

Compiler vs Interpreter

Presented by Cavad, Xəyal, Elşən

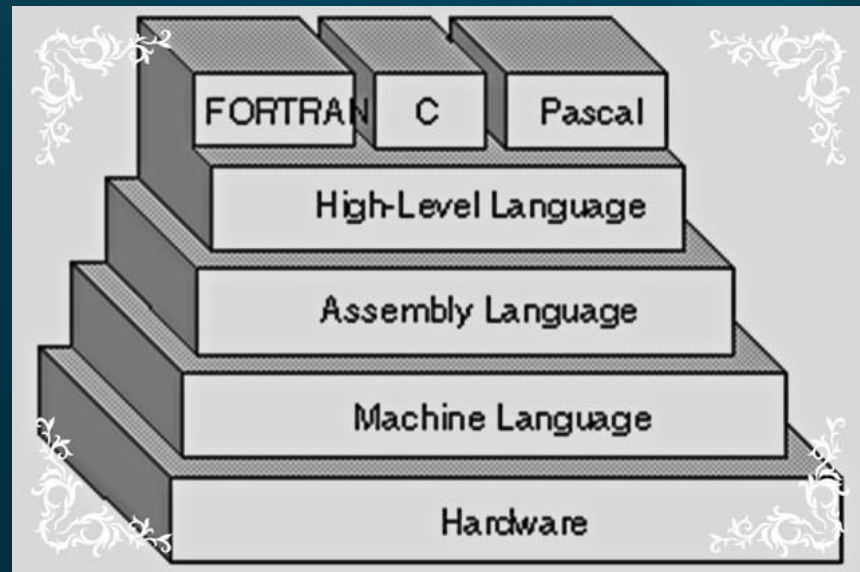




Giriş

Kompüterlər insan dilini başa düşmür. Faktiki olaraq ən aşağı "level"-də onlar yalnız əməliyyat kodlarını (op codes) təmsil edən rəqəm ardıcılıqlarını anlayır. Digər tərəfdən təbii ki, insanlar üçün (developer-lər) proqramları "op codes" şərtlərində yazmaq çətin və zaman tələb edən olardı. Beləliklə, proqramlaşdırmanı insanlar üçün daha asan hala gətirmək məqsədilə proqramlışdırma dilləri ixtira olundu.

Biz bu gün proqramı "a high-level language" şəklində sinifləndirdiyimiz dillərdə yazırıq. Və bu kod qaynaq kod ("source code") adlanır. Və kompüter bu high-level dili başa düşmür məqbul olaraq. Burda da köməyimizə source code-u machine code-a çevirmək üçün "compiler" və "interpreter" çatır. Hər ikisi high-level dildə yazılmış proqramı maşın dilinə çevirmək üçün istifadə olunur



Interpreter



A B C D E F G H I J K L M N O P Q R
S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9
* / > < & # N º ? ! @ % = + - \$ € £ (. ,)

VS

Compiler



A B C D E F G H I J K L M N O P Q R
S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9
* / > < & # N º ? ! @ % = + - \$ € £ (. ,)

Compiler

Compiler bir baxıma 2 fərqli dilə mənsub olan insanın arasındakı tərcümə rolunu oynuyur. Compiler proqramçıların ,proqram dilindən istifadə edərək yazdıqları kodu kompüterin başa düşəcəyi şəkildə ,yəni 0 ve 1 lere çevirən proqramdır.

- +Compiler`ler çoxlu proqramlama dillərinin işləməsinə kömək edir.
- +Compiler`ler olmasaydı proqramçılar kodları maşın dilində yazmağa və daha çox əziyyət çəkərək kodları daha gec yazmalı olacaqdılar.

Compiler fərqli dillərdə olan kodları kompüterin başa düşəcəyi kodlara çevirməyə kömək edən proqramlardır.

Compiler`ler executable code`lar (her hansı komputer işini həll etmək üçün istifadə olunan kod) yaradır.

- +Compiler hemde kodları üst səviyyədə olan proqramlama dilindən - alt səviyyədə olan proqramlama dilinə çevirir.Məsələn:

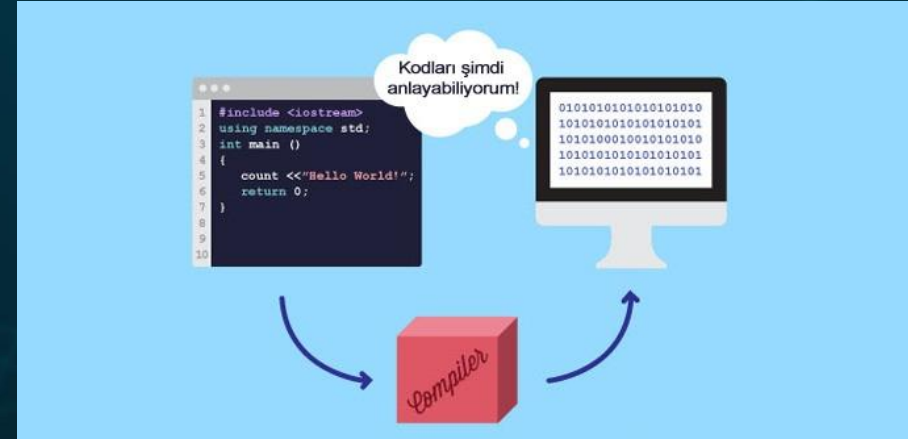
C dili ilə yazılan kodu compiler vasitəsi ilə maşın dili olan assembly ve yaxud alt seviyyə dillərinə çevirə bilər.

Compiler işləmə metodu- tərcümə eliyərkən kodlarda olan xətləri tapıb aralarndakı ünsiyətin problemlı olmaması üçün proqramçıya bildirir.

Bu da compilerin müsbət roludur ki,kodları xətasız şəkildə alt proqram dillərinə uyğun çevirir.

Məsələn, C diliylə kod yazdıq, amma bu kodu maşın dilinə çevirib kompüterə göndərməy lazımdı.Kompüterimiz C dilini birbaşa bilmədiyi üçün C dilini Pascal dilinə çevirəcək.

Bu çevirmədə olan xətləridə bizə bildirecək və compilerin çevirdiyi kod əsasında xətasız şəkildə proqramın işləməsini təmin eliyəcək.

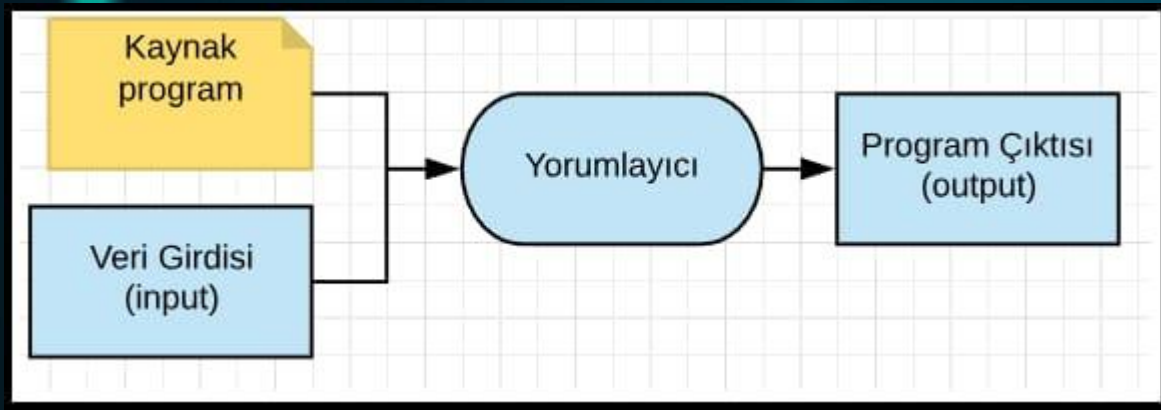


İnterprter

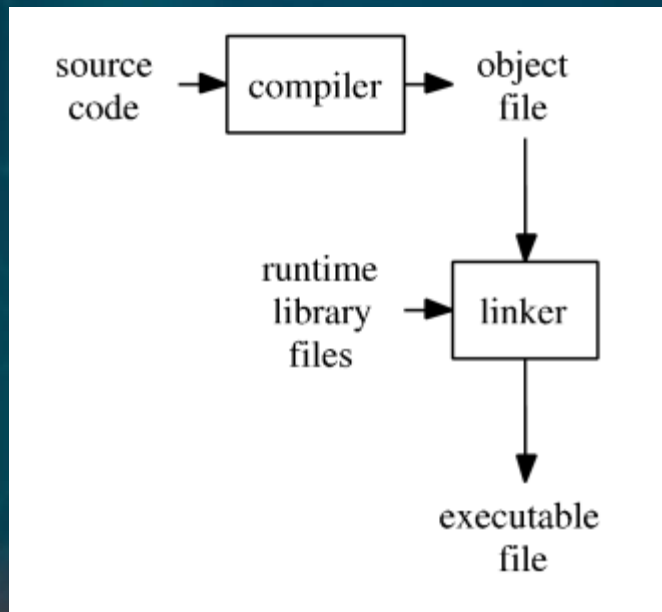
İnterpeter proqramlar yuksek seviyyeli bir dil ile yazilmis proqrami komputerin basa duseceyi dile yeni masin diline (0-1 lere) cevirir.İnterpreter olmasa idi proqrami masin dilinde yazmaga mecbur olardi buda cetin olardi.İnterpreter compilerden ferqli olaraq kodu setir-setir cevirir.Yeni her setiri sirasi gelende tercume edir.Her hansı bir setirde error varsa hemen setire catana kimi kod isleyir ve error olan setire catanda ise dayanir.

Meselen python ruby js interpreter dillerdir

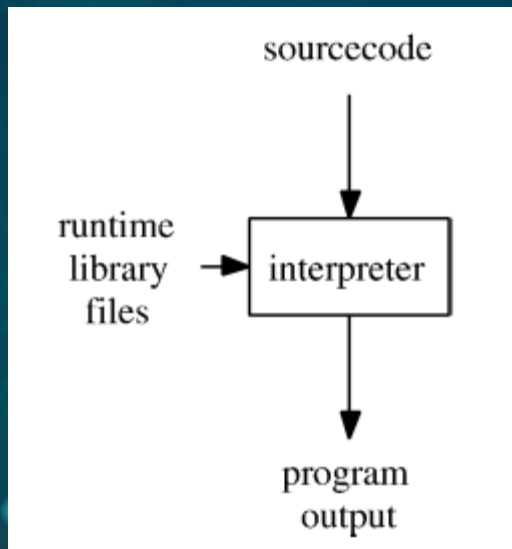
- +Compilerin eksine qaynaq kodunu oxumaga vaxt getmir
- +Error cixan setire kimi kod isleyir
- kodu isletme vaxti daha yavasdir



Fərqlər



Compiler source codu alır və machine language module-a (object file) çevirir. Başqa bir spesifik program olan "linker" bu object file-ı daha öncə compile olunmuş object files (runtime-modules) ilə birləşdirərək çalışdırıla bilən (executable file) yaradır.



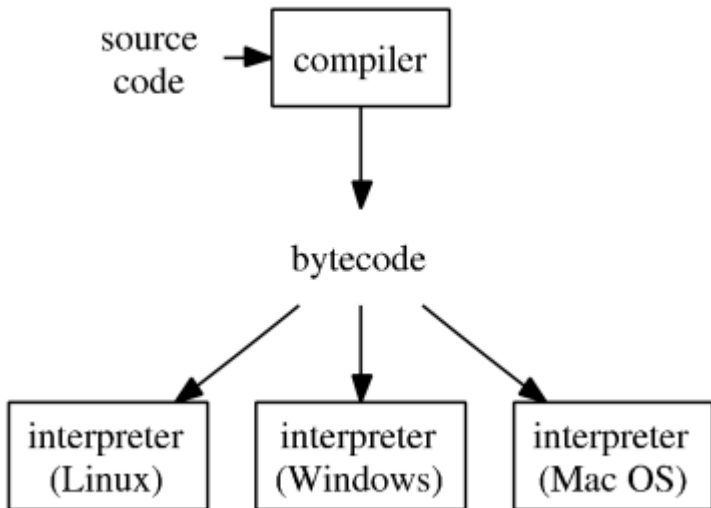
İnterpreter-də isə source code-u executable file yaradılmadan çevirmək əvəzinə interpreter source code-u program işləyərkən machine language-ə çevirir.

İnterpreter bütün runtime zamanında source code-u çevirir, runtime libraries ilə birləşdirir və machine instructions-u çalışdırır. Compiler-dan fərqli olaraq

öncədən compile olunmuş program yoxdur çalışdırmaq üçün. Burda interpreterdə program hər çalışanda bu proseslər təkrarlandığı üçün sürət fərqi var compiler ilə.

Amma bunun ona gətirdiyi müsbət bir tərəf interpreter ilə çevrilən dillər platform independent dir.

Əlavə



The Java compiler (javac) mənbə kodunu bayt koduna (bytecode) çevirir. Bytecode bir növ orta maşın dilidir. Bu bayt kodu faylı (.sınıf faylı) bu platforma üçün Java tərcüməçisindən (java) istifadə etməklə istənilən əməliyyat sistemində işlədilə bilər. Tərcüməçi Virtual Maşın adlanır. Beləliklə, Java Virtual Maşın proqramlaşdırma dilinin bir nümunəsidir. Virtual maşın dilləri tərtib edilmiş və təfsir edilən dillər arasında kompromis olmaq üçün yaradılmışdır. İdeal şəraitdə, virtual maşın dili proqramları tərtib edilmiş dil proqramlarına nisbətən daha sürətli işləyir, lakin təfsir olunan dil proqramlarının platforma müstəqilliyinə malikdir.

Diqqətiniz üçün təşəkkürlər!

