Muistinhallinnan tekniikat sulautetuissa järjestelmissä

Turun yliopisto
Tietotekniikan laitos
TkK-tutkielma
Joulukuu 2023
Matias Suksi

TURUN YLIOPISTO
Tietotekniikan laitos

MATIAS SUKSI: Muistinhallinnan tekniikat sulautetuissa järjestelmissä

TkK-tutkielma, 8 s.

Joulukuu 2023

Sovelluksen tehokas muistinhallinta on keskeisessä osassa sulavan käyttökokemuksen takaamisessa. Sulautettujen järjestelmien tuomat haasteet korostavat muistinhallinnan merkistystä entisestään. Muistin, prosessorin ja laitteiston komponenttien ominaisuuksien rajallisuus asettavat kehittäjälle haasteita, joiden ratkaisut voivat vaatia kehittäjältä hyvinkin kustomoituja ja vaativia rakenteita, jos vertaillaan PC-tietokoneille kehitettävien ohjelmien muistin rakennetta.
Kirjallisuuskatsauksessa tullaan tutkimaan sulautettujen järjestelmien muistinahallintaa näiden rajoitteiden vaikuttaessa. Päätutkimuskysymyksenä on "Millaisia muistinhallinnan tekniikoita voidaan hyödyntää sulautetuissa järjestelmissä". Katsauksessa tullaan käsittelemään sovelluksen muistin ja muistinhallinnan teoriaa, ja perustietoa sulautetuista järjestelmistä. Näiden käsitteiden ymmärtäminen on keskeistä varsinaisten muistinhallinnan tekniikoiden ja rakenteiden ymmärtämisessä. Kirjallisuuskatsaus keskittyy kehittäjän omiin henkilökohtaisiin ratkaisuihin ohjelmointikieli työkalunaan. Katsauksessa ei tulla käymään läpi kuin ainoastaan pintapuoleisesti ohjelmointikielien sisäisiä muistin allokointiominaisuuksia ja -algoritmeja. Tämä rajaus sivuuttaa muutamia tärkeitä aihepiirejä tämän katsauksen ympärillä, kuten mm. roskien keruun. Katsauksessa on valittu esimerkkiohjelmointikieleksi C konseptien havainnollistamiseksi. Valinta on perusteltu C:n alkuperäisen luonteen vuoksi sekä se on yksi yleisimmistä ohjelmointikielistä sulautetuissa järjestelmissä. Lisäksi esimerkit C:llä voidaan yleistää C:n laajennukselle C++:lle, joka on myös yksi yleisimmistä kielistä sulautetuissa järjestelmissä.
Tietoa on haettu mm. Google Scholarista, IEEE Xplore:sta sekä ACM Digital Librarysta, ja sitä on haettu hakulauksekkeella: “embedded system” AND (”memory allocat*” OR “memory manag*”) AND (technique* OR method* OR solution*).

Asiasanat: muistin allokointi, pino, keko

Sisällys

1 Johdanto

2 Muistinhallinta

Ohjelman muistin rakenteen ymmärtäminen on erityisen tärkeää tehokkaan muistin käytön saavuttamiseksi. Seuraavissa luvuissa tullaan esittelemään miten muistin allokointi ohjelmissa toimii ja millaisista muistialueista ohjelman muisti koostuu. Huomioitavaa on, että seuraavaksi esiteltävä muistirakenne on tyypillisin muistirakenne, mistä tietokoneohjelman muisti koostuu. Ohjelman muistin rakenteeseen vaikuttaa mm. käytössä oleva suoritinarkkitehtuuri, ohjelmointikielen kääntäjä sekä kääntäjien tarjoamat muistin optimointityökalut.

2.1 Ohjelman muisti

Ohjelman muisti jakautuu erilaisiin muistialueisin. Koodisegmentti sisältää varsinaisesti ajettavan ohjelman binäärin eli ohjelmatiedoston, jonka prosessori suorittaa. Lisäksi ohjelmalla on olemassa datasegmentti, joka koostuu alustetusta datasta ja alustamattomasta datasta. Alustetun datan alueeseen kuuluvat globaalit ja staattiset muuttujat sekä vakioarvoiset muuttujat, joille on alustettu jokin arvo. Alustamattomassa data-alueessa on kaikki alustamaton data eli muuttujat, jotka ovat esitelty (declare), mutta joille ei ole annettu mitään arvoa.[1] Näiden muistialueiden data allokoidaan ajettavan ohjelman muistiin jo käännön aikana (compile time) eli ohjelmointikielen kääntäjän kääntäessä lähdekoodin. Ohjelmalla on lisäksi kaksi muistialuetta pino ja keko, joiden data allokoidaan vasta ajoaikana (runtime) eli kun ohjelman itse suorittaminen aloitetaan.

/**Tähän kohtaan kuva muistista**/

/**Koodin pätkä**/

2.1.1 Pino

2.1.2 Keko

2.2 Muistin allokointi

Muistia voidaan allokoida ohjelmalla kahdella tavalla, staattisesti tai dynaamisesti.

3 Ohjelmointikielien muistinhallinnan työkalut

3.1 Mid-level kielet

3.2 Low-level kielet

4 Sulautetut järjestelmät

4.1 Mikä on sulautettu järjestelmä

4.2 Sulautettujen järjestelmien haasteet kehittäjille

5 Muistinhallinan tekniikoita ja rakenteita

6 Tekniikoiden soveltaminen sulautetuissa järjestelmissä

7 Yhteenveto

Lähdeluettelo

[1] L. Ferres, ""Memory management in C: The heap and the stack"", 2010.