

Programmering enligt Stansaab Journalen nov. 1972

Inlagt av Björn Sölving ons, 2018-08-01 22:10

Programmering - hjärnarbete med lovande framtid

Erik Åhman har skannat en intressant artikel från 1972 (redaktör Rolf Langborger). Här följer artikeln och därefter finns Bosse Lindestams kommentarer från maj i år.

Programmering - hjärnarbete med lovande framtid



Bo Lindestam, Hjärnarbetare och chef för systemprogrammeringen.

I beskrivningen av företagets verksamheter har turen nu kommit till systemsidan. System är ett begrepp som sammanfattar en stor mängd aktiviteter och kartlägger hur utrustningar samarbetar för att lösa operativa krav. En viktig del av systemarbetet är programmering. Genom program för en dator kan man styra system i avsedda funktioner, dvs. ge systemet dess tillämpningsegenskaper. Till systemarbete hör också att specificera och sammankoppla de apparatenheter som erfordras för dessa funktioner. På STANSAAB sköts detta arktitektarbete för radardatasystem av systemledningen under Ingvar Andersson. Ett mycket intimt samarbete mellan systemledning och systemprogrammering under Bo Lindestam fordras för att kunna konkurrera om de stora internationella projekten på radardatasystem. Båda funktionerna sysselsätter tillsammans drygt 40 medarbetare under ledning av systemchefen Kjell Mellberg, som därutöver svarar för entreprenader och samordning, installation, driftsättning och utprovning samt de interna datacentralerna.

Det är alltså ett stort område av företagets verksamhet, som för att bli gripbart i denna översikt kommer att belysas med programmering.

Var programmerar man idag?

Förutom systemprogrammering under Bo Lindestam bedrivs programmeringsarbete för datacentralerna, för patientdatasystem och för generell utveckling av interna ADB-rutiner av Rolf Johanson. För program för styrning och provningsändamål vid tillverkningen svarar Arne Söderkvist och för anpassning av dataterminalsystem till olika kundkrav Björn Sölving. Tyngdpunkten i de kundorienterade uppgifterna ligger på systemprogrammering för radardatasystem, där

man har alla bitarna samlade under en hatt.

Programarbetets delar

Programmerarna deltar i kundkontakter, offertarbete, uppföljning av offerter, varefter specifikation, konstruktion, produktion, integration, leveranskontroll och underhåll utföres för erhållna projekt.

Vad uppnås?

Genom att arbetet är projektorienterat uppnår man stor smidighet i att förbättra och modernisera system och även i att snabbt fånga upp och rätta till saker som av en eller annan anledning gått snett. Nackdelen, om någon, är att den ställer stora krav på personalen, som för att bli effektiv, måste täcka mycket stora kunskapsområden av programmerings- och systemarbetet. Men detta gör även att arbetet blir mycket intressant och omväxlande.

Vilka krav ställer man på programmerare?

En programmerare måste kunna tänka både översiktligt och i minsta detalj. Det är sannolikt en egenskap som alla goda programmerare har gemensamt. Om det för såväl en konstruktör som en programmerare gäller att tänka på en funktions direkta egenskaper, måste en programmerare också tänka på indirekta, helt enkelt vara listig. Han måste ständigt ställa frågan: Vad händer om operatören ej sköter sitt jobb enligt föreskrifterna? Han måste alltså konstruera programmen mot användarnas alla tänkbara missgrepp. Systemen får ju inte vara helt beroende av att människan alltid gör rätt. Dessa krav skapar en ständig öppenhet för systemstrategier och det är en egenskap som krävs. Han måste också kunna tänka i symboler och verktygen där heter

flödesplaner, källkoder, Boolsk algebra med de nya matematiska uttryckssätt, som brukar kallas mängdlära. Alla tankar måste kunna uttryckas verbalt, vilket för övrigt är en förutsättning för att kunna tänka. Programmering företer vissa likheter med språk, där t. ex. satsläran med sin analys av subjekt, predikat och objekt går att direkt omsätta till programmering. I en programsats är "jag" alltid subjekt, operatören (instruktionen) verbet och operanden (talet som skall behandlas) objektet.

För de förutbestämda eller deterministiska system vi arbetar med räcker detta. Varje tillstånd i systemet går att definiera, liksom varje övergång från ett tillstånd till ett annat. Huruvida det räcker även för s. k. icke förutbestämda system, såsom system med artificiell intelligens forskar man intensivt med världen över. Kanske blir det delar av våra framtida system.

Vilka blir programmerare?

Egentlig utbildning för programmerare bedrivs för närvarande i Sverige bara av Skolöverstyrelsen genom kurser på postgymnasial nivå och på universitetsnivå med teoretisk-praktisk inriktning.

De flesta programmerare är troligen självlärda. Från ett intresse för problemlösning väckt i gymnasium eller fackskola har de dragits till jobbet genom sitt nötknäpparintresse. Skolan är hård och kan ofta bara utvecklas genom praktiskt arbete med motot: Skriv ett program och se vad som händer. Lär av dina misstag.

På detta sätt blir man bra programmerare som sedan kan arbeta effektivt i en värld som är präglad av en långt driven systematik för att förverkliga kundernas krav på våra system. Och det är väl den personliga utvecklingen som behövs i allt nyskapande och som försvarar rubriken Programmering — Hjärnarbete med lovande framtid!

Bo Lindestam har haft vänligheten att kommentera ovanstående artikel (som nu skrevs för 46 år sen):



Bosse 2018

Erik väcker de "djupaste och mörkaste" minnen till liv genom att påminna om denna artikel.

Jag tror mig veta att ingen artikel i StansaabJournalen har fått en liknande behandling, men låt mig börja med fakta.

Som vi vet var Rolf Langborger som PR-man (numera heter det informationsdirektör) med ansvar för broschyrer och annonser även redaktör för StansaabJournalen. Han var onekligen en bra skribent, även om superlativen ibland staplades litet väl mycket på varandra, men som det ankommer en god PR-man.

Rolf och jag hade en bra relation och han hade vid ett flertal tillfällen bett om en intervju för StansaabJournalen, just när det hade hänt saker och började hända nya saker i många av buskarna, SATS, PHAROS, SEROS, ATCAS, TERCAS, i projekt där systemprogrammering var frekvent representerat.

Intervjun blev av och vi satt flera timmar och pratade, och ett tema kommer jag särskilt ihåg: att göra det begripligt för alla, eftersom det var menat för en personaltidning. Rolf skrev sin intervjuartikel, ställde några kompletterande frågor och så publicerades intervjun utan att jag fick se hans slutliga opus.

Men det dröjde inte många timmar förrän artikeln blev en stor facklig fråga. Om jag inte minns rätt reagerade SIF (inga namn nämnda) till Gunnar Wedell om karaktären på texten. "Vad tror de sig vara för några undermänniskor." Man hade reagerat på de uttalade behovet av specialkunskaper och personliga egenskaper, som återspeglades i intervjun. Och som vanligt i ett sådant fall skjuter man gärna på pianisten. Jag, som intervjuoffer, blev föremål för mycket negativa kommentarer. Hur Rolf Langborger behandlades vet jag faktiskt inte, men jag vet att som en följd av denna artikel tillsattes ett fackligt redaktionsråd (på initiativ av SIF?) som skulle vara en ryggsäck på Rolf och förhindra en upprepning av dessa elitistiska tendenser. Någon som kommer ihåg mera om detta får gärna berätta mera.

Som ni som läser detta nu år 2018 kanske under tiden har uppfattat, så var våra elitistiska ambitioner mycket låga. Vi hade fullt upp med att lösa våra systemprogrammeringsproblem utan vidare åthävor. Och tillräckligt begåvat folk kunde vi inte få nog av under denna period.

Intervjun gav faktisk även upphov till en annan diskussion om inriktningen och uppdelning av systemverksamheten, som senare skedde.

Jag hävdade med bestämdhet och till Gunnar Wedells förtret och fortsätter att hävda att en god realtidsprogrammerare bör ha god insikt i och kunskap om det målsystem, som hans programmeringsmödor är avsedda för, både logiskt/operativt och driftmässigt/dynamiskt. Våra dåvarande chefer, men sin IBM-bakgrund, hade mycket litet kunskap och känsla för verkliga realtidssystem, som ju utgjorde kärnan och nyckeln till våra system. Varje verksamhet i det stora bolaget, som jag vid några tillfällen hade tittat in i, visade sig ofta vara grundad på huvuddoktrinen att "tok gör som tok säger" (dvs. den som specificerar (tok1) har ingen feedback från verkligheten (tok 2 som kodar) så att det som tok1 specificerade inte har en chans att bli riktigt förrän det är för sent, dvs det måste "rättas" i omgångar innan det blir riktigt riktigt!!), som genom åren har visat sig ge upphov till en mängd icke-fungerande system, även i denna dag, eller har jag fel?

Men vi vet ju att chefen alltid har rätt, och så blev det.

/Bo Lindestam 2018-05-06