

# **Tidig programmering**

Transkript av ett vittnesseminarium vid Tekniska museet i  
Stockholm den 16 mars 2006

—

**Per Lundin (red.)**

Stockholm 2007

Avdelningen för teknik- och vetenskapshistoria  
Skolan för arkitektur och samhällsbyggnad  
Kungl. Tekniska Högskolan  
100 44 STOCKHOLM

Working Papers from the Division of History of Science and Technology

TRITA-HST 2007/1

Redaktör: Martin Emanuel

ISSN 1103-5277

ISRN KTH/HST/WP-2007/1-SE

ISBN 978-91-7178-601-2

Omslagsbilderna visar vittnesseminariets deltagare. Överst t.v.: Hans Riesel och Ingemar Dahlstrand. Överst t.h.: Elsa-Karin Boestad-Nilsson. Nederst fr.v.: Göran Waernér, Bengt Gällmo, Björn Kleist, Bjarne Däcker, Hans Riesel, Ingemar Dahlstrand, Elsa-Karin Boestad-Nilsson och Göran Kjellberg.

Fotograf: Nisse Cronstrand, Tekniska museet.

Tryck: Universitetsservice US-AB, Stockholm 2007

## **Abstract**

The witness seminar "Tidig programmering" [Early programming] was held at Tekniska museet [The National Museum of Science and Technology] in Stockholm on 16 March 2006 and led by Lars Arosenius. Different aspects of early programming and the development of programming languages were covered at the seminar. The physical difficulties in the programming of the relay computer BARK were observed. Furthermore the problems that the restricted memory capacity of the early computers brought to their programming were discussed. The importance of the capacity to create program libraries and manage subroutines was discussed. The first operating systems were mentioned. Attention was paid to the design of the early programming language ALFAKOD used on the BESK and FACIT EDB computers. Also the founding of the early software companies Autocode AB and Nordisk ADB AB was mentioned. Moreover the development of a compiler for the ALGOL programming language at Facit AB was dealt with. This work led to the development of the ALGOL-GENIUS language at Saab AB during the first half of the 1960s. Finally the special requirements on programming languages for telecommunication, which led to the development at Ericsson of the ERI-PASCAL and ERLANG programming languages, were debated.



## Förord

Vittnesseminariet ”Tidig programmering” ägde rum vid Tekniska museet i Stockholm den 16 mars 2006, och arrangerades inom ramen för projektet ”Från matematikmaskin till IT” som är ett samarbete mellan Avdelningen för teknik- och vetenskapshistoria vid KTH, Dataföreningen i Sverige och Tekniska museet. Det spelades in med ljud och bild samt transkriberades därefter. Per Lundin vid Avdelningen för teknik- och vetenskapshistoria har i samråd med seminariedeltagarna redigerat transkriptet. De redaktionella ingreppen är varsamma och har skett i syfte att öka tydlighet och läsbarhet. Vissa strykningar har gjorts. Det redigerade transkriptet är fackgranskat av Ingemar Dahlstrand och Sten Henriksson. Originalinspelningen finns tillgänglig på Tekniska museet i Stockholm. Seminariet finansierades med bidrag från KK-stiftelsen, Riksbankens Jubileumsfond samt Stiftelsen Marcus & Amalia Wallenbergs minnesfond.

**Deltagare:** Lars Arosenius (ordf.), Elsa-Karin Boestad-Nilsson, Ingemar Dahlstrand, Bjarne Däcker, Bengt Gällmo, Göran Kjellberg, Björn Kleist, Hans Riesel, Göran Waernér.

**Inbjudna med förhinder:** Sten Kallin, Börje Langefors, Sven Yngvell.

**Övriga närvarande vid seminariet:** Åke Bengtsson, Henrik Dahlstrand, Peter Du Rietz, Inger Gran, Tord Jöran Hallberg, Bo Hallmén, Rolf Karsvall, Kurt Katzeff, Bo Lindestam, Per Lundin, Lars-Olof Norén, Tomas Ohlin, Gert Persson, Per Olof Persson, Anders Rockström, Gunnar Stenudd, Wichartbhol Surapolbhichet, Björn Sölvig, Lars Wictorin.



## Tidig programmering

**Lars Arosenius:**<sup>1</sup> Ja, då kan vi börja. Välkomna till det här vittnesseminariet om programmering och programspråksutveckling. Vi har sagt att vi tittar på utvecklingen fram till 1970. Jag heter Lars Arosenius och blivit anförtrodd uppgiften att leda det här vittnesseminariet. Meningen med ett vittnesseminarium är att vi ska få fram personliga synpunkter från alla som har varit med om en intressant utveckling, och programmering är verkligen en intressant utveckling som vi kan se tillbaka på. Och det här är en del av ett projekt som Dataföreningen i Sverige driver, och här har vi Dataföreningens VD, Inger Gran, som är en av dem som dragit igång det här, och sen har vi primus motor i projektet Per Olof Persson.

Temat är programmering och programspråk och hur de har utvecklats, och det finns naturligtvis ett antal rätt urskiljbara skeenden i datorutvecklingen. På 40-talet var ju fokus på att överhuvudtaget få elektroniska maskiner att fungera. Det var den stora utmaningen. På 50-talet så skulle man göra de där maskinerna ekonomiskt användbara, och det var oftast stora företag som använde maskinerna. På 60-talet blev det mera en fråga om hur maskinerna skulle kunna användas även av folk som inte hade fått så väldigt djup utbildning, och det var då som programspråk och programmering på allvar kom fram som en profession. De tidiga programmen lagrades som kopplingar i propptavlor, det var vad man använde på hålkortsmaskiner, i ENIAC,<sup>2</sup> för Mark 1,<sup>3</sup> men det var när de von Neumannska koncepten<sup>4</sup> kom in som man började se programmering som en intellektuell utmaning.

Redan i september 1945 skrev ju Eckert och Mauchly,<sup>5</sup> som var det team som låg bakom ENIAC, och som börjat fundera på efterföljaren till ENIAC, EDVAC,<sup>6</sup> som väl aldrig riktigt blev färdig: ”in a typical application, the same sequence of arithmetic processes must be carried out over and over again, but each time on a new set of numbers. To repeat such arithmetic orders each time they are to be used would be wasteful of memory capacity as well as wasteful of coding time, and its obviously unnecessary [...] The directions for carrying out this work need to be inserted in the memory only once, and the basic order code must provide for finding these instructions each time that they

---

<sup>1</sup> Lars Arosenius, f. 1929, civ.ing. elektro (E), KTH 1952, tekn.lic. Han var forskningsingenjör vid Försvarets forskningsanstalt (FOA) 1951–53, forskningsingenjör och lärare vid KTH 1953–56, forskningsingenjör vid Matematikmaskinnämnden (MMN) 1956–59, teknisk chef vid ABN-bolagen samt AB Gylling & Co 1959–65. Arosenius hade därefter diverse chefspositioner vid IBM Nordiska Laboratorier 1966–77 samt vid IBM Svenska AB 1977–91.

<sup>2</sup> ENIAC, förkortning för Electronic Numerical Integrator and Computer, anses vara den första elektroniska datorn. Den konstruerades av J. Presper Eckert och John Mauchly för amerikanska armén vid University of Pennsylvania i Philadelphia, stod färdig 1946 och flyttades 1947 till Aberdeen Proving Ground, Maryland, där den var i bruk till 1957. Paul E. Ceruzzi, *A History of Modern Computing* (Cambridge, Mass., 1998), 14ff.

<sup>3</sup> Eg. Harvard Mark I, dator konstruerad av Howard Aiken vid IBM samt vid Harvard University med finansiellt stöd av IBM mellan 1939 och 1944. James W. Cortada, *Historical Dictionary of Data Processing: Technology* (New York, 1987a), 187f.

<sup>4</sup> Den ungersk-amerikanske matematikern John von Neumann, 1903–1957, utarbetade 1945 principerna för datorns konstruktion, den s.k. von Neumann-arkitekturen, som ännu är den dominerande.

<sup>5</sup> J. Presper Eckert, 1919–1995, och John Mauchly, 1907–1980, konstruerade tillsammans ENIAC under åren 1943–46. De grundade senare företaget Eckert-Mauchly Computer Corporation och byggde datorn UNIVAC.

<sup>6</sup> EDVAC, förkortning för Electronic Discrete Variable Computer, uppföljare till ENIAC som huvudsakligen konstruerades av Eckert och Mauchly 1945–46. Konstruktionsarbetet avslutades emellertid inte förrän 1951. Cortada (1987a), 151ff.

are needed, and for returning to different parts of the problem orders after each use”.<sup>7</sup> Det var alltså en klar beställning på subrutiner som man redan hade 1945.

De tidigaste datorerna hade ju enorma begränsningar, men jag tycker man ska ha i minnet att 1957 så uppskattade man att det totala antalet datorer i USA var kapabla att tillsammans åstadkomma tio miljoner instruktioner per sekund. Tio miljoner instruktioner per sekund var den totala beräkningskapacitet som USA kunde ställa upp 1957. Det är en hundradel av vad en PC som man köper i dag kan åstadkomma. Men detta var alltså den totalt installerade beräkningskapaciteten i USA på den tiden. Och det var framförallt stora begränsningar i minnet, och det gjorde att människorna var tvungna att anpassas till maskinernas språk. Det var maskinerna som dirigerade hur operatörerna och användarna skulle få nånting nyttigt ut av dem. Och så småningom såg man att det här ledde till stora kompatibilitetsproblem. Det är ganska intressant att se att 1955 då skrev National Bureau of Standards en ganska komplett beskrivning av ett system som de hade gjort som hette SEAT, och som redogjorde för alla erfarenheter som man hade haft av systemet.<sup>8</sup> Inte en rad om programmering. Det fanns inte en antydning till att man intresserade sig för programmeringssvårigheterna.

Hur tacklade vi då de här problemen i Sverige? Det är det som jag tycker att det här vittnesseminariet ska ägna sig åt. Till hjälp har vi en panel som vi tänkte skulle sätta tonen genom att ge korta inledningar, och sen hoppas jag att alla andra inbjudna kan komma in med sina egna personliga synpunkter och erfarenheter. Här har vi Göran Kjellberg som är den verkliga pionjären inom området. Du har programmerat allt från BARK och BESK till modern tid. Vi har Elsa-Karin Boestad från FOA. Hans Riesel som väl har utbildat en hel del av alla de programmerare som vi har runt oss. Vi har Ingemar Dahlstrand med sin erfarenhet från Chalmers, Saab, SARA, Facit och ALGOL-utveckling, och allt vad du hållit på med. Vi har Björn Kleist som började som flygingenjör, tror jag, och sen kom till IBM, QZ, Datema. Vi har Bengt Gällmo, labbchef på IBM, och som var en av dem som gjorde ALGOL för IBM. Vi har Göran Waernér som är applikationspionjär för vägplanering. Och sen har vi Bjarne Däcker som kommer från telefonaktiebolaget LM Ericsson och som har varit med om den speciella programmeringsutvecklingen där. Välkomna allihopa i panelen. Och vi har också en skara experter bland oss i övrigt här. Och då så hoppas jag att vi kan börja med seminariet, och Göran, du är ju den verkliga pionjären här, du började som matematiker på Ericsson tror jag. Hur kom du in på det här? Du var ju också en av industrispionerna som Sverige sände ut på –48?<sup>9</sup>

**Göran Kjellberg:**<sup>10</sup> Jaha, jag befann mig i Paris och fick brev från Stockholm av en studiekamrat som berättade att Ingenjörsvetenskapsakademien<sup>11</sup> sökte folk som ville åka till Amerika och lära sig om matematikmaskiner. Jag sökte och hade turen att få ett av

---

<sup>7</sup> Charles J. Bashe, m.fl., *IBM's Early Computers* (Cambridge, Mass., 1986), 318.

<sup>8</sup> National Bureau of Standards Circular 551, Washington, jan 25 1955.

<sup>9</sup> Arosenius syftar på de fem stipendiaterna Carl-Erik Fröberg, Göran Kjellberg, Arne Lindberger, Gösta Neovius och Erik Stemme som på uppdrag av Ingenjörsvetenskapsakademien (IVA) och Tekniska forskningsrådet (TFR) reste till USA 1947–48 för att studera matematikmaskiner.

<sup>10</sup> Göran Kjellberg, f. 1920, fil.kand. i matematik, fysik och teoretisk fysik 1943 samt fil.lic. i matematik vid Stockholms högskola 1960. Han var anställd som matematiker vid LM Ericssons transmissionsavdelning 1944–46. Kjellberg studerade matematikmaskiner vid the Computation Laboratory vid Harvard University i USA 1947–48. I november 1948 anställdes han vid nyinrättade MMN och var byrådirektör där 1953–56. Kjellberg arbetade på mikrovägsavdelningen och senare på MI-divisionen vid LM Ericsson 1956–67. Därefter var han datachef vid Metrobutikerna 1967–74 samt lektor vid KTH 1974–87.

<sup>11</sup> Ingenjörsvetenskapsakademien (IVA) bildades 1919 på initiativ av Axel F. Enström som blev dess förste VD. IVA:s målsättning har varit att överbrygga klyftorna mellan forskning, teknikutveckling och företagande, mellan högskola och industri.

dessa stipendier, och vistades sen ett år vid Harvard hos professor Aiken,<sup>12</sup> blev väldigt hjärtligt mottagen. Jag betraktades som en gäst, formellt blev jag inskriven i Harvard som ”graduate student”. Jag kunde efter behag gå och lära mig andra saker där också, så jag passade ju på med det, men jag ska försöka komma tillbaka till huvudämnet här, till programmeringen.

Det jag uppsnappade i Amerika av programmering var inte så mycket, men vad som pågick i Harvard var konstruktionen av deras tredje maskin. Kontrollenheten, styrenheten, höll på att utformas under det året, och jag fick delta i det där och planera olika villkor, olika hopp, som borde finnas. De hade redan upptäckt att programmeringen borde underlättas på något sätt, så att till maskinen Mark III<sup>13</sup> planerades en kodningsmaskin där programmerarna kunde ha ett antal knappar. Man kan föreställa sig att denna maskin kunde producera en sorts makron, så att det räckte med att kort operation, så skulle maskinen lägga ut en serie maskininstruktioner i binär kod på ett magnetband, och programmen skulle sedan läsas in från detta magnetband i minnet. Mark III hade huvudsakligen trumminne, så det var ingen särskilt snabb maskin. Den hade bara plats för fyratusen instruktioner, eller tal, i minnet. Ja, det är en lång historia, och jag ska inte uppehålla mig med det.

När vi kom hem till Stockholm, på hösten –48, så var Conny Palm<sup>14</sup> redan igång med att organisera planeringen av BARK.<sup>15</sup> Matematikmaskinnämnden<sup>16</sup> hade fattat beslut om att bygga den på hösten, och jag, Gösta Neovius<sup>17</sup> och Arne Lindberger<sup>18</sup> engagerades där som hans assistenter. Och vi började ju med att försöka hjälpa till att bestämma hur många siffror som behövdes till olika problem, om det räckte med si och så många, och om vi skulle ha flytande räkning och var decimalpunkterna skulle sitta och annat sånt. Vi kommer fram till 1950 när BARK stod färdig, och vi skulle börja programmera den. Vi bekymrade oss inte så mycket om det, för att vi hade ju operationslistan, och det var bara att skriva ner i tur och ordning vad maskinen skulle göra. Vi hade ju klart för oss att subrutiner behövdes, så vi ägnade en hel del tid åt att göra såna elementära funktioner så pass effektiva som möjligt genom att förkorta potensserierna så att det skulle bli färre termer, så att programmen inte skulle ta för stor plats. Ja, jag vill framhålla att den attityd

---

<sup>12</sup> Howard Aiken, 1900–1973, matematiker. Aiken ledde konstruktionen av Harvard Mark I. Under slutet av 1940-talet färdigställde han relämaskinerna Harvard Mark II, Mark III och Mark IV.

<sup>13</sup> Eg. Harvard Mark III, dator som konstruerades vid Harvard University under Aikens ledning och för amerikanska flottans räkning. Den stod färdig 1949/50. Cortada (1987a), 190f.

<sup>14</sup> Conny Palm, 1907–1951, docent i teletrafikteknik vid KTH. Hans doktorsavhandling från 1943 om analys av telefontrafik var banbrytande för teletrafikteorin och delar av sannolikhetsteorin. Palm ledde sedan 1948 arbetet med att konstruera en kraftfull siffermaskin vid KTH. Denna realiserades aldrig och istället konstruerade Palm och Gösta Neovius vid MMN en förenklad modell i och med relämaskinen BARK.

<sup>15</sup> BARK, förkortning för binär automatisk relä-kalkylator, Sveriges första dator, konstruerades vid MMN under ledning av Conny Palm och Gösta Neovius. Relämaskinen BARK invigdes 28 april 1950 och utnyttjades huvudsakligen av marinen för ballistiska beräkningar. Datorn avvecklades 1955. Hans De Geer, *På väg till datasambället: Datatekniken i politiken 1946–1963* (Stockholm, 1992), 24ff.

<sup>16</sup> Matematikmaskinnämnden (MMN) inrättades den 26 november 1948 med uppdraget att ”planlägga och leda arbetet med matematikmaskinutrustning för svenska behov och därvid undersöka och pröva föreliggande möjligheter, dels till inköp från utlandet, dels ock till tillverkning inom landet av dylik utrustning”. KTH upplät lokaler åt MMN på Drottninggatan 95 i Stockholm. De Geer (1992), 24.

<sup>17</sup> Gösta Neovius, 1921–2002, civ.ing. E, KTH 1946. Neovius besökte Massachusetts Institute of Technology (MIT) och Harvard University 1947–48 på ett stipendium från IVA. Han var chefskonstruktör för BARK vid MMN. Neovius gick över till LM Ericsson 1953, där han stannade till sin pensionering.

<sup>18</sup> Arne Lindberger, f. 1923, civ.ing. teknisk fysik (F), KTH 1945. Han besökte Cambridge och Manchester i Storbritannien samt IBM i New York 1948 för att studera matematikmaskiner. Lindberger arbetade vid MMN 1948–54 innan han gick över till IBM Svenska AB och senare IBM Nordiska Laboratorier. Lindberger flyttade 1964 till Seattle i USA, där han var forskare vid Boeing. Han var t.f. professor i regler-teknik vid Linköpings universitet 1974–75.

vi hade till programmering var att det var ganska lätt. De fysiska besvärligheterna med att koppla upp sladdar var besvärande och tidsödande, och i jämförelse med det var det ju ganska lätt att skriva programmen. Och en annan synpunkt var rädslan för att maskinen skulle räkna fel, för det gjorde den. Reläer kan göra tillfälliga fel och livslängden är inte så lång som på elektronrören, så tillfälliga fel var saker som faktiskt inträffade rätt ofta. Så vid programmeringen fick man ägna ordentlig eftertanke åt att ha kontroller inlagda, så att man upptäckte dessa fel, så att man kunde avbryta körningen och starta om så behövdes. Ja, det kan räcka som första introduktion till programmeringen för BARK.<sup>19</sup>

**Lars Arosenius:** Tack ska du ha. När du var på Harvard, mötte du Grace Hopper<sup>20</sup> där någon gång?

**Göran Kjellberg:** Ja då, det gjorde jag. Hon var då chefsprogrammerare och sysselsatt med programmering för Mark I och förmodligen Mark III.

**Lars Arosenius:** Grace Hopper var ju den stora pionjären, och om jag förstått det rätt så var hon den första som gjorde en ”compiler” av något slag.

**Göran Kjellberg:** Ja.

**Lars Arosenius:** På Mark III.

**Göran Kjellberg:** Nej, alltså hon flyttade ju sedan till Univac.<sup>21</sup>

**Lars Arosenius:** Ja, just det, men långt senare.

**Göran Kjellberg:** Det var ju där som den där compilern gjordes. Det måste vara någon gång mitt på 50-talet.

**Lars Arosenius:** Elsa-Karin, du är ju en av kvinnorna?

---

<sup>19</sup> ”Det kan tilläggas att programmeringen för BARK var ganska lik den som några år senare gjordes för BESK. Man planerade programmen med flödesplaner som var kända bl.a. från de rapporter som spreds från Princeton. Sen hade man som sagt maskinens operationslista som verktyg. Trots att BARK inte hade programmen i minnet var den tack vare sina reläpyramid-väljare, som kunde kopplas in mellan orderkedjan och adresshålen för variabler eller instruktioner, tillräckligt flexibel för att programmet själv skulle kunna utföra de vanligaste adressmodifikationerna, t.ex. återhopp från subrutiner till olika ställen i huvudprogrammet eller indexering vid hantering av vektorer och matriser. Programmering i maskinkod, eller kodning, som det kallades på den tiden, var ju tidsödande och pillig, men ganska lätt att lära sig. På hösten 1950 rapporterade Olle Karlqvist, som då vistades en månad vid EDSAC-laboratoriet i Cambridge, att man där ansåg att medelsvensson kunde lära sig koda på 14 dar. Och det stämde bra med vad vi sedan erfor vid MNA:s kurser i ’Kodning för BESK.’” Brev Göran Kjellberg, 17/10 2006.

<sup>20</sup> Grace Hopper, 1906–1992, matematiker och programmerare. Hopper arbetade med Aiken vid konstruktionen av Harvard Mark I. Hon anställdes 1949 vid Eckert-Mauchly Computer Corporation och var med om att ta fram datorn UNIVAC. Vid arbetet med UNIVAC utvecklade hon den första kompilatorn som kallades A-0 och fungerade 1952. Hopper tog också fram programspråket MATH-MATIC som ledde till språket FLOW-MATIC vid 1950-talets mitt, vilket i sin tur var betydelsefullt för utvecklingen av programspråket COBOL. Ceruzzi (1998), 85; James W. Cortada, *Historical Dictionary of Data Processing: Biographies* (New York, 1987b), 132f.

<sup>21</sup> Eg. Eckert-Mauchly Computer Corporation, dvs. företaget som utvecklade datorn UNIVAC. Det köptes 1950 upp av Remington Rand Corporation som 1955 gick samman med Sperry Corporation och bildade Sperry Rand Corporation. Datortillverkningen skedde vid Univac Division of Sperry Rand. James W. Cortada, *Historical Dictionary of Data Processing: Organizations* (New York, 1987c), 235ff, 248ff.

**Elsa-Karin Boestad-Nilsson:**<sup>22</sup> Ja, jag var ju på FOA.<sup>23</sup> De andra var ingenjörer, och jag hade ju bara läst en enkel fil.kand. på högskolan, och matte var mitt huvudintresse. Jag tyckte inte att teknik var roligt, och när jag kom i kontakt med BESK<sup>24</sup> och fick höra vad man kunde göra på den och hur det skulle gå till, då blev jag ju himla entusiastisk, och killarna de sa: ”Nollor och ettor, du är inte klok. Akta dig!” Men jag hoppade på det, och i början var jag nästan ensam. I början var det ju väldigt enkelt, man kunde ju använda de här sexton hexadecimala siffrorna till allting, och det var förvånansvärt vad mycket man kunde göra, det var ju bara 512 helords minne. Vi kunde lösa differentialekvationer med Runge-Kutta,<sup>25</sup> och vi gjorde analyser av tidsserier. Vi gjorde mycket såna saker. Det gick bra. Men vi hade andra problem också, det var till exempel att göra geometriska beskrivningar av stridsvagnar och flygmaskiner. För att kunna beräkna verkan av olika stridsmedel, för att kunna få de här stridsmedlen optimala, för att kunna få flygmaskiner och till exempel stridsvagnar konstruerade optimalt utifrån den typ av vapen som fanns mot dem. Och där räckte ju inte BESK alls. Vi började leta efter annat, och någon gång 1956, tror jag, började vi väl köra på IBM 650<sup>26</sup> i Arboga som var en maskin med trummor, och där fick man vara två som programmerade, en som programmerade, och en som optimerade trummans vridning, så att det inte tog för lång tid. Det kunde vi göra med programspråket som hette FLOPS,<sup>27</sup> och jag minns att det liknade något ALFAKOD,<sup>28</sup> fast det var ju en helt annan princip antar jag. Men det var mycket lättare att programmera. Och sen så tyckte vi att det inte heller räckte, utan det fanns ju maskiner i Paris. Vi programmerade i alla fall FORTRAN<sup>29</sup> för 704<sup>30</sup> och senare FORTRAN för 709<sup>31</sup> i Paris. Var nere och körde, vi programmerade ju hemma, och sen åkte vi ner och körde, och satt på nätterna. Det var överhuvudtaget väldigt mycket nattarbete med alla datorer. Ja, för man fick ju inte komma dit när de vanliga människorna skulle köra. Ja, i alla fall så var problemet med FORTRAN på de maskinerna att man inte kunde kompilera subrutiner separat, utan man måste göra hela programmet, och vi gjorde ju stora program. Ja, vi beskrev en stridsvagn inuti ganska schematiskt, men tillräckligt mycket för att inte beskrivningen skulle kunna föras över i BESK på långa vägar. Då beskrev man även vapen-

---

<sup>22</sup> Elsa-Karin Boestad-Nilsson, f. 1925, fil.kand. i matematik, mekanik och fysik. Boestad-Nilsson anställdes vid FOA 2 1948, där hon arbetade med reglertekniska och kärnfysikaliska beräkningar samt tidsserieanalyser. Hon blev 1957 chef för FOA 2:s beräkningssektion som programmerade och löste problem från stora delar av FOA, chef för FOA:s institution för tillämpad matematik och databehandling 1974 samt överingenjör 1976–90.

<sup>23</sup> FOA, Försvarets forskningsanstalt, bildades 1945 genom sammanslagning av Försvarsväsendets kemiska anstalt med bl.a. Militärfysiska institutet.

<sup>24</sup> BESK, förkortning för binär elektronisk sekvens-kalkylator, Sveriges första elektroniska dator. Konstruktionen av BESK skedde vid MMN och leddes av Stemme. BESK togs i drift i december 1953 och invigdes i januari 1954.

<sup>25</sup> Runge-Kutta-metoder, metoder för numerisk lösning av differentialekvationer.

<sup>26</sup> IBM 650 började säljas 1954 och var en decimalt räknande dator med mycket långsamt roterande trumminne. Cortada (1987a), 199f.

<sup>27</sup> Boestad-Nilsson syftar antingen på FLOP, förkortning för Floating Octal Point, ett system utvecklat i slutet av 1950-talet av Lockheed Aircraft som bl.a. implementerade flytande räkning för IBM 701, eller på programspråket FLAP som användes under 1960-talet för matematiska applikationer som matrisberäkningar och partiella differentialekvationer bl.a. för IBM 7090.

<sup>28</sup> ALFAKOD, ett s.k. lättkodsystem konstruerat av Hans Riesel, Olof Jonason och Lennart von Sydow vid firman Autocode AB. En första version presenterades i november 1957. ALFAKOD användes på maskinerna BESK och FACIT EDB. *Alfakodning för matematikmaskiner: Allmän del (preliminär version)*, Autocode AB (Stockholm, 1959).

<sup>29</sup> FORTRAN, förkortning för Formula Translation, ett programspråk för teknisk-vetenskaplig databehandling som konstruerades av John Backus på IBM och började användas tidigt 1957 för IBM 704. FORTRAN räknas som det första högnivåspråket och det vann snabbt stor spridning. Ceruzzi, 90f.

<sup>30</sup> Eg. IBM 704, IBM:s första dator med kärnminne. Den utannonserades 1955. Cortada (1987a), 203ff.

<sup>31</sup> Eg. IBM 709, dator i IBM:s 700-serie och föregångare till IBM 7090, en transistorerad dator som stod färdig 1960. Cortada (1987a), 203ff.

verkan, sen gjorde vi Monte Carlo-simuleringar<sup>32</sup> med såna här mål. Och det var jättesvårt att hålla ihop hela programmet utan att få separatkompilera en enda subrutin. Och det var vansinnigt svårt att leta efter fel i en sån där soppa. Så FOA försökte köpa en maskin. De var bland annat 1955 mest intresserade av en tysk maskin som hette ZUSE,<sup>33</sup> som hade konstruerats i början av kriget. Sen efter några år, så började man intressera sig för IBM-maskiner, och det var därför vi åkte ner till IBM för att köra, för man hade beslutat sig för man skulle köpa en matematikmaskin, och det blev IBM 7090.<sup>34</sup> Där fick vi alltså möjlighet att dela upp programmen i subrutiner och separat kompilera dem för sig.

**Lars Arosenius:** Tack ska du ha. Försvaret var en väldigt stor beställare av tid på BESK, och överhuvudtaget var det så att försvaret varit ganska viktigt i IT-sammanhang.

**Elsa-Karin Boestad-Nilsson:** Ja, vi fortsatte att köra på BESK och Facit EDB,<sup>35</sup> och vi övergick till ALFAKOD och slutade i stort sett 1962, när vi hade kört in oss ordentligt på 7090.

**Lars Arosenius:** Den som höll på med FORTRAN var ju bland annat Björn Kleist. Det ordnades 1959 en kongress i Karlskrona som hette Nordsam, *Nordiskt symposium för användning av matematikmaskiner*, och det var i maj.<sup>36</sup> Och samma sommar kom ju en kongress som UNESCO ordnade i Paris, *Information Processing*,<sup>37</sup> och Tord Jöran Hallberg<sup>38</sup> har gjort statistik, det var 63 svenskar på den där kongressen i Paris, och på Nordsam så var det väl nåt hundratal personer.<sup>39</sup> Jag vet att Göran Kjellberg var med, och han talade om extremalproblem som löstes med hjälp av sortering.<sup>40</sup> Och Björn Kleist var med där också och talade om ytterballistiska banberäkningar på IBM 650,<sup>41</sup> och jag har för mig att du var IBM:s FORTRAN-specialist redan då på slutet på 50-talet.

---

<sup>32</sup> Monte-Carlo-metoder, metoder för beräkning eller simulering, vilka utnyttjar stokastiska element. De används för att lösa problem där analytiska metoder saknas eller är alltför komplicerade, t.ex. vid beräkning av komplicerade integraler.

<sup>33</sup> ZUSE-maskinerna, eg. en serie tyska datorer som kallades för Z. Tysken Konrad Zuse, 1910–1995, byggde dem under 1930-, 40- och 50-talen, sedan 1949 vid sitt företag ZUSE KG. Cortada (1987a), 381ff.

<sup>34</sup> IBM 7090, se fotnot 31.

<sup>35</sup> Facit EDB, BESK-kopia som tillverkades av Elektronikavdelningen vid AB Åtvidabergs Industrier/Facit Electronics AB under åren 1957–62.

<sup>36</sup> Carl-Erik Fröberg & Yngve Rollof, eds., *Nordiskt symposium över användning av matematikmaskiner, anordnat av Matematikmaskinnämnden och Kungl. Örlogsmannasällskapet i Karlskrona den 14–15 maj 1959* (Stockholm, 1959).

<sup>37</sup> *Information Processing: Proceedings of the International Conference on Information Processing, Unesco, Paris 15–20 June 1959* (Paris, 1960).

<sup>38</sup> Tord Jöran Hallberg, f. 1937, civ.ing. E, Chalmers tekniska högskola (CTH) 1962. Hallberg arbetade vid det som senare blev Datasab 1963–72, mot slutet som projektledare för Saab:s sista stordator, D23. Han var från 1972 universitetslektor i datorteknik vid Linköpings högskola/Linköpings universitet, där han medverkade till starten av den första svenska civilingenjörsutbildningen i datateknik. Hallberg har även verkat som journalist samt medverkat till att bygga upp IT-ceum, Sveriges datamuseum i Linköping.

<sup>39</sup> Tord Jöran Hallberg, *IT-gryning* (Lund, 2007).

<sup>40</sup> Göran Kjellberg, ”Ett extremalproblem löses med hjälp av sortering”, i *Nordiskt symposium över användning av matematikmaskiner, anordnat av Matematikmaskinnämnden och Kungl. Örlogsmannasällskapet i Karlskrona den 14–15 maj 1959*, eds. Carl-Erik Fröberg & Yngve Rollof (Stockholm, 1959), 229–236.

<sup>41</sup> Björn Kleist, ”Ytterballistisk banberäkning på IBM 650”, i *Nordiskt symposium över användning av matematikmaskiner, anordnat av Matematikmaskinnämnden och Kungl. Örlogsmannasällskapet i Karlskrona den 14–15 maj 1959*, eds. Carl-Erik Fröberg & Yngve Rollof (Stockholm, 1959), 387–396.

**Björn Kleist:**<sup>42</sup> Ja, jag är ju flygtekniker från början, och sista året som jag gick på Teknis<sup>43</sup> så tyckte min professor, Sten Luthander<sup>44</sup> i flygteknik, att jag skulle lära mig lite grann om datorer, och jag fick gå på en kurs om såna här kopplingsboxar, proppskåp, eller vad man ska kalla dem. Jag förstod ingenting då, och jag förstod inte vad jag skulle ha det till heller. Men senare så fick jag och en kollega uppgiften att försöka programmera flygplanens rörelseekvationer på just IBM 650, och det växlade min inriktning från flygteknik till datateknik. Ett år senare så fick jag anställning på IBM och var då anställd på något som hette Applied Science. I motsats till de som höll på med administrativ databehandling var det här teknisk databehandling, och efter två år på IBM fick jag lära mig FORTRAN grundligt. Jag var nere i Paris och skrev FORTRAN-program och hjälpte kunder att få dem att fungera. Min chef, Arne Lindberger, lyckades till slut vara den som sålde 7090 till FOA. Jag var intresserad av att köra på maskiner, och jag såg att datakraften, möjligheten att programmera, nu fanns på FOA och inte på IBM. Och då var ju frågan, hur löste man det här? Och då kom jag med en idé, det utomordentliga förslaget att IBM skulle låna ut mig på två år till FOA som teknisk specialist och FORTRAN-specialist. I gengäld skulle de få 20 timmar maskintid per år på 7090.<sup>45</sup> Det var uppgörelsen. På det sättet blev jag mer eller mindre anställd på FOA under de här två åren. Och jag fick tillfälle att jobba med FORTRAN mycket, men också med det som jag tycker revolutionerade programmeringstekniken, det första riktiga operativsystemet, FORTRAN Monitor,<sup>46</sup> som bland annat gjorde att man kunde köra in flera kunders, användares, program, kompilera och leverera resultaten, och man kunde använda maskinerna väldigt effektivt, man slapp spilltid mellan varje kunds körning så att säga. Och det där utvecklade sig snabbt, operativsystemet blev bättre, krävde bättre inmatning, så vi fick skaffa en 1401:<sup>47</sup> för att sköta inläsning och utskrift, och det överfördes till magnetband som kördes in på 7090, och resultaten levererades också på ett magnetband som sen kördes på 1401 och skrivare. Det blev en ganska rationell hantering. Man kan säga min inriktning var nog mest mot operativsystem, att se till att kunderna kunde få så effektiv, snabb behandling som möjligt. Men jag var fortfarande specialist på FORTRAN, och det var alltid såna som hade problem. Det var problem på olika nivåer. Det kunde vara hur man skulle lösa en viss kodgrej, i vissa fall tog det förskräckligt lång tid. Varför gjorde det det? Och då försökte man hitta andra metoder som gjorde att det gick snabbare att beräkna. FOA hade ju väldigt tunga beräkningar på den tiden. Och sen i slutet av 60-talet, -67, så behövde FOA mer maskinkraft, och då kom man på idén att gå ihop med universiteten och skaffa en gemensam stordator. Det var då QZ<sup>48</sup> startades.

---

<sup>42</sup> Björn Kleist, f. 1933, civ.ing. skeppsbyggnad och flygteknik (S), KTH 1957. Kleist var forskningsassistent på Flygtekniska institutionen vid KTH 1957–58 och fortsatte sedan som programmerare vid IBM 1959–62. Vidare var han dataansvarig för FOA:s stordator drift 1963–66, driftschef vid QZ 1967–73, teknikchef Datema AB 1974–77, produktionschef SEB Data 1978–91. Kleist var därefter IT-konsult vid Trigon Management innan han startade egen konsultverksamhet med Theolit AB.

<sup>43</sup> Eg. Kungl. Tekniska högskolan (KTH).

<sup>44</sup> Sten Luthander, 1905–2000, civ.ing. S, KTH 1930. Luthander blev assistent i flygteknik vid KTH 1930, i hållfasthetslära 1936, avdelningschef vid Flygtekniska försöksanstalten 1940. Han utnämndes till professor i flygteknik 1947, en post han innehade till 1972. Luthander var chef för forskningsavdelningen vid Saab i Linköping 1947–49.

<sup>45</sup> Maskintiden var värderad till 6 000 kr/timme. Brev Björn Kleist, 29/9 2006.

<sup>46</sup> FORTRAN Monitor, ett av de första operativsystemen. Det utvecklades för IBM:s 7090-serie. Ceruzzi (1998), 100.

<sup>47</sup> Eg. IBM 1401, liten transistorerad dator som introducerades av 1959. Det var inte ovanligt att den användes som komplement till en stordator, då den hade hög inläsnings- och utskriftskapacitet. Ceruzzi (1998), 73ff.

<sup>48</sup> Eg. Datorcentralen för högre utbildning och forskning i Stockholm (QZ), en gemensam datorcentral för FOA och högskolorna i Stockholm som inrättades 1967. Den fungerade som en servicebyrå med stora datorsystem.

Från början hette det Stockholms datamaskincentral för högre utbildning och forskning, men blev sen omdöpt till QZ, men det är en annan historia. Operativsystemen var ännu bättre, man kunde klara flera kompileringar och exekveringar parallellt i olika ”partitioner”. Ja, det är ungefär så långt jag vill berätta nu.

**Lars Arosenius:** Tack ska du ha. Hans Riesel uppträdde ju också på Nordsamkonferensen och talade om lättkodsystem för EDB-maskiner.<sup>49</sup> Och jag tror att du hade flera inlägg på UNESCO-konferensen när du presenterade det här systemet. Vad var det som var drev fram det?

**Hans Riesel:**<sup>50</sup> Ja, det var ju så att jag tillsammans med två kompanjoner, Olof Jonason<sup>51</sup> och Lennart von Sydow,<sup>52</sup> hade en liten firma där vi åtog oss konsultuppdrag i teknisktvetenskapliga beräkningar.

**Lars Arosenius:** När bildade ni den?

**Hans Riesel:** Jag kommer inte riktigt ihåg, men det var väl 1956, eller –57, eller nåt sånt.

**Lars Arosenius:** Vad hette den?

**Hans Riesel:** Autocode AB<sup>53</sup> hette firman. Och det visade sig när man höll på med den sortens arbete att det var ju oerhört tungrott, men inte orationellt, att programmera direkt i maskinkod. En sak som vi inte har hört har ju varit avsaknaden av subrutinhantering. Det andra var ju att det var väldigt petigt med den här operationslistan, så vi tänkte att man skulle försöka göra någonting åt det där, och då hade ju Gunnar Hellström,<sup>54</sup> som ju var en föregångare i programmering på BESK, gjort några kompilatorer som han kallade för systemet för Fiktiv Adressering, FA, följt av ett nummer.

**Lars Arosenius:** FA 5.

**Hans Riesel:** Och så FA 4 och FA 5, och de använde vi ju. De där systemen var ju så att man kunde lägga in fiktiva adresser i ett program och så märka de punkter som de där fiktiva adresserna hänvisade till, och kompilera det, och då slapp man ju i alla fall det där trasslet att räkna om alla adresser, där man tvingades lägga in en eller flera instruktioner

---

<sup>49</sup> Hans Riesel, ”Alfakodning, ett lättkodsystem för EDB-maskiner”, i *Nordiskt symposium över användning av matematikmaskiner, anordnat av Matematikmaskinnämnden och Kungl. Örlogsmannasällskapet i Karlskrona den 14–15 maj 1959*, eds. Carl-Erik Fröberg & Yngve Roloff (Stockholm, 1959), 129–134.

<sup>50</sup> Hans Riesel, f. 1929, fil.lic. i matematik 1955, fil.dr i numerisk analys 1969 samt docent i ämnet 1970. Han var assistent vid KTH 1951–58, assistent och bitr. lärare vid matematiska institutionen, Stockholms högskola 1952–58, byrådirektör MMN 1959–63, Statskontoret 1963–69, universitetslektor i numerisk analys vid KTH 1969–94. Riesel bildade vidare företaget Autocode AB tillsammans med Olof Jonason och Lennart von Sydow 1957. Han var delägare där fram till 1959, då han blev byrådirektör vid MMN.

<sup>51</sup> Olof Jonason, fil.kand. i matematik, delägare i Autocode AB.

<sup>52</sup> Lennart von Sydow, fil.mag. i matematik, delägare i Autocode AB.

<sup>53</sup> Autocode AB, svenskt mjukvaruföretag grundat 1957 av Riesel, Jonason och von Sydow. Riesel avyttrade sin andel i firman när han blev byrådirektör vid MMN, och civilingenjören Lennart Ringström gick in som delägare 1963. Förutom att tillhandahålla ALFAKOD-systemet, så utförde Autocode AB också teknisk-vetenskapliga beräkningar för svensk industri. Under 1960-talet blev datoriserad sättning av trycksaker en viktig del av företagets verksamhet. Intervju Hans Riesel, 1/12 2006 (tfn); intervju Lennart Ringström, 5/12 2006 (tfn).

<sup>54</sup> Gunnar Hellström, 1922–1993, matematiker. Han var programmerare vid MMN och konstruerade bl.a. systemet för Fiktiva Adresser (FA). Hellström lämnade MMN 1956 och gick till Elektronikavdelningen vid AB Åtvidabergs Industrier. 1959 bildade han företaget Datasystem AB tillsammans med Axel Bring och Zoltán Horváth.

mitt i ett program. Och det där var en bra sak, som vi naturligtvis utnyttjade. Men så tyckte vi att man skulle väl försöka göra det lite mera användarvänligt, och det var två saker som då var viktiga. Det ena var att ha mnemotekniska beteckningar för instruktionerna, så att addition hette "add" och subtraktion hette "sub" och så vidare, och sen skulle man kunna ha mnemotekniska benämningar för variablerna, så att variablerna kunde heta "Q" och "VAD DU VIL" och så, med tre bokstäver, och så skulle man dessutom ha subrutinhantering, man skulle alltså programmera in fasta subrutiner för de elementära matematiska funktionerna och lite annat som var nyttigt. Och det var ju det som man senare skulle kalla en enkel assembler. Det visade sig sen, det kände inte vi till när vi började med det här projektet, att det fanns någonting som hade utvecklats ganska parallellt och såg rätt så likartat ut när det var färdigt, som Univac gjorde för sina maskiner. Och det där systemet kallade vi för ALFAKOD, och här råkar jag ha med mig ett häfte här, det är en beskrivning på ungefär 80 sidor, och i det står det "preliminär version, 1959",<sup>55</sup> så då var det tydligen så pass färdigt att vi vågade släppa det. Ja.

**Lars Arosenius:** Ingick inte flytande räkning också?

**Hans Riesel:** Det kom senare.

**Lars Arosenius:** Jaså.

**Hans Riesel:** Facit<sup>56</sup> ville ju av konkurrensskäl ha flytande räkning på sina maskiner. Det var ett försäljningsargument. Vi lyckades sälja det här till Matematikmaskinnämnden så att det kunde levereras med BESK-maskinerna, och flytande räkning som Göran Kjellberg nämnde, det ville Facit ha, och då gjorde vi ett system för det och la in det som en option för deras maskiner.

Problemet med det här sättet att arbeta var ju att maskinerna var så oerhört små. BESK hade ju 512 helord, och visserligen fanns det trumminne, men det var ju än svårare att administrera såna här subrutinhanterare via trumminnen, så vi försökte att klara det genom att "sko-horna" in så mycket som möjligt på dem, säg hälften av de 512 orden, så att man kunde ha resten av arbetsminnet för tal. Ja, det gjorde vi. Så höll vi på så där och det gick ju på en del år, och var ju ändå framgångsrikt på sin tid. Då fanns det visserligen parallellt andra kodningssystem som var mer avancerade, FORTRAN. Men det var ju inte tal om att skriva FORTRAN-kompilator för BESK på den tiden, därför att den teknik man senare utvecklade för att göra kompilatorer fanns ju inte, utan det här var ett arbetsprogram där man gjorde varenda grej för hand, från scratch. Senare så har man ju rationaliserat det här kompilatorbyggandet. Ja, så var det.

**Lars Arosenius:** Skulle du klassa det som ett subrutinbibliotek?

**Hans Riesel:** Det var ett subrutinbibliotek i och med att de här vanliga standardfunktionerna kvadratrotten, sinus, cosinus och så vidare, ingick i det här som order. Sedan byggde vi till ett subrutinbibliotek, så man kunde hoppa till egna subrutiner, standardsubrutiner, för allt möjligt. Det fanns ju ett stort subrutinbibliotek som Matematikmaskinnämnden administrerade och sände ut till abonnenter. Och det var ju inte gratis för abonnenterna, så vitt jag minns, utan det här var ju ett försäljningsargument för dem som hade maskiner. De kunde sälja maskintid tack vare att maskinerna gick att använda.

---

<sup>55</sup> *Alfakodning för matematikmaskiner: Allmän del (preliminär version)*, Autocode AB (Stockholm, 1959), 89 s.

<sup>56</sup> Eg. Facit Electronics AB. Elektronikavdelningen vid AB Åtvidabergs Industrier ombildades 1960 till dotterbolaget Facit Electronics AB. Detta utvecklades vid årsskiftet 1962/63 och dess verksamhet underordnades moderkoncernen.

**Lars Arosenius:** Flytande räkning byggdes ju in i hårdvaran i BESK.

**Hans Riesel:** Nej, det var ju TRASK.<sup>57</sup> Det som gjorde TRASK intressant var att den hade flytande räkning inbyggd.

**Lars Arosenius:** Vi hade flytande räkning på BESK. Vi byggde in flytande räkning på BESK omkring -58. Och det tog sedan Zoltán över till TRASK.

**Hans Riesel:** Okej.<sup>58</sup>

**Lars Arosenius:** Zoltán Horváth.<sup>59</sup>

**Ingemar Dahlstrand:**<sup>60</sup> Får jag ställa en fråga?

**Lars Arosenius:** Ja?

**Ingemar Dahlstrand:** Är det korrekt uppfattat att ALFAKOD var maskinberoende på det sättet att det hade kunnat implementerats lika väl på någon annan dator med samma kapacitet?

**Hans Riesel:** Det kunde man i och för sig ha gjort, men det är ju ett visst arbete att koda sinusfunktioner i maskinkod för en annan typ av dator, så den stora arbetsinsats som det där krävde gjorde att jag tror att det aldrig blivit etablerat för andra maskiner.

**Ingemar Dahlstrand:** Jo, men jag uppfattade det som ett vägskäl ändå, därför att före ALFAKOD så gick allting ut på att det här var nånting för BESK. Maskinberoendet är ju viktigt, för det frigör en från att vara bunden till maskintyp och att vara bunden i tiden också.

**Hans Riesel:** Jo, det är ju sant. För oss att göra maskinberoende fordras det ju ett slags standardisering där olika maskinfabrikanter kommer överens om att man ska ha samma beteckningar för en massa viktiga saker, och det är ju i grunden emot maskintillverkarnas intentioner. Maskintillverkarnas första strävan är att göra kunden bunden till sig. Och detta har ju motverkat standardiseringsarbete på alla områden i alla tider. Mer eller mindre.

**Lars Arosenius:** Mer eller mindre.

---

<sup>57</sup> TRASK, förkortning för transistorerad sekvenskalkylator, dator som konstruerades av Datasystem AB och togs i bruk 1965. TRASK användes huvudsakligen av dåvarande Nobelinstitutet för fysik, senare Forskningsinstitutet för atomfysik (AFI) och numera Manne Siegbahnlaboratoriet för fysik. E-post Lars Arosenius, 2/1 2006.

<sup>58</sup> Programmerad flytande räkning ingick från första början i ALFAKOD, man hade variabler av typ *real* och matematiska funktioner och operationer som förutsätter typ *real*. Vad som kom senare var således inte flytande räkning i sig, utan att flytande räkning byggdes in i hårdvaran på BESK och utnyttjades för att snabba upp deras upplaga av ALFAKOD. FACIT-maskinerna fick däremot aldrig inbyggd flytande räkning. Brev Ingemar Dahlstrand, 11/10 2006.

<sup>59</sup> Zoltán Horváth, elektroingenjör. Horváth kom till Sverige efter Ungernrevolten 1956 och anställdes på MMN samma år. Tillsammans med matematikerna Hellström och Bring bildade han 1959 företaget Datasystem AB.

<sup>60</sup> Ingemar Dahlstrand, se fotnot 104.

**Hans Riesel:** Men det är väl bra att pedalerna på bilar sitter likadant placerade för fötterna.

**Lars Arosenius:** Hur många manår lade ni ner på att göra ALFAKOD?

**Hans Riesel:** Ja, det kommer jag inte ihåg, men det borde i alla fall ha varit säkert tre–fyra manår. Vi hade en person anställd som gjorde maskinkodningen av olika rutiner, och testade den delen av kompilatorn. Vi höll ju på i ett par år med det här, och Olof Jonason var ju den som avgjorde hur det skulle se ut och därför föll det här arbetet för min del i glömska, för jag blev ju lektor i numeriska metoder på Teknis så småningom.

I mitten på 80-talet kom jag i kontakt med det här sättningssystemet av text som hette TeX.<sup>61</sup> Det där TeX-systemet är ju underbart för en gammal vetenskaplig programmerare att jobba med, för det är gjort av Donald Knuth<sup>62</sup> som ju är matematiker från början. Tankandet är ju rent matematiskt algoritmiskt, och han arbetar mycket med det, som man ju kan säga att ALFAKOD också var, nämligen makroinstruktioner. Man kan, om man har såna system, skriva en makroinstruktion när man vill, som utför något som man önskar få utfört, och som man då ger ett namn. Men det är ju det som man kan kalla subrutiner, eller procedurer, i ALGOL.<sup>63</sup> Den sortens funktion fanns i det här systemet, mer eller mindre indirekt i alla fall.

**Lars Arosenius:** Vet du hur många såna programsystem ni sålde?

**Hans Riesel:** Ja, vi sålde ju det till Matematikmaskinnämnden och till Facit. Inga andra avnämare fanns. Facit höll sina kunder med det, så det fanns ju ingen anledning att gå till kunderna direkt.

**Lars Arosenius:** Så ni sålde aldrig till slutanvändarna? Elsa-Karin.

**Elsa-Karin Boestad-Nilsson:** Ja, ni hyrde väl ut också? Man fick ju köra och betala per timme. Man fick betala maskintiden på BESK, och så fick man betala en avgift per timme för att man hade använt ALFAKOD när man körde.

**Hans Riesel:** Ja, det känner jag inte till.

**Elsa-Karin Boestad-Nilsson:** Så var det!

**Hans Riesel:** De som köpte rätten att använda den av oss var ju tydligen ute efter att få tillbaka lite pengar.

**Lars Arosenius:** Ja.

**Hans Riesel:** Ja, det var väl bra!

---

<sup>61</sup> TeX, formellt språk och datorprogram för utformning och framställning av typografi, utformat och utvecklat av Donald E. Knuth vid 1970-talets slut.

<sup>62</sup> Donald E. Knuth, f. 1938, amerikansk datavetare och matematiker. Knuth är professor vid Stanford University i Kalifornien. Han har bl.a. givit ut flerbandsverket *The Art of Computer Programming* (1968–73) samt skapat TeX.

<sup>63</sup> ALGOL, förkortning för Algorithmic Oriented Language, ett av de äldsta högnivåspråken. Det konstruerades under 1950-talets senare hälft av en kommitté av europeiska och amerikanska forskare, delvis som en reaktion på det under 1950-talet utvecklade FORTRAN, och slog igenom med en version från 1960, ALGOL 60. Se t.ex. ”ALGOL Session”, i *History of Programming Languages*, ed., Richard E. Wexelblat (New York, 1981), 75–172.

**Elsa-Karin Boestad-Nilsson:** Jag tyckte ju att det var väldigt smart, men då var det inte din idé?

**Hans Riesel:** Nej, jag har aldrig varit bra i affärer.

**Lars Arosenius:** I den konferensen i Paris säger Backus, tror jag, att det tar minst 30 manår att bygga en kompiler.

**Hans Riesel:** Ja, det här är ju mer av vad man skulle beteckna som en assembler.

**Lars Arosenius:** Ja. Har ni som talat hittills nån uppfattning av hur införandet av FORTRAN och ALFAKOD påverkade produktiviteten för programmerarna? Alltså blev det nån vinst? Elsa-Karin.

**Elsa-Karin Boestad-Nilsson:** Ja, åtminstone FORTRAN blev ju så att man kunde göra sånt som inte gick och göra förut.

**Lars Arosenius:** Ja.

**Elsa-Karin Boestad-Nilsson:** Och det var inte bara FORTRAN:s förtjänst. Det var de stora maskinernas förtjänst också förstås. Men det var en himmelsvid skillnad på att programmera i de här assemblerliknande språken och i FORTRAN.

**Göran Kjellberg:** Ja, jag har stött på några programsystem som parallellt med ALFAKOD fanns i Manchester som jag hälsade på 1955. De hade ett system som var någorlunda likt. Ja, jag satte mig inte in i detaljer, men det hade de faciliteter som ALFAKOD hade. Också ett programspråk för IBM 650 som togs fram vid Bell Labs<sup>64</sup> av Wolontis,<sup>65</sup> minns jag. Var inte det också med samma faciliteter som ALFAKOD, det vill säga begripliga namn på operationerna och färdiga subrutiner för flyttalsaritmetik? Eller kom det senare möjligen?

**Elsa-Karin Boestad-Nilsson:** Jag tyckte att det vi programmerade på 650 liknade ALFAKOD.

**Göran Kjellberg:** Ja, ja.

**Elsa-Karin Boestad-Nilsson:** Vad hette det där programmet som du pratade om? Kommer du ihåg vad det hette?

**Göran Kjellberg:** Vad språket hette kommer jag inte ihåg.<sup>66</sup>

**Elsa-Karin Boestad-Nilsson:** Nej, det gör du inte. FLOPS är det jag kommer ihåg, men det fanns ett till som jag inte kommer ihåg.

---

<sup>64</sup> Bell Labs, eg. Bell Telephone Laboratories Inc., en av de ledande miljöerna inom grundforskning i elektronik och fysik under 1900-talet. Under större delen av 1900-talet har det fungerat som AT&T:s forskningslaboratorium.

<sup>65</sup> Matematikern V. Michael Wolontis ledde utvecklingen av programspråket Wolontis-Bell Interpreter som fungerade på IBM 650 hösten 1955. "Wolontis-Bell Interpreter", *Annals of the History of Computing* 8 (1986), 74–76.

<sup>66</sup> "Men det språk jag fick se i Manchester kan ha hetat Manchester Autocode eller något liknande. Det fungerade förstås bara på deras dator." Brev Göran Kjellberg, 22/10 2006.

**Lars Arosenius:** Kommer du ihåg, Björn, vad det hette?

**Björn Kleist:** Nej, tyvärr.

**Bengt Gällmo:**<sup>67</sup> Fanns inte det nåt som hette SOAP?<sup>68</sup>

**Elsa-Karin Boestad-Nilsson:** Jo, SOAP!

**Björn Kleist:** Symbol optimizing assembler. Kan man få göra ett kort infall om det där med produktivitet? Som Elsa-Karin säger var det väl en kombination av FORTRAN-språket och 7090-datorn. Per Svenonius<sup>69</sup> höll tal vid invigningen av 7090 och sa att den här datorn har samma kapacitet som en miljon räknebiträden, skillnaden är att datorn går att administrera.

**Hans Riesel:** Om jag bara talar i egen sak, så ökade ju det här effektiviteten i programmeringen alldeles kolossalt. Jag använde ju själv det här för att göra väldigt stora system. Bland annat gjorde jag på konsultuppdrag åt de Laval's Ångturbin aktiebolag<sup>70</sup> ett jättesystem som räknade ut hur ångan betar sig när den rusar igenom alla stegen i en turbin. Och det hade ju aldrig gått att göra en sån grej i maskinkod. För man kunde i lugn och ro göra en subrutin för var och en av de funktioner som man behövde, testa den, och sen kunde man på ett enkelt sätt koppla ihop de där, och göra tekniska beräkningar på turbiner som var på konstruktionsstadiet.

**Lars Arosenius:** Men vad som förbinder er hittills, är att det är ganska maskinnära programmering som ni sysslade med, fram till början av 60-talet i alla fall.

**Hans Riesel:** Jomen alltså, det här språket var ju maskinprogrammering, men när man använde språket kom man ju upp på en abstraktionsnivå som låg ett steg över, man behövde inte bekymra sig om alla spillbitar och såna där konstiga detaljer som hänförde sig till maskinens hårdvarukonstruktion.

**Lars Arosenius:** Men när kom tanken på universellt språk upp?

**Hans Riesel:** Ja du, FORTRAN är ju ett universellt språk. ALGOL 60 var väl det första där man fick göra det standardiserat och med tanken att det skulle göras i ett stort antal maskiner, och den här Backus, var det inte ungefär 1960 som det första kom? Jag vet inte vad det hette. Danskarna var aktiva i detta. De hade också gjort en BESK-liknande maskin som de kallade för DASK, Dansk Sekvens Kalkulator,<sup>71</sup> och de hade gjort ett litet häfte som är en skiss av ALGOL. Det tror jag var från 1960.

---

<sup>67</sup> Bengt Gällmo, se fotnot 120.

<sup>68</sup> SOAP, förkortning för Symbolic Optimum Assembly Programming, programspråk utvecklat för IBM 650 och som började användas 1956.

<sup>69</sup> Per Svenonius, f. 1926, fil.lic., FOA 2 1955–57, chef för beräkningssektionen på FOA 4 (atomfysik) 1958. Svenonius blev senare avdelningschef på Statskontoret.

<sup>70</sup> Eg. AB de Laval's Ångturbin, svenskt verkstadsföretag grundat 1893 som huvudsakligen tillverkade ångturbiner av Gustaf de Laval's konstruktion. Det gick 1959 samman med ASEA-ägda Svenska Turbinfabriks AB Ljungströms (STAL) och fick namnet STAL-LAVAL Turbin AB.

<sup>71</sup> DASK stod även som förkortning för dansk version af besk. Datorn konstruerades vid Regnecentralen, Dansk Institut for Matematikmaskiner, och var en dansk BESK-kopia som kördes första gången hösten 1957.

**Lars Arosenius:** Hur var det med dem som skrev program, hade de svårt att lära sig det här? Hade ni kurser i det här?

**Hans Riesel:** Ja, vi höll kurser i ALFAKOD.

**Lars Arosenius:** Och IBM hade ju FORTRAN, eller hur?

**Björn Kleist:** Ja, det var några kurser där.

**Hans Riesel:** Då var det i och för sig riktat till slutanvändarna.

**Lars Arosenius:** Är det nån som har några kommentarer i auditorium här? Vi har alltså tittat på perioden fram till början på 60-talet och framför allt i Sverige. Thomas Ohlin.

**Tomas Ohlin:**<sup>72</sup> Det fanns ju ALGOL 58.<sup>73</sup> Kanske någon av er kan kommentera det?

**Ingemar Dahlstrand:** Ja, ALGOL 58 kan jag kommentera. Strax före det så hade det tyska GAMM, Sällskapet för tillämpad matematik och mekanik,<sup>74</sup> startat en kommitté för att göra ett generellt språk, och sedan så hade ACM, Association for Computing Machinery,<sup>75</sup> gjort samma samma sak. Av en tillfällighet fick de höra talas om varandra, så de bestämde sig för att göra det tillsammans. Det som blev ALGOL 58 sattes ihop på några dagar i Zürich, och sedan pågick ungefär två års debatt och utvecklingsarbete fram till ALGOL 60, så det var en gemensam europeisk-amerikansk kommitté som gjorde det.

**Lars Arosenius:** ECMA var väl igång? European Computer Manufacturers Association.<sup>76</sup>

**Ingemar Dahlstrand:** Ja, ECMA arbetade ju med de här frågorna på flera olika sätt. En av dem var att man tillsatte en kommitté för att göra ett amalgam av ALGOL och COBOL<sup>77</sup> som kunde användas för alla ändamål, både teknisk och administrativ databehandling. Jag råkade komma med på det mötet, och det mötet sprängde kommittén för att man var så oenig. Så det blev tre kommittéer istället för en. Sen arbetade ALGOL-kommittén med ”subsets” av ALGOL och standard för in- och utmatning.

---

<sup>72</sup> Tomas Ohlin, f. 1934, adj. professor i informationssystem vid Linköpings universitet. Han har bl.a. arbetat på AB Åtvidabergs Industrier/Facit AB med programmering. Ohlin medverkade i etableringen av Data- och systemvetenskaputbildningen vid KTH/Stockholms universitet 1966. Vidare har han deltagit i ett flertal statliga utredningar om informationssamhället.

<sup>73</sup> ALGOL 58, en första version av programspråket ALGOL från 1958.

<sup>74</sup> GAMM, förkortning för Gesellschaft für Angewandte Mathematik und Mechanik.

<sup>75</sup> Association for Computing Machinery (ACM) grundades 1947 och räknas som världens första datavetenskapliga samfund. Det är också den största och viktigaste organisationen inom databehandlingsindustri. Sedan 1954 ger ACM ut tidskriften *Journal of the Association for Computing Machinery* och sedan 1958 den månatliga publikationen *Communications of the ACM*. Cortada (1987c), 64f.

<sup>76</sup> European Computer Manufacturers Association (ECMA) bildades 1961 av europeisk datorindustri och verkade för standardisering av bl.a. programspråk. Sedan 1994 heter organisationen Ecma International.

<sup>77</sup> COBOL, förkortning för Common Business Oriented Language, ett programspråk för lösning av administrativa uppgifter med dator, t.ex. inom personaladministration, orderbehandling eller lagerbokföring. Det amerikanska försvarsdepartementet samlade i maj 1959 representanter för datortillverkare, användare och myndigheter i USA i syfte att skapa ett gemensamt affärsspråk. Sammankomsten resulterade i att en första version av COBOL kom 1960. Samma år deklarerade det amerikanska försvarsdepartementet att det inte skulle införskaffa eller hyra datorer som inte kunde hantera språket. Det ledde till att COBOL blev ett av de första programspråken som kunde köras på olika maskiner med samma resultat. COBOL fick stor spridning och var vid 1960-talets slut samt under 1970-talet ett av de mest använda programspråken. Ceruzzi (1998), 91f.

**Lars Arosenius:** Men kom det fram något AMALGAM?

**Ingemar Dahlstrand:** Nej, inte vad jag vet.

**Lars-Olof Norén:**<sup>78</sup> Ja, jag har varit med i ECMA-styrelsen ett antal år, och man brukar aldrig tala om nåt programspråk där. Däremot standardiserade man kodningen med alfanumeriska tecken. Tidigt.

**Lars Arosenius:** Flera frågor?

**Bo Lindestam:**<sup>79</sup> Jag har en kommentar.

**Lars Arosenius:** Bo Lindestam.

**Bo Lindestam:** Bo Lindestam här. Är det inte så med kompilatorer att det är då man låter hoppet fara? Med assembler hade man fortfarande väldigt noggrann kontroll på maskinstrukturen, det är one-to-one, medan kompilatorer normalt innebar en mycket kraftig expansion av den maskinkod som man skulle generera. Vi som jobbade på Standard Radio<sup>80</sup> kom att utveckla egna kompilatorer i slutet av 60-talet, och vårt största problem var expansionsfaktorerna som jämfört med motsvarande assemblerprogrammering ibland låg uppe på fyra, och vi strävade att komma ner på ungefär en och en halv. Fundera på detta, att man på de begränsade minnena vi hade, skulle få en expansion av kodens storlek i minnet på en faktor fyra jämfört med assemblerprogrammering. Det var en tankeställare som vi fick ta många gånger. Tack.

**Ingemar Dahlstrand:** Så stora expansionsfaktorer fick inte vi, men när jag säger vi, så menar jag vi som implementerade ALGOL 60 för Facit. Då gjorde vi också så att vi begränsade ALGOL 60 i några avseenden för att det skulle vara kompilierbart på ett rimligt sätt.

**Bo Lindestam:** Matematik kostar alltid ganska mycket.

**Ingemar Dahlstrand:** Det vinner på sikt.

---

<sup>78</sup> Lars-Olof Norén, f. 1937, civ.ing. E, KTH 1961. Norén anställdes 1961 vid LM Ericssons utvecklingsavdelning, och han fortsatte inom koncernen fram till 2001. Han var ansvarig för utvecklingen av 1:a och 2:a generationens datorstyrda telefonstationssystem (AKE) under 1960-talet och 1970-talets början. Norén var t.f. teknisk direktör för Ericsson under 1970-talet och hade därefter div. chefspositioner inom koncernen.

<sup>79</sup> Bo Lindestam, f. 1939, civ.ing. E, KTH 1962. Lindestam var programmerare vid Standard Radio 1958–70, vid Stansaab Elektronik AB 1971–78. Han arbetade därefter med marknads- och systemanalyser och som produktchef vid Datsaab AB 1978–81, Ericsson Information Systems AB 1981–87, Nokia Data Systems AB/ICL Data System AB 1988–93. Sedan 1994 har Lindestam egen konsultverksamhet.

<sup>80</sup> Eg. Standard Radio & Telefon AB (SRT), svenskt företag som bildades 1938 av världskoncernen International Telephone & Telegraph (ITT). Under 1940-talet levererade SRT radarutrustning till flygvapnet och flottan. Företaget utvecklade flygburna stridsledningsdatorer under 1950-talet och arbetade med stridsledningssystemet STRIL-60 under större delen av 1960-talet. Datadivisionen vid SRT fördes 1971 över till det nybildade företaget Stansaab Elektronik AB som 1978 gick upp i Datsaab AB, vilket i sin tur överfördes till Ericsson Information Systems AB (EIS) 1981. Kjell Mellberg, Gunnar Wedell & Bo Lindestam, *Fyrtio år av den svenska databistorien: Från Standard radiofabrik till...?* (Stockholm, 1997).

**Lars Arosenius:** Ja, vi är fortfarande inne på 50-talet och början på 60-talet. Göran Waernér, du var ju applikationsprogrammerare framförallt. Du har en bakgrund i Vägverket, eller vad det då hette?

**Göran Waernér:**<sup>81</sup> Kungliga Väg- och vattenbyggnadsstyrelsen<sup>82</sup> hette det officiellt.

**Lars Arosenius:** Och sedan hjälpte du till att sälja Facit-maskiner runt om i världen.

**Göran Waernér:** Nej, bara en till Facits Tysklandskontor. Jag var väg- och vattenbyggare och hade planerat en karriär inom Vägverket. Men så råkade jag träffa en kompis precis när jag hade börjat jobba där. Han berättade att han jobbade på Matematikmaskinnämnden och gav mig fem minuters information om vad en matematikmaskin var för någonting. Jag höll på med en mängd beräkningar om köbildning i backe och bensinförbrukning, och det var ett himla räknearbete. Då kom jag att tänka på att jag kanske skulle ringa upp Matematikmaskinnämnden och höra om det var nånting för en sån maskin. Och jag fick positivt gensvar. Så två-tre veckor efteråt så kom chefen, byråchefen,<sup>83</sup> tillbaks från en studieresa ifrån Amerika och kallade in avdelningen och berättade, ”kan ni tänka er att i Amerika så använder de nånting som heter computers och gör massberäkningar av vägar och en del andra vägberäkningar”. Och jag snabbt upp med handen och sa: ”Det heter matematikmaskiner och vi har Europas största här i Stockholm. Om fjorton dar så har de en kurs i programmering. Kunde inte jag få gå på den, så kunde jag ta reda på om vi kanske kan använda den istället för att köpa en sån där amerikansk liten maskin?” Det tyckte de var en bra idé, och då sa jag: ”Kan inte min kompis Bosse Hallmén<sup>84</sup> också få vara med, så att vi kan brainstorma? Det är ju ett helt nytt sätt att angripa massberäkningar av vägar.” Efter fjorton dagar så kom vi tillbaks och sa, ”vi tror att vi kan räkna tre kilometer väg på tio minuter”. Hur vi kom fram till det är fortfarande ett mysterium.

**Lars Arosenius:** Gjorde ni det?

**Göran Waernér:** Ja, vi fick ett år på oss att utveckla. Jag har inte varit med om att påverka själva programmeringsmetoden, utan vi fick anpassa oss efter de mycket tuffa förhållanden som fanns. Jag kommer till exempel ihåg att vi hade körtid två gånger om dygnet på BESK:en. Halvfyra på morgonen och sedan halvt på dan, så man kunde få två skott att testa programmet. Och sen innan vi hade hunnit blivit etablerade, fått egna stansar och sånt, så fick vi ju vara i god tid. Det tog ju lång tid att korrigera en sån där stor hålremsrulle, men vi lärde oss att hitta snitsiga metoder, så att man lade in ett antal blind-

---

<sup>81</sup> Göran Waernér, f. 1932, civ.ing. väg- och vattenbyggnad (V), KTH 1956. Han arbetade med databehandling vid Väg- och vattenbyggnadsstyrelsen (VoV) 1956–59. Tillsammans med Åke Bengtsson och Bo Hallmén grundade Waernér 1959 Nordisk ADB AB. Han var VD för företaget 1959–61 och 1965–70. Däremellan var Waernér VD för dotterbolaget Nordisk ADB GMBH i Düsseldorf, Västtyskland. Han var VD för Göran Waernér AB 1970–80 samt delägare och ordförande i ett antal mindre dataföretag under åren 1980–1998.

<sup>82</sup> Kungl. Väg- och vattenbyggnadsstyrelsen (VoV) inrättades 1841 som ett tillsyns- och ledningsorgan för väghållningsverksamheten. Vid vägväsendets förstatligande 1944 fick VoV det övergripande ansvaret för väghållningsverksamheten. Namnet ändrades till Statens vägverk 1967 och till Vägverket 1983.

<sup>83</sup> Harald Ekström, 1911–2003, civ.ing. V, CTH 1937. Ekström var chef för VoV:s Vägbyrå 1955–58 och dessutom överingenjör där 1958–63.

<sup>84</sup> Bo Hallmén, f. 1932, civ.ing. V, KTH 1956. Han arbetade med databehandling vid VoV 1956–59. Tillsammans med Bengtsson och Waernér grundade Hallmén företaget Nordisk ADB AB 1959, där han var företagets utvecklingschef 1960–72. Han var VD för Markdata AB som utvecklade system för digitala terrängmodeller och geodetiska beräkningar 1972–77 samt hade eget företag, Bo Hallmén Program Film AB, 1977–2006.

operationer, 0 0 0 0 5, skifta noll steg till höger. Sedan kunde man med en liten handstans plocka in operationer, så att man inte behövde kopiera om hela rullen. Efter ett år så hade vi systemet färdigt. Faktum var att det tog ungefär tio minuter att beräkna tre kilometer väg. Det tog ungefär en månad för en ingenjör att klara av samma jobb manuellt.

Sen så åkte vi som missionärer runt och lärde alla vägförvaltningar att använda metoden. Jag skulle vilja säga att från och med slutet av -57, så har samtliga vägar som projekterats i det här landet räknats med vårt system. Och det blev ju en massa terrängdata som kom in, så vi startade en avdelning på Vägverket som hette VERA, Vägbyråns elektroniska räkneavdelning. Sen så låg vi ju ganska bra till hos vägledningen, så vi fick tjäna extra genom att hjälpa norska och finska vägverken, och svenska VoV-konsulter som också ville använda tekniken. På den fritid som fanns mellan fem på eftermiddagen och halv fyra på morgonen hjälpte vi dem.

Gunnar Stenudd<sup>85</sup> hade gjort en oscillograf som kombinerades med en polaroidkamera. Den utnyttjade vi för att rita perspektivskisser. Den användes flitigt för att se hur vägarna skulle komma att se ut innan de var byggda. Senare utvecklade han en filmkamera, och vi gjorde filmer så att man kunde köra vägen innan den var byggd. Ryktet om det här spred sig över Europa. Vi hade massor med delegationer som kom upp. Bland annat så tände tyska vägverket på tekniken och framförallt det här med att göra perspektivfilmer. -58 så skickades vi, Bosse och jag och ytterligare en kille,<sup>86</sup> till Amerika för att se efter hur vi låg till, och vad man kunde få för erfarenheter från USA.<sup>87</sup> Och efter två månaders resa från öst till väst, så konstaterade vi att vi låg ju långt före amerikanerna. De hade ju bara små G-15-maskiner.<sup>88</sup> När vi såg det internationella intresset så sa vi att det här måste utnyttjas. Det här är världens chans! Vi startar eget! Och så sa vi upp oss och startade Nordisk ADB.<sup>89</sup> Sen fanns det ju mycket mer att göra inom väg- och vattenbranschen, och det vore bra att ha en intelligent kille i firman också, så vi tog vår kursätta, Åke Bengtsson,<sup>90</sup> och övertygade honom att han, som hade jobbat med konstruktioner, skulle gå med i gänget. Så det var vi tre som startade Nordisk ADB.

Samtidigt så var det ju ett stort språng att starta eget, och vi visste inte hur framtiden skulle bli, så därför tog vi ett extraknäck. Bosse och jag blev de första lärarna i ”Matema-

---

<sup>85</sup> Gunnar Stenudd, f. 1923, forskningsingenjör vid FOA 1949 och vid MMN 1950–56, där han bl.a. konstruerade BESK:s s.k. Williamsminne. Stenudd var sektionschef och projektledare vid Facit Electronics AB/Facit AB 1956–89, där han framförallt utvecklade kringutrustning som bildskärmsenheter, kärnminnen, hålkortsläsare, remsstans, remsläsare, matris skrivare.

<sup>86</sup> Carl-Olof Ternryd, f. 1928, civ.ing. lantmäteri (L), KTH 1953, tekn.lic. KTH 1971, hedersdoktor 1980. Ternryd anställdes 1957 vid VoV, där han ansvarade för utveckling av projekteringsmetodik med hjälp av fotogrammetri och databehandling. Han blev Vägverkets ställföreträdande driftschef 1967, utsågs till Vägverkets tekniska direktör 1971, till dess driftschef och ställföreträdande generaldirektör 1975 samt till dess generaldirektör 1978. Ternryd tillträdde som generaldirektör för Försvarets materielverk (FMV) 1982. Han var adjungerad professor i fotogrammetri vid KTH 1988–1995.

<sup>87</sup> Carl-Olof Ternryd, Bo Hallmén & Göran Waernér, *Fotogrammetri och datamaskiner i vägplaneringen i USA och Kanada: Erfarenheter från studieresa 3.8.–5.10. 1958* (Stockholm, 1958), 66 s.

<sup>88</sup> Eg. Bendix G-15, dator utvecklad av Bendix Corporation och som presenterades 1956. Ceruzzi (1998), 42f.

<sup>89</sup> Nordisk ADB AB, ingenjörfirma grundad av Åke Bengtsson, Bo Hallmén och Göran Waernér 1959. Nordisk ADB utvecklade dataprogram för väg- och vattenbyggnadsteknik. 1981 blev Nordisk ADB uppköpt av NordCad AB, ett företag Bengtsson startat 1980 tillsammans med Tyréns och FFNS. E-post Åke Bengtsson, 21/11 2006.

<sup>90</sup> Åke Bengtsson, f. 1931, civ.ing. V, KTH 1956. Bengtsson arbetade med byggnadskonstruktioner för Sven Tyrén AB 1956–59. Han var en av grundarna till Nordisk ADB, och där utvecklade han bl.a. datorprogram för byggnadsbranschen samt genomförde beräkningar av konstruktioner på konsultbasis. Bengtsson var VD för Nordisk ADB 1971–79. Han utvecklade CAD-teknik för arkitekt- och ingenjörsvksamhet vid NordCad AB 1980–86, och han var ansvarig för teknisk utveckling och kvalitetssäkring vid ELU-Konsult AB 1986–96. Bengtsson driver sedan 1997 Ingenjörfirma Åke Bengtsson AB.

tikmaskinens användning” på Tekniska högskolan<sup>91</sup> för avdelning V under Germund Dahlquists Tillämpad matematik.<sup>92</sup> Men då hade ju IBM kommit med 650, och för att undervisningen skulle vara helt neutral, så fick vi inte använda FA 5, som vi hade använt som programmeringsspråk, utan vi fick konstruera ett helt eget fiktivt språk, så att man skulle förstå principen.

I och med att en delegation från Bundesverkehrs ministerium kom upp och tittade på vårt system, så beställde de en perspektivfilm av motorvägen Würzburg–Nürnberg. Och den gjorde ett sånt stort intryck så att Nordisk ADB fick som första uppdrag att, eller som största uppdrag, att göra normer för tysk vägdatabehandling. Så jag har själv varit med om att skriva ”Anleitung für Elektronisches Rechnen im Strassenbau”. Och som entreprenör så tänkte man att det här måste man utnyttja. Så Bosse och jag gick till Gunnar Ericsson<sup>93</sup> på Facit och sade: ”Så här och så här ligger det till. Ni har chansen att ta över den tyska marknaden, för det är bara vi som kan räkna vägar enligt de här normerna. Ta och sätt upp en maskin i ert dotterbolag i Tyskland så ska ni se på tusan. Och så kan ni göra en servicecentral precis som Industridata.”<sup>94</sup> Och han tyckte väl att vi var framåt så. Jo, och så sa vi att, ”om du gör det så får ni 20 % av vår tyska firma under förutsättning att ni finansierar det hela”. Så startades då Nordiska ADB GMBH. Jag flyttade ner 1962, och var där mellan –62 och –65. Vi räknade ut hur mycket motorvägar som helst.

Sen skrev Bosse och jag en artikel i *Väg- och vattenbyggaren* om vägprojektering med datamaskin.<sup>95</sup> Den läste en svensk ingenjör i Spanien. Han skrev tillbaks och sa: ”Vill ni räkna vägar i Spanien?” Självklart sa vi. Sen hörde vi inte något på ett antal månader, men så kom det tillbaks en fråga: ”Kan ni projektera också?” Det var vi inte så duktiga på, men vi kände ju många i branschen, så vi åkte till Kjessler & Mannerstråle,<sup>96</sup> och sade: ”Vill ni vara med och dela? Här har ni chansen.” ”Ja, vad är villkoren?” ”Ja, 50–50 men du finansierar.” ”Javisst.” Så åkte vi ner och fick motorvägen Málaga–Torremolinos i uppdrag och sedan översiktsplaneringen ända ner till Cádiz. Så vi hade 40–50 ingenjörer som var nere och projekterade under en massa år. Sen så, om jag får snudda lite grann på 60-talet också, så vi gjorde om systemet till FORTRAN. Åke Bengtsson utvecklade massor med statiska system som också blev en förebild inom europeisk hållfasthetsberäkning. Vi gjorde en mängd olika system som täckte nästan hela väg- och vattenbyggarområdet. I mitten av –65, så sålde vi hälften av Nordisk ADB till ett konsortium av tretton av de största ingenjörbyråerna, för att få en samordnad utveckling inom branschen. Tyvärr var det dålig timing. Byggkrisen kom –66, –67, ungefär och man upplevde att datorer tog jobb ifrån de svenska ingenjörerna. Hade inte det inträffat så

---

<sup>91</sup> Tekniska högskolan i Stockholm, eg. Kungl. Tekniska högskolan (KTH).

<sup>92</sup> Germund Dahlquist, 1925–2001, fil.lic. i matematik 1949 samt fil.dr i ämnet 1958. Dahlquist var verksam som matematiker vid MMN 1950–59, där han bl.a. arbetade med uppbyggnad av programvara för BESK och med programmeringsutbildning. Han var professor i numerisk analys vid KTH 1963–90 samt gästprofessor vid Stanford University 1982–86.

<sup>93</sup> Gunnar Ericsson, f. 1919, VD för AB Åtvidabergs Industrier/Facit AB 1957–70 samt styrelseordförande där 1970–82.

<sup>94</sup> Waernérs hänvisning till Industridata AB är en liten anakronism, eftersom Facits Düsseldorfcentral inrättades 1962, medan Industridata AB bildades först 1964 av AB Åtvidabergs Industrier, ASEA, Datasaab samt AB Skandinaviska Elverk. Företaget var servicebyrådelen av nedlagda Facit Electronics AB, och affärsidén bakom Industridata var att sälja överskottstid på delägarnas datorer. Ingemar Dahlstrand, opublicerat manuskript.

<sup>95</sup> Bo Hallmén, ”Databehandlingssystem för vägbyggare”, *Väg- och vattenbyggaren* 8 (1962), 215–219.

<sup>96</sup> Kjessler & Mannerstråle AB, svensk konsulterande ingenjörfirma i bl.a. väg- och vattenbyggnad grundad 1934. Kjessler & Mannerstråle köptes 2000 av J&W som i sin tur togs över av den internationella koncernen WSP Group 2001. Se t.ex. Lars Wickström, *60 år med KM: Kjessler & Mannerstråle – rapsodi 1984–1994* (Stockholm, 1994).

hade svensk ingenjörskonst haft ett utomordentligt försprång och hade kunnat bli vad som helst. Ja.

**Lars Arosenius:** Men alltihopa körde ni i Düsseldorf?

**Göran Waernér:** Ja, vi var ju stora kunder först på BESK:en under utvecklingen, och sen så när man öppnade datacentralen Facit på Karlavägen,<sup>97</sup> så körde vi där. De tyska och spanska vägberäkningarna körde vi nere i Düsseldorf.

**Lars Arosenius:** Hade ni problem med att ni var maskinbundna?

**Göran Waernér:** Nej. Det spelade ingen roll. De som ville använda systemen, ville ju ha resultat. Var det var kört struntade de i.

**Lars Arosenius:** Men amerikanarna nappade aldrig?

**Göran Waernér:** Nej, vi spände inte bågen längre än runt här i Europa.

**Lars Arosenius:** FORTRAN-versionen kom ganska sent, sa du?

**Göran Waernér:** Jag skulle tippa att det var -64, -65, som vi började programmera om systemet.

**Lars Arosenius:** Vad hände med den sedan?

**Göran Waernér:** Ja, vi körde på 7090, och sedan så gick vi över till UNIVAC 1108.<sup>98</sup> Sen så fanns det ju fler möjligheter. När Facit tog ner Facit EDB hade vi sålt företaget till amerikanska Leased. Då var vi inte längre med i den utvecklingen.

**Lars Arosenius:** Vad spännande.

**Göran Waernér:** Och verksamheten har ju sen fortsatt inom Vägverket. De har ju nu en stor datoravdelning.

**Lars Arosenius:** Vem hade rättigheterna till programmet?

**Göran Waernér:** Det snackade vi aldrig om.

**Lars Arosenius:** Nej.

**Göran Waernér:** Jamen, vi vidareutvecklade ju programmet, och sålde sen tillbaks nya versioner till Vägverket ett antal gånger på 60-talet.

**Lars Arosenius:** Det var egentligen ett tjänsteföretag, och inte ett programvaruföretag.

**Göran Waernér:** Ja, så blev det.

---

<sup>97</sup> Elektronikavdelningen vid AB Åtvidabergs Industrier hade ett kontor på Karlavägen 62 i Stockholm. Den ombildades till dotterbolaget Facit Electronics AB 1960 och flyttade samma år ut till Solna i norra Stockholm.

<sup>98</sup> UNIVAC 1108 introducerades 1964 och var den andra datorn i Sperry Rand Corporations UNIVAC 1100-serie.

**Lars Arosenius:** Som ni märker så har vi glidit in på 60-talet. Ingemar, du har ju varit med om alla de här generationssprången, du har gått på Chalmers 1959, du står listad som Chalmers ADB<sup>99</sup> på Nordsam 1959, och sedan vet jag att du har hållit på med SARA,<sup>100</sup> D21,<sup>101</sup> D22<sup>102</sup> och Facit.<sup>103</sup> Vad har du dina erfarenheter?

**Ingemar Dahlstrand:**<sup>104</sup> Ja, när jag tänkte igenom det här nyligen, så slog det mig en sak och det var det att det gick så fort. BESK gick ju igång i december –53, och bara åtta år senare så hade vi den första kursen i ALGOL 60 för Facit EDB. Det var stora kliv. Åtta år från ren maskinkod till ett högre språk som man fortfarande skulle kunna använda idag utan att skämmas allt för mycket.

Så när jag började –55 var det fortfarande rå maskinkod. Vi hade inte fått FA4, FA5 ännu, utan det första bekymret var att få in programmet i datorn. Ifall man räknade fel på antalet ord i programmet, så skrek maskinen på mera remsa ifall det var ett ord för litet, och var det ett ord för mycket, så tolkade den det sista ordet som etiketten för nästa remsa, och det kunde bli precis vad som helst. Och väldigt ofta var det så att ett tecken fattades på remsan därför att man hade kopierat den några gånger på de här RLS-stanserna,<sup>105</sup> och det var lika galet det.

Och vad hade man för sätt att ta reda på vad som hade gått fel? Ja, man fick sätta igång maskinerna att skriva ut vad som stod i minnet hexadecimalt, och så sitta med remsan och se var glappet var någonstans. Så när FA4, FA5, kom, så var det fantastiskt. Det var inte bara det att man slapp numrera om hela tiden, som Hans nämnde förut, utan när man hade skrivit programmet och kompilerat det och fått ut en remsa, så gick det normalt att mata in direkt och köra. Halva jobbet var gjort.

Och sen gick det ju lite långsamt en tid just på Facit. Vi köpte in ALFAKOD, men det hann liksom inte komma i bruk, och vi hade aldrig någon assembler med alfabetiska adresser av det skälet att de första åren kunde maskinen inte läsa alla fem kanalerna på remsan. Den läste bara fyra kanaler och det var bara hexadecimala siffror. Men någon gång under de här åren i slutet på 50-talet, så lärde den sig att läsa den femte kanalen, och vad jag vet så var det firma Autocode som först utnyttjade detta till ALFAKOD.

Sen så kom, som jag nämnde förut, de här kommittéerna som arbetade fram ALGOL 60, och jag hade turen att få ansvaret för att implementera det tillsammans med en kamrat som hette Sture Laryd.<sup>106</sup> Och han tog på sig att göra det som var alldeles nytt

---

<sup>99</sup> Eg. ADB Institutet vid Chalmers tekniska högskola. Det inrättades 1957 och var en donation från finansmannen Axel Wenner-Gren till Chalmers. Donationen omfattade datorn Alwac III-E samt bemanning med sex personer under tre år. 1960 förlängdes donationen, men 1965 lades ADB Institutet slutligen ned. Ingemar Dahlstrand, opublicerat manuskript.

<sup>100</sup> SARA, förkortning för Saab:s aritmetiska räkneautomat, BESK-kopia vid Saab i Linköping som var i bruk årsskiftet 1957/58.

<sup>101</sup> D21, dator som konstruerades av Saabs elektronikavdelning. Chefskonstruktör var Viggo Wentzel och det första exemplaret stod klart 1962. Tord Jöran Hallberg, ed., *Tema D21* (Linköping, 1994).

<sup>102</sup> D22, en efterföljare till D21 som presenterades 1966. Hallberg (1994).

<sup>103</sup> Eg. Facit EDB, se fotnot 35.

<sup>104</sup> Ingemar Dahlstrand, f. 1932, fil.mag. i matematik, fysik och mekanik, Uppsala universitet 1955. Dahlstrand arbetade med Börje Langefors vid Saab:s beräkningskontor i Linköping 1955–57. Därefter gick han över till ADB Institutet vid Chalmers i Göteborg 1957–59. Dahlstrand ledde implementeringen av ALGOL 60 för Facit EDB vid Facit Electronics AB/Facit AB i Göteborg 1959–64. Han var systemchef vid Industridata AB:s Göteborgscentral 1964–66 samt distriktschef där 1966–68. Dahlstrand var driftschef för Lunds datacentral för forskning och högre utbildning 1968–79 och programmerare där 1980–85. Vidare var han adjunkt vid Institutionen för datalogi och numerisk analys vid Lunds universitet 1985–1997.

<sup>105</sup> RLS-stans, IBM-skrivmaskin kopplad till remsläsare och remsstans.

<sup>106</sup> Sture Laryd, f. 1932, statistiker. Han implementerade ALGOL 60 tillsammans med Dahlstrand vid Facit Electronics AB/Facit AB 1960–64. Laryd arbetade sedan med datakommunikationsproblem på Facits kontor i Solna utanför Stockholm fram till 1967, då han gick över till Datasystem AB.

för oss, nämligen att göra en formelöversättare. Assemblering hade jag hållit på med tidigare och det var liksom mera välkänt, men det här var helt ny mark. Och vi reste till Köpenhamn, till Peter Naur<sup>107</sup> och Bech<sup>108</sup> och de människorna där. De lärde oss väldigt mycket. Sen fick vi en artikel från den tysk-schweiziska ALCOR-gruppen med Samelson, Bauer och Rutishauser som författare, som gav oss lite elementa om hur man skulle översätta en formel till elementära operationer.<sup>109</sup> Och så fick vi det här klart på ungefär ett år, och sen var det en dramatisk effekt av detta. Jag har kvar en artikel som jag skrev i *BIT* ett halvår efteråt.<sup>110</sup> Då hade vi redan 80 % av nyprogrammeringen i ALGOL och 50 % av beläggningen totalt. Vi hade gjort ALGOL så att det skulle gå fort och köra det bildade programmet. Vi avstod från finesser i ALGOL 60 för den sakens skull. Och det gjorde att det var konkurrenskraftigt, och folk gick över direkt.

Sen för att kommentera lite mer det som du [Bo Lindestam] nämnde från Standard Radio, det var en skillnad mellan amerikanerna och européerna, och det berodde på att amerikanerna på ett tidigt stadium hade börjat bygga flytande räkning i centralenheten. Så när de skulle göra FORTRAN så var de ganska kräsna. När vi gjorde ALGOL, så kunde vi trösta oss med att de flytande operationerna som vi programmerade ändå tog 30, 40, 50 additionstider, och så lite extra för administration, 15 % eller så, men amerikanerna var vana vid att det skulle gå fort hela vägen, så där måste de göra optimerande kompilatorer redan från början, och lade ner mycket jobb på det här. Därför lade de ner så mycket fler manår på FORTRAN än vi gjorde på ALGOL. Och det gjorde också att när det kom till kritan, så slog inte ALGOL igenom, mer än under några år.

Kring -64, -65, så kom flera amerikanska superdatorer in, ASEA köpte den som vi kom att köra på, och universitet och högskolor köpte superdatorer, och allihopa hade FORTRAN-miljö. Så det blev FORTRAN som man översatte programmen till. Men på UNIVAC 1108, som jag kom att jobba på sen, där var ALGOL-miljön och FORTRAN-miljön kompatibla med varandra. Man kunde ta en subrutin för FORTRAN och sätta in i ett ALGOL-program. Man var inte avskuren från FORTRAN-världen om man använde ALGOL. Men så var det inte överallt, på GE-625<sup>111</sup> till exempel var det vattentäta skott och där dog ALGOL ut ganska fort.

**Lars Arosenius:** Med vad du vet nu, skulle du ha startat med ALGOL-utveckling, eller skulle du ha gått på FORTRAN-kompilator istället?

**Ingemar Dahlstrand:** Ja, det är en intressant fråga, och jag har inget självklart svar på den. FORTRAN kom ju att utvecklas vartefter, så att många av de bra sakerna i ALGOL kom också in i FORTRAN. Men det tog ända till 1977 innan "if", "then", "else" kom in i FORTRAN. Fram till dess var det en massa villkorliga hopp. Programmen såg ut som rishögar. Väldigt många människor, bland annat Mossberg<sup>112</sup> på Flygmotor<sup>113</sup> och någon

---

<sup>107</sup> Peter Naur, f. 1928, astronom samt professor i datalogi vid Köpenhamns universitet sedan 1969. Han var i slutet av 1950-talet en av de ledande gestalterna i utvecklingen av ALGOL 60. Mellan 1959 och 1969 var Naur anställd vid Regnecentralen i Köpenhamn.

<sup>108</sup> Niels Ivar Bech, 1920–1975, var chef för Regnecentralen 1955–71 och en drivande kraft i den tidiga datautvecklingen i Danmark.

<sup>109</sup> Klaus Samelson & Fritz L. Bauer, "Sequentielle Formelübersetzung", *Elektronische Rechenanlagen* 1 (1959), 176–182; idem, "Sequential Formula Translation", *Communications of the ACM* 3 (1960), 76–83.

<sup>110</sup> Ingemar Dahlstrand, "A Half Year's Experience with the Facit-Algol 1 Compiler", *Nordisk tidskrift för informationsbehandling: BIT* 1962:3, 137–142.

<sup>111</sup> GE-625, dator i General Electric's GE-600-serie som presenterades 1965.

<sup>112</sup> Lars G. Mossberg, ingenjör vid Svenska Flygmotor AB som konstruerade en förkompilator för FORTRAN.

<sup>113</sup> Eg. Svenska Flygmotor AB, svenskt flygmotorföretag i Trollhättan. Volvo köpte 1941 Nohab Flygmotorfabriker AB och bildade Svenska Flygmotor AB som sedan 1994 heter Volvo Aero Corporation.

på Saab, vars namn jag inte minns,<sup>114</sup> gjorde förkompilatorer, så man kompilerade från ett förbättrat FORTRAN med ”if then else” till standard-FORTRAN. Så att det var inget svårt, inget som kostade effektivitet i det slutliga programmet egentligen. Men det tog tid innan kommittén hade jobbat färdigt och fått fram en standard.

**Lars Arosenius:** Men man lade ju ner mycket tid på att göra de här alternativa program-språken, du hade ALGOL, du hade SIMULA...<sup>115</sup>

**Ingemar Dahlstrand:** Ja, utvecklingen av SIMULA har jag inte haft mycket kontakt med. Det gjorde norrmännen huvudsakligen. Men däremot så fick jag en del att göra med ALGOL-GENIUS<sup>116</sup> för att Langefors<sup>117</sup> grupp hade definierat GENIUS. Ja, ALGOL-GENIUS är detta amalgam av ALGOL och COBOL som ECMA efterlyste. Det gjordes ju verkligen. Då gjorde vi en deal med Saab att vår ALGOL-grupp skulle göra ALGOL-delen till ALGOL-GENIUS.

**Lars Arosenius:** Vilka var vi?

**Ingemar Dahlstrand:** Facit. Vår ALGOL-grupp skulle lämna över flödesplanerna för ALGOL-kompilatorn, och hjälpa till att tolka dem och göra om dem till D21. Och sedan skulle folket i Linköping göra GENIUS-delen. Och så skedde, och det kom fram på tid, och det fungerade bra. Och sen fanns ALGOL-GENIUS som ett mycket användbart språk. Och när Saab, Datasaab,<sup>118</sup> såldes till Univac, 1975 ungefär, då åtog sig Univac att implementera ALGOL-GENIUS. Så det gjordes det också. Så det fortsatte att finnas länge. Jag tror inte vi skulle varit lyckliga om vi hade varit utan ALGOL-projektet för det gav väldigt många rotskott och nya idéer och befruktade även FORTRAN på sikt.

**Elsa-Karin Boestad-Nilsson:** Och SIMULA.

**Ingemar Dahlstrand:** Och SIMULA, ett av de bästa språken som finns. Och jag ska genast bekänna att från 1977 så arbetade jag aktivt med standardisering av FORTRAN. Jag tänkte att om det nu är de som vinner, så får man jobba inifrån. Så jag var med där

---

<sup>114</sup> Dahlstrand syftar på Lars Jeneskog vid Saab-Scania som gjorde en förkompilator för datorn PDP-11. Se även fotnot 134. Brev Ingemar Dahlstrand, 23/1 2007.

<sup>115</sup> SIMULA, programspråk för datorer baserat på ALGOL 60, utformat av de norska datalogerna Ole-Johan Dahl och Kristen Nygaard under 1960-talet. Språket konstruerades ursprungligen för simulering och systembeskrivning. Det räknas som det första objektorienterade språket. ”SIMULA Session”, i *History of Programming Languages*, ed., Richard E. Wexelblat (New York, 1981), 439–493.

<sup>116</sup> ALGOL-GENIUS, ett derivat av programspråket ALGOL 60 med inslag av COBOL. Idégivare till ALGOL-GENIUS var Börje Langefors som utvecklade programvaran till Saabs dator D21. Han föreslog att ALGOL bl.a. kompletterades med ett generellt in/utsystem (GENIUS) modellerat efter COBOL. Utifrån Ingemar Dahlstrands implementering av ALGOL 60 för Facit EDB utvecklade Langefors grupp därefter ALGOL-GENIUS, och den första versionen presenterades 1964. ALGOL-GENIUS användes för Saabs datorer D21 och D22 samt UNIVAC 1100. Bengt Asker, ”ALGOL-GENIUS: An Early Success for High-level Languages”, i *History of Nordic Computing: IFIP WG9.7 First Working Conference on the History of Nordic Computing (HiNC1), June 16–18, 2003, Trondheim, Norway*, eds. Janis Bubenko, Jr, John Impagliazzo & Arne Solvberg (New York, 2005), 251–260.

<sup>117</sup> Börje Langefors, f. 1915, civ.ing. teknisk fysik (F). Langefors var ingenjör vid Nordiska Armaturfabrikerna (NAF) AB i Stockholm och Linköping 1944–49, vid Saab 1949–65, docent i byggnadsstatik vid Chalmers tekniska högskola 1963, professor i informationsbehandling, särskilt den administrativa databehandlingens metodik, vid KTH 1967–80. Som ingenjör på Saab ledde han bl.a. arbetet med konstruktionen av SARA.

<sup>118</sup> Datasaab, division inom Saab-Scaniakoncernen som formerades vid 1960-talets början med utgångspunkt i Saabs data- och elektronikdivision. Dess verksamhet låg inom datortillverkning och systemkonstruktion.

och puttade in ALGOL-idéer var helst det gick. Och det finns nu som FORTRAN 90.<sup>119</sup> Vi skojade när FORTRAN 90 skulle komma ut, ”att det här är inte en ansiktslyftning, det här är en huvudtransplantation”.

**Lars Arosenius:** Och så gav ju ALGOL samarbete mellan Saab och Facit. Som ju var ett kors i taket på den tiden.

**Ingemar Dahlstrand:** Ja, det var ett väldigt intimt samarbete på systemsidan, tyvärr inte lika intimt på den tekniska sidan.

**Lars Arosenius:** En D21, sedan fick du D22.

**Ingemar Dahlstrand:** Ja, det var detta om ALGOL-GENIUS.

**Lars Arosenius:** Det var ALGOL-GENIUS som kom. Bengt Gällmo, jag tror att du var med och skrev en av ALGOL-kompilatorerna för IBM?

**Bengt Gällmo:**<sup>120</sup> Jo, som jag minns historien, så kom väl FORTRAN och ALGOL till ungefär samtidigt. Sisådär –57, –58, nåt i den stilen, va? Därför att –59 när jag körde mitt första program i FLOPS på IBM 650 i Germund Dahlquists kurs ”Siffermaskinens användning” då fanns det faktiskt en FORTRAN i miniatyr, en bantad FORTRAN på 650 som hette FORTRANSIT.<sup>121</sup> Det var derivat av FORTRAN redan –53. Och ALGOL har vi ju redan hört om, ALGOL 58, ALGOL 60.

Sedan som IBM uppfattade situationen, så var det att ALGOL var nånting formfulländat, pryddigt och fint, men ganska opraktiskt. Det gick inte att kompilera till direkta program om man inte gjorde ganska godtyckliga restriktioner i språket, till exempel som vi gjorde rekursiva procedurer. Men då försvann ju den där skönheten och formfulländningen, och det blev olika varianter av ALGOL som inte var kompatibla med varandra. FORTRAN å andra sidan, redan från början hade man insett att man måste begränsa språket för att det skulle gå att göra effektiva program av det, och därför tillät inte FORTRAN till exempel rekursiva procedurer i språket, och hade en del andra restriktioner, de här enkla ”if” till exempel.

Att IBM kom att göra en ALGOL-kompilator, som jag alltså hade fingret med i, och ett visst ansvar för, det var nog mest en marknadsföringsgrej. Den här striden mellan ALGOL och FORTRAN rasade i ganska många år. IBM var inne och körde med FORTRAN, men hur det nu än var så tyckte man att man i alla fall måste ha en ALGOL-kompilator i sortimentet. Så vi fick i uppdrag på IBM Nordiska Laboratorier<sup>122</sup>

---

<sup>119</sup> FORTRAN 90, en FORTRAN-version som släpptes som ANSI-standard 1992.

<sup>120</sup> Bengt Gällmo, f. 1937, civ.ing F, KTH 1963. Gällmo började vid IBM Svenska AB 1960 som tekniskvetenskaplig och systemprogrammerare. Han blev senare chef för IBM Nordiska Laboratorier på Lidingö i Stockholm. Därefter gick han över till Stansaab AB/Datsaab AB/Ericsson Information Systems AB, där han var utvecklingschef samt innehade stabspositioner 1975–86. Gällmo var på LM Ericssons koncernstab Teknik 1986–97. Sedan 1994 driver han egen verksamhet, Indak HB.

<sup>121</sup> FORTRANSIT, ett s.k. subset av FORTRAN som användes för IBM 650 och IBM 704 under senare delen av 1950-talet. David A. Hemmes, ”FORTRANSIT Recollections”, *Annals of the History of Computing* 8 (1986), 70–73.

<sup>122</sup> IBM Nordiska Laboratorier var ett produktutvecklingslaboratorium och utgjorde del av IBM:s omfattande forskningsorganisation. Det etablerades 1960 och lokaliserades till Lidingö i Stockholm. Dess första chef var Cai Kinberg som tidigare arbetat på IBM:s laboratorium i Zürich. IBM Nordiska laboratorier fick en inriktning mot processkontroll och arbetade bl.a. med styrning av pappersmaskiner på Billerud. Forskare från flera nordiska länder arbetade vid laboratoriet. Hans Andersin, ”The Role of IBM in Starting up Computing in Nordic Countries”, i *History of Nordic Computing: IFIP WG9.7 First Working Conference on the*

att utveckla en ALGOL-kompilator, och sen tog det IBM på vanliga sättet ett stort antal månår, jag kommer inte ihåg hur många, och vi beundrade Ingemar som enligt ryktet tillverkade en ALGOL-kompilator egenhändigt på ett halvår. Så effektiva var inte vi. Men i alla fall, vi tog fram en ALGOL-kompilator, och sen var det ingen som använde den heller. Det var alltså för att det skulle synas i sortimentet.

Men annars så har jag en sak som jag skulle vilja lyfta fram, och det är trickkodning. Maskinerna var ju så väldigt begränsade. Och det är inte bara det att vi ser i backspegeln idag, utan det kände vi nog redan då. Det var ont om minne, väldigt ont om minne. Och det tog en evig tid om man inte passade sig och skrev effektiva program. Man lade ner väldigt mycket arbete på att utveckla algoritmer, speciellt sorteringsmetoder. Numeriska algoritmer och allt möjligt sånt där. Men sen trickkodade man också. IBM 1401<sup>123</sup> var skräckexemplet, därför att den hade såna lustiga egenskaper. Den hade ju variabel ordlängd, och om man höll reda på vad som fanns kvar i adressregistren efter en instruktion, så behövde man inte skriva några adresser för instruktionen efter, då kunde man bara skriva operationskoden, med ett tecken istället för 6 eller 7 eller 8. På det viset kunde man skriva långa haranger med bara operationskoder, och ingen människa kunde förstå det där programmet, inte ens författaren själv, så det var ett elände.

Jag kommer ihåg en liten historia när jag fanns på FOA tillsammans med Björn Kleist och drog igång IBM 7090. Jag skulle försöka få in ett reaktorprogram av någon sort på IBM 7090. Det där programmet hade tidigare körts på en IBM 709 eller 7090, vilken det nu var, och fått plats på minnet, men på FOA fick det inte plats. För FOA hade ett rudimentärt operativsystem som hette FORTRAN Monitor, och det tog i alla fall något litet plats. Så jag satte igång och bantade Monitor för att få plats för reaktorprogrammet. Och då upptäckte jag att i Monitor ingick en subrutin som hette EXEM<sup>124</sup> på 500–600 ord, tror jag, det var mycket, som bara bestod av felmeddelanden. Och när nånting gick åt skogen i programmet, så hoppade man till EXEM med en numerisk kod i ackumulatorn, och så tog EXEM fram felmeddelandet. Jag tänkte, det här programmet är välbeprövat, det ska inte förekomma division med noll i det här programmet. Det har körts i många herrans år. Så jag tog bort EXEM och satte in en kort snutt som bara skrev ut ”felmeddelande nummer si och så, för närmare information, kontakta FOA datacentral!” Detta var si så där –61 skulle jag tro. –69 var det så att jag på IBM som chef fick ett telefonsamtal från QZ eftersom de hade fått ett sånt mystiskt felmeddelande, och då trodde de att jag hade nånting med det att göra. Det hade dividerats med noll i alla fall! Och man var alltså tvungen att göra såna där saker för att att få rimlig effektivitet och få in programmen överhuvudtaget. Det är för mig ett bestående minne, och vi hade som sagt på IBM 1401 skräckexemplet, att själva definitionen av en dator var ungefär ”en kalkylator som kan modifieras med sitt eget program”. Och programmen låg ju lagrade med data på precis samma sätt, med precis samma typ av minne, ibland huller om buller. Så man kunde, avsiktligt eller oavsiktligt, råka modifiera sitt program istället för data. Det började bli ordning på det när Dijkstra skrev det berömda pappret i slutet av 60-talet, –69, tror jag, om ”go to”-statement.<sup>125</sup> Och jag har hävdad att egentligen hade han fel. Det var inte ”go to”-statementet som var problemet, utan det var den förbaskade adressmodifieringen. Det var det som gjorde programmen oläsliga.

Ja, för övrigt började som sagt min historia med Germund Dahlquist under kursen. Vi skrev i det här fiktiva språket som du [Göran Waernér] tydligen var varit med om att

---

*History of Nordic Computing (HiNC1), June 16–18, 2003, Trondheim, Norway*, eds. Janis Bubenko, Jr, John Impagliazzo & Arne Solvberg (New York, 2005), 33–43; E-post Lars Arosenius, 4/12 2006.

<sup>123</sup> IBM 1401, se fotnot 47.

<sup>124</sup> EXEM, förkortning för Execution Error Monitor.

<sup>125</sup> Edsger W. Dijkstra, ”Go To Statement Considered Harmful”, *Communications of the ACM* 11 (1968), 147–148.

utforma. Vi fick välja mellan att göra ett riktigt exempel på antingen BESK, Facit eller IBM 650. Och till mina kurskamraters förvåning så valde jag IBM 650, för den var inte alls lika glamorös som BESK och Facit. Men jag gjorde i alla fall ett program, och även i slutet på 50-talet så kunde man inte riktigt lita på hårdvaran. Jag löste en ekvation som jag inte alls kommer ihåg vad det var för nånting. Jag gjorde så att jag räknade först fram alla resultat, och sen så satte jag igång och stansa ut dem för senare tabulering, man gjorde så. Programmet gick i fem minuter, till synes utan att det hände nånting. Det var bra där att min handledare slog av maskinen. Man gjorde inte så, utan det gällde att programmera så att man antingen slängde in kontroller, som någon sa tidigare, eller så lät det hända någonting med jämna mellanrum, så att operatören kunde se att programmet gick.

Sen efter det så fick jag åka till USA och lära mig IBM 709, och sen kom jag hem och var en av fyra personer i landet ungefär, jämte Björn, som förstod FOA:s nya dator. Så jag och Björn jobbade ihop. Ja, det kan väl räcka.

**Lars Arosenius:** Tack ska du ha, spännande tider. Bjarne, ska du också berätta dina erfarenheter? Bjarne kommer ju från telefonaktiebolaget Ericsson, och du har ju en liten annan vinkel än de övriga här, eftersom du har sysslat med programsystem som ska se till att våra telefonstationer fungerar rätt. Eller hur?

**Bjarne Däcker:**<sup>126</sup> Ja, jag tror inte att jag är fullt så pionjärartad som många andra här i panelen. Jag började ju inte jobba förrän 1966. Det var ju faktiskt väldigt idylliska tider jämfört med vad ungdomarna har nu. Jag hade varit utomlands ett par år. Så sökte jag jobb, och så gick jag på Statistiska centralbyrån, IBM och på LM, som det kallades då, inom en vecka. På Statistiska centralbyrån så blev jag satt på ett prov för programmerare. Så vi satt där. Och det var på den tiden man trodde att programmering var ungefär som att lösa korsord, så det var ju såna här frågor i stil med, fylld triangel, tom triangel, fylld cirkel, vad kan nästa vara och så vidare. Dagen efter så var jag på IBM och de sa: ”Ja, vi har ett prov här”. Det visade sig att det var faktiskt samma prov. Dagen efter det var jag på LM i Älvsjö och då frågade de: ”När kan du börja jobba?” ”Ja, jag kan börja jobba imorn.” ”Okej, vi börjar åtta.” Ja, så gick det till, och jag fick förstås ringa till de andra och berätta att jag faktiskt hade fått ett jobb. Och de var väldigt ledsna på IBM för ingen hade klarat det här provet så bra. Jag hade ju inte hjärta att skvallra om att jag faktiskt fått köra ett genrep först.

Och då så går jag in i det här klockhuset i Älvsjö där det fanns en programmeringsavdelning, och de första åren så arbetade jag med AKE 12,<sup>127</sup> där den egentlige initiatören var Kurt Katzeff<sup>128</sup> som ju sitter här. Det handlade om det datorstyrda telefonväxelsystemet AKE 12 i Tumba. På den tiden kallades det inte datorstyrt utan programminnes-

---

<sup>126</sup> Bjarne Däcker, f. 1942, tekn.lic. KTH 2000, hedersdoktor vid Linköpings universitet 1993. Däcker anställdes vid LM Ericsson 1966, där han stannade fram till 2002. Han arbetade huvudsakligen med CAD för hårdvara och systemstöd. Däcker utvecklade bl.a. mjukvaran till ett testsystem för kretskort som användes för AXE 10-växlarna samt programspråket ERIPASCAL. Han var initiativtagare och chef för Datalogilaboratoriet vid Ericsson 1984–2002. Under Däckers ledning utvecklade Datalogilaboratoriet bl.a. programspråket ERLANG. Han är sedan 1990 ledamot av IVA.

<sup>127</sup> AKE 12, elektronisk och datorstyrd telefonväxel som började utvecklas 1962 och som användes för en av Televerkets lokalstationer i Tumba söder om Stockholm. AKE 12-stationen prövades första gången i drift 1967.

<sup>128</sup> Kurt Katzeff, f. 1926, civ.ing. E, KTH 1950, hedersdoktor i Lund 1994. Katzeff arbetade vid LM Ericssons telefonstationsdivision 1950–72 och blev teknisk chef för divisionen 1962. Han var teknisk chef för ITT Europe:s televerksamhet 1972–80, teknisk chef för Televerket 1980–91 och startade där Telelogic. Katzeff var chef för det europeiska forskningsinstitutet EURESCOM i Heidelberg. Vidare är han ledamot av IVA och Life Fellow i IEEE.

styrt. Och sen arbetade vi också med ett väldigt tidigt CAD-system, men avsett för relä-satser som kallades för AKORD.<sup>129</sup>

Men det minne som jag har framförallt av den tiden var just hur långt borta datamaskinen var. Nu har man ju den i knät, bokstavligen talat. Men då såg man den ju sällan, och särskilt när det gällde att programmera telefonväxlar. Det handlade om att skriva något program som skulle stansas, skicka iväg det och så skulle programmet kompileras på en ORION-maskin.<sup>130</sup> Det var en maskin utvecklad av Ferranti i Storbritannien.<sup>131</sup> Och resultatet skulle i sin tur skickas till den dator som styrde telefonstationen, och då handlade det ju mest om programtest på en anläggning i Västberga. Man gick till nån som hette Niklasson som körde programmet. Och hela den här hanteringen var så bökelig att det bara fanns ett sätt, och det var att sitta och lusläsa sitt program så att det inte skulle vara några fel. Det var helt enkelt så arbetsamt.

Men när jag höll på med det här AKORD-systemet för reläsatser satt jag i Älvsjö, men datorn fanns på ICL<sup>132</sup> i Solna. Det betydde att det kom en budbil två gånger om dan, en liten kille som såg ut ungefär som en överårig raggare som körde en Volvo Duett. Och allt som oftast så kastade man sig över sina utskrifter, och så fick man se, ”jamen vänta, där och där är det ju fel”, gick och stansade ett par kort, för han var ju där i alla fall i fem-tio minuter, och så åkte han iväg. Lagom som han hade åkt så insåg man plötsligt: ”Herregud, felet låg ju här istället!” Jag hade kvar det ursprungliga felet plus att jag hade lagt till ytterligare ett. I hopp om att snabba upp det hela så hade jag alltså fördröjt det. Och det mest larviga inträffade ju förstås den gången då han gjorde en rivstart på Karlbergsvägen och bakdörrarna öppnade sig och alla våra hålkort spreds ut över gatan. Men just att man hade datorn väldigt långt borta.

Vi upplevde ett par tillfällen då det vände. Inte minst var jag involverad med att utveckla ett testsystem för kretskort för AXE 10<sup>133</sup> som byggde på PDP-11.<sup>134</sup> Här var en helt ny sorts dator, och det var liksom en revolution. Det var den första minidatorn, och den hade vi på vårt kontor. Det var bara det att det inte fanns något operativsystem, så vår avdelningschef Stig Larsson,<sup>135</sup> som sen hamnade på SJ, kom över till mig, och han var lite bekymrad. Jag var ju chef för en liten grupp som skulle utveckla det här väldigt viktiga testsystemet för kretskorten för AXE 10, så han undrade vad det fanns för program för den här PDP-11:an. Ja, det visade sig att det fanns faktiskt 16 program, varav 12 var spel. Men just att man inte hade ett operativsystem hade åtminstone den fördelen att man ju faktiskt hade total kontroll, och sen var det betydligt mindre att lära sig.

Så småningom så hamnade vi i situationen att det blev en slags maktkamp på Ericsson kring datorresurserna. För vi hade ju stordatoravdelningar som ju förstås körde ekonomisystem, lönesystem och sånt där, men de skulle också serva Ericssons konstruktionsavdelningar med program. Dessa hade ju lite annorlunda slags behov, men det

---

<sup>129</sup> AKORD, förkortning för automatisk konstruktion av reläatsdokumentation.

<sup>130</sup> ORION, dator konstruerad av det brittiska företaget Ferranti kring 1960. Den var en efterföljare till Ferrantis bästsäljande dator PEGASUS som introducerades 1956. Cortada (1987a), 299f.

<sup>131</sup> Eg. Ferranti Company, Ltd., ett av de första brittiska företagen som producerade och sålde datorer. Den första datorn som företaget utvecklade var Mark I som började säljas 1951. Ferranti sålde 1963 sin datorverksamhet till International Computers and Tabulators (ICT). Cortada (1987c), 128f.

<sup>132</sup> Se fotnot 153.

<sup>133</sup> AXE 10, datorstyrd digital växel som började utvecklas 1972 under ledning av Bengt-Gunnar Magnusson vid företaget Ellemtel Utvecklings AB vilket ägdes gemensamt av LM Ericsson och Televerket. Den första AXE-stationen togs i drift i början av 1977.

<sup>134</sup> PDP-11, en minidator som introducerades 1970 av Digital Equipment Corporation (DEC). Cortada (1987a), 296ff.

<sup>135</sup> Stig Larsson, f. 1931, ingenjör. Han kom till Ericsson 1960 och arbetade där med signalsystem för väg- och järnvägstrafik, datorisering av telefonstationer samt med AXE-systemet. Larsson blev chef för koncernens bolag RIFA 1979, chef för Ericsson Information Systems 1985 och slutligen generaldirektör för SJ under åren 1988–98.

förstod inte datoravdelningen som ju bara ville sälja MIPS.<sup>136</sup> Så jag var med om att införskaffa en av de första VAX:arna,<sup>137</sup> såna där ”midi”-datorer, och det var mycket bråk om den. Men det hör till saken att till slut fick vi faktiskt okej för att köpa den här VAX:en, för vi uppfattades som helt ofarliga, för på den så skulle vi köra nåt konstigt universitetssystem som hette UNIX,<sup>138</sup> och de räknade med att ingen annan nånsin skulle vilja använda det.

I samband med det här så blev jag intresserad av teknik och programspråk för realtidsprogrammering. Om man jämför med ALGOL och FORTRAN, så krävs ju ett par ytterligare aspekter för realtidsprogrammering. Något slags modulbegrepp, därför att bygger man ett stort program så kan man ju inte göra det bara som en stor monolit, det är ju många människor som programmerar. Men framför allt så måste det finnas något slags processbegrepp, eftersom program ju inte bara går i en sekvens: startar, läser indata, gör beräkningar, skriver ut utdata, slutar, utan det ska ju avverka en mängd parallella processer, och framför allt så är det en parallellitet som jag tror många människor har svårt att inse. Det handlar inte om tio parallella processer, utan det handlar om hundratal, tusentals, så det krävs att programspråket och hela implementationstekniken måste vara väldigt bra på att kunna hoppa mellan dem. Och sen så krävs det ju tekniker för felhantering. Det är ju så att om man har ett program och jag råkar dividera med noll, då brukar ju ett vanligt operativsystem bryta programmet. Man blir utslängd helt enkelt. Men om man har en telefonväxel som går, och ett debiteringsprogram av nån anledning dividerar med noll, så får det ju inte stanna hela växeln. Just den processen får stanna och kastas ut, men alla andra måste kunna köras vidare. Ett av de verkligt stora lyften med Ericssons AXE-system stod en person som heter Göran Hemdahl<sup>139</sup> för. Det var ju det första system som verkligen skrevs i ett högnivåspråk, och det var det egenutvecklade språket PLEX.

**Elsa-Karin Boestad-Nilsson:** PLEX?

**Bjarne Däcker:** Ja, det stod för Programming Language for Exchanges. Men det intressanta var ju att i det tidigare systemet AKE 12 för Tumba kunde det liksom hända väldigt otrevliga saker eftersom systemet var totalt flexibelt. Alla program kunde skriva och läsa i hela minnet. Och jämfört med att bygga reläsats, en reläsats kan ju inte bli större än en viss storlek för sen blir den ju ohanterbar, så hade man inga begränsningar. Och så småningom visade det sig att det ju kunde leda till otäcka fel. Ett program kunde gå fel på grund av att det var nåt annat program som varit inne och rotat i dess data av misstag, därför att man hade haft ett pekarfel. Och det var det som var så fiffigt med AXE-systemet PLEX, det var ju inte så att antalet pekarfel minskade med nån viss faktor eller nåt sånt, utan man kunde i princip eliminera dem.

Vi hade ett experimentspråk som hette PASTEL, och det var en förkortning för PASCAL for Telecommunications. Ja, ni kan ju tänka er vilka tjusiga manualer vi skulle kunna göra. Medan det här PLEX-språket påminde mer om BASIC och FORTRAN, det hade lika tråkiga ”if”-satser och så vidare, så byggde vi nu mer på ALGOL och PASCAL.

---

<sup>136</sup> MIPS, miljoner instruktioner per sekund, mått på exekveringshastighet hos datorer.

<sup>137</sup> VAX-11, dator som konstruerades av DEC och utannonserades 1977. Den var en uppföljare till DEC:s PDP-11-dator. Ceruzzi (1998), 243ff.

<sup>138</sup> UNIX, operativsystem som började utvecklas 1969 vid AT&T Bell Laboratories av Ken Thompson och Dennis Ritchie.

<sup>139</sup> Göran Hemdahl, en av de ledande ingenjörerna bakom utvecklingen av AXE-systemet. Han lämnade Ellemtel för ITT 1977 för att senare starta eget företag.

Det första språk som jag var med om att utveckla och som verkligen blev nåt var PL163<sup>140</sup> för ett datorsystem som hette APN163.

Det här med programspråk uppfattas numera som väldigt svårt, eller som standard. Det är i princip ett område som har stagnerat. Men då var det idylliska tider, det var ingen som bekymrade sig över det: ”Vi har en ny processor här, klart vi ska ha ett nytt språk.” Och om det sedan behövdes något operativsystem. Ja, det var inte så svårt, vi var ju i alla fall ett par man.

Det intressanta med PL163-språket är bland annat att det visar hur stort inflytande som Niklaus Wirth<sup>141</sup> har haft. PL163 kommer ju från Niklaus Wirths PL360,<sup>142</sup> vilket var en fullständigt genialisk idé. Det är lika strukturerat som ALGOL med ”if”, ”and”, ”else” och hela det köret, men alla data är på assemblernivå. Det står saker som ”if ’register 5’ > 105 then”, så det är ett högnivåspråk, men det är så extremt effektivt. Sen hade vi processer. Nästa språk som jag var involverad med var ERIPASCAL, som var ett ”subset” av CHILL men med PASCAL-syntax,<sup>143</sup> och det ledde så småningom till att jag blev involverad med Ericssons arbete kring ADA,<sup>144</sup> och vi hade mycket med FOA att göra. Jag var med om att starta Föreningen Ada i Sverige.<sup>145</sup> Och då hade vi förhoppningar om ett standardrealtidsspråk. Tyvärr så var det ju som så att det var jättestort intresse för ADA, men så gick det några år då det inte kom fram någon riktig kompilator, och så när det äntligen gjorde det hade liksom intresset svalnat och då hade ju C seglat upp däremellan.

**Lars Arosenius:** Nu är det kaffepaus.

---Kaffepaus---

**Lars Arosenius:** Låt oss börja den här andra sessionen med att säga att meningen med att göra programspråk var ju att man skulle få kompatibilitet mellan olika maskiner, så att man så småningom skulle komma till den ideala världen där varje program kunde köras på vilken annan maskin som helst. Och Bengt var ju inne lite på det här att man fick vara ganska smart som programmerare på 50-talet för att utnyttja det begränsade minnet som fanns, och det ställde ju till oändliga problem för alla oss som höll på med hårdvara. För varje gång det kom en ny generation hårdvara så upptäckte man att det var nån programmerare som hade utnyttjat nån okänd faktor i systemen, så att systemen inte kunde gå. Men när programspråken kom, så sa man, ”nu har den ideala världen kommit, vi kan använda högnivåspråk”. Var det så? Åke Bengtsson här var ju med att göra den där FORTRAN-versionen av vägplaneringsprogram. Fungerade det? Kunde du köra på olika system sedan?

**Åke Bengtsson:**<sup>146</sup> Ja, vi utvecklade program som statistiska beräkningar...

---

<sup>140</sup> PL163, programspråk utvecklat på Ericsson för datorsystemet APN163 under projektledning av Ingemar Ahlberg. ”Datalogilaboratoriet – de första 10 åren”, Elmetel Utvecklings AB, 1994.

<sup>141</sup> Niklaus Wirth, f. 1934, professor i informatik vid ETH i Zürich, Schweiz, sedan 1969. Wirth har lett utvecklingen av ett flertal programspråk, däribland ALGOL W under åren 1964–67, PASCAL 1968–72, MODULA 1973–76, MODULA-2 1977–80, och OBERON 1986–90.

<sup>142</sup> PL360, programspråk utvecklat av Niklaus Wirth under 1960-talets senare hälft. Det var ett ALGOL-derivat avsett för datorerna IBM 360 och IBM 370.

<sup>143</sup> Utvecklingen av språket ERIPASCAL leddes av Däcker och bestod av en utvidgning av PASCAL med moduler, processer, processkommunikation m.m. hämtade från språket CHILL, som är en förkortning för CCITT High Level Language. ”Datalogilaboratoriet – de första 10 åren”.

<sup>144</sup> ADA, ett programspråk vars första version togs fram av det amerikanska försvarsdepartementet 1975–79. Språket blev ANSI-standard 1983 och är sedan 1984 amerikanska försvarets vedertagna programspråk. ADA är ett starkt strukturerat språk av högnivåtyp och används bl.a. inom realtidsprogrammering.

<sup>145</sup> Föreningen Ada i Sverige, intresseförening bildad 1983 för att främja programspråket ADA.

<sup>146</sup> Åke Bengtsson, se fotnot 90.

**Lars Arosenius:** Det var inte vägplanering?

**Åke Bengtsson:** Nej, och det gjorde vi tillsammans med en schweizisk firma.<sup>147</sup> Det skrevs i FORTRAN, men det första vi gjorde var att vi försökte undersöka de olika FORTRAN-kompilatorerna som fanns på marknaden, och så sa vi det att vi ska bara använda såna delar som fanns gemensamt, vi skulle inte använda några finesser, utan vi skulle försöka och hålla oss till en ”bas”-FORTRAN som fungerade för alla. Och det var mycket framgångsrikt. Under den här perioden, 1960–65, började vi ju med att programmera det här. Det tog kanske fem–sex år innan vi var riktigt färdiga med hela systemet och under den tiden så kom det nya versioner av kompilatorer och så vidare, och det var fördel att vi inte hade använt några finesser. Vi kunde flytta programmen från en maskin till en annan utan större svårigheter. Och alldeles tydligt blev det när vi började installera på PC. Det visade sig att det gick ganska enkelt att föra över de här programmen som var skrivna för en stordator till PC. Så det var nog ett ganska vist beslut vi tog den gången. Det är faktiskt så att de programmen som vi skrev mellan –65 och –70 fortfarande används. Det finns idag ungefär 50 installationer av de här programmen i Sverige. Så vi har lyckats klara av att föra de här programmen från en generation av datorer till flera andra.

**Lars Arosenius:** Björn, vad säger du? Vilka var erfarenheterna på QZ och Datema?<sup>148</sup>

**Björn Kleist:** Ja, vi gjorde inte så mycket modifikationer på senare år. I början på 60-talet var vi till och med in i monitorerna och ändrade. Det betydde att det kom en ny version av monitorsystemet, och då fick vi göra om allt arbete vi hade gjort. Men på språksidan, på applikationssidan, tror jag att det var få egentligen som utnyttjade odokumenterade finesser. Ja, man lärde sig rätt snabbt att om man använde de här funktionerna då fick man jobb när det kom en ny version. Och det kom ju nya versioner av kompilatorerna rätt så ofta.

Och man kan väl säga att gränslinjen går väl någonstans vid –65, eller nåt sånt, då började man lära sig det här med att standard är viktigt. Man försökte hålla sig till de medier som fanns. Då blev också maskinerna lite mer kraftfulla. Och nya maskiner som kommit sen. Men då var ju inte minnet längre lika hårt begränsande.

**Lars Arosenius:** Vad tycker Ingemar?

**Ingemar Dahlstrand:** Ja, jag har inte råkat ut för så stora bekymmer med såna här omläggningar fast de till och med har varit från ett språk till ett annat. När till exempel SMIL<sup>149</sup> i Lund lades ned, så gick omläggningen till UNIVAC 1108 utan några märkbara bekymmer. Jag tror det beror på att väldigt många program i universitetsmiljöer är kortlivade till sin natur. Det är inte stora reguljära körningar utan ett forskningsprojekt, och när forskaren har doktorerat, så är det inte intressant längre, så det beror mycket på miljön också. Sen när man ska skriva stora program som ska köras på många ställen, får man ju sätta sig in i det ordentligt, som ni gjorde.

---

<sup>147</sup> Det schweiziska bolaget som Nordisk ADB AB samarbetade med hette Digital AG. E-post Åke Bengtsson, 21/11 2006.

<sup>148</sup> Datema AB, svenskt IT-företag.

<sup>149</sup> SMIL, förkortning för siffermaskinen i Lund, BESK-kopia vid Lunds universitet som stod färdig 1956. Konstruktionsarbetet leddes av Gunnar Wahlström vid MMN.

**Elsa-Karin Boestad-Nilsson:** Jag undrar varför Ericsson hade så mycket egen programvara, ni hade egen PASCAL. Hur kom det sig att ni väldigt sent gick över till att använda samma programvara som alla andra?

**Lars Arosenius:** Lars-Olof Norén, kan du svara på det?

**Lars-Olof Norén:** Ja, det var så här att vi bland annat ju hade problem med både processorkapacitet och minnesstorlek. Då tittade vi på vilken typ av operationer som behövdes för att på ett effektivt sätt styra telefonprocesser. Det kanske inte var så att flyttalsräkning var en mer logisk operation och så vidare. Och det ledde så småningom till att man fick en egen datorarkitektur, men också egen programvaruteknik, och vi såg tidigt, i de jämförelser som vi gjorde, att vi faktiskt vann ett antal faktorer i kostnadseffektivitet. Så det var kanske grundförutsättningen. Programspråken som är specifika för Ericsson, det är PLEX, som Bjarne talade om, det är ERLANG,<sup>150</sup> som Bjarne varit intresserad av, det är ERIPASCAL, som enkelt uttryckt kan sägas vara en kombination av PLEX och PASCAL, och vi använde det i produkter, bland annat dataväxeln ERIPAX.

Men den egna programvarutekniken ledde också fram till att vi behövde egenutvecklade stödsystem för programutveckling. I början gjordes dessa för ORION som Ericsson också var återförsäljare av under 60-talet. Så småningom så räckte ju inte kapaciteten på den ORION vi hade, utan då letade vi reda på en ORION-anläggning i England på ett försäkringsbolag. Vi hade ett gäng som hade ansvar för den utväxlingen. Vi hade ett gäng som åkte ut dit varje vecka och härjade, och gjorde körningar, kom hem tillbaka. Så småningom så skulle det här gänget i England byta till IBM och lägga ner det där, och vi var väl också på den vägen. Men i alla fall behövde jag datakraft, och då hittade vi en ny ORION hos en guldgruva i Sydafrika, så vi hade en kille som åkte till Sydafrika varje vecka, vill jag minnas. Och kontentan: det visade sig att guldgruvan också skulle gå över till IBM vilket innebar att vi måste konvertera våra stödsystem. På guldgruvan var det en smart programmerare som hade gjort en automatkompilator från ORION till IBM, och då frågade vi: ”Får vi använda den?” ”Javisst”, svarade han, ”det får du göra”, och det gjorde jag också. Under mellantiden så hade vi använt IBM:s servicebyrå i New York och lagt dit många tiotals miljoner på att konvertera. De lyckades inte riktigt med det där, utan det blev ständigt förlängning, så vi sa till dem, ”nu tar vi hem alltihopa, så gör vi det själv”, och det gick ganska smidigt. Så jag tror att alla de där resorna och allt det där betalade sig genom att vi kom i kontakt med den där killen i Sydafrika, faktiskt.

**Lars Arosenius:** Kurt Katzeff.

**Kurt Katzeff:** Ja, jag skulle kanske ge lite kommentarer till vad som har sagts här. Det var ju faktiskt på det sättet att LM Ericsson ägde ett dotterbolag som levererade datorer, Driftkontrollbolaget,<sup>151</sup> och de levererade Ferranti-datorer, därav var vi ganska bundna av det. Jag begärde, som teknisk chef för telefonstationsdivision, att få sätta in användarna i IBM-datorer och vi tjafsade om det här ett tag, och så tog två vice verkställande direktörer<sup>152</sup> och reste runt till olika användare i Sverige, och kom tillbaka och sa att de flesta användarna var missnöjda med IBM. Och då var det inte så mycket jag kunde göra. Vi

---

<sup>150</sup> ERLANG, förkortning för Ericsson Language, funktionellt programspråk utvecklat av Datalogilaboratoriet vid Ericsson under ledning av Däcker. De första versionerna kom 1987/88. ERLANG släpptes som ”open source” 1998.

<sup>151</sup> Driftkontrollbolaget var ett dotterbolag till LM Ericsson som sysslade med fjärrstyrning och automatisering av trafiksignaler och järnvägssignaler.

<sup>152</sup> Dessa var 1:e vice VD Malte Patricks och vice VD Christian Jacobæus.

gjorde oss av med Driftkontrollbolaget en tid och gick över till ICT.<sup>153</sup> Men så fort Hans Werthén<sup>154</sup> kom till som produktionsdirektör, så lyckades jag få igenom att vi fick använda IBM-datorer, för de var faktiskt bäst vid det tillfället. Det var det här att LM Ericsson ägde ett dotterbolag som trasslade till alltihop. Så man får komma ihåg att problemen är ibland, så att säga, oförståeliga för utomstående.

**Göran Waernér:** Om jag får komplettera, så hade Nordisk ADB också en viss konsultverksamhet, och vi fick ett programmeringsuppdrag från LM Ericsson tillsammans med Autocode. Och då skulle vi köra på den här ICT-maskinen. Ja, och jag fick alltid kompilera två gånger för att se efter om vi fick samma resultat, och om det inte var det, så fick vi köra en tredje gång. Och när vi väl hållit på så ett tag, så sa jag att det här kan vi inte ta ansvar för. Så jag och Lennart von Sydow i Autocode skrev ett gemensamt brev till Björn Lundwall,<sup>155</sup> och vi påpekade att våra erfarenheter från UNIVAC var mycket bättre. Och då ändrade Ericsson strategin efter ett tag och gick över från ICT till UNIVAC.

**Lars Arosenius:** Ingemar.

**Ingemar Dahlstrand:** Ja, jag måste passa på här och berätta att 1982 års nobelpristagare i fysik, Wilson,<sup>156</sup> skrev en ganska intressant artikel där han sa att bristen på kompabilitet hotar vetenskapens grundvalar därför att väldigt många vetenskapliga resultat inom fysiken bygger på stora beräkningar, och om man inte kan reproducera dem någon annanstans, så bryter man ju mot detta viktiga villkor för vetenskap, att resultaten ska vara reproducerbara av någon annan än författaren.<sup>157</sup> Det har uppmärksammats på, så att säga, högsta nivå att det inte är riktigt bra det här.

**Lars Arosenius:** Men fördelen med det här är väl att du faktiskt kan köra det på olika ställen och pröva? Är inte det så att det här ju ökar möjligheterna till utbyte?

**Ingemar Dahlstrand:** Jo, just det, men om man får olika resultat, så har man ju en besvärlig situation.

**Lars Arosenius:** Bjarne.

**Bjarne Däcker:** Ja, när det gäller ERIPASCAL så borde kanske jag nämna att inom telekomvärlden så utvecklades det ett språk som hette CHILL.<sup>158</sup> Det var så att säga motsvarigheten till ADA, men ADA var ju på den militära sidan och CHILL var ju på telekomsidan. I själva verket så hade de faktiskt ganska mycket likheter. Bägge hade moduler, parallella processer, processkommunikation och så där. Medan CHILL hade

---

<sup>153</sup> ICT, förkortning för International Computers and Tabulators, företag som bildades 1959 när British Tabulating Machine Company (BTM) fusionerades med Powers-Samas Accounting Machines, Ltd. ICT köpte 1963 Ferrantis datorverksamhet. Det gick samman med English Electric 1968 och formade International Computers Ltd. (ICL). Cortada (1987c), 83f.

<sup>154</sup> Hans Werthén, 1919–2000, svensk industriman. Han hade ett flertal chefsbefattningar inom Ericsson-koncernen. Werthén var VD för Electrolux-koncernen 1967–74 och styrelseordförande där 1974–91.

<sup>155</sup> Björn Lundwall, 1920–1980, civ.ing. E, KTH 1930. Lundwall anställdes vid LM Ericsson 1943, blev chef för dess transmissionsavdelning 1960, vice VD 1963 och VD 1964, en post han innehade fram till 1977. Därefter var han styrelseordförande i koncernen.

<sup>156</sup> Kenneth G. Wilson, f. 1936, amerikansk fysiker, professor i teoretisk fysik vid Cornell University, Ithaca, New York, 1971–87, vid Ohio State University, Columbus, sedan 1987. Wilson erhöill nobelpriset i fysik 1982.

<sup>157</sup> Kenneth G. Wilson, "A Program for Computing Support for Scientific Research", Newman Laboratory of Nuclear Studies, Cornell University, Ithaca, N.Y., 6ff.

<sup>158</sup> Se fotnot 143.

asynkron processkommunikation, så att säga, med signaler, ”messages”, precis som i CCITT-SDL,<sup>159</sup> så hade ADA synkron processkommunikation vilket ansågs säkrare, men det har visat sig göra att ADA-program i själva verket blir väldigt stela. Sen upptäckte vi att CHILL var ungefär som era erfarenheter av FORTRAN, det var för stort, det fanns för mycket. Philips hade fått in sina ”events”, och ITT hade fått in sina ”buffertar”, och Ericsson hade fått in sina ”messages”, och skulle man använda alla de här begreppen samtidigt skulle man få ett helt hopplöst system. Det betydde att alla valde ut sina ”subsets”, så vi skar ut ett ”subset” av ERICHILL, men för att snabbt komma fram, så lyckades vi komma över en PASCAL-kompilator som byggde på San Diego-PASCAL,<sup>160</sup> och utvidgade den så att den i alla fall skulle vara semantiskt ekvivalent. Tanken var att köra ERIPASCAL, men i princip kunna skriva något som såg ut som CHILL. Det var liksom ett skäl. Så vi hittade inte på ERIPASCAL bara för att vara besvärliga, utan för att det hjälpte oss att lösa problem och för att det hängde ihop med vårt arbete med CHILL.

**Lars Arosenius:** Vi har talat om FORTRAN, PASCAL, COBOL var ett annat sånt där språk, ALGOL och AMALGAM. Men vart har alla språk tagit vägen? IBM hade ju APL.<sup>161</sup> PL/I<sup>162</sup> var ju ett väldigt elegant språk. APL var ju oerhört elegant, men som aldrig riktigt slog igenom. Men det var oerhört lätt att använda, det fanns ju på en liten IBM-maskin som hette 1130.<sup>163</sup> Och det...

**Elsa-Karin Boestad-Nilsson:** Men kunde det användas till allt?

**Lars Arosenius:** Ja, framför allt matriser. APL var ju ett väldigt fint språk för alla som sysslade med matrisberäkningar. Björn vet allt om det här.

**Björn Kleist:** Ja, jag är väl kanske en av de få som kört APL i större miljöer. Jag var väl en av de som införde APL på S-E-Banken och använde väl APL i mer än tio år. Vi hade vårt första mailsystem baserat på APL. Ja, det var väldigt populärt. Man kunde göra användargränssnitt som man testade på användarna utan att ha ett trädsystem bakom, alltså man gjorde ett användargränssnitt som användarna kunde ta ställning till. Sen byggde man stora system baserade på COBOL och assembler. Om det här systemet visade sig vara det man ville ha, så att säga.

**Lars Arosenius:** Bjarne.

**Bjarne Däcker:** Ja, jag tror det väl har utvecklats kanske sju- till tiotusen olika program-språk. Det har gått så pass långt att det finns en speciell disciplin som heter History of Programming Languages. Man kan alltså surfa in på deras hemsidor, så får man ut kopi-öst stora släkträd över alla de här programspråken.<sup>164</sup> Men det är en utveckling som

---

<sup>159</sup> CCITT-SDL, förkortning för CCITT Specification and Description Language.

<sup>160</sup> Eg. UCSD PASCAL, en variant av PASCAL utvecklad vid University of California San Diego under 1970-talet.

<sup>161</sup> APL, förkortning för A Programming Language, ett programspråk för matematiska applikationer utvecklat under 1960-talet. Det kunde bl.a. hantera vektorer och matrisoperationer. ”APL Session”, i *History of Programming Languages*, ed., Richard E. Wexelblat (New York, 1981), 661–691.

<sup>162</sup> PL/I, uttals Programming Language One, programspråk som utvecklades i början av 1960-talet. Det baserades på delar av FORTRAN, COBOL och ALGOL. PL/I stöddes av IBM, men nådde ändå aldrig någon större popularitet. Ceruzzi (1998), 104ff.

<sup>163</sup> IBM 1130, en mindre IBM-dator för kommersiellt och vetenskapligt bruk som presenterades 1965.

<sup>164</sup> Se t.ex. hemsidan: [http://www.oreilly.com/pub/a/oreilly/news/languageposter\\_0504.html](http://www.oreilly.com/pub/a/oreilly/news/languageposter_0504.html). Information hämtad 29/11 2006.

verkar ha stannat av. Med LISP<sup>165</sup> och PROLOG<sup>166</sup> var man ju faktiskt ganska långt framme här i Sverige. Jag hade ju trott en gång i tiden, när jag jobbade med det här, att inom programspråk skulle man ha en utveckling ungefär som av maskinvara: maskinspråk, assembler, FORTRAN, LISP-språk, AI-språk, och så skulle det gå vidare. Men det verkar liksom ha stannat av. Vi hade ju verksamheten som Erik Sandewall<sup>167</sup> bedrev, datalogilaboratoriet i Uppsala. Sen när han fick en professur i Linköping och flyttade dit, då flyttade ju i princip hela institutionen med. Familjerna och fruarna hittade nya jobb. Det var ganska intressant. Sen i Uppsala så hade vi Sten-Åke Tärnlund<sup>168</sup> och PROLOG.

Jag gick en gång en sån där sommarskola i Uppsala under två veckor och lärde mig LISP, och då hade jag arbetat ett par år med Kurts Katzeffs system för reläsatser för att uppfinna en databas i FORTRAN för att grafiskt representera reläsatsscheman. Men så upptäckte jag att det var ju som att stjäla godis från småbarn med ett språk som LISP bakom sig. Jag hade suttit för hand och utvecklat dynamisk minneshantering i FORTRAN bara för att kunna hantera det här problemet, när jag insåg att det fanns färdiga teknologier för sådant. Symbolisk programmering börjar komma tillbaka, men då heter det såna här groteska saker som XML<sup>169</sup> och sånt. Men att LISP försvann, fastän det var så pass kraftfullt, det är en tråkig utveckling.

**Lars Arosenius:** Bengt.

**Bengt Gällmo:** Ja, om jag får gå tillbaks lite grann så saknades nämligen i stort sett utbildning på 50- och 60-talen. Alltså, min lilla kurs som jag nämnde förut, ”Siffermaskinernas användning”, läsåret –58, –59. Det var ingen stor kurs. Det var liten bagatellartad kurs. Sen kom det så småningom en ettårig kurs i ADB, och Elsa-Karin berättade tidigare idag att det tydligen fanns en parallell ettårig kurs i teknisk-vetenskaplig databehandling som jag hade glömt. Men hur som helst, den ettåriga kursen i ADB var ju inte särskilt mycket att ha. Ännu 1964, –65, så anställde vi ganska mycket folk på IBM som programmerare utan att hitta någon som hade relevant utbildning. Utan vi frågade om de spelade schack, om de löste korsord, om de hade läst teoretisk filosofi, och så hade vi det här ”programming aptitude”-testet som Bjarne hade råkat ut för två gånger. Det tog en timme vill jag minnas, och det var inte något särskilt omfattande test heller. Sen anställde vi folk baserat på det här och hoppades att schackspelare och liknande skulle fungera som programmerare. Ännu 1968, eller –69, så vet jag att jag anställde någon, i varje fall, som inte hade programmerarutbildning. Det här satte också sin prägel på utvecklingen ganska mycket. Vi lärde oss i småkurser. Tillverkare som IBM och andra höll kurser om sina maskiner och sina språk. Vi läste manualer och lärde oss själva. Vi gjorde övningsexempel själva och lärde oss dem. Men det fanns ingen sammanhållen utbildning. Det påverkade utvecklingen.

---

<sup>165</sup> LISP, förkortning av list processing language, ett funktionellt programspråk som började utvecklas under senare delen av 1950-talet vid MIT av matematikern John McCarthy. Det har bl.a. använts inom fältet artificiell intelligens (AI). ”LISP Session”, i *History of Programming Languages*, ed., Richard E. Wexelblat (New York, 1981), 173–197.

<sup>166</sup> PROLOG, förkortning för Programmation en Logique, ett programspråk som bygger på logiska uttryck och som utvecklades i början av 1970-talet i Marseilles, Frankrike. ”Prolog Session”, i *History of Programming Languages: II*, eds., Thomas J. Bergin, Jr. & Richard Gibson, Jr. (New York, 1996), 331–367.

<sup>167</sup> Erik Sandewall, f. 1945, fil.lic. 1966, fil.dr 1969, docent 1970, professor i informationsbehandling särskilt datalogi sedan 1975. Sandewall är ledamot i IVA sedan 1981.

<sup>168</sup> Sten-Åke Tärnlund, f. 1941, professor i datalogi vid Uppsala universitet.

<sup>169</sup> XML, förkortning för Extensible Markup Language, standard för strukturmärkning av textbaserade elektroniska dokument. Språket används t.ex. för elektronisk dokumenthantering på World Wide Web.

**Lars Arosenius:** Absolut. Jag kommer ihåg att vi gjorde en undersökning på IBM på 70-talet, och då hittade vi 22 olika akademiska discipliner bland programmerarna. 22 olika discipliner. Det var precis som Bengt säger att det fanns ingen utbildning, så man plockade in de som man trodde hade förutsättningar att klara jobbet. Och det här är ju ett dilemma. Alltså, programmeringen har gått från att vara en intellektuell utmaning för dem som gillar schack och så vidare, till att bli en användning av verktyg. För det kanske är en hel del människor som är lite olyckliga över att de inte får tillämpa sina kunskaper idag. Ingemar.

**Ingemar Dahlstrand:** I Göteborg började utbildningen på Chalmers ungefär -62, i och med att ALGOL blev tillgängligt, och i Lund bör det också ha börjat ungefär -62. Sen tog det ju några år innan de kom ut på arbetsmarknaden. Så före -65 kan de ju inte ha varit ute. Men då var det ju så att universiteten inte genomgående hade tillgång till dataresurser. Vi fick ju på Facit Electronics donera maskintid för de här kurserna. I Lund var det lite bättre, för där ägdes datorn av institutionen som hade den här utbildningen, så att det var ju inga såna problem.

**Lars Arosenius:** Lars-Olof Norén.

**Lars-Olof Norén:** Jo, det här med utbildningen. Jag vill ta upp en annan aspekt av utbildningen också. Det är frågan hur man arrangerar, så att säga, stora programvaruprojekt överhuvudtaget. I vårt fall, i telekommunikationsfallet, rörde det sig under 60-talet om projekt på flera år som ingenjörer var inblandade i. Hur arrangerar man det? Hur organiserar man programvaran? Hur kan man träna det egentligen? Det fanns ju ingen utbildning, det fanns ingenstans man kunde lära sig. Vad vi gjorde var att vi besökte kollegor, Bell Labs. Jag var på Bell Labs minst en gång om året under den här perioden. Vi snackade och utbytte erfarenheter. Det var väl det sättet som man lärde sig på helt enkelt. Men sen också vilka programvarutekniker man ska använda, det finns ju också lednings- och organisationsaspekter på hur man arrangerar det, den här tekniken är lättare att arrangera, den andra är svårare att arrangera. Det fanns ingen utbildning i det överhuvudtaget, och jag tror inte det finns nu heller.

**Lars Arosenius:** Gunnar Stenudd sen.

**Gunnar Stenudd:** Ja, jag heter Gunnar Stenudd. Jag pratade med min gamle vän Gunnar Hellström och Olle Karlqvist<sup>170</sup> om det här ämnet för en 40 år sen, minns jag. I Facit Electronics var det väl då, och jag frågade: "Vilket underlag kräver ni för att få en duktig programmerare?" Jag minns inte om det var Gunnar Hellström eller Olle som sa: "Helst ska de vara helt utbildade. Konstnärer med egen fantasi är till exempel mycket mera formbara."

**Lars Arosenius:** Ja. Bengt.

**Bengt Gällmo:** Ja, det slår mig att det finns en gräns i tiden här någonstans, därför att när Björn och jag satt på IBM:s Applied Science, ungefär 1960, då var vi som kunde programmera kungar. Vi gick ut till ASEA, och talade om för dem hur man skriver reaktortorprogram. Det var nämligen vi som kunde skriva program. Det var viktigare än att

---

<sup>170</sup> Olle Karlqvist, 1922–1976, civ.ing. E, KTH 1949, tekn.lic. 1953. Han deltog i utvecklingen av datorerna BARK och BESK vid MMN 1949–56. Karlqvist gick 1956 över till Elektronikavdelningen vid AB Åtvidabergs Industrier, där han var chef för datacentralverksamheten, och 1964 till nybildade Industridata AB som chef för teknisk databehandling. Han hade egen konsultverksamhet fr.o.m. 1971.

förstå sig på reaktorerna. Vi var kvasiexperter på allt mellan himmel och jord bara för att vi kunde skriva program. Någonstans under den här tioårsperioden, från -60 till -70 så gick det en gräns. Därefter började teknikerna, vetenskapsmännen, och vad det nu var för något, ta över, då hade de kommit ifatt, och såg att de kunde ta över. Då tog de över ledningen så att säga, och då var det inte fullt så glamoröst att vara programmerare.

**Lars Arosenius:** Omkring slutet på 60-talet och början på 70-talet, säger du?

**Bengt Gällmo:** Ja, jag skulle tro det.

**Lars Arosenius:** Elsa-Karin.

**Elsa-Karin Boestad-Nilsson:** På FOA så hade vi räknebiträden, och många av dem hade ett betyg i matte, sen hade de misslyckats med studierna på resten. Sen 1954–55 så började vi utvidga oss till en större servicebyrå, och jag hade väldigt stort inflytande på vilka som anställdes, för det hade man på den tiden om man var chef. Och jag anställde ju nästan ingen som inte hade åtminstone läst ett betyg på matematik eller hade väldigt bra studentbetyg. Det här var alltså på 50-talet. Och sen omkring 1960 så anställde jag väl fyra, fem stycken akademiker med matte i sin examen. Och sen 1960 började undervisning i numerisk analys, och jag kunde anställa dem som hade läst det. Sen antogs nästan bara manliga studenter till den utbildningen, så i slutet av 70-talet så blev vår kvinnodominerade grupp plötsligt blandad, för då anställde vi män, för det var bara såna som sökte. Och sen på 70-talet så hade alltså utvecklingen gått så långt så att i alla möjliga ämnen så läste man databehandling. Man kom ju till FOA och hade helt andra professioner, men programmera kunde man, och man kunde ta hand om färdiga program, så då ändrade gruppen som jag hade karaktär. Vi gjorde sen mera stora verktyg till de här som programmerade själva, och som framför allt tog hand om sina programsystem själva, så efter 1970 så fanns det helt annan sorts folk att tillgå, och helt andra kunskaper i databehandling, så det liksom blev en helt annan inriktning på det hela. Men alla på FOA kunde köra sina egna program i stort sett.

**Lars Arosenius:** Ja, Bo Lindestam.

**Bo Lindestam:** Jag ska ta och bryta ämnet lite grann, för jag skulle vilja ta upp en annan aspekt som jag tycker är väsentligt när vi diskuterar det här från 50- och 60-talen. Majoriteten, jag exkluderar nu Bjarne, har diskuterat batchprogrammering. Idag så finns det knappast någonting som kallas batchprogrammering utan allting är 24-timmars drift, mer eller mindre. Vi slår på datorn, och sen går PC:n, och stordatorerna går 24 timmar och så vidare. Det här är ur programmeringssynpunkt en väldigt viktig aspekt. Batchprogrammering är ett sätt att jobba, realtidsprogrammering är ett annat sätt. Och idag så är det väl oftast realtidsprogrammering som man lägger ihop i form av rutiner som man länkar till varandra i en kontinuerlig loop. När, enligt er mening, har man börjat att lämna batchprogrammeringen och gå över till 24-timmars drift? Jag skulle vilja ta upp en fråga till. Ställ samma fråga vad gäller operativsystem. Vilka operativsystem fanns under 1950-talet i era maskiner? Vilka operativsystem fanns under 1960-talet som inte var IBM-system? Och hur har det påverkat er uppläggning av programmering? När tog operativsystemet hand om vårt sätt att programmera? För IBM:s doktrin var ju att det fanns inga begränsande resurser. Operativsystemet löser det här åt er. Medan vi som jobbade i realtid hade en helt annan grunduppfattning om vilka resurser vi hade. Vi hade vår dator som gick där, och det var inget operativsystem som vi hade råd med på den tiden. Vi pratar ju om, säg, mitten av 60-talet fram till mitten av 70-talet. Tack för ordet.

**Lars Arosenius:** Ja, Hans Riesel.

**Hans Riesel:** Får jag svara på det första, den där man övergav batchprogrammering. Batchprogrammering är ju så pass tråligt att hålla på med, så det övergav man ju så fort man hade chansen, och det var väl när det började komma datorterminaler. Och när var det? Var det i början av 70-talet?

**Lars Arosenius:** 70-talet. Operativsystemen fick väl sitt genombrott när man började titta på IO-processer, till exempel när man började lägga ut intelligens utanför centralenheterna, så var man tvungen att samordna de perifera processerna. Det var väl med 709 alltså? Var det inte så, Björn?

**Björn Kleist:** Jo, det stämmer.

**Lars Arosenius:** Bengt.

**Bengt Gällmo:** Får jag kommentera Bosses inlägg också? Alltså, vi har väl två utvecklingar här, mer eller mindre parallella. För att för det första för batchbetonade jobb där man gör en sak och sen gör man en annan sak, så att säga. Under 50-talet hade man inte några operativsystem och stoppade in varje gång. Början på 60-talet så var det fortfarande samma sak, fast vi började få ett operativsystem som tog i sig jobben ett i taget, och kunde köra dem efter varandra utan operatörsingripande. Så kom någonting som hette SPOOL, Simultaneous Peripheral Operations On-Line, det var en slags rudimentär multiprogrammering. Man kunde åtminstone köra "dataprinters", "datapunchers" och såna där saker parallellt med andra program. Sen så kom terminaler, så man behövde inte lämna kortbuntar och sånt, utan man kunde sitta vid sin terminal istället. Fram till dess hade datorerna inte en sådan kapacitet att det var meningsfullt att låta flera användare komma in samtidigt, utom just i det där specialfallet SPOOL. Men ungefär samtidigt som vi fick terminaler, så hade vi också sån kapacitet på datorerna att man kunde köra multiprogrammering och låta flera användare dela maskinen samtidigt.

Parallellt med allt detta har vi realtidsprogrammering som följde ett annat spår. Och det började väl också på 60-talet någon gång med Ericssons växlar, med Standard Radios flyg- och trafikledningssystem, processtyrning på IBM och på många andra ställen. Och där har vi alltså parallella processer och "interrupt"-hantering, och såna där saker. Men den utvecklingen har liksom inte så mycket med batchmultiprogrammering att göra. Det är en annan grej.

**Lars Arosenius:** Ja, Bjarne.

**Bjarne Däcker:** Alltså, vad vi behöver tänka på är ett system, typ bankomater. Vad som brukade vara ekonomisystem som banken körde efter tre, eller vid månadsslut, ligger uppe 24 timmar om dygnet, och det är transaktioner ungefär som telefonsamtal, med likartade aspekter. Och då har ju plötsligt de här systemen glidit samman, det är inte klart vad som är ett batchekonomisystem och vad som är online-transaktionssystem.

**Lars Arosenius:** Göran.

**Göran Waernér:** Jag är nästan övertygad om att vi körde "timesharing" på 60-talet och då gjorde man ju programmering direkt online. Jag har också för mig att vi redan under slutet av 60-talet hade "teletype"-terminaler som vi körde timesharing på UNIVAC 1108.

Vi hade också batchterminaler, så att vi kunde ladda in och ta ut resultaten hemma på kontoret. Och 1972 var jag med och utvecklade ett bokningssystem för Viking Lines terminaler som gick i realtid parallellt med timesharing och batch, körde alla tre parallellt.

**Lars Arosenius:** Både terminaler och timesharing kom väl i mitten på 60-talet. Jag kommer ihåg att jag var i San José. Vi hade en uppkoppling till ett General Electric-system med terminaler, så att man kunde köra BASIC<sup>171</sup> på de där terminalerna, och jag vet inte var datorn stod någonstans, och det fungerade alldeles utmärkt då.

Ja, tiden börjar gå. Men hur ser ni på programmeringen och programmeringsutvecklingen, har den stannat? Gav 50- och 60-talen någonting? Björn?

**Björn Kleist:** Ja, jag tycker att huvudpunkten är produktivitet. Från början stod man vid datorn och matade in sin program. Sen kom de första operativsystemen, FORTRAN Monitor System är ett exempel. Då ökade produktiviteten, folk behövde inte stå och vänta så länge. Sen på senare år så har produktiviteten ökat hela tiden. Man kan bygga system fortare, men kraven har samtidigt ökat.

**Lars Arosenius:** Bengt.

**Bengt Gällmo:** Det här är mer en observation än någonting man kan göra någonting åt. Jag tycker, som någon har sagt, att det har stagnerat lite grann. På 70-talet trodde jag att ADA var den sista dinosaurien, och efter det skulle LISP och PROLOG ta över som de första däggdjuren. Det blev inte så. Vi har fortfarande, som någon också har sagt, ingen arvtagare till ALGOL egentligen. C inbjuder till trickprogrammering och det tycker jag är bedrövtligt, så många, decennier efteråt, så att säga. Som sagt, jag vet inte vad man kan göra åt det. Men vi arbetar fortfarande med stora, ganska klumpiga algoritmiska språk, istället för att använda de här som jag trodde var de första däggdjuren på 70-talet.

**Lars Arosenius:** Tack ska ni ha allihopa. Tack ska ni ha alla som har kommit hit. Tack särskilt till panelen för att ni kom. En applåd för panelen.

---

<sup>171</sup> BASIC, förkortning för Beginner's All Purpose Symbolic Instruction Code, ett programspråk som ursprungligen utvecklades vid Dartmouth College under det tidiga 1960-talet. Under 1970-talet utvecklade William Gates och Paul Allen en version av språket för persondatorn Altair, och det ledde till bildandet av Microsoft Corporation. Ceruzzi (1998), 203–205, 232–236.