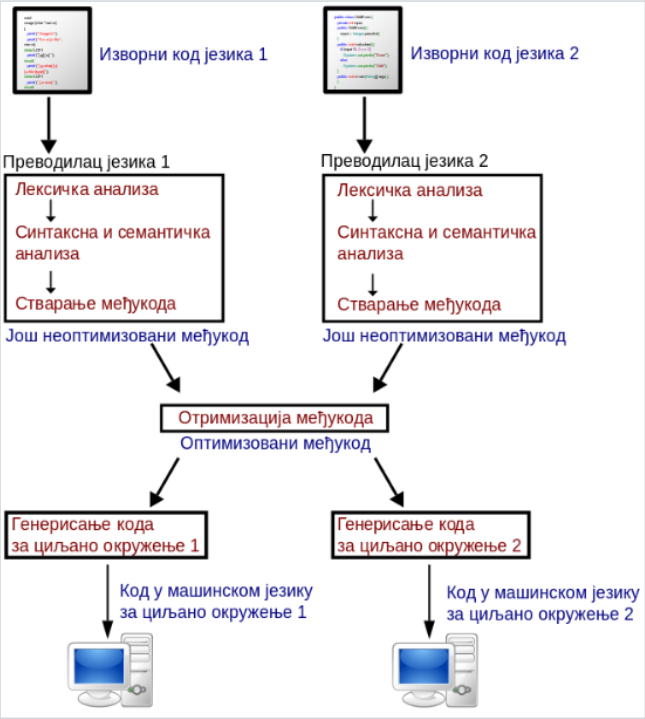
Програмски преводиоци

Програмски преводилац (compiler) чини низ програма који трансформишу код једног програмског језика у други програмски језик. Код који се преводи се назива изворни код (source code) а добијени код је машински код.
Циљ рада преводиоца је најчешће прављење извршног програма.
У новије време користи се израз **компилатор** који означава програмски преводилац са вишег програмског језика на нижи програмски језик.
Постоје и **декомпилатори** који преводе са нижег програмског језика на виши програмски језик.
Програмски преводилац који преводи са једног вишег на други виши програмски језик се назива **конвертор**.
На слици се виде фазе превођења програма.

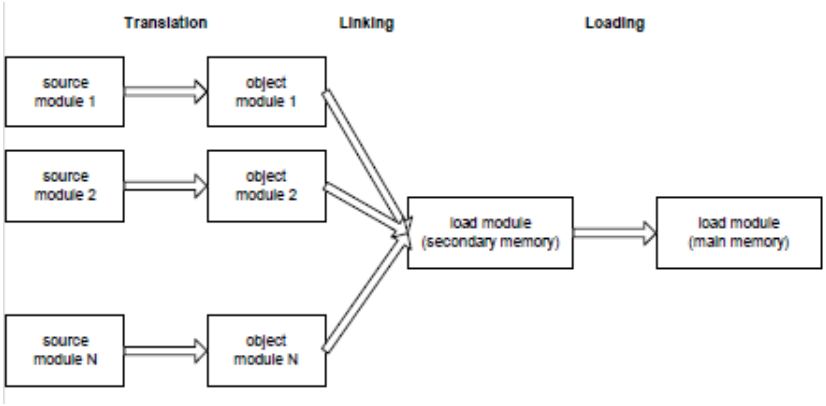


Лексичка анализа дели изворни код на мање делове – токене. Сваки токен је најмања јединица језика (кључна реч, идентификатор, име симбола).
Синтаксна анализа обухвата анализу секвенце токена и испитивање граматичке исправности кода.
Током **семантичке анализе**, преводилац гради табелу симбола и проверава тип података и повезивање референци са промењивима и функцијама.

Линекери су програми који узимају један или више објеката створених помоћу компајлера и састављa их у један извршни егзекутабилан програм. Повезивање које се при томе обавља може бити статичко (сви објекти су повезани у једну датотеку) или динамичко (повезивање појединих објеката се врши у тренутку самог извршења програма).

Лоадери су део оперативног система који учитавају програме и библиотеке у радну меморију пре њиховог извршавања. Учитавање се састоји у читању садржаја извршног фајла који садржи инструкције које ће се извести.

Пример лоадера у ОС Windows 7 (LdrInitializeThunk је функција унутар ntdll.dll):



1. иницијализује саму себе унутар dll библиотеке
2. врши валидацију извршног фајла којег треба да учита
3. креира heap (ограничену меморијску област којој се приступа као целини) преко функције RtlCreateHeap
4. алоцирање промењиве блок и пута до њега (path block)
5. додавање извршног фајла у листу објектних модула
6. учитавање Karnel32.dll библиотеке за могућност рада са додатним функцијама (BaseThreadInitThunk)
7. учитавање иморта извршних фајлова (динамичке библиотеке)
8. у оквиру дибаг фазе, ствара системски breakpoint
9. иницијализује dll библиотеке
10. сакупљање смећа (garbage collection)
11. позива NtContinue и стартује извршни фајл