



Miksi C on vieläkin käytössä?

Aihemääritelmä

Jaakko Hannikainen

24. marraskuuta 2016

C on ollut vallitseva ohjelmointikieli järjestelmäohjelmoinnissa C:n alkuaikojista lähtien. Useita ohjelmointikieliä on luotu historian saatossa, joiden oli tarkoitus syrjäyttää C. C on vieläkin johtavana kielenä varsinkin sulautetuissa järjestelmissä ja UNIX-pohjaisten käyttöjärjestelmien vallitsevana järjestelmäohjelmointikielenä. Tarkoituksena tutkia, miksi vaihtoehtoisista huolimatta C on vieläkin laajalti käytössä myös uusissa projekteissa. Minkälainen ohjelmointikieli voisi syrjäyttää C:n.

Määritetään C:tä paremmaksi kieleksi jokin kieli, mikä C:hen verrattuna:

- on yhtä nopea tai nopeampi
- käyttää saman verran muistia tai vähemmän
- on helpompi käyttää
- toimii järjestelmissä, joissa C toimii

Tällainen kieli on selkeästi olemassa, sillä C:stä voi tehdä helpomman käyttää poistamalla C:n ominaisuuksista trigrafit, joiden poistaminen ei vaikuta kielen ominaisuuksiin.

C:n vaihtoehtoisiksi tutkitaan seuraavia kieliä: C++, Go, Ada, D, Rust. Näistä kielistä tutkitaan, mikä tai mitkä ominaisuudet ovat estäneet C:n korvauksen, ja mitkä ominaisuudet ovat olleet parannuksia C:hen verrattuna. Lisäksi tutkitaan muista suosituista ohjelmointikielistä ominaisuuksia, jotka ovat hyödyllisiä matalan tason ohjelmoinnissa ja jotka voi toteuttaa korvaavan kielen rajoitteissa. Suosituista ohjelmointikielistä tutkitaan seuraavia kieliä: Python, Java, Haskell. Erityisesti tutkitaan kielen mahdollisuuksia

- tyyppiturvallisuuteen (Haskell)
- kääntäjän suorittamaan optimointiin (esim. vektorisointi)
- kääntöseen koodin varmistamiseen (alustamattomat muuttujat)
- funktionaaliseen ohjelmointiin

- sivukanavahyökkäysten torjumiseen (ajoitushyökkäykset, välimuistihyökkäykset)

Tutkittavana on myös mitä optimointeja C:ssä ei voi tehdä helposti johdettujen kielen rajoitteista, miten tämän voisi korjata. Näitä ominaisuuksia ovat esimerkiksi sivuvaikutuksettoman ohjelmakoodin merkitseminen, optimointivinkien alustariippumaton ilmaiseminen (asettaminen funktion parametreihin) ja useat eri funktiot riippuen parametrien arvoista, mikäli ne voidaan kääntöaikaisesti päätellä.

Viitteet

- [1] D programming language. <http://dlang.org/spec/spec.html>, 2016.
- [2] The go programming language. <https://golang.org/ref/spec>, 2016.
- [3] Information technology – Programming languages – C++. Standard, International Organization for Standardization, Geneva, CH, Dec. 2014.
- [4] Information technology – Programming languages – Ada. Standard, International Organization for Standardization, Geneva, CH, Dec. 2012.
- [5] Information technology – Programming languages – C. Standard, International Organization for Standardization, Geneva, CH, Dec. 2011.