



# INSTITUTIONEN FÖR DATA- OCH INFORMATIONSTEKNIK

## **DIT162 Realtidssystem, 7,5 högskolepoäng**

Real-Time Systems, 7.5 credits

*Avancerad nivå / Second Cycle*

---

### **Fastställande**

Kursplanen är fastställd av Institutionen för data- och informationsteknik 2017-12-19 att gälla från och med 2018-08-19, höstterminen 2018.

*Utbildningsområde:* Teknik 50 % och Naturvetenskapligt 50 %

*Ansvarig institution:* Institutionen för data- och informationsteknik

### **Inplacering**

Kursen erbjuds inom flera utbildningsprogram. Den ges även som fristående kurs vid Göteborgs universitet.

Kursen kan ingå i följande program: 1) Datavetenskapligt program (N1COS), 2) Computer Science, Master's Programme (N2COS), 3) Software Engineering and Management masterprogram (N2SOF), 4) Applied Data Science masterprogram (N2ADS) och 5) Game Design & Technology masterprogram (N2GDT)

#### *Huvudområde*

Datavetenskap

#### *Fördjupning*

A1F, Avancerad nivå, har kurs/er på avancerad nivå som förkunskapskrav

### **Förkunskapskrav**

För tillträde till kursen krävs att studenten har klarat följande kurser, eller motsvarande:

- DIT391 Principer för parallell programmering, 7,5 hp
- DIT401 Operativsystem, 7,5 hp
- DIT151 Maskinorienterad programmering, 7,5 hp

## Lärandemål

Efter godkänd kurs ska studenten kunna:

### *Kunskap och förståelse*

- formulera krav för inbyggda datorsystem med strikta önskemål på fördröjning och periodicitet hos datorberäkningarna
- kategorisera och beskriva de olika lagren i en systemarkitektur för inbyggda realtidssystem
- beskriva de principer och mekanismer som används för att konstruera run-time system och kommunikationsnätverk för realtidstillämpningar
- beskriva hur de generella principerna bakom programmering av realtidssystem realiserar i olika högnivåspråk

### *Färdigheter och förmåga*

- konstruera parallellt exekverande processer (programvarumoduler) för realtidstillämpningar med gränssnitt mot hårdvaruenheter (sensorer/ställdon)
- tillämpa de grundläggande analysmetoder som används för att verifiera den tidsmässiga korrektheten hos en uppsättning processer

### *Värderingsförmåga och förhållningssätt*

- reflektera och argumentera i skrift om etiska aspekter gällande valet av systemimplementering

## Innehåll

Denna kurs syftar till att ge grundläggande kunskaper om metoder för konstruktion och analys av realtidssystem. Exempel på realtidssystem är styrsystem för bilar, flygplan och rymdfarkoster, samt datorspel och multimediatillämpningar.

Ett inbyggt system är ett datorsystem som är konstruerat för att utföra en eller ett fåtal specifika funktioner. Det är inbyggt i den mening att det är en del av en komplett enhet, ofta omfattande elektrisk hårdvara och mekaniska delar. Av säkerhets- och användbarhetsskäl har vissa inbyggda system strikta krav på icke-funktionella beteenden såsom fördröjning och periodicitet hos datorberäkningarna. Sådana datorsystem kallas realtidssystem. Exempel på realtidssystem är styrsystem för bilar, flygplan och rymdfarkoster, samt datorspel och multimediatillämpningar. Denna kurs syftar till att ge grundläggande kunskaper om metoder för konstruktion och analys av realtidssystem.

På grund av de extremt höga kostnader som följer av en sen upptäckt av problem i inbyggda system är det viktigt att god designmetodik används under utvecklingen av mjuk- och hårdvaran. En sådan metod är att använda en systemarkitektur som tillhandahåller bra abstraktioner av mjukvarukomponenter och som understödjer enkel

sammankoppling av sådana komponenter. Filosofin bakom systemarkitekturer dikterar att mjukvaran i ett inbyggt system organiseras i multipla parallellt exekverande processer, där varje process (eller grupp av processer) implementerar en specifik funktionalitet i systemet. Uppdelningen i processer gör det möjligt att på ett naturligt sätt bryta ner ett komplext system i mindre delar som är enkla att förstå, implementera och underhålla.

I kursen används programspråket C, utökat med ett mjukvarubibliotek som tillhandahåller stöd för programmering av parallella processer med tidskrav. Ett viktigt syfte med kursen är att demonstrera hur det utökade programspråket C används för att realisera kommunikation och synkronisering mellan processer, resurshantering och ömsesidig uteslutning. Eftersom andra programspråk använder monitorer och semaforer för att realisera dessa funktioner, innehåller kursen också en genomgång av sådana tekniker. Dessutom demonstreras det i kursen hur man använder lågnivåprogrammering i C för att realisera avbrottsstyrd interaktion med hårdvaruenheter. För att demonstrera de generella principerna bakom programmering av realtidssystem ger kursen också exempel på hur dessa principer realiseras i andra programspråk, som Ada och Java.

För att möjliggöra exekvering av ett program som innehåller parallella processer används ett runtime-system (realtidskärna) som fördelar den totala tillgängliga beräkningskapaciteten hos systemets mikroprocessor(er) mellan processerna. Kursen beskriver hur ett enkelt runtime-system är organiserat. Exekveringsordningen mellan processerna bestämmer runtime-systemet med hjälp av en schemalägningsalgoritm. Kursen presenterar tekniker baserat på såväl cykliska tidtabeller som statiska eller dynamiska processprioriteter. Dessutom presenteras protokoll för hantering av delade hård- och mjukvaruresurser. Eftersom många av dagens realtidstillämpningar är distribuerade över multipla datornoder innehåller kursen också en introduktion till topologier och mediamaccessstekniker för några vanligt använda kommunikationsnätverk.

I realtidssystem, där processer har strikta tidskrav, är det nödvändigt att göra a priori analys av systemets schemalägningsbarhet. Kursen presenterar tre olika analysmetoder för system som schemalägger processer med statiska eller dynamiska prioriteter: utnyttjandegradsbaserad analys, svarstidsanalys och beräkningsbehovsanalys. I samband med detta ges också en insikt i hur det maximala resursbehovet (värsta-fallets exekveringstid) för en process beräknas.

### *Delkurser*

1. **Skriftlig tentamen** (*Written examination*), 4,5 hp  
Betygsskala: Väl godkänd (VG), Godkänd (G) och Underkänd (U)
2. **Laboration** (*Laboratory work*), 3 hp  
Betygsskala: Väl godkänd (VG), Godkänd (G) och Underkänd (U)

**Former för undervisning**

Kursen utgörs av en föreläsningsserie samt ett antal salsövningar där programmeringmetoder och teorier från föreläsningarna praktiseras. Kursen examineras medelst en skriftlig tentamen. Dessutom ingår en obligatorisk laborationsuppgift som syftar till att eleverna skall ta fram programvara för en inbyggd realtidstillämpning som exekveras på ett hårdvarusystem bestående av multipla datornoder som kommunicerar via ett bussnätverk. Uppgiften omfattar, förutom programmering av samverkande parallella processer, även maskinnära programmering av hårdvaruenheter som realtidsklockor och nätverkskontrollers.

*Undervisningsspråk:* engelska

**Former för bedömning**

Studenten examineras medelst en skriftlig tentamen och en obligatorisk laborationsuppgift.

Om student som underkänts två gånger på samma examinerande moment önskar byte av examinator inför nästa examinationstillfälle, ska sådan begäran inlämnas skriftligt till kursansvarig institution och bifallas om det inte finns särskilda skäl däremot (HF 6 kap § 22).

I det fall en kurs har upphört eller genomgått större förändringar ska studenten i normalfallet garanteras tillgång till minst tre provtillfällen (inklusive ordinarie provtillfälle) under en tid av åtminstone ett år med utgångspunkt i kursens tidigare uppläggning.

**Betyg**

På kursen ges något av betygen Väl godkänd (VG), Godkänd (G) och Underkänd (U). För godkänt betyg på hel kurs krävs godkänt betyg på samtliga delkurser.

För att få betyget Väl godkänd (VG) på kursen krävs betyget VG på båda delkurserna.

**Kursvärdering**

Kursen utvärderas genom möten både under och efter kursen mellan lärare och studentrepresentanter. Därutöver används en anonym enkät för att få skriftlig information. Resultatet av utvärderingen används för att förbättra kursen genom att visa på delar som kan läggas till, förbättras, ändras eller tas bort.

**Övrigt**

Kursen är samläst med Chalmers.

Kurslitteratur kommer att publiceras senast 8 veckor innan kursstart.

Kursen ersätter kursen DIT161, 7,5 hp. Den här kursen kan inte ingå i en examen som innehåller DIT161. Den kan inte heller ingå i en examen som bygger på en annan examen där DIT161 ingår.