

# Ohjeita kääntäjien käyttöön

Luokassa KO130, jossa kaikki kurssin opetus järjestetään, on tietokoneille asennettuna Microsoft Visual Studio 6 -C-kääntäjä (oikeastaan se on Visual C++-kääntäjä, mutta C++-kieli sisältää periaatteessa C-kielen – ehkä tälläkin kursilla tulevaisuudessa siirrytään C++-kieleen). Pääasiassa kurssilla käytetään tätä kääntäjää.

Lisäksi kurssilla tutustutaan Linux-ympäristön gcc-kääntäjään (GNU Compiler Collection). gcc:tä voi käyttää kätevästi myös kotikoneelta, jos kotona on internet-yhteys.

## Pikaopas

Selitän ensin lyhyesti, miten em. kääntäjillä saa ajettavan ohjelman aikaiseksi.

### Visual Studio

Visual Studio 6:ssa lähdekooditiedostot kuuluvat aina johonkin projektiin, ja projektit kuuluvat johonkin työtilaan (workspace). Visual Studiolla ohjelmitaessa onkin kätevintä ensin luoda uusi työtila, johon luodaan uusia projekteja, joihin sitten luodaan lähdekooditiedostoja. Käytännöllinen tapa harjoitusvuoroilla on luoda workspace, jonka nimeää esim. ATK1. Se kannattaa luoda suoraan omaan kotihakemistoon paju:lle (ks. unix-ohjeita.pdf: kotihakemiston suora liittäminen), josta sen voi aina harjoitusvuorojen alussa avata. Kurssin kaikkia ohjelmointitöitä varten voi luoda uuden projektin, ja ne ovat kaikki samassa workspaceissa, helposti selattavissa.

Valitse File→New..., avaa Workspaces-välilehti, jolta valitse Blank Workspace (ainut vaihtoehto). Anna työtilalle nimi ja määrää kansio, johon se tallennetaan. Paina OK.

Paina hiiren oikeanpuoleista nappia työtilan kohdalla (Workspace 'xxx': 0 project(s)) ja valitse Add New Project to Workspace. Valitse projektin tyyppi Win32 Console Application, anna projektille nimi ja kansio (yleensä oletusarvoinen kansio on hyvä – se sijoittaa projektin työtilan alihakemistoon), varmista että optio 'Add to current workspace' on valittu ja paina OK. Seuraavasta valikosta valitse An empty project ja paina Finish.

Paina oikeanpuoleista nappia projektin nimen kohdalla (xxx files) ja valitse Add new files to project. Jos aiot luoda uuden lähdekooditiedoston, navigoi kansioon

johon haluat sen luoda (yleensä oletuskansio on hyvä), kirjoita tiedoston nimi (muista tiedostotunnus `.c`, esim. `koodi.c`, jotta kääntäjä tunnistaa tiedoston nimenomaa C-kieliseksi lähdekooditiedostoksi) ja paina OK. Jos haluat aukaista jo olemassaolevan tiedoston, etsi ja valitse se ja paina OK. Utta tiedostoa luottaessa kääntäjä kysyy, lisääkö viittaus tiedostoon projektiin. Vastaa kyllä. Lähdekooditiedosto ilmestyy kansioon Source Files. Kaksoisklikkaamalla tiedoston nimeä saat sen sisältämän koodin näkyviin. Jos tiedosto on uusi, se luodaan tässä yhteydessä (vastaa kysymykseen taas kyllä).

Kannattaa luoda jokaista harjoitustehtävää varten uusi projekti (mutta liittää se ATK1-työtilaan – näin koodi-, ym. tiedostot pysyvät hyvässä järjestyksessä).

Koodi (projekti) käännetään (luodaan ajettava ohjelma) valitsemalla Build→Build projekti.exe. Mahdolliset virheilmoitukset tuleva ruudun alalaitaan pitkulaiseen ikkunaan, jota voi selata ylöspäin ja oikealle.

Ohjelma ajetaan valitsemalla Build→Execute projekti.exe.

## **GCC**

GCC-kääntäjää voi käyttää esim. paju- tai tuomi-palvelimella. Näihin palvelimiin voi ottaa yhteyden kotikoneelta SSH Secure Shell -ohjelmalla, joka on saatavilla esim. yliopiston tietohallinnon ohjelmistojakelu-sivustolta ([www.oulu.fi/tietohallinto](http://www.oulu.fi/tietohallinto)).

GCC:ssä ei ole omaa tekstieditoria, joten ohjelma on kirjoitettava esim. Vimillä, Emacsilla, Picolla, notepadilla, tms. tekstinkäsittelyohjelmalla, joka ei muotoile tekstiä. Ohjelma käännetään kirjoittamalla `gcc ohjelma.c`, ja käännetty ohjelma ajetaan kirjoittamalla `./a.out` (`a.out` on oletusarvoinen tiedostonimi – se on siis suoritettava ohjelma).

Jos käytät ohjelmassa matemaattisia funktioita (`math.h`-kirjastotiedostoa), niin ohjelma on käännettävä käskyllä `gcc ohjelma.c -lm`. Parametri `-lm` liittää ohjelmaan matematiikkakirjaston.

## **Virheiden metsästys**

Kääntäjissä on mukana ns. debug-ohjelma, jolla oman ohjelman suoritusta voi seurata, esim. ajamalla sen rivi kerrallaan.

## **Visual Studio**

Ohjelma ajetaan rivi kerrallaan valitsemalla Build-> Start Debug-> Step Over (tai Step Into). Toinen tapa on asettaa ns. breakpointeja, jolloin valittaessa Start Debug-> Go ohjelman suoritus pysähtyy breakpointtien kohdalla. Break-

pointteja asetetaan painamalla F9, kun kursori on rivillä, johon breakpointin haluaa asettaa.

Breakpointin kohdalla ohjelman suoritus pysähtyy (ajettaessa rivi kerrallaan ohjelman suoritus taukoaa jokaisella rivillä). Tällöin voi esim. tarkastella muuttujien arvoja.

## **GCC**

`gcc:n` yhteydessä käytetään debug-ohjelmaa `gdb`. Jotta `gdb` toimisi, on ohjelma käännettävä komennolla `gcc -g ohjelma.c`. `gdb` käynnistetään kirjoittamalla `gdb a.out`. Seuraavassa tärkeimpiä käskyjä:

- `run` Ajaa ohjelman pysähtyen mahdollisten breakpointtien kohdalla
- `step` Aja rivi kerrallaan.
- `break ohjelma.c:n` Asettaa breakpointin tiedoston `ohjelma.c` riville `n`.
- `watch muuttuja` Seurataan muuttujan muuttuja arvoa.