



Kursplan för läsåret 2010/2011
(Genererad 2010-06-28.)

MULTICOREPROGRAMMERING

Multicore Programming

EDAN25

Antal högskolepoäng: 6. **Betygsskala:** TH. **Nivå:** A (Avancerad nivå). **Huvudområde:** Teknik. **Undervisningsspråk:** Kursen ges på svenska. **Överlappar följande kurs/kurser:** EDA116. **Valfri för:** D4, D4is, D4pv, F4, Pi4. **Kursansvarig:** Universitetslektor Jonas Skeppstedt, Inst f datavetenskap. **Förkunskapskrav:** EDAA01 Programmeringsteknik - fördjupningskurs eller EDA027 Algoritmer och datastrukturer, samt de obligatoriska momenten i kursen EDAF15 Algoritmimplementering. Får läsas samtidigt som EDAF15. **Prestationsbedömning:** Skriftlig tentamen. För godkänt betyg krävs godkänd tentamen, godkända datorlaborationer samt godkänd inlämningsuppgift (projekt). Projekt utgörs av ett program som ska skrivas dels för en multiprocessor med fyra processorer, och dels för en Cellprocessor. Slutbetyg i kursen grundar sig på resultatet av den skriftliga tentamen. Kursen får läsas samtidigt som EDAF15. **Hemsida:** <http://cs.lth.se/>.

Syfte

Kursens syfte är att studenterna ska använda kunskapen från EDAF15 Algoritmimplementering vid programmering på multicores, dvs att lära sig att implementera en algoritm så effektivt som möjligt på en given multicoredator, dels för traditionella multiprocessorer med delat minne, och dels för multicores där kommunikation mellan cores hanteras explicit i mjukvara som t.ex med DMA på Cellprocessorn, samt att förstå fördelar och nackdelar med dessa två olika multicorearkitekturer.

Mål

Kunskap och förståelse

För godkänd kurs skall studenten

- förstå hur cacheminnen fungerar på multicores
- förstå vikten av att minska antal cachemissar
- förstå vikten av load-balancing mellan olika processorer
- förstå hur Cellprocessorn och dess DMA-baserade kommunikation fungerar
- förstå hur man kan dela upp ett sekventiellt C program med hänsyn tagen till ovanstående
- förstå fördelar och nackdelar med olika multicorearkitekturer dels med avseende på (1) prestanda och dels på (2) hur komplicerad programmeringen blir

Färdighet och förmåga

För godkänd kurs skall studenten

- kunna använda Pthreads för att skapa trådar och synkronisera dessa på en multiprocessor
- kunna använda OpenMP för att parallellisera beräkningsintensiva sekventiella C program
- kunna använda Cellprocessornas DMA-primitiver för att i varje core uppnå parallellism mellan beräkning och kommunikation
- kunna förbättra prestandan på ett givet sekventiellt program genom att parallellisera detta på ett sätt som maximerar prestandan dels på en traditionell multiprocessor och dels på Cellprocessorn.

Värderingsförmåga och förhållningssätt

För godkänd kurs skall studenten

- kunna hitta en fördelaktig version av en algoritm med hänsyn taget till tillämpningens krav och arkitekturen hos den aktuella multiprocessorn som ska användas
- kunna välja rätt multicorearkitektur för en given algoritm.

Innehåll

Multiprocessorer, klassificering av cachemissar, parallellisering av ett sekventiellt program genom decomposition, assignment, orchestration, och mapping, owner-computes regeln, memory consistency models, sequential consistency, weak ordering, release consistency, Pthreads, OpenMP, Cellprocessorn, aktuell forskning inom multicores.

Litteratur

Skeppstedt, Söderberg: Writing Secure and Efficient C Code - a Thorough Introduction for Java Programmers (ISBN ej klart).