

Cap.1 –Introdução às Linguagens de Programação e Compiladores

Nome: Isaque Hollanda Gonçalves

Curso: GES - 166

1.O que são os Paradigmas das Linguagens de Programação e como são classificados?Explique.

Os paradigmas de Lps são o meio de dividir os diferentes tipos. Um exemplo disso seria a divisão entre Linguagens Imperativas e Declarativas.

2.Quanto a geração das Linguagens de Programação, como elas podem ser classificadas? Cite um exemplo de linguagem pertencente a cada uma delas.

As gerações de LPs são: 1ª Geração, que corresponde à linguagem binária; a 2ª geração, que corresponde à linguagem Assembly, que é específica para cada CPU; 3ª Geração corresponde às LPs de alto nível, tais como Python, que se aproximam da linguagem humana. Na 4ª Geração temos as linguagens para fins específicos, à exemplo do SQL, utilizado em Banco de Dados. Já a 5ª Geração é caracterizada pelas linguagens com foco em IA e maior base matemática, tais como Prolog.

3.Oque são Tradutores de Linguagens de Programação? Quais os tipos mais populares e suas principais diferenças?

Os tradutores de linguagens de programação, basicamente, podem ser divididos entre Compiladores e Interpretadores. O primeiro possui a característica de analisar o código inteiro e, logo após, traduz em linguagem de máquina(Assembly). Já o Interpretador não tem um modelo claramente definido, porém a maior diferença é sua análise do código que é feita linha a linha, sem otimização, traduzindo, o que resulta em uma menor velocidade comparado ao Compilador.

4.O que são Bibliotecas (libraries ou simplesmente libs)? Todas elas podem ser facilmente adicionadas ou removidas pelos desenvolvedores em seus códigos?Explique.

As bibliotecas possuem dois principais tipos, sendo elas: Estáticas, que são adicionadas antes da fase de compilação e são adicionados pelo desenvolvedor. Outro tipo são as dinâmicas, das quais surgem durante ou após compilar. Estas, por sua vez, não podem ser removidas, caso contrário geram mal funcionamento no software.

5.Que estratégias podem ser usadas pelos Compiladores ao detectarem possíveis erros durante o processo de Compilação? Explique.

Os Compiladores utilizam de duas decisões: Abortar o processo ou então tentam recuperar, tentando ignorar tal erro.

6.Em um processo de Compilação, quais são as fases geralmente envolvidas? Explique sucintamente o papel de cada uma delas.

As fases que envolvem tal processo são: Análise Léxica, responsável por identificar os Tokens do código(palavras-chave); Análise Sintática, caracterizada por analisar a validade das operações entre os Tokens; Análise Semântica, que verifica a compatibilidade das operações, com, por exemplo, somar Float com Int. Após isso, é gerado um código intermediário, ele é otimizado, removendo variáveis não utilizadas, por exemplo; e transformado em Assembly, finalizando o processo.

7.O que você entende por Compilação Cruzada? Dê exemplos de dispositivos que podem/devem utilizar desta técnica para executar seus programas.

Compilação Cruzada é definida por casos em que deseja-se desenvolver para outros dispositivos, tais como Embarcados. Nesse caso, utilizando um PC, produz um código que é compilado para o hardware do aparelho desejado, que é diferente do hardware fonte.

8.O que você entende por Compiladores Híbridos e qual o “pulo do gato” desta abordagem? Explique.

Compilador Híbrido é uma técnica que utiliza de Compiladores e Interpretadores, como o caso do Java. Nesse processo o código é compilado para uma VM. Após isso, a VM gera um código Assembly através de um Interpretador.

9.Além do processo de tradução de Linguagens de Programação, cite duas outras áreas que a Teoria de Compiladores pode ser utilizada para resolver problemas computacionais.

A Teoria de Compiladores pode ser aplicada em casos como CyberSegurança, cifrando uma mensagem. Pode também ser utilizada em detecções de padrões, como na área de IA.

10.Existe a Linguagem de Programação perfeita?Para ajudar a responder esta pergunta, analise as Propriedades Desejáveis das Linguagens de Programação e dê sua opinião.

Não existe uma LP perfeita, pois nenhuma Linguagem é capaz de satisfazer as 9 propriedades desejáveis. O que deve ser feito é analisar o objetivo de um projeto e verificar a que melhor se adapta aos requisitos necessários.