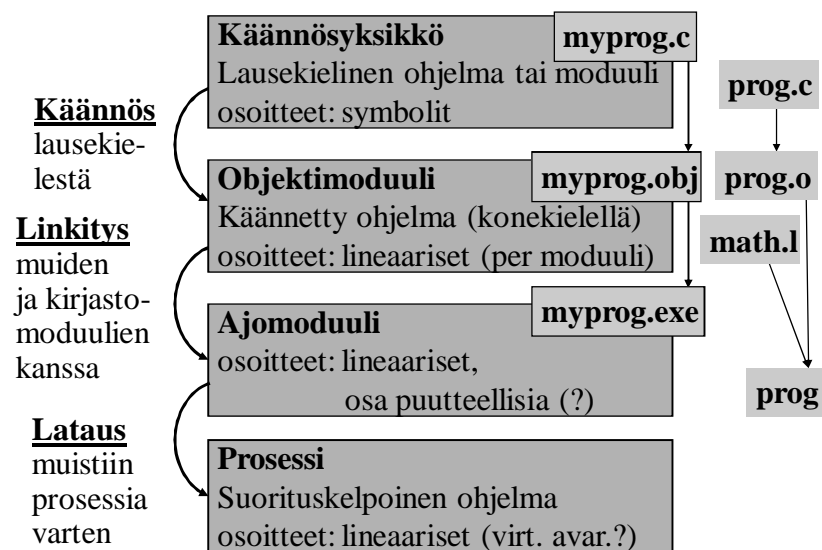


Luento 9 (verkkoluento 10)

Käännös, linkitys ja lataus

Ohjelmasta prosessiin
 Käännösyksikkö
 Kääntämisen vaiheet
 Makrot, literaalit
 Staattinen ja dynaaminen
 linkitys
 Nimien sidonta

Lausekielestä suoritukseen



15.11.2012

Copyright 2012 Teemu Kerola

Keskustele 2

Objektimoduuli

- Konekielinen koodi
 - moduulin sisäiset viitteet paikallaan (linearisessa muistiavaruudessa)
 - moduulin ulkopuoliset viitteet merkitty
- Linkitystä varten:
 - tiedot niiden osoitteiden **RELOCATION TABLE** sijainneista, jotka täytyy päivittää, kun moduulin osoiteavaruus yhdistetään jonkin toisen moduulin osoiteavaruuden kanssa linkityksessä
 - tiedot viittauksista moduulin ulkopuolelle **IMPORT**
 - tiedot moduulin kohdista, joihin saa viitata ulkopuolelta
 - symbolitaulu **SYMBOL TABLE** **EXPORT**

15.11.2012

Copyright 2012 Teemu Kerola

3

Symbolitaulu

- Kääntäjä generoi
- Ylläpidetään linkityksen aikana
- Joskus ylläpidetään myös latauksen jälkeen virheilmoitusten tekemistä varten
 - ohjelmien kehitysympäristöt ylläpitävät symbolitaulua koko ajan
- Jätetään pois valmiista ohjelmasta
 - vie turhaa tilaa

15.11.2012

Copyright 2012 Teemu Kerola

4

Makrot

- Helpottavat ohjelmointia, toistuva koodisarja
- Voivat sisältää parametreja
 - useimmiten nimiparametreja (call-by-name)
- Käsitellään ennen kääntämistä
 - eivät kuulu konekieleen
 - makron ”kutsu” (käyttö) korvataan makron rungolla
- Esimerkkejä
 - swap
 - aliohjelmien prologi ja epilogi
 - itse tehdyt, kääntäjän käyttämät
- Erot aliohjelmiin
 - Kutsu ajankohta, call/return, koodien lukumäärä

15.11.2012

Copyright 2012 Teemu Kerola

Keskustele 5

Literaalit

- Korkean tason kielissä kaikki isot vakiot ovat literaaleja `N := 35000;` `var myStr = "literal"`
 - kääntäjän pitäisi estää literaalien muuttaminen
 - `FortranX: 5 = 6;` `LOAD R1, six` `STORE R1, five` `???`
 - literaalia ei saisi välittää viiteparametrina
 - aliohjelma voisi muuttaa sen arvoa? `Java string?`
- Joissakin symb. konekielissä literaalien implisiittinen (automaattinen) määrittely
 - helpommin luettavaa koodia
 - literaalin 234567 tilanvaraus automaattisesti
 - `Load R14, =F'234567'`

15.11.2012

Copyright 2012 Teemu Kerola

6

Assembler käänös

- 1. vaihe:
 - laske käskyjen tilanvaraukset
 - generoi symbolitaulu, muut taulut
- 2. vaihe
 - generoi lopullinen objektimoduuli
 - tulosta symbolinen konekielinen listaus
 - generoi taulut linkitystä varten
 - anna virheilmoitukset
- 3. vaihe
 - koodin optimointi
 - voi olla oikeasti ennen 2. vaihetta tai sen yhteydessä

15.11.2012

Copyright 2012 Teemu Kerola

Keskustele 7

Korkean tason kielen käänös

- Enemmän vaiheita
 - Syntaktisten alkioden etsintä **(front end)**
 - Syntaksipuun generointi ja jäsenitys
 - Lauseiden tunnistaminen syntaksipuun avulla
 - Välikielen (välikoodin) generointi (ei aina)
 - Koodin generointi **(back end)**
 - ei (yleensä) Java-ohjelmille

Välikieliesitys ja symbolitaulut

Lisää tietoa?

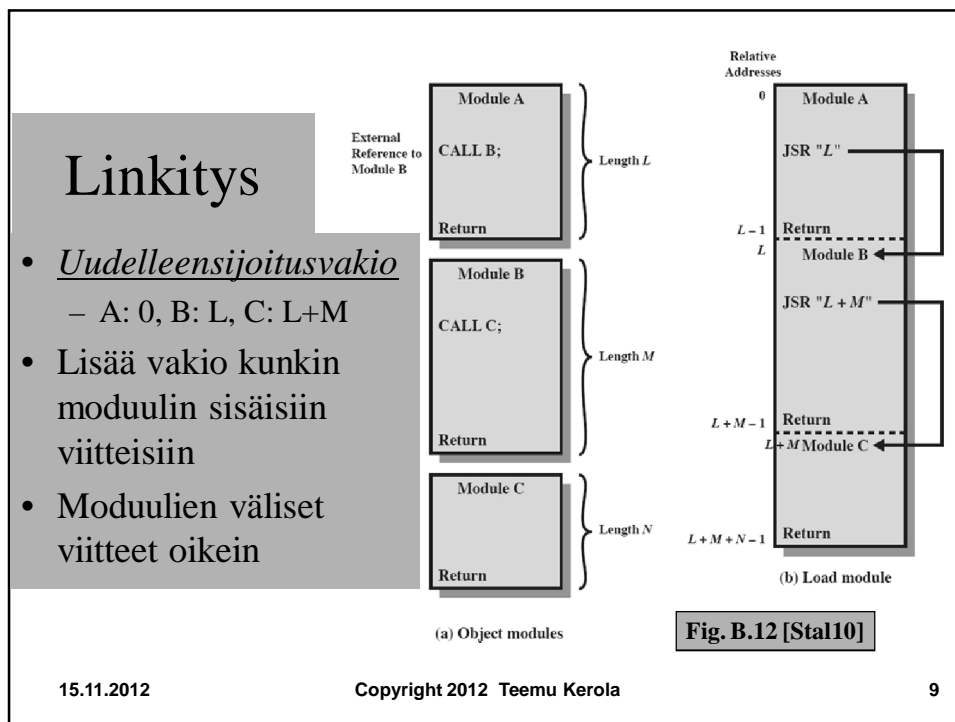


**Kääntäjien
ja ohj. kielten
kurssit**

15.11.2012

Copyright 2012 Teemu Kerola

Keskustele 8



Staattinen ja dynaaminen linkitys

- Staattinen linkitys
 - Kaikki ohjelmakoodissa viitatus moduulit ja kirjastorutiinit on linkitetty ennen suoritusta
 - Iso ajomoduli
 - mukana moduuleja, joihin ei yhdellä suorituskerralla tule lainkaan viittauksia
- Dynaaminen linkitys
 - Kutsukohdat muihin moduuleihin jätetään auki
 - Pienempi ajomoduli, mutta hitaampi suorittaa
 - Viittaus ”ratkaisemattomaan” (eli ei-linkitettyyn) moduuliin ratkotaan suoritusaikana
 - suoritus keskeytyy ja puuttuva moduuli linkitetään paikalleen (kaikki viittaukset siihen korjataan kuntoon)

Lataus

- Ajomoduulista luodaan suorituskelpoinen prosessi (rakennetaan PCB ja sen viitteet kuntoon)
- Prosessin koodi- ja data-alueet ladataan muistiin, prosessi siirretään ready (Ready-to-Run) jonoon
- Eri tyyppejä
 - Absoluuttinen – aina samaan paikkaan muistia
 - Uudelleensijoitettava – joustava paikan valinta
 - Dynaaminen ajoaikainen – paikat vaihtelevat ajoaikana

15.11.2012

Copyright 2012 Teemu Kerola

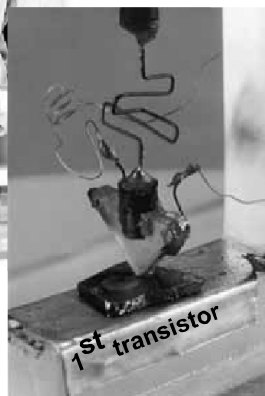
11

-- Luennon -- -- loppu --

- Transistori
 - J. Bardeen, W.B. Shockley ja W. Brattain, Bell Labs, 1948
 - TX-0, MIT, 1956
- Integroitu piiri (ei enää johtoja)
 - Jack Kilby, Texas Instruments, 1958
 - Robert Noyce, Fairchild Semiconductor, 1959
 - IBM S/360, 1964

Nobel
1956

Nobel
2000



15.11.2012

Copyright Teemu Kerola 2001

12