



## Arquitetura de Computadores

### Aula 5

Prof. André Roberto Guerra

## Organização da Aula

### ISA – *Instruction Set Architecture*

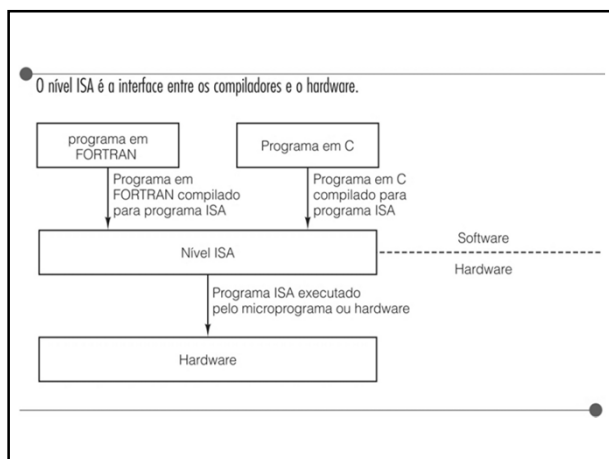
- Visão geral
- Tipos de dados
- Formatos e tipos de instruções
- Endereçamento

## Sistemas Operacionais

- Visão geral de sistema operacional
- Objetivos e funções de sistemas operacionais
- Tipos de sistemas operacionais

## Visão Geral ISA

- ISA – *Instruction Set Architecture* (arquitetura do conjunto de instruções) está entre os níveis da microarquitetura e do sistema operacional
- **Fronteira** e/ou **interface** entre o **hardware** e o **software**



- ISA foi desenvolvido antes de quaisquer outros níveis
- Originalmente era o único nível
- Denominado como “**a arquitetura**” da máquina ou (**incorretamente**) como “linguagem de montagem”



- ISA é adotado para traduzir programas escritos nas mais diversas linguagens para uma forma intermediária comum, e para construir *hardware* que os execute diretamente nesta forma

- ISA define a interface entre os compiladores e o *hardware*
- **É a linguagem que ambos têm de entender**

- **Compatibilidade:** o desafio é projetar máquinas melhores e mais velozes garantindo a compatibilidade com sistemas e aplicativos antigos

- Fatores essenciais a uma “boa” ISA:
  - definir conjunto de instruções que possa ser facilmente implementado
  - fornecer um alvo claro para o código compilado

### Tipo de Dados ISA

- ISA utiliza dados dos tipos: **numéricos** e **não numéricos**
- Dados numéricos são:
  - **inteiros** (com e sem sinal)
  - **inteiro decimal** (código binário)
  - **ponto flutuante**

- Dados não numéricos são:
  - **ponteiros** (endereço de máquina)
  - **caracteres ASCII** e **UNICODE**
  - **booleanos** (de dois valores: V ou F, 0 ou 1, M ou F etc.)



### Formatos, Endereçamento e Tipos de Instrução ISA

- Instruções ISA são formadas por **opcodes** (códigos de operação), usualmente em conjunto com alguma informação adicional, tais como de onde vêm os operandos e para onde vão os resultados

### Formatos de Instrução ISA

- Diversos formatos (possíveis) de instrução
- Instruções sempre têm um *opcode* que indica o que ela faz
- Instruções podem não ter endereço ou ter 1, 2, ou 3

- Em algumas máquinas todas as instruções têm o mesmo comprimento; em outras, pode haver muitos comprimentos diferentes

- Vários fatores considerados na escolha do formato de instruções
- A dificuldade na decisão não deve ser subestimada e o formato deve ser decidido no início do projeto
- O conjunto de instruções pode sobreviver por muitos anos

### Endereçamento de Instrução

- Instruções possuem operandos e o **endereçamento** especifica onde eles (operandos) estão
- **Modos de endereçamento** como os bits de um campo de endereço são interpretados para encontrar o operando

- São diversos os modos:
  - **Endereçamento Imediato** (o modo mais simples)
  - **Endereçamento Direto** (especifica o endereço completo na memória)



- **Endereçamento de Registrador** (especifica um registrador)
- **Endereçamento Indireto de Registrador ou Ponteiro** (endereço em registrador, sem endereço na instrução)

- **Endereçamento Indexado** (registrador e deslocamento constante)
- **Endereçamento de Base Indexado** (dois registradores)
- **Endereçamento de Pilha**

### Tipos de Instrução ISA

- Instruções ISA são divididas em grupos, com similaridade entre as máquinas, diferentes nos detalhes
- Instruções fora do comum, para manter a compatibilidade com modelos anteriores

- Dentre elas destacam-se:
  - instruções para movimento de dados
  - operações diádicas
  - operações monádicas

- comparações e desvios condicionais
- instruções de chamada de procedimento
- controle de laço
- entrada/saída

### Visão Geral de Sistema Operacional

- Segundo **Tanenbaum**, **sistema operacional** é um programa que, do ponto de vista do programador, acrescenta uma variedade de novas instruções e características, acima e além do que o nível ISA fornece



## Sistema Operacional

- Normalmente, o SO é implementado em *software*, sem razão teórica por não ser colocado em *hardware* (microprogramas)
- **OSM** (*Operating System Machine*) é denominado nível de máquina de sistema operacional

- Stallings define o **sistema operacional** como um programa que **gerencia os recursos do computador**, fornece serviços para os programadores e estabelece uma ordem de execução de outros programas

- O conhecimento sobre SO é essencial para o entendimento dos mecanismos da CPU para controle do computador
- Em particular, **interrupções e gerenciamento de memória**

- O SO controla a execução de programas em um processador, principalmente as funções:
  - **escalonamento de processos**
  - **gerenciamento de memória**
- Só executadas se o SO possuir suporte do *hardware* da CPU

- CPU tem também *hardware* para:
  - **gerenciamento de memória virtual e de processos**
  - registradores especiais e áreas de armazenamento temporário
  - conjunto de circuitos de gerenciamento de recursos

- **Gerenciamento de memória** é outra importante função do SO
- SO atuais possuem a capacidade de **memória virtual**, o que traz dois benefícios:



1. processo executado na memória principal sem que todas as instruções e dados estejam armazenados na memória principal
2. espaço de memória total disponível pode exceder o tamanho da memória principal do sistema

- Embora o gerenciamento de memória seja feito pelo *software*, o sistema operacional conta com suporte do *hardware* do processador, incluindo paginação e segmentação da memória

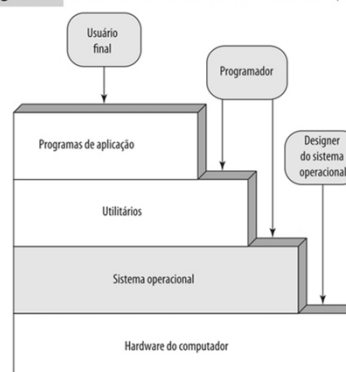
- **Escalonamento de processos** é uma das funções mais importantes do SO que determina os processos a executar a cada instante
- Possibilita compartilhar o tempo de CPU entre determinado número de processos de modo imparcial

### Objetivos e Funções dos Sistemas Operacionais

- Stallings define o SO como programa que controla a execução de programas aplicativos e age como **interface** entre o **usuário** e o *hardware* do **computador**

- Possui basicamente os objetivos:
  - **conveniência:** tornar o uso do computador mais conveniente
  - **eficiência:** utilização mais eficiente dos recursos do sistema

Figura 8.1 Camadas e visões de um sistema de computação





- O usuário final vê o sistema de computação como uma aplicação, sem interessar-se pelo modo como executa suas complexas e difíceis funções

- Um conjunto de programas de sistema (utilitários) implementam as principais funções

- O SO é o mais importante. Esconde os detalhes do *hardware* e fornece uma interface conveniente, agindo como um mediador, tornando mais fácil o acesso e uso desses recursos e serviços

- Tipicamente o SO fornece serviços para a realização das seguintes atividades:
  - **criação de programas**
  - **execução de programas**
  - **acesso a dispositivos de E/S**

- **acesso controlado a arquivos**
- **acesso ao sistema**
- **detecção e reação aos erros**
- **monitoração**

### **O sistema operacional como gerente de recursos**

- Computador é um conjunto de recursos (processar, transferir e armazenar dados e controlar essas funções) controlados pelo SO



- O SO é um programa como outro qualquer, executado pela CPU
- Renuncia o controle da CPU para, em seguida, obter o controle novamente

- Uma parte do sistema operacional reside na memória principal – o núcleo (*kernel*), que contém as funções do SO usadas mais frequentemente, além de outras partes do SO que estão em uso naquele momento

- O SO também decide quando um dispositivo de E/S pode ser usado pelo programa em execução e controla o acesso e o uso de arquivos

- A CPU também é um recurso controlado pelo SO, que determina quanto tempo da CPU deve ser dedicado à execução de cada programa de usuário

- No caso de um sistema de computação com múltiplos processadores, essa decisão se estende a todos os processadores

### **Tipos de Sistema Operacional**

- Algumas características básicas diferenciam os vários tipos de sistemas operacionais
- Essas características são relativas a dois aspectos independentes





- O primeiro aspecto especifica se o sistema de computação é **interativo** ou é um sistema de processamento em lotes (**batch**)
- No interativo, o programador/ usuário interage diretamente com o computador. No *batch*, o oposto

- Outro aspecto independente especifica se o sistema de computação emprega **multiprogramação** ou não
- Uma tentativa de deixar a CPU ocupada o maior tempo possível, mantendo-a trabalhando em mais de uma tarefa (programa)

## Síntese

### Referências de Apoio

- TANENBAUM, A. S. **Organização Estruturada de Computadores**. 6. ed. Prentice-Hall, 2013.
- STALLINGS, W. **Arquitetura e Organização de Computadores**. 8. ed. Prentice-Hall, 2010.