

Universitetens databehandling

Tomas Ohlin

Kritiken mot IBM från användarna inom Stockholm i samband med A-maskinaffären har ingalunda varit en enstaka företeelse. Under utredningsarbetet i samband med affären inlöpte till Universitetet mycken information om likartade problem vid bl a utländska universitet och högskolor. Detta bestyrkte uppfattningen om det befogade i att påvisa vissa av de brister som det aktuella maskinsystemet, IBM 360/75, var och är behängt med för aktuell användningsmiljö.

Naturligt nog är det främst programvaror som står i centrum för dylik debatt. Dessa är ju konstruerade för att medge möjlighet till modifikation i en väsentligt högre grad än maskinvaror.

Utomlands har alternativa önskemål rörande existerande operativsystem av standardtyp ofta tvingats leda till att speciella, egna system tagits fram. I Sverige har vi tyvärr ännu några år otillräckliga möjligheter att klara dylik specialkonstruktion av större operativsystem inom universitetet. Utan tvekan vore något sådant mycket önskvärt ur utbildnings- och forskningssynpunkt, men personal, anslag etc vill ännu inte räckta till.

Vilka är då skälen till att universitetet ofta kräver speciella programvaror, och inte låter sig nöja med leverantörernas standardiserade system?

För det första kan vi utgå från att intresse ur utbildningssynpunkt och vilja till satsning tillsammans med viss fackkännedom a priori möjliggör dylik specialsömnad har betydelse. Det är ingen tvekan om att även många användare inom industrin

skulle vilja ha tillgång till effektivare programvaror, men ser sig nödsakade acceptera standardprodukter av resurs-skäl. I universitetsmiljö, där man berättigat kan locka med examensmeriter o d för systemutvecklingsarbete, och där ekonomin därför inte på samma sätt är avgörande, är läget annorlunda.

Universitetsprofilen

För det andra är universitetens användarprogramprofil av ett speciellt utseende. Låt oss betrakta denna ett ögonblick.

Vi kan till att börja med göra en uppdelning på tre olika användarkategorier. För det första har vi den dominerande gruppen användare av typen icke-specialister inom ADB-området. Det gäller för dessa att få stödande hjälp av maskinen vid verksamhet inom ett starkt ökande antal ämnesområden, matematik, fysik, kemi, biologi, beteendevetenskap, ekonomi, lingvistik osv. Dessa användare nöjer sig i sin ADB-utbildning med en kort grundorientering om elementär databehandlingsteknik plus ett programmeringsspråk, ofta av typen Algol. De program som produceras är här korta och i avsaknad av specialistegenskaper, för maskinen således enkla ting. Dessa program tillbringar förhållandevis lång tid i sin rättnings- och inkörningsfas, och behandlas därefter endast någon enstaka gång i produktion.

Den andra kategorin användarprogram kännetecknas av större maskinresursbehov (tid, minne etc) och en mångfald produktionskörningar. Det är här fråga om omfattande bearbetningar av stor betydelse för forsk-

ningens framåtskridande inom olika ämnesområden. Till denna grupp kan vi också räkna tillämpningar som kräver mera speciella resurser, av typen on-line-bearbetningar, sådana som använder plotters eller bildskärmar med ljuspenna etc.

Den tredje kategorin är anknuten till undervisning och forskning inom ADB-facket, i vårt land sorterande under ämnet Informationsbehandling. Här ställs högre krav på maskinens flexibilitet än för de båda övriga kategorierna. Speciella egenskaper hos program och maskinvara prövas, effektivitet studeras, nya tillämpningsfält angrips osv.

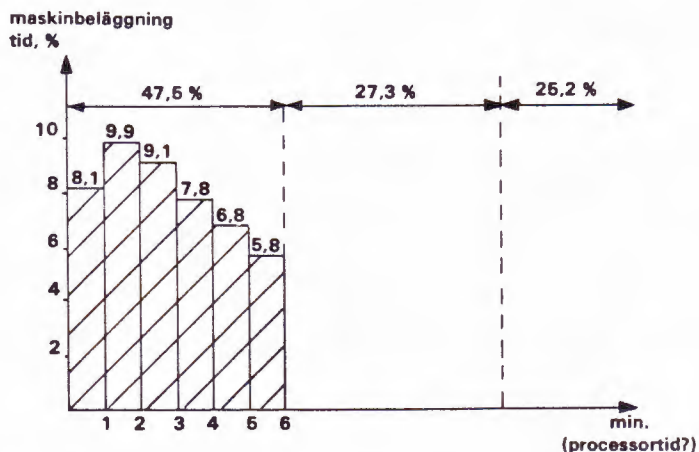
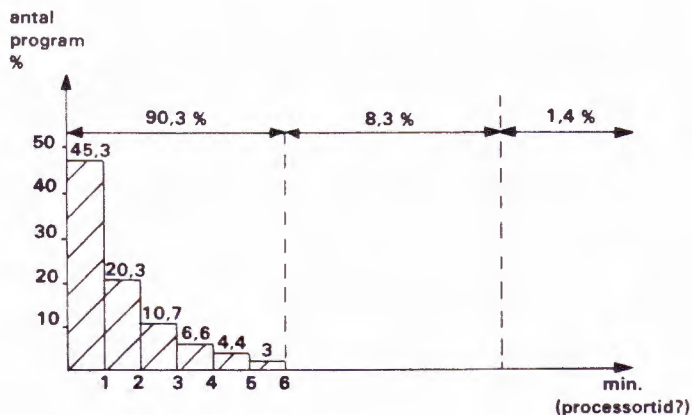
Det är ett faktum att den första kategorin av de tre nämnda dominerar klart mängdmässigt. Ett för profilen sammanfattande kännetecken är alltså

- 1) en dominerande mängd små program med ringa resursbehov
- 2) en mindre mängd mera resurskrävande program
- 3) ett även det mindre antal flexibilitetskrävande, mera avancerade program, oftast av måttlig storlek.

Det bör emellertid noteras att kategorierna 2 och 3 på intet sätt får betraktas som mindre viktiga än kategori 1.

Ett empiriskt belägg för uppdelningen redovisas av Samet, University College London, i augustinumret 1968 av Computer Journal. Åtta månaders statistik från en IBM 360/65 innebär enligt Samet: se fig. sid. 5.

Vi ser här bl a att över 45 % av alla program vardera krävt mindre än 1 min maskintid (processortid?), men



tillsammans belagt maskinen i bara 8 % av totaltiden. Dessutom anges att endast 1,4 % av programmen körts i mer än 20 minuter vardera, dock summerande sig till mer än 25 % av hela totaltiden. Dessutom noteras av Samet märkbart stora in/ut-mängder, dagligen 150 000 lästa kort och 400 000 tryckta rader. Inte minst detta senare faktum, med de stora in/ut-mängderna, synes utgöra en intressant arbetsbas för en expanderande time-sharing-verksamhet.

Våra egna erfarenheter inom landet synes vara på väg mot den av Samet visade programfördelningen.

Maskinsituationen

Det är inte lätt att hitta en existerande maskin som samtidigt är idealisk för att klara alla de tre ovan nämnda programkategorierna. De många små programmen klaras troligen bäst/mest ekonomiskt av en mindre till medel-

stor maskin med ett något statiskt operativsystem. De resurskrävande programmen behöver en mycket stor och snabb maskin, om vi tillfälligt bortser från on-line-bearbetningarna. De flexibla programmen kräver emellertid ett operativsystem som smidigt och effektivt hanterar även sällan använda maskinfaciliteter. Det är också klart att vissa av den tredje kategorins uppgifter önskar tillgång till en eller ett par små maskiner, att användas i huvudsakligen figursytt syfte.

Det har emellertid inom universitetsmiljö, inte bara i vårt land i maskinanskaffningstider, helt enkelt sagts att emedan en enda stor maskin verkligen klarar alla tre programkategorierna är anskaffning av en dylik berättigad. Vad man underlåtit att fördjupa sig i är HUR den klarar uppgifterna. Med vilken ekonomi, med vilken lätthanterlighet för användar-

na? Ett grundförbiseende synes vara underlåtenheten att värdera den mänskliga arbetsinsats som erfordras för att rutinmässigt använda maskinen. Med bilden kompletterad av en adekvat värdering av den arbetstid som kvalificerade forskare behöver för att få ut sina resultat från maskinen kan den mest detaljerade tekniska maskinprestandaanalys förskjutas starkt. Här är det verkligen tal om suboptimering.

Programkategori 2:s snabbhetskrav och kategori 3:s flexibilitetskrav leder inom samma maskin till totalt sett långa spiltider vid hantering av de många små programmen. Är det inte troligt att en mera differentierad maskinpark, även med hänsyn tagen till drifts- och användningsproblemen, skulle kunna innebära en bättre anpassning till användarmiljön? Vi utgår då från att kompatibiliteten bör vara klart bättre än den i många fall är idag.

Vi skulle å andra sidan kunna fråga oss varför inte bättre anpassade universitetsmaskiner existerar på marknaden. Då bör vi emellertid komma ihåg att universitetens förhållandevis sett är fåtaliga. Leverantörerna måste rimligtvis koncentrera sig på att tillhandahålla program- och maskinvaror för de större kategorierna användare. För de flesta leverantörskunderna är maskinprogramseffektivitet viktigare än hög kompilershastighet.

Programvarorna

Ett av skälen till allmänt sett mindre god möjlighet till optimalt utnyttjande av dagens maskiner rör samspelet mellan operativsystem och begreppet maskinfamilj. Det anses idag hos leverantörerna av flera skäl lämpligare att lansera familjer av maskiner än enstaka modeller. Vissa synpunkter talar för att leverantörerna har rätt däri. En nackdel med familjefilosofin är emellertid att programvarorna därmed måste konstrueras för att kunna täcka hela spektrum. Ett programsystem som kan köras på en liten modell med minimikonfiguration kommer, installerat på en större modell med mera fullständig konfiguration, naturligt nog att utnyttja maskinvarorna på ett mindre tillfredsställande sätt.

Vi kommer här in på problemet med tillhandahållande av ett operativsystem som en färdig svart box eller som en mängd små byggklotsar.

Detta är ett komplicerat avvägningsproblem. Jag tror mig personligen vilja rösta för en kompromiss, ett litet antal färdiga boxar i olika storlekar. Av de två alternativen tror jag snarast att bristen på byggklotsspecialister väger tyngre än nackdelarna med den statiska boxfilosofin.

Ett annat skäl till dagens standardiserade operativsystems mindre goda effektivitet härrör från det faktum att de ofta tillverkas av mycket stora programmerargrupper. Parkinsons lag ger här utslag. Svårigheterna med kontinuerlig intern kontakt mellan programmerarna även av nära anknutna programblock leder lätt till en "gyttjig" slutprodukt.

För att återgå till universitetsproblemet kan vi konstatera att önskemål är:

- 1) människovänliga, enkelt hanterbara system
- 2) snabba kompilatorer för de högre språken
- 3) operativsystem med kortast möjliga spiltider.

Låt oss ägna en stund åt att studera var och en av dessa punkter separat.

Människovänligheten

Dagens situation är, mildt uttryckt, knappast tillfredsställande. Ett antal bläckplumpar dyker omedelbart upp:

- a) Styrkortsutformningen
- b) Felsökningsproblemen
- c) Resursåtkomstmöjligheterna

En biprodukt från den hos leverantörerna uppdelade systemprogramframställningen blir lätt att snart sagt varje arbetsgrupp lanserar skäl för något eller några egna styrkort, och genast syr in dessa djupt i sina programsegment. Tillsammans med en hög parameterflexibilitet blir re-

sultatet att systemanvändaren krävs på kännedom om utformning av ett otal dylika kort, även för den enklaste tillämpning. Det är i sanning berättigat att tala om krav på användaren om programmering i två språk, styrspråket såväl som det använda programmeringsspråket. — Vi bör kunna utgå från att leverantörerna i framtiden lägger ned tillräckligt arbete på styrkortsutformningen, så att t ex flera nivåer av styrinstruktioner tas fram. För en användare som så önskar skall **ett** styrkort vara tillfyllest för att klara ett antal standardiserade tillämpningar.

Vad gäller felsökningsproblemen är situationen likartad. Alldeles för mycket specialkunskaper krävs idag för att kunna lokalisera och rätta enkla programfel. Låt oss notera ett exempel på ett önskemål. Många enkla grammatikaliska programmeringsfel görs och kommer att göras. Även ett enda kommatecken, placerat på fel ställe, medför att vi idag efter kompileringen tvingas ta emot fullständiga programlistor, en hög av papper. Användaren är bara intresserad av satsen med kommatecknet. Här kan man verkligen tala om redundans. Låt oss begära att framtidens kompilatorer igenkänner dylika specialsituationer och levererar en utmatning vars omfång står i paritet med vad användaren är intresserad av. Detta blir fundamentalt i time-sharing-miljö. Dagens utmatning av systemmeddelanden är ofta tio, ja hundra gånger så omfattande som den borde vara.

Resursåtkomstmöjligheterna är ett mera mångfacetterat problem. Vi kan vara överens om att det idag är alltför komplicerat att komma till tals

med maskinerna. Leverantörernas time-sharing har, med ett par undantag, visat sig dröja med sin ankomst. Det är ingen tvekan om att time-sharing-service medför revolutionerande förbättringar för användarna, givetvis även inom universiteten. Jag skulle emellertid vilja komplettera bilden genom att dra en lans för parallell tillgång till satsvis bearbetning med korta omloppstider även i framtiden. Ett väl konfigurerat system med satsvisa omloppstider på c:a 10 minuter, alls ingen orimlighet, kommer troligen för ett stort antal användare att vara en god konkurrent till time-sharing. Direkt styrning av exekveringens gång via t ex parametrar är inte väsentligt för alla.

Beträffande människovänligheten hos dagens system krävs ett nytänkande från tillverkarna. Låt oss än en gång ta fram klyschan om att maskinerna skall närma sig människan, och inte tvärtom. Det kan inte sägas en gång för ofta.

Kompilatorerna

Utgående från vår användarprogramprofil ovan är det naturligt att universiteten mera kommer att intressera sig för korta kompileringstider än effektiva maskinprogram. Detta alltså p g a att det övervägande antalet program kompileras många gånger för att sedan produktionsköras endast vid något enstaka tillfälle.

Korta kompileringstider har emellertid hittills inte varit leverantörernas främsta intresse. Detta är i och för sig begripligt, men för universiteten mindre lyckligt. I några fall har därför vid USA-universitet tagits fram egna, snabba kompilatorer. Exempel är WATFOR (University of Waterloo

Produktionsplanering



-fråga
DATEMA AB
- ett av Skandinavien's största dataforetag.
Huvudstagan 1, Box 1056, 171 21 Solna 1
tel. 08, 83 40 20

Fortran Compiler) och PUFFT (Purdue University Fast Fortran Translator). Det är sant att man vid användning av dessa måste göras uppmärksam på vissa språkmässiga restriktioner, men dessa torde knappast i de flesta fall medföra avgörande olägenheter. Att det går att kompilera även Algol förhållandevis snabbt har Burroughs visat, med redan idag existerande standardkompilatorer.

Kompileringshastigheten får emellertid inte, t ex via den eviga primärminnesbristen, drivas upp på bekostnad av diagnostikens fullständighet. Fullständig diagnostik är ett krav idag, både i kompileringsfasen och i exekveringsfasen.

Eventuell framgång för nyare högre språk av typen Algol 68 eller PL/I kommer troligen, bland mycket annat, att vara relaterat till existensen av snabba och vänliga kompilatorer i samband med språkens utläring i större skala.

Operativsystem

Dagens målsättning vid konstruktion av ett större operativsystem synes oftast vara att det skall kunna klara alla sorters arbeten, vad som än be-

gärs. Det kan synas tveksamt om detta alltid är en riktig linje att arbeta efter. En konsekvens blir en i varje arbetsfas nödvändig genomgång av en omfattande checklista, av den typ flygplanskaptenen genomgår före start. Om systemet från början vetat hur liten del av denna lista som i de flesta fall är intressant, hade tidsåtgången väsentligt kunnat minskas.

Vi bör i framtiden kunna hoppas på operativsystem (Hardware?) som före laddning av användareprogrammet igenkänner dess typ mera fullständigt än idag, och därefter utför en starkt begränsad facilitetskontroll under utförandet.

Sammanfattning

Vi har alltså ovan noterat universitetsmiljöns speciella programprofil. Det framgår att leverantörernas standardiserade programvaror idag inte passar denna profil särskilt väl. Av flera skäl kan emellertid leverantörerna rekommenderas att titta närmare på dessa universitetsproblem.

För det första befinner sig den högre utbildningen i en starkt expanderande fas, såväl inom ämnet Informationsbehandling som i stort. Under sin

utbildning kommer studenten idag och framför allt i morgon i nära kontakt med databehandlingsproblemen, och en grundinställning formas hos dem. Det eventuella säkertroende som där grundläggs kan säkerligen vara av betydelse när de senare efter sin utbildning kommer i position att utöva inflytande över program- och maskinval i olika sammanhang.

Än viktigare kan det faktum sägas vara att universitetens programprofil idag troligen är en bild i väsentligt större dimensioner längre fram. Allt eftersom databehandling i större utsträckning görs tillgänglig för folket på gatorna, i hemmen, kommer den okvalificerade användarkadern att expandera i förhållande till den kvalificerade. En utomordentligt väsentlig användarprofil kommer således undan för undan att i sina konturer närma sig vad vi idag kallar universitetsprofil. Det kommer givetvis fortfarande att finnas andra mycket viktiga grupper att ta hänsyn till, men en förskjutning åt universitetsprofilen är trolig. Dimensionerna kommer att vara stora. Det kan komma att visa sig vara av värde att i tid då ha tänkt över de problem som för närvarande diskuteras i universitetskretsar.

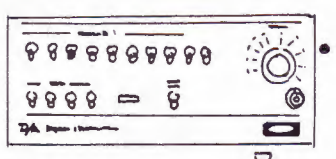


COMPUTER

Raytheon Computer can supply a powerful, inexpensive IC system computer, analog and digital IC modules and package analog instruments.

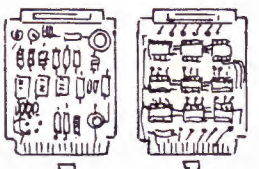
ANALOG o DIGITAL MODULES

There are more than 60 in the line, all compatible with each other, with our analog instruments and with the 703 IC systems computer. You design the logic. We'll produce a wire list and wire wrap, assemble and deliver checked-out hardware.



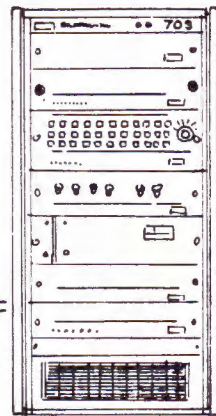
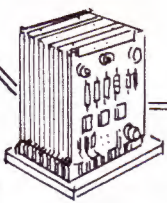
DIGITAL-TO-ANALOG CONVERSION

Several versatile digital-to-analog converters. You select number of channels and resolution. Built-in test and calibration.



ANALOG-TO-DIGITAL CONVERSION

Multiplexing, sample-and-hold and analog-to-digital converters. With our MULTIVERTER and MINIVERTER, all you do is choose number of channels, resolution and speed.



703 IC COMPUTER

The \$15,000 computer with 15,000 spares. The 703's central processor is a single wire-wrapped master module board with plug-in IC's. A unique self-diagnosis locates circuit malfunctions, tells which IC to replace.

- 16-bit word length
- 1.75 μsec cycle
- Up to 32k Memory
- Word and byte commands
- Wide range of peripherals
- Software includes executive, real-time monitor and real-time FOSTRAN IV.

Kontakta oss för ytterligare information och datablad.

08/82 0410 · SCANDIA METRIC AB · FACK SOLNA 3