

Algoritmos

Paulo Torrens

paulotorrens@gnu.org

Departamento de Ciência da Computação
Centro de Ciências e Tecnologias
Universidade do Estado de Santa Catarina

2020/1

Introdução à Programação

- Um **programa** de computador é composto de uma sequência de comandos, capazes de instruir um computador a realizar um conjunto de operações desejadas
- Embora possamos dar um sentido mais amplo à palavra, a **programação** geralmente se refere ao processo de **escrita e manutenção** de um programa de computador
- O processo de programação é uma aplicação importante para as áreas da tecnologia da informação e ciência da computação
 - Inclui desde o desenvolvimento de sites, jogos e aplicativos, ao controle de máquinas industriais, sistemas aeroespaciais e desenvolvimento científico
- Mas como um programa é executado dentro de um computador? Como o *hardware* é organizado, e como ele pode executar operações descritas pelos programadores?

Introdução à Programação

- Um **programa** de computador é composto de uma sequência de comandos, capazes de instruir um computador a realizar um conjunto de operações desejadas
- Embora possamos dar um sentido mais amplo à palavra, a **programação** geralmente se refere ao processo de **escrita e manutenção** de um programa de computador
- O processo de programação é uma aplicação importante para as áreas da tecnologia da informação e ciência da computação
 - Inclui desde o desenvolvimento de sites, jogos e aplicativos, ao controle de máquinas industriais, sistemas aeroespaciais e desenvolvimento científico
- Mas como um programa é executado dentro de um computador? Como o *hardware* é organizado, e como ele pode executar operações descritas pelos programadores?

Introdução à Programação

- Um **programa** de computador é composto de uma sequência de comandos, capazes de instruir um computador a realizar um conjunto de operações desejadas
- Embora possamos dar um sentido mais amplo à palavra, a **programação** geralmente se refere ao processo de **escrita** e **manutenção** de um programa de computador
- O processo de programação é uma aplicação importante para as áreas da tecnologia da informação e ciência da computação
 - Inclui desde o desenvolvimento de sites, jogos e aplicativos, ao controle de máquinas industriais, sistemas aeroespaciais e desenvolvimento científico
- Mas como um programa é executado dentro de um computador? Como o *hardware* é organizado, e como ele pode executar operações descritas pelos programadores?

Introdução à Programação

- Um **programa** de computador é composto de uma sequência de comandos, capazes de instruir um computador a realizar um conjunto de operações desejadas
- Embora possamos dar um sentido mais amplo à palavra, a **programação** geralmente se refere ao processo de **escrita** e **manutenção** de um programa de computador
- O processo de programação é uma aplicação importante para as áreas da tecnologia da informação e ciência da computação
 - Inclui desde o desenvolvimento de sites, jogos e aplicativos, ao controle de máquinas industriais, sistemas aeroespaciais e desenvolvimento científico
- Mas como um programa é executado dentro de um computador? Como o *hardware* é organizado, e como ele pode executar operações descritas pelos programadores?

Introdução à Programação

- Um **programa** de computador é composto de uma sequência de comandos, capazes de instruir um computador a realizar um conjunto de operações desejadas
- Embora possamos dar um sentido mais amplo à palavra, a **programação** geralmente se refere ao processo de **escrita** e **manutenção** de um programa de computador
- O processo de programação é uma aplicação importante para as áreas da tecnologia da informação e ciência da computação
 - Inclui desde o desenvolvimento de sites, jogos e aplicativos, ao controle de máquinas industriais, sistemas aeroespaciais e desenvolvimento científico
- Mas como um programa é executado dentro de um computador? Como o *hardware* é organizado, e como ele pode executar operações descritas pelos programadores?

Organização do computador

- Um computador é um equipamento capaz de executar alguma operação sobre uma entrada de dados, e produzir uma saída
- Embora o exemplo mais óbvio seja o computador pessoal, existem computadores com diversos propósitos e tamanhos
- Um computador é composto de um conjunto de *hardware* e *software*
 - O *hardware* representa as partes físicas do computador, como a memória, o processador, e as unidades de entrada e saída
 - O *software* são os programas contidos dentro do computador, incluindo o sistema operacional e demais aplicativos
- Podemos citar como unidades de entrada o *mouse* e o teclado, e como unidade de saída o monitor e a impressora
- Além disso, podemos dividir a memória em memória principal (a memória RAM) e a memória secundária (discos rígidos e discos de estado sólido)

Organização do computador

- Um computador é um equipamento capaz de executar alguma operação sobre uma entrada de dados, e produzir uma saída
- Embora o exemplo mais óbvio seja o computador pessoal, existem computadores com diversos propósitos e tamanhos
- Um computador é composto de um conjunto de *hardware* e *software*
 - O *hardware* representa as partes físicas do computador, como a memória, o processador, e as unidades de entrada e saída
 - O *software* são os programas contidos dentro do computador, incluindo o sistema operacional e demais aplicativos
- Podemos citar como unidades de entrada o *mouse* e o teclado, e como unidade de saída o monitor e a impressora
- Além disso, podemos dividir a memória em memória principal (a memória RAM) e a memória secundária (discos rígidos e discos de estado sólido)

Organização do computador

- Um computador é um equipamento capaz de executar alguma operação sobre uma entrada de dados, e produzir uma saída
- Embora o exemplo mais óbvio seja o computador pessoal, existem computadores com diversos propósitos e tamanhos
- Um computador é composto de um conjunto de *hardware* e *software*
 - O *hardware* representa as partes físicas do computador, como a memória, o processador, e as unidades de entrada e saída
 - O *software* são os programas contidos dentro do computador, incluindo o sistema operacional e demais aplicativos
- Podemos citar como unidades de entrada o *mouse* e o teclado, e como unidade de saída o monitor e a impressora
- Além disso, podemos dividir a memória em memória principal (a memória RAM) e a memória secundária (discos rígidos e discos de estado sólido)

Organização do computador

- Um computador é um equipamento capaz de executar alguma operação sobre uma entrada de dados, e produzir uma saída
- Embora o exemplo mais óbvio seja o computador pessoal, existem computadores com diversos propósitos e tamanhos
- Um computador é composto de um conjunto de *hardware* e *software*
 - O *hardware* representa as partes físicas do computador, como a memória, o processador, e as unidades de entrada e saída
 - O *software* são os programas contidos dentro do computador, incluindo o sistema operacional e demais aplicativos
- Podemos citar como unidades de entrada o *mouse* e o teclado, e como unidade de saída o monitor e a impressora
- Além disso, podemos dividir a memória em memória principal (a memória RAM) e a memória secundária (discos rígidos e discos de estado sólido)

Organização do computador

- Um computador é um equipamento capaz de executar alguma operação sobre uma entrada de dados, e produzir uma saída
- Embora o exemplo mais óbvio seja o computador pessoal, existem computadores com diversos propósitos e tamanhos
- Um computador é composto de um conjunto de *hardware* e *software*
 - O *hardware* representa as partes físicas do computador, como a memória, o processador, e as unidades de entrada e saída
 - O *software* são os programas contidos dentro do computador, incluindo o sistema operacional e demais aplicativos
- Podemos citar como unidades de entrada o *mouse* e o teclado, e como unidade de saída o monitor e a impressora
- Além disso, podemos dividir a memória em memória principal (a memória RAM) e a memória secundária (discos rígidos e discos de estado sólido)

Organização do computador

- Um computador é um equipamento capaz de executar alguma operação sobre uma entrada de dados, e produzir uma saída
- Embora o exemplo mais óbvio seja o computador pessoal, existem computadores com diversos propósitos e tamanhos
- Um computador é composto de um conjunto de *hardware* e *software*
 - O *hardware* representa as partes físicas do computador, como a memória, o processador, e as unidades de entrada e saída
 - O *software* são os programas contidos dentro do computador, incluindo o sistema operacional e demais aplicativos
- Podemos citar como unidades de entrada o *mouse* e o teclado, e como unidade de saída o monitor e a impressora
- Além disso, podemos dividir a memória em memória principal (a memória RAM) e a memória secundária (discos rígidos e discos de estado sólido)

Organização do computador

- Um computador é um equipamento capaz de executar alguma operação sobre uma entrada de dados, e produzir uma saída
- Embora o exemplo mais óbvio seja o computador pessoal, existem computadores com diversos propósitos e tamanhos
- Um computador é composto de um conjunto de *hardware* e *software*
 - O *hardware* representa as partes físicas do computador, como a memória, o processador, e as unidades de entrada e saída
 - O *software* são os programas contidos dentro do computador, incluindo o sistema operacional e demais aplicativos
- Podemos citar como unidades de entrada o *mouse* e o teclado, e como unidade de saída o monitor e a impressora
- Além disso, podemos dividir a memória em memória principal (a memória RAM) e a memória secundária (discos rígidos e discos de estado sólido)

- Embora estejamos habituados com o sistema **decimal**, um sistema de base 10, onde usamos dez símbolos distintos para representar números, esse sistema não costuma ser adequado para salvar dados digitais
- O principal sistema numérico usado pelos computadores convencionais é o sistema **binário**, ou seja, um sistema de base 2, contendo apenas dois símbolos: 0 e 1
 - Então nós contamos: 0, 1, 10, 11, 100, 101, 110, 111, 1000...
 - O termo *bit* se refere a um dígito binário, sendo ele zero ou um, e é usado internamente dentro da memória
 - O termo *byte* representa o menor conjunto de bits que podem ser acessados simultaneamente pelo sistema: em praticamente todos os casos, isso significa 8bits
 - O tamanho do barramento do computador, usado para transferir dados entre a memória e o processador, define o tamanho da **palavra binária** (e.g., 32bits e 64bits)

- Embora estejamos habituados com o sistema **decimal**, um sistema de base 10, onde usamos dez símbolos distintos para representar números, esse sistema não costuma ser adequado para salvar dados digitais
- O principal sistema numérico usado pelos computadores convencionais é o sistema **binário**, ou seja, um sistema de base 2, contendo apenas dois símbolos: **0** e **1**
 - Então nós contamos: 0, 1, 10, 11, 100, 101, 110, 111, 1000...
 - O termo *bit* se refere a um dígito binário, sendo ele zero ou um, e é usado internamente dentro da memória
 - O termo *byte* representa o menor conjunto de bits que podem ser acessados simultaneamente pelo sistema: em praticamente todos os casos, isso significa 8bits
 - O tamanho do barramento do computador, usado para transferir dados entre a memória e o processador, define o tamanho da **palavra binária** (e.g., 32bits e 64bits)

- Embora estejamos habituados com o sistema **decimal**, um sistema de base 10, onde usamos dez símbolos distintos para representar números, esse sistema não costuma ser adequado para salvar dados digitais
- O principal sistema numérico usado pelos computadores convencionais é o sistema **binário**, ou seja, um sistema de base 2, contendo apenas dois símbolos: **0** e **1**
 - Então nós contamos: 0, 1, 10, 11, 100, 101, 110, 111, 1000...
 - O termo *bit* se refere a um dígito binário, sendo ele zero ou um, e é usado internamente dentro da memória
 - O termo *byte* representa o menor conjunto de bits que podem ser acessados simultaneamente pelo sistema: em praticamente todos os casos, isso significa 8bits
 - O tamanho do barramento do computador, usado para transferir dados entre a memória e o processador, define o tamanho da **palavra binária** (e.g., 32bits e 64bits)

- Embora estejamos habituados com o sistema **decimal**, um sistema de base 10, onde usamos dez símbolos distintos para representar números, esse sistema não costuma ser adequado para salvar dados digitais
- O principal sistema numérico usado pelos computadores convencionais é o sistema **binário**, ou seja, um sistema de base 2, contendo apenas dois símbolos: **0** e **1**
 - Então nós contamos: 0, 1, 10, 11, 100, 101, 110, 111, 1000...
 - O termo *bit* se refere a um dígito binário, sendo ele zero ou um, e é usado internamente dentro da memória
 - O termo *byte* representa o menor conjunto de bits que podem ser acessados simultaneamente pelo sistema: em praticamente todos os casos, isso significa 8bits
 - O tamanho do barramento do computador, usado para transferir dados entre a memória e o processador, define o tamanho da **palavra binária** (e.g., 32bits e 64bits)

- Embora estejamos habituados com o sistema **decimal**, um sistema de base 10, onde usamos dez símbolos distintos para representar números, esse sistema não costuma ser adequado para salvar dados digitais
- O principal sistema numérico usado pelos computadores convencionais é o sistema **binário**, ou seja, um sistema de base 2, contendo apenas dois símbolos: **0** e **1**
 - Então nós contamos: 0, 1, 10, 11, 100, 101, 110, 111, 1000...
 - O termo *bit* se refere a um dígito binário, sendo ele zero ou um, e é usado internamente dentro da memória
 - O termo *byte* representa o menor conjunto de bits que podem ser acessados simultaneamente pelo sistema: em praticamente todos os casos, isso significa 8bits
 - O tamanho do barramento do computador, usado para transferir dados entre a memória e o processador, define o tamanho da **palavra binária** (e.g., 32bits e 64bits)

- Embora estejamos habituados com o sistema **decimal**, um sistema de base 10, onde usamos dez símbolos distintos para representar números, esse sistema não costuma ser adequado para salvar dados digitais
- O principal sistema numérico usado pelos computadores convencionais é o sistema **binário**, ou seja, um sistema de base 2, contendo apenas dois símbolos: **0** e **1**
 - Então nós contamos: 0, 1, 10, 11, 100, 101, 110, 111, 1000...
 - O termo *bit* se refere a um dígito binário, sendo ele zero ou um, e é usado internamente dentro da memória
 - O termo *byte* representa o menor conjunto de bits que podem ser acessados simultaneamente pelo sistema: em praticamente todos os casos, isso significa 8bits
 - O tamanho do barramento do computador, usado para transferir dados entre a memória e o processador, define o tamanho da **palavra binária** (e.g., 32bits e 64bits)

- A base 2 é importante por podermos definir unidades em termos de bits: ligado ou desligado, sim ou não...
- A consequência é que muitas vezes tratamos de valores baseados em uma potência de 2 para representar grandezas
 - K (kilo): 2^{10} ($\approx 10^3$)
 - M (mega): 2^{20} ($\approx 10^6$)
 - G (giga): 2^{30} ($\approx 10^9$)
 - T (tera): 2^{40} ($\approx 10^{12}$)
 - P (peta): 2^{50} ($\approx 10^{15}$)
- Além do sistema binário, os sistemas octais e hexadecimais são muito comuns dentro de sistemas computacionais
- O sistema **octal**, de base 8, é facilmente convertido para o sistema binário (visto que cada três dígitos binários equivalem a um dígito octal) e vice-versa
- O sistema **hexadecimal**, de base 16, é facilmente convertido para o sistema binário (visto que cada quatro dígitos binários equivalem a um dígito hexadecimal) e vice-versa

- A base 2 é importante por podermos definir unidades em termos de bits: ligado ou desligado, sim ou não...
- A consequência é que muitas vezes tratamos de valores baseados em uma potência de 2 para representar grandezas
 - K (kilo): 2^{10} ($\approx 10^3$)
 - M (mega): 2^{20} ($\approx 10^6$)
 - G (giga): 2^{30} ($\approx 10^9$)
 - T (tera): 2^{40} ($\approx 10^{12}$)
 - P (peta): 2^{50} ($\approx 10^{15}$)
- Além do sistema binário, os sistemas octais e hexadecimais são muito comuns dentro de sistemas computacionais
- O sistema **octal**, de base 8, é facilmente convertido para o sistema binário (visto que cada três dígitos binários equivalem a um dígito octal) e vice-versa
- O sistema **hexadecimal**, de base 16, é facilmente convertido para o sistema binário (visto que cada quatro dígitos binários equivalem a um dígito hexadecimal) e vice-versa

- A base 2 é importante por podermos definir unidades em termos de bits: ligado ou desligado, sim ou não...
- A consequência é que muitas vezes tratamos de valores baseados em uma potência de 2 para representar grandezas
 - K (kilo): 2^{10} ($\approx 10^3$)
 - M (mega): 2^{20} ($\approx 10^6$)
 - G (giga): 2^{30} ($\approx 10^9$)
 - T (tera): 2^{40} ($\approx 10^{12}$)
 - P (peta): 2^{50} ($\approx 10^{15}$)
- Além do sistema binário, os sistemas octais e hexadecimais são muito comuns dentro de sistemas computacionais
- O sistema **octal**, de base 8, é facilmente convertido para o sistema binário (visto que cada três dígitos binários equivalem a um dígito octal) e vice-versa
- O sistema **hexadecimal**, de base 16, é facilmente convertido para o sistema binário (visto que cada quatro dígitos binários equivalem a um dígito hexadecimal) e vice-versa

- A base 2 é importante por podermos definir unidades em termos de bits: ligado ou desligado, sim ou não...
- A consequência é que muitas vezes tratamos de valores baseados em uma potência de 2 para representar grandezas
 - K (kilo): 2^{10} ($\approx 10^3$)
 - M (mega): 2^{20} ($\approx 10^6$)
 - G (giga): 2^{30} ($\approx 10^9$)
 - T (tera): 2^{40} ($\approx 10^{12}$)
 - P (peta): 2^{50} ($\approx 10^{15}$)
- Além do sistema binário, os sistemas octais e hexadecimais são muito comuns dentro de sistemas computacionais
- O sistema **octal**, de base 8, é facilmente convertido para o sistema binário (visto que cada três dígitos binários equivalem a um dígito octal) e vice-versa
- O sistema **hexadecimal**, de base 16, é facilmente convertido para o sistema binário (visto que cada quatro dígitos binários equivalem a um dígito hexadecimal) e vice-versa

- A base 2 é importante por podermos definir unidades em termos de bits: ligado ou desligado, sim ou não...
- A consequência é que muitas vezes tratamos de valores baseados em uma potência de 2 para representar grandezas
 - K (kilo): 2^{10} ($\approx 10^3$)
 - M (mega): 2^{20} ($\approx 10^6$)
 - G (giga): 2^{30} ($\approx 10^9$)
 - T (tera): 2^{40} ($\approx 10^{12}$)
 - P (peta): 2^{50} ($\approx 10^{15}$)
- Além do sistema binário, os sistemas octais e hexadecimais são muito comuns dentro de sistemas computacionais
- O sistema **octal**, de base 8, é facilmente convertido para o sistema binário (visto que cada três dígitos binários equivalem a um dígito octal) e vice-versa
- O sistema **hexadecimal**, de base 16, é facilmente convertido para o sistema binário (visto que cada quatro dígitos binários equivalem a um dígito hexadecimal) e vice-versa

- A base 2 é importante por podermos definir unidades em termos de bits: ligado ou desligado, sim ou não...
- A consequência é que muitas vezes tratamos de valores baseados em uma potência de 2 para representar grandezas
 - K (kilo): 2^{10} ($\approx 10^3$)
 - M (mega): 2^{20} ($\approx 10^6$)
 - G (giga): 2^{30} ($\approx 10^9$)
 - T (tera): 2^{40} ($\approx 10^{12}$)
 - P (peta): 2^{50} ($\approx 10^{15}$)
- Além do sistema binário, os sistemas octais e hexadecimais são muito comuns dentro de sistemas computacionais
- O sistema **octal**, de base 8, é facilmente convertido para o sistema binário (visto que cada três dígitos binários equivalem a um dígito octal) e vice-versa
- O sistema **hexadecimal**, de base 16, é facilmente convertido para o sistema binário (visto que cada quatro dígitos binários equivalem a um dígito hexadecimal) e vice-versa

- Computadores executam tarefas grandes e complexas através de instruções básicas simples
- Para tal, o funcionamento é organizado através de camadas: o *hardware*, o sistema operacional, e aplicações
- Os usuários do sistema operam diretamente sobre aplicações, as quais são gerenciadas pelo sistema operacional, que abstrai as funcionalidades do *hardware*
- Aplicações são desenvolvidas através de **linguagens de alto nível**: linguagens formais usadas por seres humanos para descrever as funcionalidades desejadas
 - Mas o *hardware* não entende linguagens de alto nível!
- A fim de se executar uma linguagem de alto nível, precisamos de um compilador ou um interpretador, ferramentas usadas para permitir a execução de um programa

- Computadores executam tarefas grandes e complexas através de instruções básicas simples
- Para tal, o funcionamento é organizado através de camadas: o *hardware*, o sistema operacional, e aplicações
- Os usuários do sistema operam diretamente sobre aplicações, as quais são gerenciadas pelo sistema operacional, que abstrai as funcionalidades do *hardware*
- Aplicações são desenvolvidas através de **linguagens de alto nível**: linguagens formais usadas por seres humanos para descrever as funcionalidades desejadas
 - Mas o *hardware* não entende linguagens de alto nível!
- A fim de se executar uma linguagem de alto nível, precisamos de um compilador ou um interpretador, ferramentas usadas para permitir a execução de um programa

- Computadores executam tarefas grandes e complexas através de instruções básicas simples
- Para tal, o funcionamento é organizado através de camadas: o *hardware*, o sistema operacional, e aplicações
- Os usuários do sistema operam diretamente sobre aplicações, as quais são gerenciadas pelo sistema operacional, que abstrai as funcionalidades do *hardware*
- Aplicações são desenvolvidas através de **linguagens de alto nível**: linguagens formais usadas por seres humanos para descrever as funcionalidades desejadas
 - Mas o *hardware* não entende linguagens de alto nível!
- A fim de se executar uma linguagem de alto nível, precisamos de um compilador ou um interpretador, ferramentas usadas para permitir a execução de um programa

- Computadores executam tarefas grandes e complexas através de instruções básicas simples
- Para tal, o funcionamento é organizado através de camadas: o *hardware*, o sistema operacional, e aplicações
- Os usuários do sistema operam diretamente sobre aplicações, as quais são gerenciadas pelo sistema operacional, que abstrai as funcionalidades do *hardware*
- Aplicações são desenvolvidas através de **linguagens de alto nível**: linguagens formais usadas por seres humanos para descrever as funcionalidades desejadas
 - Mas o *hardware* não entende linguagens de alto nível!
- A fim de se executar uma linguagem de alto nível, precisamos de um compilador ou um interpretador, ferramentas usadas para permitir a execução de um programa

- Computadores executam tarefas grandes e complexas através de instruções básicas simples
- Para tal, o funcionamento é organizado através de camadas: o *hardware*, o sistema operacional, e aplicações
- Os usuários do sistema operam diretamente sobre aplicações, as quais são gerenciadas pelo sistema operacional, que abstrai as funcionalidades do *hardware*
- Aplicações são desenvolvidas através de **linguagens de alto nível**: linguagens formais usadas por seres humanos para descrever as funcionalidades desejadas
 - Mas o *hardware* não entende linguagens de alto nível!
- A fim de se executar uma linguagem de alto nível, precisamos de um compilador ou um interpretador, ferramentas usadas para permitir a execução de um programa

- Computadores executam tarefas grandes e complexas através de instruções básicas simples
- Para tal, o funcionamento é organizado através de camadas: o *hardware*, o sistema operacional, e aplicações
- Os usuários do sistema operam diretamente sobre aplicações, as quais são gerenciadas pelo sistema operacional, que abstrai as funcionalidades do *hardware*
- Aplicações são desenvolvidas através de **linguagens de alto nível**: linguagens formais usadas por seres humanos para descrever as funcionalidades desejadas
 - Mas o *hardware* não entende linguagens de alto nível!
- A fim de se executar uma linguagem de alto nível, precisamos de um compilador ou um interpretador, ferramentas usadas para permitir a execução de um programa

Linguagens de programação

- Processadores executam operações básicas descritas em **código de máquina**, que instruem a **unidade de controle** do processador
- As **linguagens de baixo nível** são diretamente mapeáveis para código de máquina, sendo possível usar essas linguagens de montagem para programar um computador
- Linguagens de baixo nível não foram projetadas para serem usadas por seres humanos; linguagens de alto nível incluem abstrações que facilitam a implementação de código
- Um **compilador** é um aplicativo que é capaz de converter um programa escrito em uma linguagem de alto nível em uma linguagem de baixo nível
 - Nesse ponto, outro aplicativo, o **montador**, traduz a linguagem de baixo nível para código de máquina
- Um **interpretador**, entretanto, é um programa que executa diretamente código escrito em uma linguagem de alto nível, sem a necessidade de convertê-lo para código de máquina

Linguagens de programação

- Processadores executam operações básicas descritas em **código de máquina**, que instruem a **unidade de controle** do processador
- As **linguagens de baixo nível** são diretamente mapeáveis para código de máquina, sendo possível usar essas linguagens de montagem para programar um computador
- Linguagens de baixo nível não foram projetadas para serem usadas por seres humanos; linguagens de alto nível incluem abstrações que facilitam a implementação de código
- Um **compilador** é um aplicativo que é capaz de converter um programa escrito em uma linguagem de alto nível em uma linguagem de baixo nível
 - Nesse ponto, outro aplicativo, o **montador**, traduz a linguagem de baixo nível para código de máquina
- Um **interpretador**, entretanto, é um programa que executa diretamente código escrito em uma linguagem de alto nível, sem a necessidade de convertê-lo para código de máquina

Linguagens de programação

- Processadores executam operações básicas descritas em **código de máquina**, que instruem a **unidade de controle** do processador
- As **linguagens de baixo nível** são diretamente mapeáveis para código de máquina, sendo possível usar essas linguagens de montagem para programar um computador
- Linguagens de baixo nível não foram projetadas para serem usadas por seres humanos; linguagens de alto nível incluem abstrações que facilitam a implementação de código
- Um **compilador** é um aplicativo que é capaz de converter um programa escrito em uma linguagem de alto nível em uma linguagem de baixo nível
 - Nesse ponto, outro aplicativo, o **montador**, traduz a linguagem de baixo nível para código de máquina
- Um **interpretador**, entretanto, é um programa que executa diretamente código escrito em uma linguagem de alto nível, sem a necessidade de convertê-lo para código de máquina

Linguagens de programação

- Processadores executam operações básicas descritas em **código de máquina**, que instruem a **unidade de controle** do processador
- As **linguagens de baixo nível** são diretamente mapeáveis para código de máquina, sendo possível usar essas linguagens de montagem para programar um computador
- Linguagens de baixo nível não foram projetadas para serem usadas por seres humanos; linguagens de alto nível incluem abstrações que facilitam a implementação de código
- Um **compilador** é um aplicativo que é capaz de converter um programa escrito em uma linguagem de alto nível em uma linguagem de baixo nível
 - Nesse ponto, outro aplicativo, o **montador**, traduz a linguagem de baixo nível para código de máquina
- Um **interpretador**, entretanto, é um programa que executa diretamente código escrito em uma linguagem de alto nível, sem a necessidade de convertê-lo para código de máquina

Linguagens de programação

- Processadores executam operações básicas descritas em **código de máquina**, que instruem a **unidade de controle** do processador
- As **linguagens de baixo nível** são diretamente mapeáveis para código de máquina, sendo possível usar essas linguagens de montagem para programar um computador
- Linguagens de baixo nível não foram projetadas para serem usadas por seres humanos; linguagens de alto nível incluem abstrações que facilitam a implementação de código
- Um **compilador** é um aplicativo que é capaz de converter um programa escrito em uma linguagem de alto nível em uma linguagem de baixo nível
 - Nesse ponto, outro aplicativo, o **montador**, traduz a linguagem de baixo nível para código de máquina
- Um **interpretador**, entretanto, é um programa que executa diretamente código escrito em uma linguagem de alto nível, sem a necessidade de convertê-lo para código de máquina

Linguagens de programação

- Processadores executam operações básicas descritas em **código de máquina**, que instruem a **unidade de controle** do processador
- As **linguagens de baixo nível** são diretamente mapeáveis para código de máquina, sendo possível usar essas linguagens de montagem para programar um computador
- Linguagens de baixo nível não foram projetadas para serem usadas por seres humanos; linguagens de alto nível incluem abstrações que facilitam a implementação de código
- Um **compilador** é um aplicativo que é capaz de converter um programa escrito em uma linguagem de alto nível em uma linguagem de baixo nível
 - Nesse ponto, outro aplicativo, o **montador**, traduz a linguagem de baixo nível para código de máquina
- Um **interpretador**, entretanto, é um programa que executa diretamente código escrito em uma linguagem de alto nível, sem a necessidade de convertê-lo para código de máquina

- Embora seja possível executar programas direto na máquina (algo conhecido como *baremetal*, ou "direto no metal"), aplicativos costumam ser executados com o auxílio de um **sistema operacional**
- Sistemas operacionais abstraem operações repetitivas, como o controle do *hardware*, facilitando portabilidade e diminuindo a carga necessária para aplicações
- Além disso, os sistemas operacionais gerenciam múltiplos usuários e múltiplas tarefas, permitindo a execução de vários aplicativos simultaneamente
- Um sistema operacional muito popular no passado foi o UNIX, que foi usado de base para diversos sistemas operacionais modernos, como Linux, Solaris e os sistemas BSD (FreeBSD, NetBSD, OpenBSD e macOS)

- Embora seja possível executar programas direto na máquina (algo conhecido como *baremetal*, ou "direto no metal"), aplicativos costumam ser executados com o auxílio de um **sistema operacional**
- Sistemas operacionais abstraem operações repetitivas, como o controle do *hardware*, facilitando portabilidade e diminuindo a carga necessária para aplicações
- Além disso, os sistemas operacionais gerenciam múltiplos usuários e múltiplas tarefas, permitindo a execução de vários aplicativos simultaneamente
- Um sistema operacional muito popular no passado foi o UNIX, que foi usado de base para diversos sistemas operacionais modernos, como Linux, Solaris e os sistemas BSD (FreeBSD, NetBSD, OpenBSD e macOS)

- Embora seja possível executar programas direto na máquina (algo conhecido como *baremetal*, ou "direto no metal"), aplicativos costumam ser executados com o auxílio de um **sistema operacional**
- Sistemas operacionais abstraem operações repetitivas, como o controle do *hardware*, facilitando portabilidade e diminuindo a carga necessária para aplicações
- Além disso, os sistemas operacionais gerenciam múltiplos usuários e múltiplas tarefas, permitindo a execução de vários aplicativos simultaneamente
- Um sistema operacional muito popular no passado foi o UNIX, que foi usado de base para diversos sistemas operacionais modernos, como Linux, Solaris e os sistemas BSD (FreeBSD, NetBSD, OpenBSD e macOS)

- Embora seja possível executar programas direto na máquina (algo conhecido como *baremetal*, ou "direto no metal"), aplicativos costumam ser executados com o auxílio de um **sistema operacional**
- Sistemas operacionais abstraem operações repetitivas, como o controle do *hardware*, facilitando portabilidade e diminuindo a carga necessária para aplicações
- Além disso, os sistemas operacionais gerenciam múltiplos usuários e múltiplas tarefas, permitindo a execução de vários aplicativos simultaneamente
- Um sistema operacional muito popular no passado foi o UNIX, que foi usado de base para diversos sistemas operacionais modernos, como Linux, Solaris e os sistemas BSD (FreeBSD, NetBSD, OpenBSD e macOS)