

hyväksymispäivä

arvosana

arvostelija

## C- ja C++-ohjelmointikielten kehitys

Juha Huotari

Helsinki 11. helmikuuta 2003

Seminaariesitelmä

HELSINGIN YLIOPISTO

Tietojenkäsittelytieteen laitos

# Sisältö

<b>1 Johdanto</b>	<b>1</b>
<b>2 C-kielen juuret</b>	<b>2</b>
2.1 Uusi käyttöjärjestelmä . . . . .	2
2.2 Ongelmia B-kielen kanssa . . . . .	4
<b>3 C-kielen kehitys</b>	<b>5</b>
<b>4 C++-kielen synty</b>	<b>6</b>
4.1 Lähtökohdat . . . . .	7
4.2 Myöhempi kehitys . . . . .	8
<b>5 Nykyaika</b>	<b>9</b>
5.1 C++-kielen tulevaisuus . . . . .	9
<b>6 Yhteenveto</b>	<b>10</b>
<b>Viitteet</b>	<b>10</b>

# 1 Johdanto

C-ohjelmointikieli sai alkunsa 1970-luvun alussa yhdysvaltalaisen puhelinyhtiön tutkimuskeskuksessa Bell Telephone Laboratories (Bell Labs). C-kieli suunniteltiin alunperin järjestelmäohjelmointiin, ja sitä käytettiin aluksi vain Unix-käyttöjärjestelmässä. Mikrotietokoneiden yleistyessä 1980-luvun alussa C-kieltä alettiin käyttää useissa erilaisissa tietokoneissa ja käyttöjärjestelmissä.

1980-luvun alussa alettiin kehittämään C++-ohjelmointikieltä lisäämällä C-kieleen ohjelmointia parantavia ja uusia ohjelmointitekniikoita tukevia ominaisuuksia, ensisijaisena lisäyksenä luokat. Joitain yksityiskohtia lukuunottamatta C säilytettiin C++-kielen ”alijoukkona” eli C-kielellä kirjoitettu ohjelma on myös kelvollinen C++-kielinen ohjelma.

C-kielen menestykseen ovat vaikuttaneet Unix-käyttöjärjestelmän suosio sekä kielen ja kääntäjän yksinkertaisuus. Vaikka C-kieli on riippumaton laiteympäristöstä, se sopii hyvin järjestelmäohjelmointiin, sillä C-kielen tietotyypit perustuvat tietokoneiden oikeisiin tietoalkioihin; bitteihin, merkkeihin, sanoihin ja muistiosoitteisiin. C++-kielellä on samat hyvät ominaisuudet kuin C-kielellä – yksinkertaisuutta lukuunottamatta. Se tuo parannuksia C-kieleen, tarjoaa mahdollisuudet parempiin ohjelmointitekniikoihin ja on nykyään yksi suosituimmista ohjelmointikielistä.

Esitelmän pääpaino on C-kielen kehityksen alkuvaiheissa. Tarkoituksena on antaa jonkinlainen kuva ohjelmointikielen kehityksestä 1970-luvun alussa, jolloin alan huippututkimus oli usein muutaman miehen harteilla.

Luvuissa 2 ja 3 kerrotaan C-kielen synnystä ja kehityksestä 1960- ja 1970-lukujen vaihteesta alkaen. Luvussa 4 kerrotaan C++-kielen kehityksestä ja luvussa 5 C- ja C++-kielten käytöstä nykyaikana.



Kuva 1: Ken Thompson

## 2 C-kielen juuret

C-kielen juuret johtavat 1960-luvun loppupuolelle, jolloin Bell Labs vetäytyi Multics-projektista (*Multiplexed Information and Computing Service*). Multics oli Bell Labsin, MIT:n (*Massachusetts Institute of Technology*) ja General Electricin yhteishanke monen käyttäjän käyttöjärjestelmälle. Bell Labsin johto kuitenkin epäili, että projekti tuottaisi tuloksia liian myöhään ja liian kalliilla hinnalla. [Ric93]

### 2.1 Uusi käyttöjärjestelmä

Projektissa mukana ollut Ken Thompson (kuvassa 1) alkoi ryhmänsä kanssa kehittämään vaihtoehtoista järjestelmään Bell Labsissa. Tavoitteena oli luoda helppokäyttöinen ohjelmointiympäristö heidän omiin tarpeisiinsa. Vuonna 1968 Thompson alkoi käyttämään uutta DEC PDP-7 -tietokonetta, jossa oli 8 kilon 18-bittinen muisti, eikä yhtään käyttökelpoista ohjelmistoa. Ensitöikseen Thompson ohjelmoi PDP-7:lle käyttöjärjestelmän, jota voidaan pitää varhaisena versiona Unixista. Ohjelmoinnin

```

get 'HEAD1A'
let Formdigit(x) = x+48    // the result is an ASCII digit
and Unpackstring(S, V) be
    $(1 let n = S*(0) rshift Byte1shift
        let i,j = 0,0
        until i gr n do
            $( let W = S*(i)
                V*(j) := W rshift Byte1shift
                V*(j+1) := W rshift Byte2shift & Bytemax
                V*(j+2) := W rshift Byte3shift & Bytemax
                V*(j+3) := W & Bytemax
                i,j := i+1,j+4 $)1
and Packstring(V, S) be
    $(1 let n = V*(0)
        and i,j = 0,0
        V*(n+1), V*(n+2), V*(n+3) := 0, 0, 0
        until j gr n do
            $( S*(i) := V*(j) lshift Byte1shift logor
                V*(j+1) lshift Byte2shift logor
                V*(j+2) lshift Byte3shift logor V*(j+3)
                i,j := i+1,j+4 $)1

```

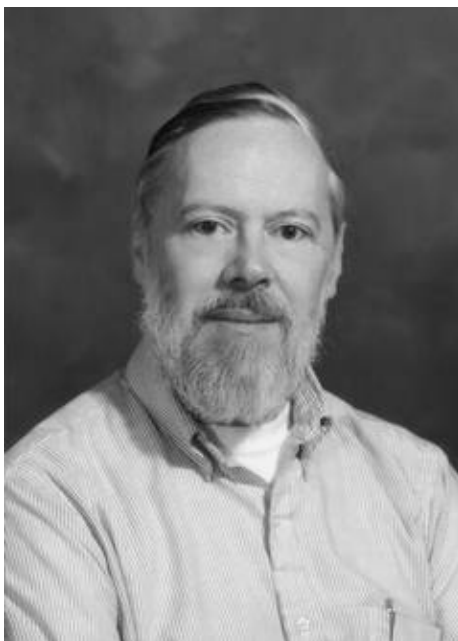
Kuva 2: Esimerkki BCPL-kielestä [Ric67]

hän suoritti GE-635-koneella. Siitä saatu paperinauhatuloste annettiin syötteenä ja testattiin PDP-7:ssä, kunnes käyttöjärjestelmän ydin, tekstieditori, assembler ja komentotulkki muutamalla komennolla varustettuna oli valmis vuonna 1969. [Ric93]

Koneesta puuttui vielä korkean tason ohjelmointikieli. Epäonnistuneiden Fotranin kanssa tehtyjen kokeilujen jälkeen Thompson kehitti PDP-7:lle kielen nimeltään B. Kieli perustuu Martin Richardsin MIT:ssä kehittämälle BCPL-kielelle (*Basic Combined Programming Language*). BCPL-kieli oli suunniteltu kääntäjien ja käyttöjärjestelmien ohjelmointiin. BCPL:stä on poistettu edeltäjänsä CPL:n tyypit ohjelmien kääntämisen helpottamiseksi. Kuvassa 2 on esimerkki BCPL-kielisestä koodista [Ric67].

Samaan tapaan kuin BCPL, B on tyyppitön kieli (tai yksityyppinen), joka käyttää samanlaista, laitteistosta riippumatonta osoitinaritmetiikkaa kuin BCPL-kieli.<sup>1</sup> B-

<sup>1</sup>Dennis Ritchien, C-kielen kehittäjän sanoin: "B can be thought of as C without types; more accurately, it is BCPL squeezed into 8K bytes of memory and filtered through Thompson's brain." [Ric93]



Kuva 3: Dennis Ritchie

kielen syntaksi on jo hyvin samankaltainen kuin C-kielessä. B-kieltä ei kuitenkaan voitu käyttää PDP-7:llä paljon muuhun kuin pieniin kokeiluihin, sillä B-kääntäjä vei liikaa tilaa PDP-7:n 8 kilon muistista.

Vuonna 1970 Bell Labs hankki testikäyttöön uuden PDP-11:n. Kone tilattiin DEC:ltä heti kun se valmistui tehtaalta, mutta sen kiintolevy saapui Bell Labsiin vasta kolmen kuukauden päästä. Sitä odotellessa Thompson ohjelmoi uudelle koneelle Unixin ja Thompsonin työkaveri Dennis Ritchie (kuva 3) B-kielen kääntäjän, jonka avulla koneessa voitiin hyödyntää edelliselle koneelle B-kielellä ohjelmoitua koodia. Vuoteen 1971 mennessä PDP-11:llä oli useita käyttäjiä, ja kirjastonsa ansiosta B-kieli osoittautui mielenkiintoisten ohjelmien tekoon käytännöllisemmäksi kuin symbolinen konekieli. [Ric93]

## 2.2 Ongelmia B-kielen kanssa

Dennis Ritchie ei kuitenkaan ollut tyytyväinen B-kieleen. B-kielen ainoa tietoaalkiotyyppi oli solu (*cell*), joka vastasi tietokoneen käytössä olevaa tietoyksikköä; PDP-

7:ssä 18-bittinen, PDP-11:ssä 16-bittinen sana. Etenkin kirjainten käsittely oli hankalaa, kun teksti oli pakattu soluihin. Teksti piti purkaa niin, että muistissa oli yksi kirjain per solu, jotta tekstiä voitiin käsitellä B-kielisessä ohjelmassa.

Toinen ongelma liittyi liukulukuihin. 18-bittisessä PDP-7:ssä liukuluku pystyttiin esittämään yhdessä solussa, mutta 16-bittisessä PDP-11:ssä se ei ollut mahdollista. Oli selvää, että ohjelmointikielessä olisi hyvä pystyä määrittelemään muuttujan tyyppi.

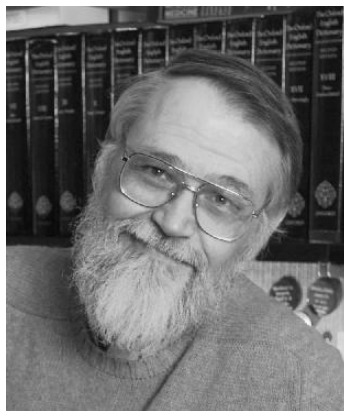
### 3 C-kielen kehitys

Ritchie alkoi kehittämään B-kielestä uutta ohjelmointikieltä NB (*new B*). Kielessä oli muuttujatyypit `int` ja `char`, sekä niille osoittimet. Rakenteisten muuttujien (`struct`) lisääminen uuteen kieleen vaati koko BCPL- ja B-kielistä perityn staattisen osoitinmekanismin saattamista uuteen uskoon. Uusi systeemi salli rakenteiset muuttujat, jotka sisälsivät toisia rakenteisia muuttujia, osoittimet osoittimiin ja osoittimet funktioihin. Osoitinsysteemin uudistus muodostui sellaiseksi kuin se on C-kielessä edelleen ja on paljon velkaa ohjelmointikielelle Algol 68. Muut ratkaisevat muutokset kielen syntaksiin olivat uudenlaiset loogiset operaattorit (`&`, `|`, `&&`, `||`).

Kieli erosi B-kielestä jo niin paljon, että Ritchie antoi sille uuden nimen C. Vuoteen 1973 mennessä muitakin Bell Labsin tutkijoita oli liittynyt C-kielen kehitykseen ja merkittävimmät ominaisuudet kieleen olivat olemassa. Kääntämiseen liittyvä suuri uudistus oli esiprosessorin lisääminen, mikä mahdollisti lähdetiedostojen lisäämisen (`#include`), merkkijonojen korvaamisen (`#define`), makrot argumenteilla ja ehdollisen kääntämisen. [Ric93]

Seuraavina vuosina C-kielen kehityksessä keskityttiin siirrettävyyteen, tietotyyppien toimivuuteen (*type safety*) ja C-kielellä kirjoitettujen ohjelmien yhteensopivuuteen olemassaolevien ohjelmien kanssa. Unix pystyttiin kirjoittamaan C-kielellä ja siirtämään se muihin laiteympäristöihin.

Vuonna 1978 Dennis Ritchie kirjoitti työkaverinsa Brian Kernighanin (kuvassa 4) kanssa kirjan *The C Programming Language*. Vaikka siinä ei ollut C-kielen myöhem-



Kuva 4: Brian Kernighan

piä ominaisuuksia (mm. `void` ja `enum`) eikä se ollut usein riittävän tarkka, niin se toimi ainoana ”virallisena” C-kielen käsikirjana useita vuosia. Kirjan tekijöiden nimitystä tuleekin vanhan C-kielen nimitys K&R C.

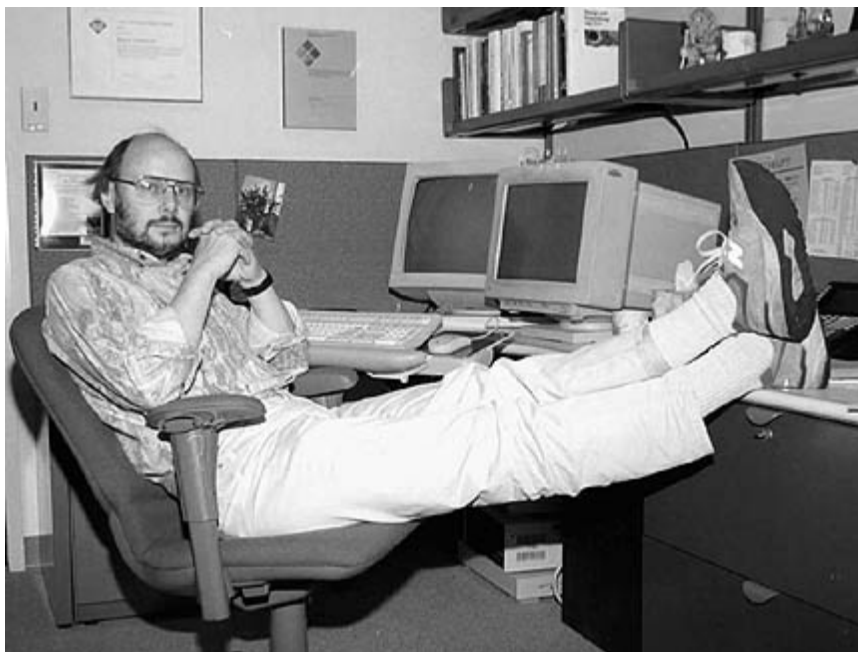
Vuonna 1982 C-kieli oli – osaksi Unix-käyttöjärjestelmän suosion ansiosta – tullut suosituksi useissa laiteympäristöissä, mutta oli edelleen ilman muodollista standardia. ANSI (*American National Standards Institute*) perusti komitean X3J11 kehittämään ANSI C -standardia. Raportti oli valmis vuonna 1989, ja vuotta myöhemmin standardin hyväksyi ISO (*International Organization for Standardization*) tunnuksella ISO/IEC 9899-1990.

## 4 C++-kielen synty

Tanskalainen Bjarne Stroustrup (kuvassa 5) alkoi kehittämään C-kieltä. Tavoitteena hänellä oli luoda ohjelmointikieli, joka olisi sekä tehokas että elegantti. Hänen mielestään siihen asti oli voinut valita vain toisen [Str02].

Stroustrup aloitti työnsä vuonna 1979, jolloin hän muutti perheineen yhdysvaltoihin ja aloitti työt Bell Labsissa. Silloin uuden kielen työnimi oli ”C with Classes”. Vuonna 1984 Bell Labs siirtyi AT&T:n (*American Telephone and Telegraph Company*) alaisuuteen ja ensimmäinen versio C++-kielestä tuli AT&T:n sisäiseen käyttöön. En-





Kuva 5: Bjarne Stroustrup

simmäinen kaupallinen versio C++-kielestä ilmestyi vuonna 1985 [Str02]. Seuraavana vuonna ilmestyi ensimmäinen painos Stroustrupin kirjasta *The C++ Programming Language* [Str86].

#### 4.1 Lähtökohdat

Stroustrup halusi lisätä C-kieleen uusia ominaisuuksia niin, että sen tehokkuus ja yksinkertaisuus säilyisivät [Str02]. Kieli ei – C-kielen tapaan – sisällä korkean tason tietotyyppejä, kuten matriiseja ja merkkijonoja, vaan sellaiset on käyttäjän itsensä määriteltävissä. Kielen tietorakenteet eivät myöskään sisällä metadataa (*“housekeeping information”*), vaan esimerkiksi rakenne (`struct` tai `class`), joka sisältää kaksi 16-bittistä kokonaislukua, vie muistia vain 32-bittiä.

C++-kielen yhteensopivuus C-kieleen haluttiin säilyttää. C-kieltä voidaan edelleen pitää joitain yksityiskohtia lukuunottamatta C++-kielen ”alijoukkona” eli C-kielellä kirjoitettu ohjelma on myös kelvollinen C++-ohjelma. C-kieliset ilmaisut, jotka eivät ole C++-kieltä, ovat lähinnä ”huonoa” C-kieltä, jota ei olla haluttu sallia C++-

kielessä. Tärkein syy yhteensopivuuden säilyttämiseksi oli se, että C-kiehellä kirjoitettua koodia oli paljon ja sitä haluttiin pitää myös C++-ohjelmoijien käytettävissä [Str02].

Ensimmäisenä ja tärkeimpänä lisäyksenä C-kieleen oli luokka, jonka esikuvana oli Simula67:n luokat [Str97]. Luokat lisäsivät C-kieleen monia kaivattuja ominaisuuksia:

- paransivat tiedon abstrahointia ja piilottamista sekä kielen modulaarisuutta
- takasivat tietoalkioiden alustuksen (konstruktorit)
- implisiittiset tyyppimuunnokset
- operaattoreiden kuormitus [Str86]

Simula67 olisi muuten ollut sopiva Stroustrupin testatessa tapahtumaohjattuja simulaatioita (*event-driven simulations*), mutta se oli liian hidas. Siksi Stroustrupin oli lisättävä nämä ominaisuudet C-kieleen [Str97].

Muita C++-kieleen lisättyjä tärkeitä ominaisuuksia olivat muuttujien parempi ja dynaaminen tyyppitys, dynaaminen muistin hallinta (komennot `new` ja `delete`), kuormitetut funktiot ja viitteet [Str86].

## 4.2 Myöhempi kehitys

Pitkään C++ kehittyi niin, että suunnittelu, dokumentointi ja toteutus kulkivat käsikädessä. Kielen kehittäminen tapahtui yhteistyössä Stroustrupin ja hänen kollegoidensa kanssa. C++ tuli niin suosituksi, että vuoden 1987 aikaan oli jo selvää, että se tarvitsi muodollisen standardin.

AT&T – jolla siihen aikaan oli oikeudet C++:aan – salli Stroustrupin jakaa vedoksia C++-kielen määrittelystä sadoille eri organisaatioiden edustajille saadakseen ehdotuksia ANSI C++:a varten. Vuonna 1989 ANSI-komitea X3J16 kokoontui standardisoimaan C++-kielen, ja vuonna 1991 kielestä tuli kansainvälinen ISO-standardi.

Komitean puheenjohtajana Stroustrup on edelleen vastuussa standardoinnista ja hyväksyi suurimmat kieltä koskevat muutokset [Str02].

## 5 Nykyaika

Vaikka C-kieli ei enää ole käytännöllisin kieli moniin tarpeisiin, sitä käytetään edelleen paljon. Yksi syy on Unixin ja muiden sen kaltaisten käyttöjärjestelmien suosio, sillä niiden mukana C-kieli on kulkenut mukana jo sen takia että niin paljon ohjelmia on kirjoitettu C-kielillä. Lisäksi C-kieli on käyttökelpoinen monissa pienissä koneissa ja sulautetuissa järjestelmissä, ja sen kääntäjät ovat pieniä ja yksinkertaisia. Sen tietotyypit perustuvat sellaisiin tietoalkioihin, jotka tietokoneista löytyvät fyysisestikin. Silti se on riittävän abstrakti niin, ettei ohjelmoijan tarvitse liikaa ajatella koneen teknisiä yksityiskohtia.

### 5.1 C++-kielen tulevaisuus

C++ pysyy edelleen yhtenä suosituimmista ohjelmointikielistä suurissa projekteissa, ja kun ohjelmistolta vaaditaan nopeutta. C++-kielestä löytyvät työkalut moderniin olio-ohjelmointiin ja tiedon abstrahointiin, ja silti siinä on samoja hyviä puolia kuin C-kielissä, kuten koneläheisyys ja siirretävyys. Viimeisimmät uudistukset C++-kieleen ovat

- nimiavaruudet
- poikkeukset
- kaavaimet (*templates*)
- ajonaikainen tyyppitunnistus (*run-time type identification*)

Lisäksi ISO C++:n standardikirjasto sisältää paljon valmiita korkean tason tietotyyppejä – kuten lista, kartta (*map*) ja matriisi – joiden avulla ohjelmoija voi aloittaa korkeammalta tasolta. C++ on edelleen kehitystyön alla, ja ainakin Stroustrupin

mielestä kehityksen painon pitäisi olla enemmän standardikirjastoissa kuin kielessä itsessään. Hänen mielestään C++:sta pitäisi tehdä parempi kieli järjestelmien ja kirjastojen ohjelmointiin sekä helpompi opettaa ja oppia [Str02].

## 6 Yhteenveto

Tässä esitelmässä on kerrottu C- ja C++-kielten kehityksestä 1960- ja 1970-lukujen vaihteesta tähän päivään asti. C-kieli on tullut aikanaan suosituksi osaksi Unix-käyttöjärjestelmän menestyksen johdosta, mutta on pysynyt käytännöllisenä ohjelmointikielenä näihin päiviin asti myös omilla ansioillaan.

C++-kieli on tuonut C-kieleen useita parannuksia ja tuen uusille ohjelmointitekniikoille. C++ on nykyään yksi suosituimmista ohjelmointikielistä, ja sen kehitys jatkuu edelleen.

## Viitteet

- Ric67 Martin Richards, The BCPL Reference Manual, 1967, MIT.
- Ric93 Dennis Ritchie, The Development of the C Language, 1993, Second History of Programming Languages conference.
- Str86 Bjarne Stroustrup, The C++ Programming Language, 1986, Addison-Wesley.
- Str97 Bjarne Stroustrup, The C++ Programming Language, 1997, Addison-Wesley.
- Str02 Bjarne Stroustrup's FAQ, 2002, [http://www.research.att.com/~bs/bs\\_faq.html](http://www.research.att.com/~bs/bs_faq.html)