

Disciplina de Linguagens de Programação
1ª Lista de Exercícios
Curso de Engenharia de Computação
UEMG Ituiutaba
<https://github.com/mauro-hemerly/UEMG-2019-1>

Assunto: Paradigmas de Programação, Compilação, Interpretação

1. Programar o computador significa **dar ordens a ele**, as quais são executadas seqüencialmente. Em tal paradigma, **representar** a solução de um problema para ser resolvido pelo computador envolve escrever uma série de ações que, se executadas seqüencialmente, levam à solução. Trata-se do paradigma:
 - (a) Lógico.
 - (b) Funcional.
 - (c) Orientado a Objetos.
 - (d) Procedural ou imperativo.
 - (e) Orientado a Símbolos.
2. A respeito das características dos paradigmas e das linguagens de programação, assinale a opção correta.
 - (a) As linguagens do paradigma de programação funcional, como o Prolog, não apresentam grandes restrições ao uso de estruturas de controle (goto), o que pode reduzir a legibilidade dos programas construídos sem limitações.
 - (b) No paradigma lógico, que é suportado por linguagens de programação não imperativas, como o Lisp, os programas gerados são embasados em funções matemáticas.
 - (c) O paradigma de programação orientado a objetos reúne linguagens, como o C++, que são declarativas, isto é, o foco está na especificação dos resultados desejados ao invés dos procedimentos para produzi-los.
 - (d) As linguagens imperativas, como o Pascal, são voltadas para a especificação da solução do problema, por meio do detalhamento do algoritmo e da especificação da ordem das instruções.
 - (e) As linguagens procedimentais que dominaram o mercado antes da programação estruturada, tal como o COBOL, caracterizaram-se por utilizar amplamente os tipos abstratos de dados.
3. Com relação aos softwares básicos envolvidos no processo de transformação de um programa escrito em linguagem de alto nível para um código executável ou interpretado, assinale a alternativa correta.
 - (a) No processo de análise semântica da compilação, é verificada a consistência dos tipos de variáveis envolvidas em operações aritméticas.
 - (b) Os montadores buscam referências externas em rotinas de bibliotecas e inserem-nas no código-objeto.
 - (c) Compilação e montagem são processos semelhantes; contudo, o processo de montagem é bem mais complexo.
 - (d) Os link-editores são usados para resolver as referências aos símbolos internos de um programa, produzindo um código-objeto simplificado.
 - (e) No processo de compilação, o analisador léxico e o sintático estão preocupados com a semântica e os significados dos programas como um todo.
4. No programa em linguagem de alto nível, os interpretadores executam os passos definidos para cada instrução e produzem o mesmo resultado que o do programa compilado. Entretanto, a execução de um programa em linguagem de alto nível com o uso de interpretadores é mais lenta que a execução de um programa compilado, uma vez que precisa examinar cada instrução no programa-fonte, à medida que ela ocorre, e desviar para a rotina que executa a instrução.
 - (a) Certo
 - (b) Errado
5. Na programação empregando uma linguagem de alto nível, a utilização de um compilador implica o uso de um ligador e de um carregador para a correta execução do programa; por outro lado, a utilização de um interpretador, que simula a existência de um processador cujas instruções são aquelas da linguagem de alto nível empregada, torna desnecessárias as etapas de ligação e carga.

- (a) Certo
(b) Errado
6. Durante a compilação de um código-fonte, a fase do compilador que é responsável por produzir uma sequência de tokens é a
- (a) análise léxica.
(b) análise semântica.
(c) análise sintática.
(d) geração de código executável.
(e) verificação de tipos.
7. Em relação a compiladores, ligadores (link-editores) e interpretadores, é correto afirmar que
- (a) uma função importante do tradutor é a realocação.
(b) o link-editor é o utilitário responsável por carregar, na memória principal, um programa para ser executado.
(c) o depurador é o utilitário responsável por gerar, a partir de um ou mais módulos-objeto, um único programa executável.
(d) o link-editor é o utilitário responsável por gerar, a partir de um ou mais módulos-objeto, um único programa executável.
(e) a grande vantagem do uso de interpretadores é o tempo gasto na tradução de suas instruções sempre que é executado.
8. O interpretador
- (a) tem, como vantagem, o tempo gasto toda vez que for executado.
(b) é considerado um tradutor que gera módulo-objeto.
(c) é o mesmo que loader.
(d) é considerado um tradutor que não gera módulo-objeto.
(e) tem, como desvantagem, permitir a implementação de dados dinâmicos.
9. Tradutores de linguagem são programas que traduzem os programas-fonte escritos em linguagem de alto nível para código de máquina. Com relação a tradutores, assinale a opção correta.
- (a) Compiladores são projetados para um tipo específico de hardware e de sistema operacional.
(b) A interpretação nada mais é do que uma compilação cruzada.
(c) Toda linguagem orientada a objetos pura é compilada.
(d) Java é uma linguagem fundamentalmente interpretada.
(e) A linguagem Delphi pode ser tanto interpretada quanto compilada.
10. O compilador, em contraste com o montador, opera sobre uma linguagem de alto nível, enquanto o montador opera sobre uma linguagem de montagem.
- (a) Certo
(b) Errado
11. Um analista está responsável pelo projeto de um compilador para uma nova linguagem que sua empresa está desenvolvendo. Este compilador deverá receber como entrada o código fonte do programa e gerar como saída o código de linguagem de máquina para um determinado processador e sistema operacional. O analista sabe que um processo de compilação é dividido em fases e que seu compilador deverá possuir todas as fases de um típico processo de compilação mesmo que elas NÃO sejam obrigatórias em outros compiladores. A fase que não faz parte de um processo de compilação e que, portanto, não seria necessária no projeto do analista é a
- (a) análise léxica.
(b) análise sintática.
(c) geração de código intermediário.
(d) ligação dos módulos.
(e) otimização de código.
12. Em um compilador, o analisador léxico
- (a) cria uma estrutura de dados.
(b) cria o código objeto.
(c) cria uma sequência de símbolos.
(d) cria uma gramática livre de contexto.
(e) cria um código intermediário.
13. O utilitário responsável por gerar, a partir de um programa escrito em linguagem de alto nível, um programa em linguagem de máquina não executável é o
- (a) montador
(b) interpretador
(c) compilador
(d) linker
(e) loader
14. Os desenvolvedores (programadores) utilizam ambientes de desenvolvimento compostos por ferramentas que auxiliam na produção de softwares. Dentre essas ferramentas, o compilador é responsável por :
- (a) gerar um programa em linguagem de máquina não executável a partir de um programa escrito em linguagem de alto nível.
(b) gerar um único programa executável a partir dos módulos objetos.
(c) traduzir um programa-fonte em linguagem de montagem em um módulo objeto não executável.

- (d) traduzir e executar imediatamente cada uma das instruções do programa-fonte escrito em linguagem de alto nível durante a execução do programa.
- (e) carregar na memória principal um programa para ser executado.
15. Compiladores, montadores e ligadores são softwares que convertem programas de um formato de código (entrada) para um mais próximo ao formato executável compreendido pela máquina (saída). Os ligadores geram como saída
- (a) programas objeto.
- (b) bibliotecas de programas semicompilados.
- (c) programas em formato bytecode.
- (d) programas executáveis em linguagem de máquina.
- (e) programas compilados em código intermediário, mas ainda não executáveis.
16. A compilação é o processo de análise de um programa escrito em linguagem de alto nível, denominado programa-fonte, e sua conversão em um programa equivalente, escrito em linguagem binária de máquina, denominado programa-objeto.
- (a) Certo
- (b) Errado
17. Rotinas em linguagem de montagem são bastante utilizadas em sistemas de trens metropolitanos para controlar atividades críticas. As afirmativas **I** e **II** estão corretamente relacionadas às linguagens de montagem, montadores e suas aplicações **APENAS** em:
- (a) I. O termo assembly refere-se à linguagem de programação, que é também designada por linguagem de macro, uma vez que se encontra intimamente relacionada com a macroprogramação do processador a que se destina.
- II. Cada processador de cada fabricante (Intel, AMD, Motorola etc), tem o seu próprio assembly, já que cada um tem estrutura interna diferente, portanto, não há um termo assembly que se aplica a todos eles, mas há linguagem assemblyIntel, assemblyAMD etc.
- (b) I. É muito complicado para os programadores humanos escreverem programas usando a linguagem pura do processador ou a linguagem de máquina, que é constituída por um conjunto mais ou menos extenso de bits.
- II. Os programas executáveis gerados por um assembler são do tipo .EXE, arquivos em linguagem de baixo nível, que devem ser carregados e executados a partir do endereço 10FFh. Assim, todos os programas assembly devem começar com a instrução org 10FFh, seguidos de três seções (text, data e code).
- (c) I. O termo assembler refere-se a um programa que permite facilitar o trabalho com a linguagem assembly, fazendo com que esta se assemelhe um pouco mais com os mnemônicos de uma linguagem de alto-nível.
- II. As instruções do assembler são traduzidas para os conjuntos de bits que constituem as instruções do processador. O assembly executa um conjunto de ações necessárias para que o programa assembler possa ser executado, escondendo essas operações do programador.
- (d) I. A programação em assembly permite escrever programas que executam muito mais rapidamente que programas escritos em linguagens de alto-nível. Os compiladores ou interpretadores destas linguagens, ao traduzirem as suas instruções para assembly, fazem-no de forma menos eficiente, gerando mais instruções assembly do que um programador humano pode conseguir se programar diretamente em baixo-nível.
- II. O assembly proporciona um maior controle sobre os componentes de hardware, em particular do processador, permitindo usar melhor as suas funcionalidades e capacidades.
- (e) I. Todas as linguagens de alto-nível permitem que se possam incluir instruções assembly entre as instruções da própria linguagem, visando acelerar rotinas críticas do programa. Isso porque as linguagens de alto-nível impedem o acesso a certos componentes de hardware, a fim de evitar que possam ser desencadeadas ações potencialmente perigosas.
- II. Existem vários assemblers, dentre os quais os mais conhecidos são o TASM (da Intel), o NASM (da Microsoft), que são propriedade dos respectivos fabricantes, e o MASM, que é de uso livre.
18. Para que a máquina entenda os comandos de um programa desenvolvido na linguagem C, o programa deve ser compilado e traduzido para a linguagem da máquina.
- (a) Certo
- (b) Errado
19. Um programa em linguagem de alto nível deve ser compilado para se transformar em linguagem de montagem e, posteriormente, montado para se transformar em linguagem de máquina.
- (a) Certo
- (b) Errado
20. Interpretador é um tradutor de linguagem que executa o programa fonte de imediato, em vez de gerar um código objeto a ser executado após o término da tradução, enquanto o compilador recebe um programa fonte e produz programa equivalente na linguagem alvo. No caso da linguagem Java, os processadores combinam compilação e interpretação.

- (a) Certo
 - (b) Errado
21. De uma forma geral, um compilador é um programa que recebe como entrada o texto de um programa (fonte) em uma certa linguagem e produz como saída outro texto de programa em uma linguagem diferente da utilizada no fonte. Para isso, compiladores são estruturados internamente em uma série de módulos que produzem, em sequência, representações intermediárias do programa fonte ou de suas partes.

A quebra do texto do programa fonte em unidades básicas da linguagem denominadas de tokens é realizada pelo módulo denominado

- (a) Interpretador
 - (b) Analisador Léxico
 - (c) Analisador Sintático
 - (d) Tratador de Contexto
 - (e) Gerador de Código.
22. A programação em linguagens de alto nível pressupõe a utilização de alguma ferramenta que permita a transformação do programa original em instruções que o processador da máquina seja capaz de executar.
- Compiladores e Interpretadores são dois desses tipos de programa, que têm como principal diferença o fato de que os compiladores
- (a) geram, a partir do código fonte, um novo programa, chamado programa executável, que só então poderá ser executado; já os interpretadores fornecem um ambiente que permite a leitura e tradução direta do programa fonte, permitindo sua execução sem a necessidade de gerar um novo programa executável.
 - (b) são capazes de operar em diversos ambientes, por exemplo, em máquinas virtuais, o que não é possível com interpretadores, que são feitos para uma determinada plataforma.
 - (c) são dependentes da linguagem utilizada, enquanto os interpretadores podem ser utilizados para qualquer linguagem de programação, já que não precisam gerar código e sim traduzir o código fonte.
 - (d) são mais confiáveis, pois o resultado final fica armazenado em um novo programa; já os interpretadores podem ocasionar a perda de informações.
 - (e) são mais genéricos do que interpretadores, porque podem ser utilizados para qualquer tipo de linguagem de programação, ao contrário dos interpretadores, que só podem ser construídos para tipos específicos de linguagens.