DASTURLASHDA TRANSLYATORLARDAN FOYDALANISH

[1] Qulmatov Qurvonali Zokirali o'g'li
[2]Anvarjonov Boburjon
[1,2]Muhammad al-Xorazmiy nomidagi Toshkent axborot texnologiyalari universiteti talabasi

https://doi.org/10.5281/zenodo.7983711

Annotatsiya. *Dasturlash sohasida til o'rganuvchilar uchun tranlyatorlarning qanday ishlashi va ularning turlari haqida umumiy ma'lumot. Translyatorlarning dastrulash sohalarida qo'llanilishi va o'zaro farqlari.*

Kalit so'zlar. *Kompilyator, interpretator, kompilatsiya, interpretatsiya, translyator.*

KIRISH.

Axborot texnologiyalari sohasida dasturlash tillari orqali dasturiy ilovalar ishlab chiqishda translyatorlarning o'rni juda muhimdir. Yuqori darajali yoki quyi darajali dasturlash tillarida yozilgan kodlarni mashina tiliga o'tkazish uchun ishlatiladigan programmalar — *translyatorlar* deb ataladi. Translyatorlarning uch xil turi mavjud:

- Assembler
- Kompilyator
- Interpretator

Assembler bu quyi darajali dasturlash tillarida yozilgan kodni mashina tiliga o'giruvchi translyatorlardir. Bu jarayonlar assemblatsiya (assembling) deb nomlanadi.

Kompilyator bu dasturlash tilini mashina kodiga yoki assembler tiliga yoki past darajadagi tilga osonlikcha tarjima qiladigan kompyuter dasturi. U har bir dasturni kompyuter tushunadigan va kodga mos keladigan vazifani bajaradigan ikkilik (1 va 0) ga tarjima qiladi. Interpretator bu dastur ko'rsatmalarini mashina kodiga aylantiruvchi kompyuter dasturi. Dastur bayonotlari manba kodi, oldindan tuzilgan kod va skriptlarni o'z ichiga oladi. Umumiy qilib aytganda, kompilyator va interpretatorlar yuqori darajli dasturlash tillarida yozilgan kodlarni mashina tiliga o'giruvchi translyatorlardir. Kompilyatorlar va interpretatorlar o'rtasidagi farqni tushunish uchun ularning ishlash usullarini tahlil qilish kerak. Kompilyator - bu yuqori darajadagi dasturlash tili kodini bevosita kompyuter protsessori tomonidan bajarilishi mumkin bo'lgan mashina kodiga o'tkazuvchi dastur. Interpretatorlar manba kodining har bir satrini o'qiydi va uni bajariladigan fayl yaratmasdan darhol bajaradigan dasturdir. Kompilyatorlar va interpretatorlarda mashina kodiga o'tkazish jarayoni mos tarzda kompilatsiya va interpretatsiya deb nomlanadi.

Kompilatsiya jarayoni bu dasturlash tilida yozilgan kodni mashina tushunadigan tilga to'g'ridan-to'g'ri o'girish hisoblanadi. Mashina tiliga o'giruvchi programma esa kompilyator deb nomlanadi. Kompilyator butun kodni bir vaqtda o'qib oladi ya'ni skannerlaydi va agar kodda xatoliklar bo'lsa koddagi barcha xatoliklarni ko'rsatadi, agar xatolik mavjud bo'lmasa uni ishga tushiradi. Kompilyatorga misollar: javac (Java kompilyatori), c (C kompilyatori), cpp (c++ kompilyatori).

Interpretatsiya jarayonida esa dasturlash tilida yozilgan kod mashina tiliga to'g'ridan-to'g'ri o'girilmaydi. Uning o'rniga boshqa bir programma kodni o'qiydi va uni ishga tushiradi. Ya'ni o'rtada bir boshqa programma vositachi bo'lib turadi. O'sha vositachi programmaga interpretator deyiladi. Misol uchun siz kodda «+» amalini bajarmoqchi bo'lsangiz, interpretator

siz kodda yozgan «+» amalini o‘zining versiyasi bilan almashtiradi va so‘ngra uni mashina tiliga o‘giradi. Interpretator kodni satrma-satr o‘qib oladi va agar biror satrda xatolik mavjud bo‘lsa, darhol xatolikni ko‘rsatadi va skannerlashni ya’ni o‘qib olishni to‘xtatadi, agar satrda xatolik topilmasa darhol o‘sha satrdagi kodni ishga tushiradi. Interpretatorlarga misollar: Ruby, Phyton, GW Basic.

ADABIYOTLAR TAHLILI VA METODOLOGIYA.

Kompilyatorning asosiy ishi dasturni mashina kodiga o‘girish va dasturda xatoliklar, diapazonlar, chegaralar va hokazolar, ayniqsa, sintaktik xatolar mavjudligini dasturchiga bildirishdir. U butun dasturni tahlil qiladi va uni mashina kodiga aylantiradi. Kompilyatorning ishlashini quyidagi bosqichlarga bo‘lish mumkin:

- Leksik tahlil : manba kodini leksema deb nomlanuvchi mavhum bo‘lakka bo‘lish. Leksemalarning har biri uchun kalit so‘z, qator yoki boshqa o‘zgaruvchi bo‘ladimi, degan ma’noda token hosil bo‘ladi.
- Sintaksis tahlili : Belgilangan tokenlar mavhum sintaksis daraxtini (Abstract Syntax Tree - AST) hosil qilish uchun tuzilgan va sintaksisdagi xatolar tekshiriladi.
- Semantik tahlil : AST noto‘g‘ri tayinlangan o‘zgaruvchi, e’lon qilinmagan o‘zgaruvchi, kalit so‘zlarni o‘zgaruvchi sifatida ishlatish va hokazo kabi semantik xatolar uchun tekshiriladi.
- Oraliq kodni yaratish : Kompilyatsiya jarayoni ikki yoki undan ortiq oraliq kod shakllarini yaratadi.
- Optimallashtirish : Kompilyatsiya jarayoni vazifani yaxshilashning bir nechta usullarini izlaydi.
- Kod ishlab chiqarish : Kompilyator oraliq optimallashtirilgan kodni mashina kodiga aylantiradi, shundan so‘ng manba dasturi ob’ekt dasturiga aylanadi.

Interpretator ishlash jarayoni kompilyatornikiga juda o‘xshashdir. Ularning ishlashi orasidagi yagona farq shundaki, interpretator hech qanday oraliq kod shakllarini yaratmaydi, xatolarni tekshirish uchun dastur qatorini o‘qiydi va dasturni bir vaqtning o‘zida ishga tushiradi.

NATIJALAR.

Kompilyator va interpretator dasturlarni mashina kodiga tarjima qilish orqali bir xil ishlayotganga o‘xshasa-da, ularni bir-biridan farqlovchi muhim jihatlari mavjud. Bular :

1. Kompilyatsiya va interpretatsiya : Kompilyator bir vaqtning o‘zida butun dasturni mashina kodiga aylantiradi va kompilyatordan mustaqil ravishda bajarilishi mumkin bo‘lgan exe faylni hosil qiladi. Interpretator esa kodni satr bo‘yicha o‘qiydi va bajaradi, har bir bayonotni ketma-ket mashina kodiga aylantiradi.

2. Bajarish tezligi : Kompilyator tomonidan tuziladigan dasturlar interpretator tomonidan tuziladigan dasturlardan tezroq ishlaydi, chunki kompilyatorda tuzilgan dasturlar bajarilishidan oldin mashina kodiga aylantirgan bo‘ladi.

3. Xatolarni aniqlash : Kompilyatorlar interpretatorlarga qaraganda xatolarni tekshirishni kuchliroq amalga oshiradilar, chunki ular bajariladigan faylni yaratishdan oldin butun dasturni tahlil qiladilar. Interpretatorlar odatda xatolarni satrga qarab aniqlaydilar.

4. Portativlik : Kompilyatsiya qilingan dasturlar, odatda, ma’lum bir platforma yoki operatsion tizim uchun kompilyatsiya qilinganligi sababli, interpretatsiya qilingan dasturlarga qaraganda kamroq portativdir.

5. Ishlab chiqish vaqti : Dasturni kompilyatsiya qilish uni interpretatsiyalashdan ko‘ra ko‘proq vaqt talab etadi, chunki bajarish boshlanishidan oldin butun dasturni tahlil qilish va mashina kodiga aylantirish kerak. Interpretatorlar kod yozilgandan so‘ng darhol uni bajarishni boshlashlari mumkin.

6. Xotiradan foydalanish : Interpretatsiya qilingan dasturlar kompilyatsiya qilingan dasturlardan ko‘ra ko‘proq xotiradan foydalanadi, chunki interpretator ijro paytida xotiradagi manba kodini ham, unga mos keladigan mashina kodini ham kuzatib borishi kerak.

MUHOKAMA.

Kompilyator yoki interpretatordan foydalanishni tanlash, ishlab chiqilayotgan dastur turi, mavjud resurslar va loyiha talablari kabi turli omillarga bog‘liq. Umuman olganda, kompilyatsiya qilingan dasturlar interpretatsiyalangan dasturlarga qaraganda tezroq va samaraliroq, lekin kompilyatsiya paytida ko‘proq xotira resurslarini talab qiladi. Interpretatsiya qilingan dasturlarni o‘zgartirish osonroq, lekin o‘zgarishlar kuchga kirishi uchun tez-tez qayta kompilyatsiya qilishni talab qilishi mumkin.

XULOSA.

Kompilyatorlar ham, interpretatorlar ham loyiha talablariga qarab o‘zlarining afzalliklari va kamchiliklariga ega. Kompilyatorlar tezroq ishlaydigan, lekin yaratish uchun ko‘proq vaqt talab qiladigan, bajariladigan fayllarni ishlab chiqaradilar. Interpretatorlar xatolarni chuqur tekshirmaydigan, lekin tezroq ishlab chiqish vaqti bilan kodni satr bo‘yicha bajaradilar.

REFERENCES

1. "Compilers: Principles, Techniques, and Tools" by Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman.

2. "Crafting Interpreters: A Handbook for Making Programming Languages" by Robert Nystrom.

3. Guru99

URL : https://www.guru99.com/difference-compiler-vs-interpreter.html