

Lista de Exercício - Exercício II

Rafael Gonçalves de Oliveira Viana

2º semestre de 2017

1. Qual a diferença entre compilador e interpretador?

R: Para entendermos suas diferenças devemos entender para que serve um compilador e um interpretador.

Os dois são definições dada pela arquitetura da linguagem de programação.

A compilação é o processo de análise e possivelmente transformação do código fonte em código alvo, ou seja, o uso (execução, por exemplo) se dá em processo separado posterior.

Quando utilizamos uma linguagem compilada, o desenvolvedor deve executar a compilação (Tradução da linguagem atual pra linguagem de máquina), para poder testar a aplicação, porém esse processo de tradução é custoso, ocupando ciclos de CPU e espaço em memória.

Como esse método visa a geração do código fonte atual direto para uma linguagem de máquina, uma vez compilado um código, esse mesmo não precisa ser recompilado para ser executado novamente na máquina, ganhando desempenho pelo baixo custo computacional para executor o arquivo diretamente em linguagem de máquina.

Alguns programas quando compilado pode demora horas para finalizar sua compilação, um exemplo seria o Kernel do Linux.

A interpretação ocorre quando o uso (comummente a execução) do código se dá junto à análise do mesmo.

Enquanto o compilador tem que ser executado pelo desenvolvedor o interpretador lê uma linha de cada vez e executa o comando, sendo assim a tradução da linguagem de alto nível para de baixo nível utilizando o interpretador (linguagem intermediária) acontece em tempo real

Então aplicações que rodam interpretadas precisam do código fonte enquanto que as compiladas só precisam do código alvo para funcionar.