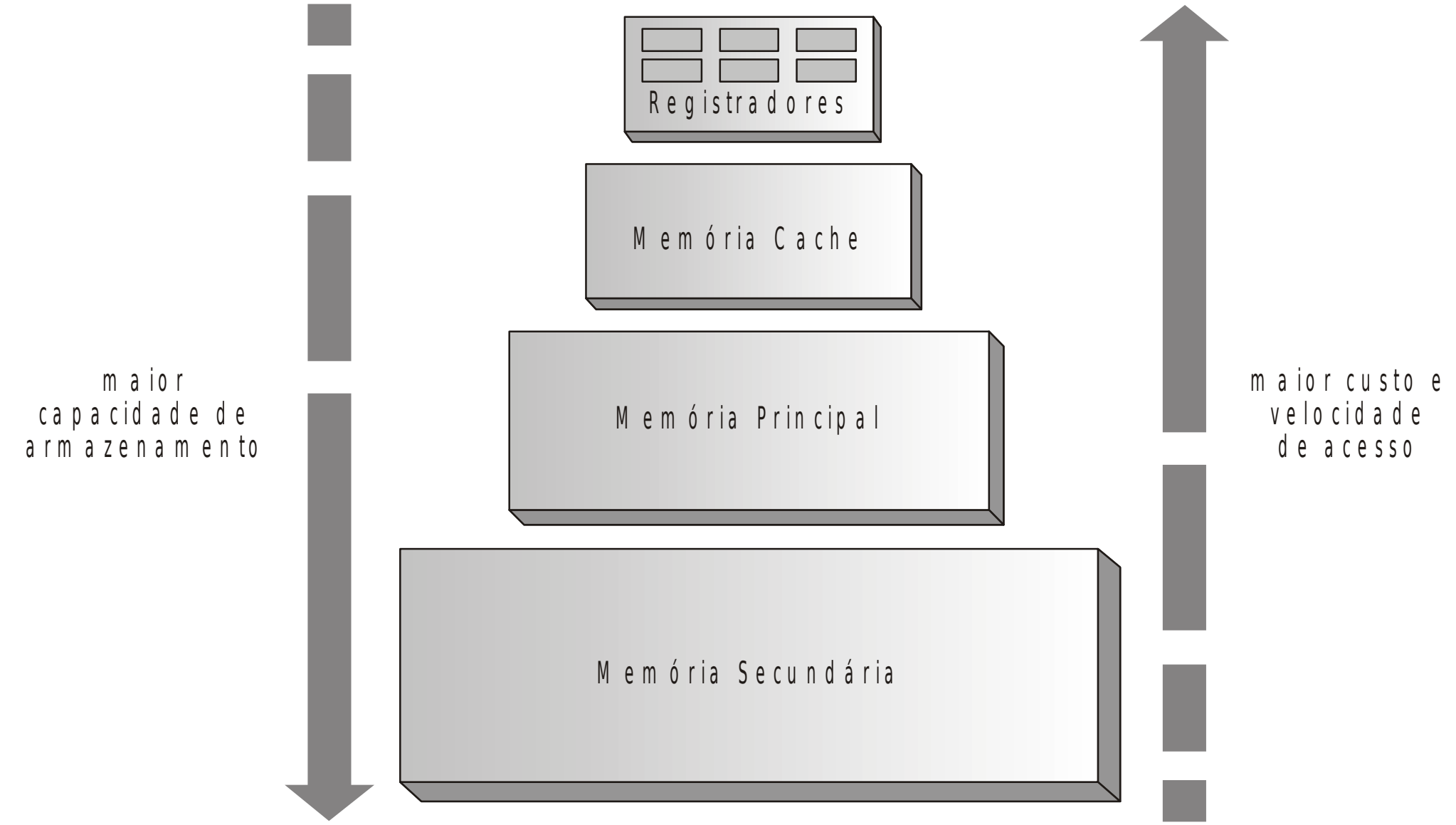
Caminho de Dados Completo (MIPS Monociclo)



O Processador

- Acesso à Memória (Dados e Instruções)
- Programa armazenado
- Instruções em código binário
  - Operação
    - Lógica e aritmética
    - Desvio de controle (condicional e incondicional)
    - Acesso à memória
  - Operandos
    - Registradores
    - Memória (endereço)
    - Imediato

Subsistema de Memória



Subsistema de Memória

- Fundamentado nos princípios de localidade temporal e espacial

```
int main(void)
{
   int i, x = 0, v[100];
   ReadVector( v, 100); //Leitura do Vetor v
   for (i = 0; i < 100; i = i + 1)
      x = x + v[i];
}
```

Subsistema de Memória → Registradores

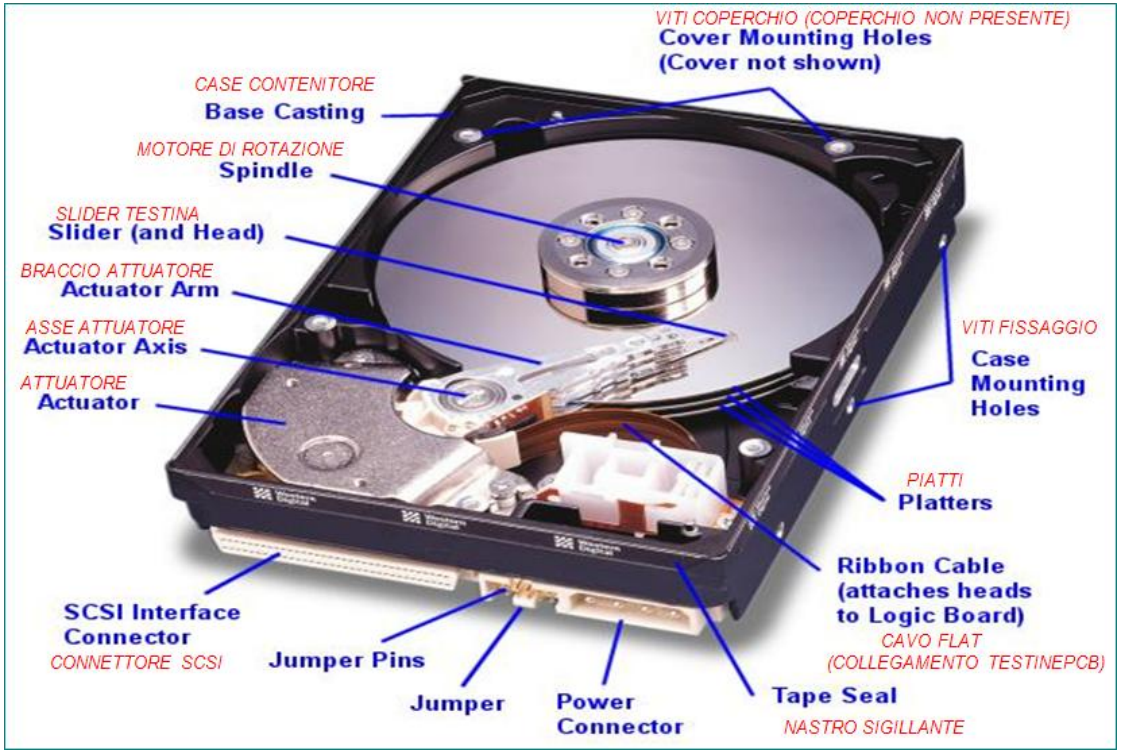
- Localizados dentro do processador (caminho de dados)
- São acessados pelas instruções
- Muito rápidos
- Quantidade Limitada
- Gerenciados pelo programador Assembly (ou compilador)

Subsistema de Memória → Cache

- A memória é muito mais lenta que o processador
- Memórias caches são mais rápidas (e caras).
- Elas guardam os dados mais frequentemente acessados para que o processados não pague o preço de ir até a memória (cache miss)
- Gerenciado pelo processador

Subsistema de Memória → Cache

- O conceito de cache é usado em outros lugares
  - Cache de navegador
  - Cache de disco usando SSD



Subsistema de Memória → Memória Principal

- RAM (Random Access Memory).
- Acessado por instruções específicas do processador (Load e Store, por exemplo).
- A memória pode ser vista como um vetor de bytes
  - Cada elemento do vetor é numerado
  - Esse número é o endereço de memória daquele dado
- Instruções de um programa também ficam na memória

Subsistema de Memória → Memória Principal

- O Sistema Operacional aloca uma parte da memória para cada programa em execução (processo).
- Os processos tem a ilusão que a memória é toda deles.

Mais detalhes

Memória Virtual: Arquitetura de computadores II

Gerência de processos: Sistemas Operacionais I

Gerência de Memória: Sistemas Operacionais II

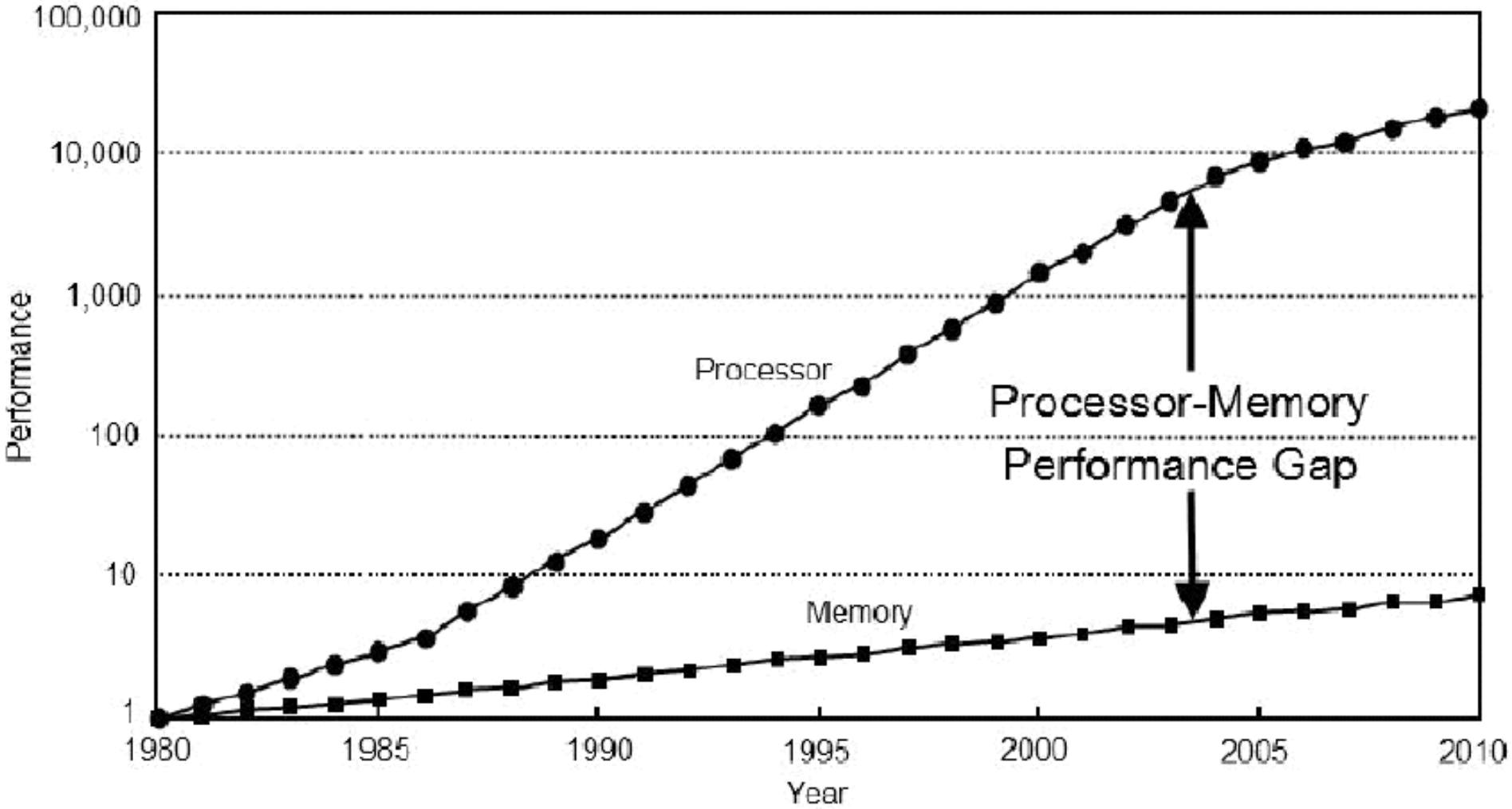
Subsistema de Memória → Memória Secundária

- Disco Rígido, CD, DVD, PenDrive
- Acessados por chamadas ao sistema operacional
- Processador não faz acesso direto a estes dispositivos
- Device drivers
- Controladoras
- Lentos, mais baratos e maior capacidade

Disciplinas Sistemas Operacionais I e II

Memory Wall

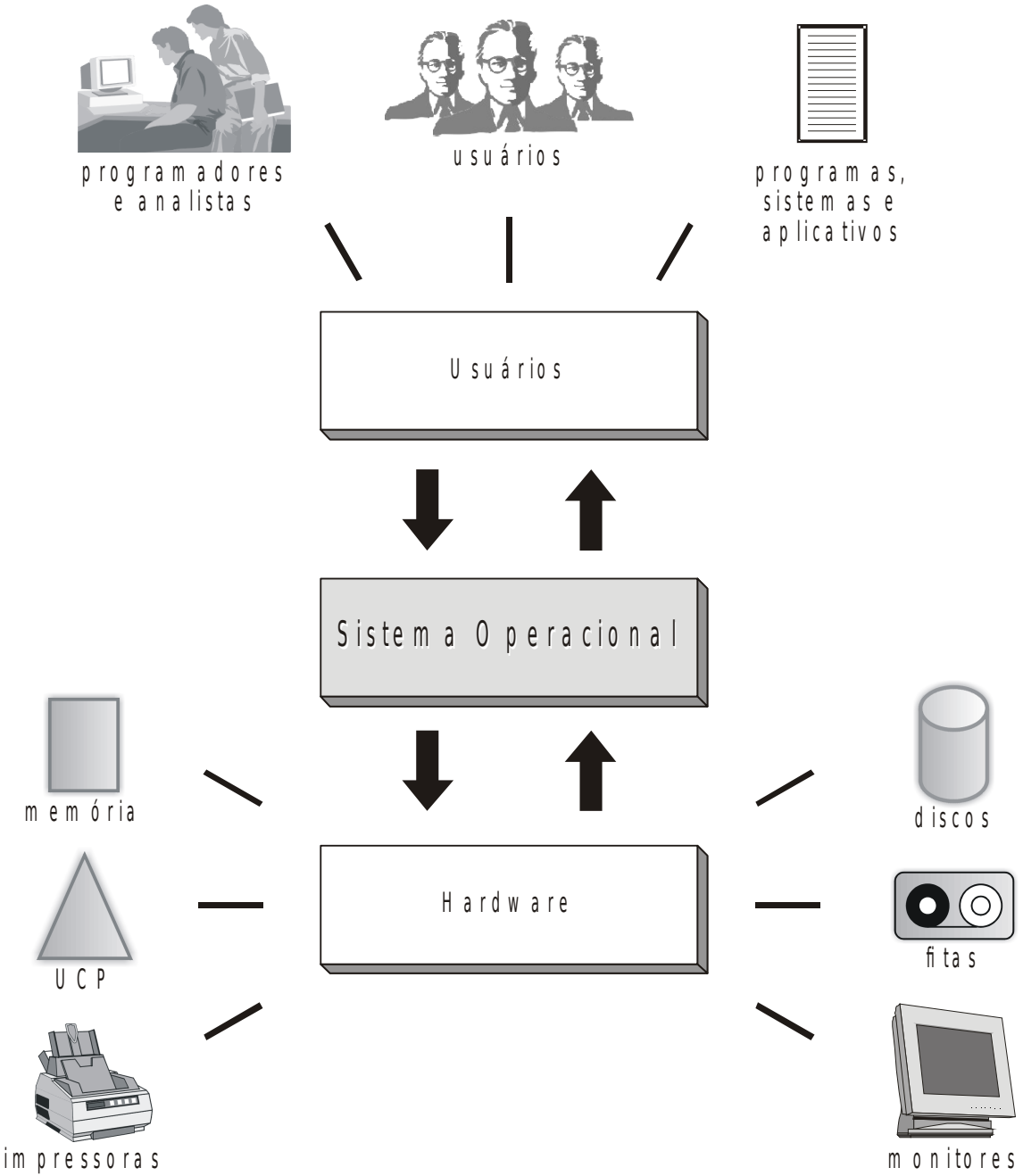
Performance: CPU X Memória



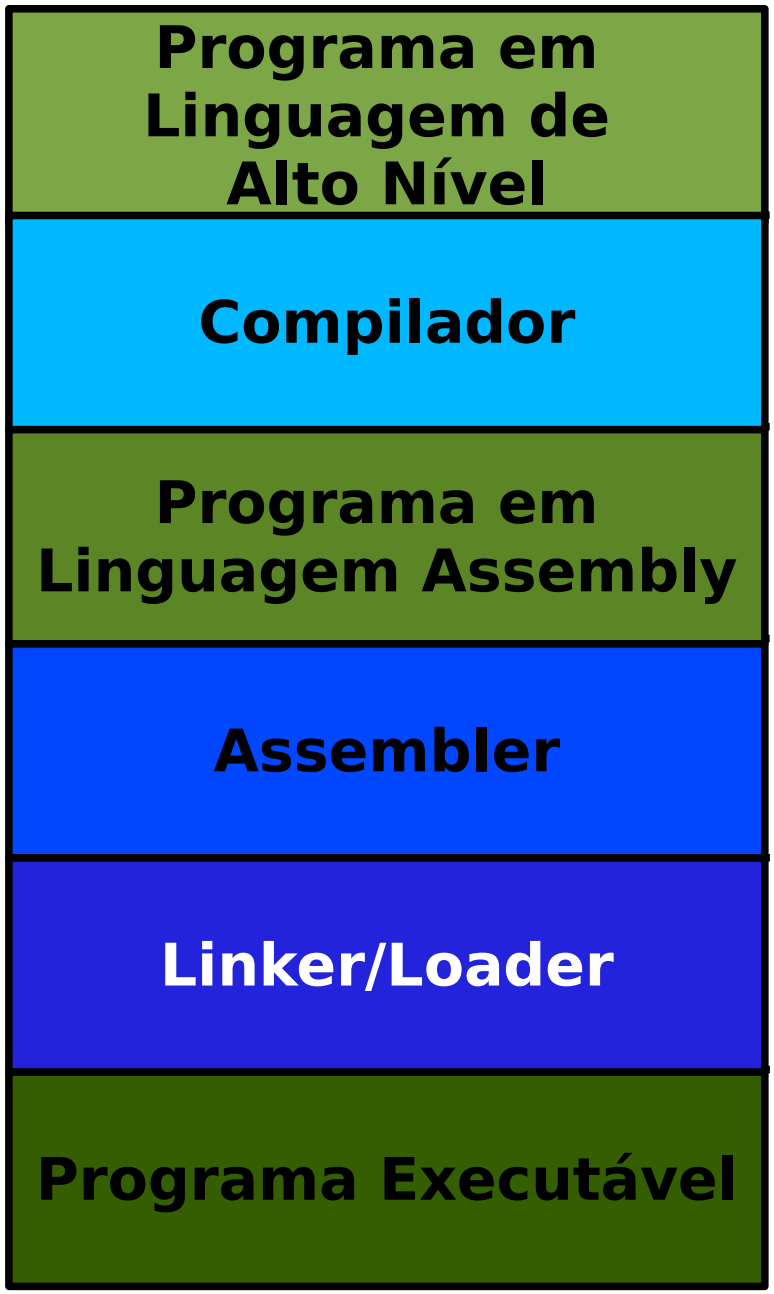
Fonte: Computer Architecture: a quantitative approach
John L. Hennessy, David A Patterson, Andrea C Arpaci-Dusseau

Conceitos básicos Software

Sistema Operacional



Compilador, Montador, Loader e Linker



Comandos de alto nível, mais próximos da nossa linguagem natural (Pascal, C, C++, Delphi, C#, VB, entre outros)

Programa que faz a tradução da linguagem de alto nível para a linguagem de montagem. Deve conhecer o conjunto de instruções da máquina.

A linguagem de montagem possui é a interface exposta pelo processador ao usuário. Programar em assembly é mais complexo.

Tradutor que converte o código de montagem em binário. É muito mais simples que o compilador (cada linha de assembly vira uma de binário).

Linker junta vários arquivos em um só binário e o loader carrega o binário na memória para execução (ajuste de endereços de memória).

Programa já carregado na memória para execução pelo processador.

A linguagem C e o seu uso

Histórico da Linguagem C

- Criada em 1972 por Dennis Ritchie
  - Um dos autores do nosso livro texto é o próprio criador da linguagem.
  - Também é um dos criadores do Unix
- No AT&T Bell Labs
- Para desenvolver o sistema operacional Unix (escrito originalmente em Assembly)
- A linguagem começou a se tornar popular depois do lançamento do primeiro livro

ANSI C

- No final dos anos 70, a linguagem C começou a substituir o BASIC nos microcomputadores
- Na década de 80 foi adaptado para uso em IBM PC
- Nesta época surgiu também o C++ (OO)
- Em 1983 o American National Standards Institute (ANSI) formou um comitê para estabelecer um padrão para a linguagem – finalizado em 1989

Algumas Características da Linguagem

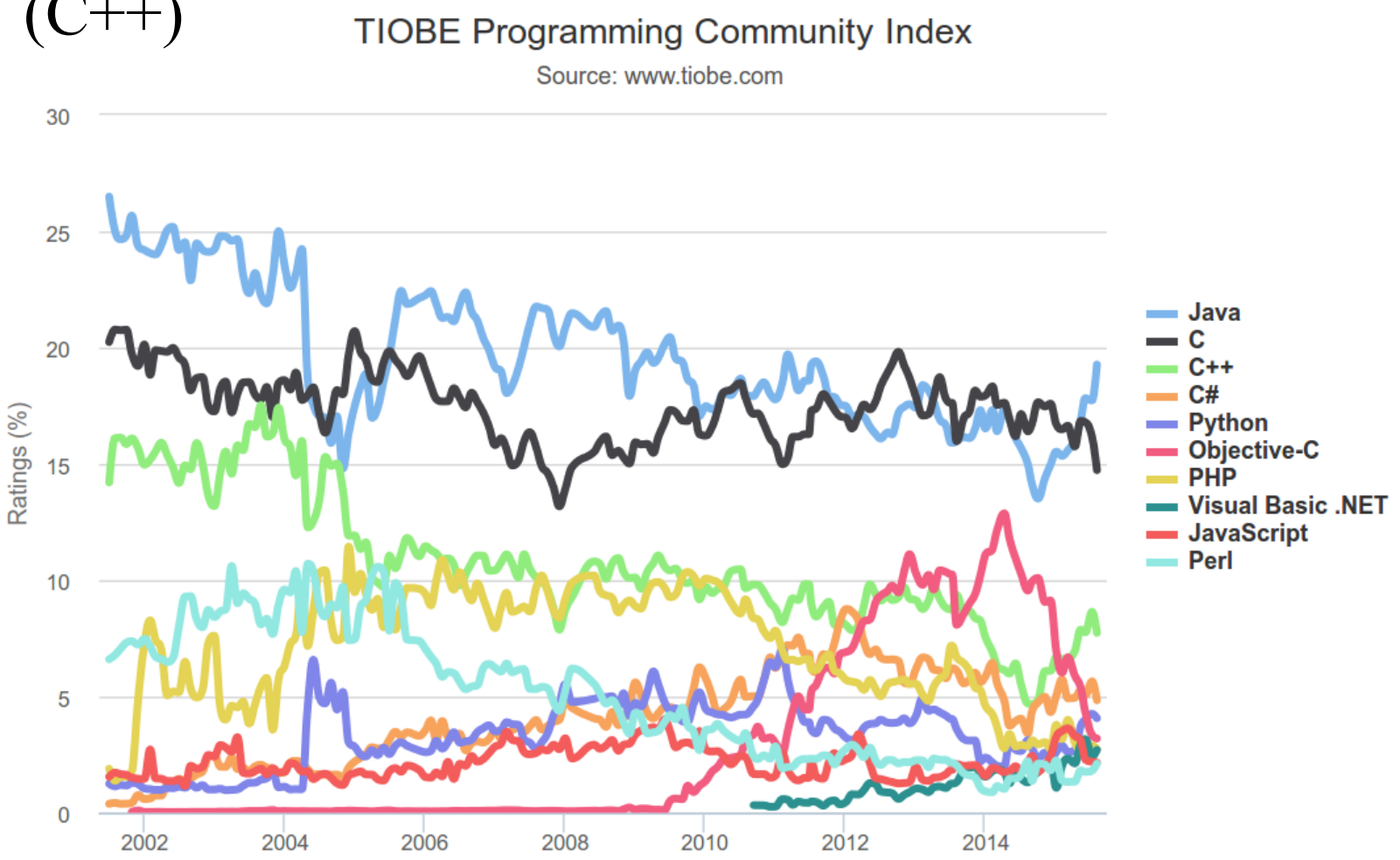
- Linguagem de alto nível
- Imperativa
- Procedural
- Tipagem estática (pré-declaração) e forte (fixo)
- Acesso de baixo nível à memória
- Baixos requerimentos de hardware
- Possibilita reaproveitamento de código
- Case sensitive

Algumas Características da Linguagem

- Simples
- Funcionalidades não essenciais fornecidas por um conjunto de bibliotecas padronizadas de funções
- Sistema de tipos simples
- Suporte a pré-processamento
  - Macros
  - Compilação condicional
- Apontadores
- Possibilidade de inclusão de código assembly (otimização manual)

Quem usa C?

- Computação de alto desempenho (C++ também)
- CUDA (C++)
- Unix
- Linux



Fim