Innehåll:

Kort ordförandespalt

av Jens Larsson

Jag kanske ska ta och presentera mig. Mitt namn är alltså Jens Larsson och studerar på Y-linjen (andra året sägs det, men det är svårt att förstå när man läser min studiemeritförteckning) och jag är eder nye vice ordförande. Då Lysator inte har någon ordinarie ordförande för tillfället så är det alltså jag som har makten. :-)

När jag inte läser LysKOM så försöker jag tillbringa lite tid med min flickvän. I övrigt rekommenderar jag att ni läser min presentation i LysKOM.

Nu lämnar jag över ordet till vår förnämlige redaktör.

Nyheter inom Lysator

av Leif Stensson

Det har hänt en del sen senaste numret av Garb kom ut. Bland annat så har vi, som synes, fått en ny ordförande, Jens Larsson.

Vi har även fått en del nya maskiner:

- Bodil: en Sun4/280 med 32MB RAM. Donerad av Sun.
- Dell: en Dell 433TE (i486) med 8MB RAM. Utlånad av Dell.
- Hennes: en Sun4/110 med 8MB RAM. Donerad av Sun.
- Kajsa: en Sun3/280 med 24MB RAM. Här körs NannyMUD numera.
- Konrad: en Sun4/65 med 16MB RAM. Donerad av Sun.
- Mauritz: en Sun4/110 med 8MB RAM. Donerad av Sun. Här körs LysKOM-servern.
- Quetzalcoatl: en Pyramid 90x med 8MB RAM. Agerar NEWS-server.
- Varg: Sun3/280 med 16MB RAM.

Alla Sunmaskinerna kör SunOS 4.1, utom Konrad, som kör Solaris. Dell kör Linux, och Quetzalcoatl k—r Pyramid OSx.

GARB är organ för datorföreningen Lysator vid Linköpings Universitet och Tekniska Högskola. Detta nummer har tryckts på LinTek REPRO, och har framställts med hjälp av LaTeX och Emacs. Omslagsbilden är gjort av Jonas Yngvesson vid Institutionen för Systemteknik. Redaktör och ansvarig utgivare för detta nummer är Leif Stensson.

Medlemsskap i Lysator är öppet för alla studerande och anställda vid Linköpings Universitet och Tekniska Högskola. Vid pressläggningen hade Lysator ungefär 350 medlemmar. Medlemsskap för tolv månader kostar 128 kronor som betalas till postgiro 88 56 69–2.

Lysator har postadressen Lysator, c/o Institutionen för Systemteknik, Universitetet, 581 83 Linköping. Elektronisk postadress: `lysator@lysator.liu.se`.

Datasaabs hemvändardag

av Lars Aronsson

På eftermiddagen fredagen den 16 oktober gick vi till Collegium på Mjärdevi för att vara med på Datasaabs hemvändardag. Det var jag, Mikael Cardell, Magnus Redin och vår ordförande Calle Dybedahl. Bara Calle var bjuden av arrangörerna som representant för Lysator, men vi andra inbjöds av ISY genom Tord-Jöran Hallbergs försorg. Det betydde att vi alla fyra kunde vara med på eftermiddagens föredrag, men vi hade bara en biljett till middagen som följde på kvällen. Den tog Magnus Redin hand om.

Det är kanske inte alla som vet vad Datasaab var, eller varför Lysator skulle vara med, så låt mig dra historien helt kort.

Datasaab formades som en del av Saabs flygdivision i slutet av 1950-talet. De tillverkade och sålde både små och stora datorer och blev så småningom ett eget aktiebolag. Mot slutet av 1970-talet gick affärerna sämre och Datasaab har sedan dess haft många ägare och namn: Ericsson, Nokia och nu senast ICL. Datasaabs första stordator D21 började levereras 1962, och hemvändardagen ordnades för att fira detta trettioårsjubileum.

Lysator bildades 1973 när Datasaab skänkte en D21 som gått i pension. Lysator har också kört andra maskiner från Datasaab, och när fackföreningen på Nokia Data för ett par år sedan ordnade en museiutställning, fick de många av delarna från Lysator.

I Collegiums foajé en trappa upp var det redan fullt med folk när jag kom. När man registrerade sig fick man en namnskylt, och dessutom en diskret kavajnål i emalj som det stod Datasaab på. Dessa rariteter hade legat i lager sedan 1975 då de tillverkades. Nu var det dags för alla 350 deltagarna att samlas i stora hörsalen. Ur högtalarna strömmade musik som gjorts på D21 eller D22. Den finns inspelad på en grammofonskiva som Lysator har, om någon är intresserad.

Dagens konferenciär var Bernt Magnusson (Ericsson Radio i Mjärdevi) och han började med att hälsa hedersgästerna välkomna. Bland dessa fanns Skandinaviska Elverk som var den första D21-kunden, ytterligare någon tidig Datasaabskund, samt Lysator.

Dagens förste föredragshållare var tekn dr Tore Gullstrand, som arbetat med datorer på Saab sedan 1953. I början använde Saab statliga Matematikmaskinnämndens datorer BARK och BESK. Senare började man bygga sina egna maskiner, vilket ledde till bildandet av Datasaab. Nu för tiden köper Saab datorer av bland annat Digital Equipment och Cray, och dessa används till avancerade beräkningar som finita elementmetoden (FEM) och Computer Fluid Dynamics. Ett par områden där Saab var tidiga att använda datorer är matematisk formgivning och numeriskt styrda verkstadsmaskiner, kort och gott CAD/CAM.

1953 var utvecklingsarbetet på flygplan 35, Draken, i full gång. Men trots att Draken var långt ifrån färdig, fick Tore göra en utredning om "plan efter 35". Detta var alltså 15 år innan Viggen fanns. Hans-Olof Palme och Tore Gullstrand övertalade Saabs direktör Tryggve Holm om vikten av att även utveckla sidosystem som robotar och elektronik. Gunnar Lindström anställdes för att bygga upp en elektronikavdelning som fick namnet R-system (R för robot).

Den första uppgiften för R-system var en navigationsdator till en robot och nästa var utvecklingen av centraldatorn till Viggen (CK-37). Det byggdes också en stor analogimaskin för att prova ut styr- och reglersystemen till robotarna. Något som man tidigt insåg vikten av var hur informationen från datorerna skulle presenteras för piloten. Saabs försprång inom utvecklingen av människa-maskininterface märks idag när Saab planerar att sätta in Gripens head up-display i vanliga personbilar.

Tore Gullstrand avslutade sitt föredrag med ett par ”what if” om hur svensk elektronikindustri missat några enastående tillfällen. Tidigare nämndes BESK från Matematikmaskinnämnden[1]. Saab byggde i slutet av 1950-talet en kopia av denna elektronrörsdator och kallade den Saabs Räkneautomat, SARA. FACIT i Åtvidaberg byggde också kopior som de sålde och kallade FACIT EDB. När Saab föreslog ett samarbete runt utvecklingen av datorn, avslog FACIT och menade att de kunde klara sig själva. Det vet vi idag att de inte gjorde.

Av R-system som utvecklade elektronik och robotar, har bildats Saab Combitech som av historiska skäl finns både i Linköping och Jönköping. Samarbetet mellan dessa platser fungerade aldrig som det borde, och det är synd, menade Tore.

På 1970-talet fanns planer på att skapa ett stort svenskt halvledarföretag genom att bland annat Saab och ASEA skulle slå sina avdelningar ihop. Detta gick i stöpet och idag har flera svenska företag sina egna små halvledaravdelningar, men det mesta av produktionen går bara till det egna företagets behov.

Slutligen avrundade Tore Gullstrand med att sammanfatta 1950- och 60-talen som en gyllene epok för ingenjörer: Svårigheten var alltid tekniken, aldrig ekonomin. Det gick alltid att få pengar till nya projekt.

Det andra föredraget hölls på engelska av Edward O'Hara från International Data Corporation (IDC). Jag blev inte klar över exakt vad det företaget sysslar med, men det verkar röra sig om teknikbevakning och strategisk planering. Edward försökte ge perspektiv på datautvecklingen genom att berätta anekdoter om vad som hänt och skissa riktlinjer för framtiden. Det var en mycket intressant och underhållande framläggning.

En av anekdoterna som Edward drog var denna: 1954 hade IBM planer på att placera en, säger en, datamaskin i London och låta den betjäna hela Europas behov av datakraft.

Något som man kan fråga sig, när datorerna blir mindre och mer utspridda, är om de ändrar samhällsförhållandena. Kommer de att leda till en informationsrevolution, på samma sätt som ångmaskinen ledde till den industriella revolutionen? Edward menade att så kommer att ske, men att vi inte riktigt är där än. Fortfarande används datorerna huvudsakligen som en modern ersättare för pennan, medan förhållandet mellan chef, mellanchefer och anställda i stort är desamma som förr.

Varför införs datorer på allt fler kontor? Jo, naturligtvis för att rationalisera och skapa effektivare informationshantering. Men vinner man egentligen något på att datorisera? Idag används, har IDC kommit fram till, 80% av alla datorer (antagligen mätt i pengar som datorerna kostat) till ekonomiska system som redovisning och fakturering. När man mäter vinsterna av datoriseringen, bör man inte mäta på datorerna, utan på de som använder datorerna. Man ska alltså mäta chefernas och ekonomernas effektivitet.

[1] När BESK var ny 1953 var den en kort tid världens snabbaste dator. Här missade svensk industri en chans att komma 25 år före Silicon Valley.

Det man kommer att finna då, är att det med mer datorer går att klara sig med färre chefer, menade Edward. När informationen ökar i värde, blir cheferna arbetslösa. Detta är den informationsrevolution vi står inför. Chefer och mellanchefer kommer att bytas mot programvara.

Han berättade vidare om ett större flygbolag som utrett lönsamheten i de olika delverksamheterna. Det man fann var att flygtrafiken bar sina kostnader, men inte mer. Att transportera folk och varor med flygplan gav inte en dollar i vinst. Det som gav vinsterna var det datoriserade flygbokningssystemet. Där kunde man beställa hotellrum och hyrbilar och blommor och kemtvätt. Resenärerna betalade för detta och pengarna gick genom flygbokningssystemet till hotellen och hyrbilsfirmorna, som betalade en avgift för att få vara med i bokningssystemet. Man kan ana att pengarna också låg en dag hos flygbolaget och gav ränta.

Detta sätt att förmedla affärer, utan att egentligen sälja något själv, kallade Edward för "capturing transactions". Det är framtidens melodi, enligt honom. Han gav en vision av ett företag där anställda utan eget skrivbord sätter sig vid en dataskärm och förmedlar affärer hela dagen, utan att ha något väldefinierat yrke. Det låter som ett mellanting mellan aktiebörsen och Lysator, tycker jag.

Det tredje föredraget hölls av tekn lic Viggo Wentzel, och måste nog betraktas som dagens höjdpunkt. Efter licentiatexamen från Chalmers tekniska högskola 1957, där han arbetat med analogimaskiner, kom Viggo till Datasaab och var kvar där till 1987 då han gick i pension. Viggo var bland annat chefsingenjör bakom D21 och har haft nära kontakt med Lysator både när föreningen bildades och på senare år.

Redan 1957 hade SARA byggts. 1959 kom Gunnar Lindström till Saab, som Tore Gullstrand berättade ovan, och blev elektronikchef. Under den här tiden utvecklades de första transistoriserade datorerna. 1963 ändrades organisationen, så att Datasaab blev en självständigare avdelning. En av de första produkterna var CK-37, centraldatorn till den första modellen av flygplan 37 Viggen. Senare modeller av Viggen hade en amerikansk maskin som kallades dator 107.

Det var en dynamisk tid och bland annat lär Saabs direktör Tryggve Holm vid ett kundbesök ha uttryckt "Jag vaknade en morgon och fann att vi var datamaskintillverkare."

Saabs första D21 stordator levererades 1962. 1966 ställdes D21 ut på en mässa i Prag. Det resulterade i flera sålda maskiner i Östeuropa, och dessa kom sedan att användas länge när exportreglerna blev hårdare.

1969 kom minidatorn D5. 1974 övertogs datorn 6501 från FACIT i Åtvidaberg och kallades D15. Samma år togs beslut om utveckling av D16, delvis i samarbete med Computer Automation i Kalifornien. Datasaab hade nu 3000 anställda. 2000 minidatorer tillverkades om året och var tionde anställd var en minidator; man levde som man lärde.

Under åren 1972-75 hade Datasaab dotterbolag i sex länder och USA. Exporten utgjorde 44% av försäljningen. 1975 gick försäljning och orderingång mycket bra och företaget expanderade kraftigt. Detta ledde tyvärr till växtvärk och följdes av bakslag de följande åren. 1980 var banksystemens exportandel 93% medan andra produkter exporterades till 59%.

1975 bildades samarbetsbolaget Saab Univac för att ta hand om Saabs större datormodeller i D20-serien. Minidatorerna blev kvar i Datasaab, som var en division av Saab. Genom sammanslagning med statliga Stansaab 1978 bildades Datasaab Aktiebolag som ägdes till hälften var av Saab och svenska staten. 1981 såldes detta bolag till L-M Ericsson och fick namnet Ericsson Information Systems. Efter diverse äventyr såldes bolaget 1987 vidare till finska Nokia och döptes om till Nokia Data. Senare har det sålts till brittiska ICL.

Under Datasaabs tid som Datasaab fanns det tre speciellt framstående produkter som Viggo ville nämna, och tre anledningar till att de lyckades. Anledningarna var att tekniken var ny, att produkterna kom rätt i tiden, samt att man lyckades göra spännande nya tillämpningar. Exempel på det sista är styrning och övervakning av elkraftnätet samt väderprognoser. Idag är det självklart att använda datorer till sådant, men så var det inte när Datasaab sålde sina första system till Skandinaviska Elverk och SMHI.

Den första produkten är D20-linjen, stordatorerna. D21 kom 1962. Redan från början användes högnivåspråket Algol Genius, Saabs variant av Algol-60 med generellt in- och utmatningssystem, för både tekniska och administrativa tillämpningar. D21 såldes i totalt 30 exemplar. Efterföljaren D22 kom 1968 och var helt kompatibel med D21. Den såldes i 70 exemplar. Nästa modell baserades på en mikroprogrammerad Flexible Central Processing Unit, FCPU. Den levererades i 10 exemplar som kompatibel till D22, under namnet D23, men var egentligen tänkt för mer avancerad användning.

Bland stordatorkunderna märks Skandinaviska Elverk, SMHI, Kockums varv, Vägverket och Riksskatteverket. Här måste jag återge några av de kunddialoger som Viggo nämnde. Från Skandinaviska Elverk kom ett telefonsamtal: ”Viggo, skicka gärna hit servicemännen. Men låt dem bara inte röra maskinerna!” Det var Saabs direktör Tryggve Holm som förhandlade med finansminister Gunnar Emanuel Sträng om försäljningen av datorer till skatteverket: ”Tryggve, man säger att ni inte har COBOL!?” – ”Hör du Emanuel, det ska jag säga dig att vi har COBOL i varje bil.”

Som den andra succéprodukten nämnde Viggo Wentzel CK-37, som levererades i 200 exemplar till AJ37 Viggen. Med vissa modifikationer kommer den att finnas i drift in på 2000-talet.

Den tredje framgångsrika produkten var minidatorn D5. Det började 1968 med att Nordiskt Spardata, ett samarbetsorgan för sparbanker i de nordiska länderna, gav en offertinfordran på banksystem till 15 företag. På den här tiden dominerade amerikanska NCR med försäljning av stora datorer till bankerna.

Det var en liten grupp på Datasaab som fick uppdraget, och de valde att utveckla D5/20 och det interpretativa språket DIL. I september 1969 fick huvudkonkurrenten Olivetti ge upp, för då undertecknade Datasaab kontraktet i Köpenhamn. Den sista D5 togs ur drift i slutet av 1980-talet.

Viggo Wentzel avslutade sitt föredrag genom att påpeka att ingen ännu har skrivit Datasaabs historia. När Viggo gick i pension gjorde han tillsammans med fackföreningen på Nokia Data den utställning som kan ses i entren till nuvarande ICLs lokaler på Mjärdevi. En del gammal dokumentation från Datasaab har också lagrats i landsarkivet i Vadstena.

Ingen har emellertid sammanställt allt material till en bok, och detta efterlyste Viggo nu. Kanske blir han bönhörd, för efter föredraget kom en kille från Universitetets Tema teknik och social förändring fram och presenterade sig. Jag vet att Mikael Cardell också är mycket intresserad av vår lokala datateknikhistoria.

Sen följde en kaffepaus och efter den presenterades hemvändardagens organisationskommitte och sponsorer. Många sponsorer var sådana som varit tryckerier och reklamagenter åt Datasaab, eller före detta Datasaabare som nu fanns i andra företag.

Tre föredrag återstod. Dessa hölls av halvledarprofessorn Christer Svensson och krypteringsprofessorn Ingemar Ingemarsson samt av Alfred Friman. Det sista var skojigast, för Alfred talade om det svenska kynnet under trettio år. Han drog otaliga vitsar om hur vi svenskar beter oss, och jag hade så roligt att jag inte har några anteckningar kvar från det.

Ingemar berättade hur han 1971 kom som ny professor till Linköping samtidigt som Datasaab utvecklade de första uttagsautomaterna åt sparbankerna. Han deltog då i arbetet med att finna algoritmer för kryptering med den personliga koden som knappas in vid sedeluttag. Huvuddelen av föredraget hade samma innehåll som första föreläsningen i kryptologikursen, och det orkar jag inte återge här.

Christer Svensson, som är professor vid Institutionen för fysik och mätteknik, berättade om halvledarteknikens utveckling historiskt och i framtiden. De integrerade kretsarna i dagens datorer använder fälteffekttransistorer som kallas MOS, Metal Oxide Semiconductor. MOS-transistorn uppfanns redan 1934, men fungerade inte. Den fungerande MOS-transistorn uppfanns 1962.

1962 hade man alltså en ”krets” med en transistor. 1987 rymdes en miljon transistorer på samma krets. Utvecklingstakten är densamma idag som då, menade Christer, men hur länge till kommer den att kunna vara det? Kommer utvecklingen att nå en gräns, eller kan den fortsätta länge än?

Generellt är det så, sade Christer, att beräkningshastigheten och effektförbrukningen hos en krets ökar med inverterade kubiken på transistorernas längd. När man kan göra en transistor hälften så stor, blir kretsen åtta gånger snabbare och alstrar åtta gånger mer värme. Men den minnesmängd man kan bygga på en minneskrets ökar bara med inverterade kvadraten på transistorlängden. Samtidigt som beräkningskraften åttafaldigas, kan man bygga fyra gånger mer minne på en krets.

Det finns flera faktorer som kan begränsa den här utvecklingen. Om transistorerna arbetar med för små energisteg, blir de känsliga för brus. Men vi har långt kvar ännu till så små energisteg. Transistorerna kan inte heller arbeta med alltför små spänningar.

Alltför små avstånd ger tunnelingeffekter, men den gränsen ligger vid ungefär 10 nanometer och dagens kretsar har avstånd runt 1 mikrometer, hundra gånger större än det kritiska värdet. En faktor hundra i längd gör faktorn en miljon i beräkningskraft eller tio tusen i minnesmängd.

Så långt kan man alltså utveckla datorernas kretsar enbart genom att tränga ihop dagens teknik. Detta ger ett andrum på ungefär tjugofem år, om utvecklingen går i samma takt, menade Christer. Det finns alltså inga hinder de närmaste tio eller femton åren. Genom att ändra principerna för hur kretsarna fungerar, borde man kunna fortsätta långt in i framtiden. Och det låter ju betryggande, för oss som vill ha snabbare maskiner att programmera.

Fun With MUD

av B. W. Smith

Följande artikel är klippt från januarinumret 1985 av Computing Today*, och handlar om Essex-MUD, världens första MUD.*

It's so cheap to kill your friends, after 12 and at weekends. Multi-User Dungeons, played remotely at the end of a telephone line, could herald the future of computer gaming.

Every adventure game written for computers has its roots in the original role-playing fantasy games, where a Dungeon-master created a whole fantasy world on a table top, and the players had to explore it while acting out their chosen personas. The whole concept can be curiously compulsive, and hardened role-players have been known to comment that what happens in the game is real life: the fantasy world is the one that you and I inhabit. This type of game is also part acting, and a lot of pleasure comes from interacting with the other members of your team, all of whole have their own make-believe characters.

Transfering this sort of game to a home computer presents problems. A role-playing game can be as big as the Dungeon-master cares to make it, while micros are limited to relatively small databases. On top of that is the problem of hooking up all the players simultaneously; networking is still relatively rare in the home market. Consequently the direction taken by adventure games has been the single-user puzzle-solving programs which we all know and love.

Glorious MUD

For some time now, however, a true multi-user role-playing computer game has been available to those with the right equipment. The game is called Multi-User Dungeons (not surprisingly!), or MUD for short, and it runs on a DEC System 10 minicomputer in Essex University. If you aren't actually on site to play in person, then you can play remotely using your home computer and a modem as a terminal.

This explains why MUD is a nocturnal or weekend activity: the telephone charges are cheaper then. Also, the DEC is required for rather mundane tasks during working hours! Unfortunately some restraint is still called for: nighttime play didn't prevent one player from racking up a massive £3000 phone bill, which not surprisingly brought participation to a sudden end.

It may seem odd that a serious institution would devote so much equipment, time and programming effort into such an apparently frivolous pursuit. But MUD has hidden depths. From a computer science viewpoint MUD provides a testbed for artificial intelligence routines (for example, the parser which interprets the English sentences input by the players), and for communications (MUD has probably one of the most advanced user-to-user communication channels in public service: better than Micronet, for example). Co-author of MUD Richard Bartle belives that MUD has put Essex in the forefront of this field.

Play Time

MUD is a real-time game, and he who hesitates will probably find that another player has beaten him to the treasure. Any player who is on-line can manipulate the database, so if you drop an object somewhere and go back to collect it later, it's quite likely that someone else will have found it and taken a fancy to it.

You can talk directly to any or all of the other players by using the person's name or the command Shout. You can fight them as well if you wish, but novices are warned that this could be a quick way to die early. Cooperation is also possible, and you can ask another, more powerful player to perform some action that is beyond you: he, of course, may or may not oblige. Or he might demand payment of a treasure.

Just like the 'real' role-playing game, your character develops with experience. Players start off as novices and can work their way up through the ranks to the heady heights of Wizard. Players at this level have quite extraordinary powers, and can spy on lesser players, move them around and generally be the sort of nuisance that wizards are supposed to be.

The human players are not the only participants in MUD's adventures: the game has its own characters called mobiles, which are creatures, usually hostile, which are controlled by the program. They can be quite nasty, and a group of players may need to combine their strengths in order to vanquish one.

Getting In

MUD is free to play, but you require a modem and a computer configured as a terminal. You will also need an account with the PSS (Packet Switching System) of British Telecom, which doesn't come cheap. (Compunet users will be getting their own version of MUD). You're probably looking at a start-up cost of £150 if you only have a computer at present. On the other hand, the fascination that MUD has on its players may be worth this outlay. I know one Wizard, Thor by name, who gets a lot of useful information through his MUD contacts: sometimes even paying for real-world data with fantasy-world treasure!

He also points out that MUD has a disturbing tendency to reinforce character traits in certain players, which is reflected in their daytime behaviour. So if you do get the MUD bug, try to keep a grip on reality: whatever that is...

Sett i LysKOM:

```
231281 1992-08-07  09:56  /1 rad/ Dr Winston O'Boogie Jr
Kommentar till text 231198 av Örjan Berglund
Mottagare: Religioner, världsåskådningar och sånt <4235>
Ärende: Kristendom utan att straffas med döden för det?
------------------------------------------------------------
...vidare diskussion förutsätter en bra definition av definition...
(231281) -----------------------------------
```

PPP - ett Punkt-till-Punkt-Protokoll

av Pell Pell Pell

När man talar om nätverk handlar det numera oftast om LAN av olika typ, framförallt implementerade med Ethernet. Om man däremot tittar på dataöverföring i allmänhet så är givetvis serieförbindelser ytterst vanliga. Varenda dator har en serieport och det finns minst 2*N kommunikationsprogram, där N är det högsta slumpmässiga tal du kan tänka dig. Trots det är det inte många som använder nätverksprotokoll (typ TCP/IP) för att transportera data över sina serielinor. En orsak är att det inte har funnits någon standard för hur det ska gå till. Vissa ad-hoc-protokoll som SLIP och CSLIP har använts ett tag, men inte blivit riktigt populära. De kan också endast användas för IP. För DECnet finns DDCMP och för synkrona länkar HDLC, men en övergripande Internet-standard har saknats.

Evangelium enligt RFC 1331

För att avhjälpa denna brist skapades PPP, dvs Point-to-Point Protocol, som beskrivs i RFC 1331. PPP är ordentligt influerat av HDLC (länknivån i X.25) och kan användas över både synkrona och asynkrona kanaler, som t.ex. RS-232C, RS-422/423 och V.35 (RydNät-länken). verföringen måste vara full-duplex och 8-bits ren, men det finns en mekanism för att undvika att skicka vissa tecken, t.ex. XON/XOFF, samt att filtrera bort dem om de skulle råka smyga sig in i det mottagna meddelandet. Feldetektering sker med ett 16-bitars CRC.

PPP har ett eget länkprotokoll, LCP (Link Control Protocol). Med det kan PPP förhandla med motparten om vilka tecken som inte får skickas, max paketstorlek, koppla upp/ner förbindelsen, rapportera fel, mm. PPP sköter bara en fast länk, så vill man använda modem måste man ha extra programvara som ringer upp, loggar in, startar en PPP-länkprocess på motdatorn, mm. Mer om det senare.

PPP kan transportera flera protokoll, nämligen ca 16000 stycken. Normalt brukar en ”protokollnivå” inte bry sig om vad det är för data som transporteras, men det bygger på att mottagaren kan titta på dataklumpen och se vart den ska, vilket funkar bra om man håller sig inom samma protokollfamilj. Eftersom PPP stödjer flera familjer, måste PPP veta till vilket protokoll den ska leverera datamängden och till det avsätts två bytes i PPP-headern. (Snuskiga detaljer: Den första byten är alltid jämn och den andra alltid udda. Så har t.ex. IP nummer 0x0021 och DECnet-IV 0x0027. De ”populära” protokollen har alltid 0 som mest signifikant byte och för dem räcker det att bara specificera den minst signifikanta byten. Eftersom den är udda kan mottagaren direkt avgöra om det är ett 1-bytes eller ett 2-bytes nummer som kommer.)

I övrigt har PPP väldigt lite ”overhead”, förutsatt att komprimering enligt ovan accepteras i förhandling med motpartens PPP (även andra komprimeringar av PPP-ramen finns). Använder man sen TCP/IP med Van Jacobsons header-komprimering blir det bara 5-10 byte overhead.

Hur gör man?

Då är frågan hur man kör PPP på sin dator. Det finns ett paket för PPP på fasta linor från CMU, som fungerar på Sun och 386/BSD.

För uppringd PPP (över modem) finns ett paket från Purdue, som har CMU-PPP som huvudbeståndsdel och som fungerar på Sun.

För PC tror jag KA9Q har PPP. Sen finns förstås kommersiella varianter.

En annan fråga är om det räcker med ett 2400-modem för att köra PPP, eller om man måste gradera upp sig. Lite statistik kommer senare, fast du som vill hålla fast ditt gamla 75/1200-modem kan nog sluta läsa här.

Så här funkar det för mig

Jag har kört med Purdue-paketet i drygt en månad. Det heter för tillfället dp-2.2beta (dp står för Dialup PPP) och "beta" är inget att vara rädd för enär det fungerar mycket stabilt. dp har Van Jacobsons TCP/IP-kompression (VJC). (Samma som i CSLIP, såvitt jag vet)

dp har som enda paket en laddbar device-driver (loadable module) som implementerar PPP som en STREAMS driver. Det innebär att ingen omlänkning av UNIX-kärnan behövs. En modload(8) bara och så har man plötsligt PPP som ett nätverks-interface. Vad modulen i kärnan gör är att få paket från nätverksdelen i kärnan, konvertera det till VJC och skicka det till STREAMS-drivern för ttyporten modemet hänger på. Och tvärtom från andra hållet. Den fångar också upp alla LCP-paket, men behandlar dem inte själv. I stället skickar den ut dem till en användarprocess, som heter "dp". Den processen sköter alla förhandlingar med motpartens "dp" och den kan förstås också skicka ner LCP-paket till kärnan (kommunikationen sker via en device i /dev).

För inloggningsförfarandet finns två andra processer: dpd och dplogin. I begynnelsen finns ingen "dp"-process utan det är i stället dpd som lyssnar på /dev/dp. Ett PPP-interface är alltid markerat som UP, även om det inte finns någon förbindelse. När det kommer ett paket (t.ex. efter en "telnet lysator" och routen till lysator pekar på PPP-länken) och länken är nere, signalerar PPP-drivern till dpd att det är dags att ringa upp. dpd forkar då av sig en kopia som ringer upp motmodemet.

På motsystemet ligger en helt vanlig getty och lyssnar på modemet, så att "normala" inloggningar kan ske helt som vanligt. Däremot finns en speciell login-ID som har "dplogin" som shell! När dpd får kontakt så loggar den in med ID och lösenord som vanligt. Därefter svarar dplogin "PPP-start" som är början på en kort handskakning med dpd. Både dpd och dplogin gör sedan en exec på varsin "dp"-process vilka omedelbart börjar förhandla med varann och snart är förbindelsen uppkopplad och TCP/IP-paketen börjar flyta.

PPP-drivern tittar lite grann på TCP-protokollet för att se när en (virtuell) förbindelse kopplas upp och ned. Denna information används sedan som indata till några olika timers för nedkoppling.

Man kan konfigurera hur snabbt länken ska kopplas ner efter att sista TCP-förbindelsen stängs, samt dessutom specificera att länken ska kopplas ner efter en viss "idle-tid". Den kopplas alltså ner trots att det finns flera aktiva förbindelser, men så fort det kommer ett nytt paket på någon av dem ringer dpd upp igen, enligt ovan. Allt görs helt transparent.

Hur blir det nu med mitt hembyggda 300-baud modem?

Så till hastighet, med mera. Själv kör jag ett 2400-baud modem med MNP1-5 och V.42/V.42bis. Jag visar nedan lite statistik med användning av olika modemprotokoll.

Konfiguration

Maskin	Dator	Modem	Serieport
Lokal	Sun 3/50	PM2400SA	9600 baud
Remote	Sun 3/60	Octocom	14400/38400 baud

Inga av maskinerna var belastade. Flödeskontroll fungerade (få omsändningar eller borttappade paket).

Tester

För tester 2-7 användes programmet tcpspray som skickar 1024-byte-block till discard- (eller echo-) porten på målmaskinen. Alla block är identiska, men kan fyllas med olika godis. Alla värden för dataöverföring är i bytes/s med två signifikanta siffror, såvitt inget annat anges.

Protokoll	1	2	3	4	5	6	7
Ren 2400	900ms	220	180/170	150	150	160	150
V42	1000ms	250	260/240	180	160	170	170
MNP4+5	1100ms	880	840/810	230	290	170	170
V42+V42bis	680ms	960	840/800	570	520	560	260

1. Ping. Medelvärde av 20 paket (stabilt för alla konf.). Värde i ms.
2. 100 paket nollor (NUL). (*)
3. 100 paket nollor, men till echo-porten. (Endast 50 paket för ren 2400 och V42.) Resultatet är här två värden: transmit/receive
4. 20 paket av de första 1k av tcpspray-binären. (*)
5. 20 paket av de första 1k av källkoden till tcpspray. (*)
6. 20 paket av de första 1k av en 8-bits Sun rasterfil. (*)
7. 20 paket av de första 1k av samma rasterfil men komprimerad (16-bit).

(*): Data i dessa tester kan komprimeras så här mycket med compress: 2: 95% :-) 4: 22%, 5: 28%, 6: 25%. (OBS: endast själva 1k-blocket)

Sammanfattning

Som synes av statistiken ovan är V.42+V.42bis en bra kombination och helt acceptabel för interaktiv trafik, medan en ren 2400-förbindelse känns ganska seg. Det går att köra en del X-program sin Xeyes och liknande, men till exempel FrameMaker kan man glömma.

Kör man en FTP till en maskin nära motmaskinen låser sig länken helt (ur interaktiv synvinkel) och vaknar inte upp igen förrän filen är överförd. Problemet är att FTP inte ACKar paket utan litar på TCP (vilket i och för sig är helt rätt). Tyvärr medför det att den sändande FTP-processen gör en jätte-write på hela filen och låter sen nätverksdelen i kärnan stycka upp det i mindre paket. Man kan se det som en jättepropp i ledningen, eftersom andra paket inte får nån större chans att komma till. De gånger jag har behövt köra FTP har jag därför skickat över filerna till USA och sen FTP:at dem därifrån[2]. Det långa avståndet gör att FTP-paketen får en tidslucka där mindre paket kan slinka emellan och det fungerer grejt.

FTP-problemet kommer att fixas i en framtida version av dp genom att de paketen märks med en annan TOS (Type Of Service - se TCP-RFCn). Man kan då skilja på FTP och övrig trafik och ge företräde åt den senare. Det här är egentligen nåt som borde ha fixats ordentligt för länge sen. TCP har alltid haft en TOS i headern (såvitt jag vet), men den har aldrig använts till nåt. Det hade varit mycket roligare på alla nätverk om FTP hade haft en lägre TOS än t.ex. telnet-trafik. Fixen i dp ser jag som en kludge. Fast funkar det är det ju pent.

Jag har inte provat NFS, av förklarliga skäl.

För mer information, läs newsgruppen comp.protocols.ppp.

[2] Fy på mig!

Parsning av naturliga språk med länkgrammatik

av Pell Pell Pell

Jag tänker inte låtsas som om jag har läst datalingvistik, för det har jag inte. Däremot hittade jag ett programpaket för parsning av språk som verkade skojigt och jag kunde inte låta bli att roa mig lite med det. Det är bara syntaktisk parsning, men parsningen sker efter vissa grammatiska regler som gör att man ofta kan få god semantisk information tillbaka.

Att grammatiken är länkbaserad innebär att parsern binder samman ord, eller följder av ord, med länkar. Det finns sedan vissa regler, t.ex. att länkar inte får korsas. Varje ord, samt vissa kombinationer av ord, har en eller flera "kontakter" som länkar kan kopplas till. Intresserade hänvisas till artikeln som medföljer programpaketet (se nedan).

För att visa hur man kan bygga en grammatik tar jag svenska som exempel. Först vill vi ha lämpliga subjekt i en mening:

```
jag hund katt: S+;
```

Den här deklarationen säger att "jag", "hund" och "katt" kan användas som subjekt (`S`) i en mening. Plustecknet efter `S` innebär att där kan kopplas en `S`-länk, men bara riktad mot höger. Här är predikat (verb):

```
hoppar sover flyger: S-;
```

De orden accepterar en tt S-länk från vänster. Vi testar parsern med den här grammatiken:

```
> jag sover
 +--S-+
 |    |
jag sover
```

```
> sover jag
Not accepted (no linkage exists)
```

```
> hund flyger
  +--S--+
  |     |
hund flyger
```

"Sover jag" accepterades inte, eftersom länkarna för den meningen blir:

```
S-+    +-S
  |    |
sover jag
```

vilket inte går ihop. Det är förstås en korrekt svensk mening (fråga), men inte enligt den här minigrammatiken. Något annat som inte heller är så bra är att substantiv i singularis behöver "en" eller "ett" i subjektsställning. Det är väl lika bra att införa bestämningsord direkt och visa fler egenskaper hos grammatiken:

```
jag: S+;
en:  Don+; ett: Dot+; den: Dbn+;
det: Dbt+;
de:  Dp+;
```

```
hund:               Don- & S+;
äpple:               Dot- & S+;
hunden:               {Dbn-} & S+;
äpplet:              {Dbt-} & S+;
hundar äpplen:   S+;
hundarna äpplena:{Dp-} & S+;
hoppar sover flyger: S-;
```

Jag tar de engelska namnen för länkarna (därav `D` för determiner) även fast de kan väljas helt fritt. Versaler står för en grammatisk huvudkonstruktion, medan gemener är en liten modifiering av nåt slag. Vi ser att bestämningsord nu krävs i singularis. I plural obestämd form står däremot namnet för "bestämningslänken" inom {}. Det betyder att länken är tillåten, men inte nödvändig. Vi testar att parsa några meningar:

```
>en hund flyger
 +-D-+--S--+
 |   |     |
en hund flyger


> ett hund flyger
Not accepted (no linkage exists)


> den hunden flyger
 +--D--+---S--+
 |     |      |
den hunden flyger


> hundarna flyger
    +---S---+
    |       |
hundarna flyger


> de hundarna flyger
 +--D--+---S---+
 |     |       |
de hundarna flyger
```

Det gick ju bra, men grammatiken är något begränsad. Vi testar att införa transitiva verb, samt objekt. Objektlänken kallar vi `O`:

```
hund:  Don- & (S+ or O-);
matar: S- & O+;
```

Det här betyder att hund kan vara subjekt eller objekt, men inte båda samtidigt. De andra substantiven modifieras på liknande sätt. Ordningen mellan kontakterna är betydelsefull. I det här fallet betyder det att ett bestämningsord alltid ska stå före substantivet, oberoende av om det senare är subjekt eller objekt. Matar blir ett ord som behöver både subjekt- och objektlänk.

```
> jag matar
Not accepted (no linkage exists)


> jag matar hunden
 +--S-+---O--+
 |    |      |
jag matar hunden


> jag matar en hund
      +----O---+
 +--S-+    +-D-+
 |    |    |   |
jag matar en hund


> en hund matar en hund
          +----O---+
 +-D-+--S-+    +-D-+
 |   |    |    |   |
en hund matar en hund
```

Som nästa exempel kan vi lägga till adjektiv, som ju modifierar substantiv. Adjektiv böjs tyvärr beroende på genus och antal och dessutom kan flera adjektiv verka på samma substantiv.

```
hund:   {@Asn-} & Don- & (S+ or O-);
äpple:  {@Ast-} & Dot- & (S+ or O-);
hunden: {@Ap-} & {Dbn-} & (S+ or O-);
äpplet: {@Ap-} & {Dbt-} & (S+ or O-);
hundar äpplen:
        {@Ap-} & (S+ or O-);
hundarna äpplena:
        {@Ap-} & {Dp-} & (S+ or O-);

svart:              As+;
ful gammal:         Asn+;
fult gammalt:       Ast+;
svarta fula gamla:  Ap+;
```

Här visas några nya konstruktioner. @ framför en kontakt anger att kontakten kan ta en eller flera liknande länkar. Ett substantiv kan alltså kopplas mot flera adjektiv. ”Svart” kan kopplas både mot den- och det-ord (men endast i singularis) och behöver därför ingen n/t-specifikation.

```
> gamla hunden flyger
  +---A--+---S--+
  |      |      |
gamla hunden flyger

> den gamla hunden flyger
 +-----D-----+
 |    +---A--+---S--+
 |    |      |      |
den gamla hunden flyger

> jag matar en svart ful hund
      +---------O--------+
      |    +------D------+
      |    |   +----A----+
 +--S-+    |   |    +--A-+
 |    |    |   |    |    |
jag matar en svart ful hund
```

Okej, det ska vara komma mellan adjektiven och det kan man fixa med grammatiken. Det här är emellertid främst menat som en enkel introduktion. Slutligen kan vi ta adverb:

```
hoppar sover flyger: S- & {E+};
matar:       S- & {E+} & O+ & E+;
snabbt skickligt:    E-;
```

Det är kanske lite tveksamt om man ska tillåta adverbet mellan verbet och objektet, men det är en ganska vanlig konstruktion så det kan vi göra.

```
> jag sover snabbt
 +--S-+---E--+
 |    |      |
jag sover snabbt

> jag matar hunden skickligt
      +-------E------+
 +--S-+---O--+       |
 |    |      |       |
jag matar hunden skickligt

> jag matar snabbt hunden skickligt
      +----------E----------+
      +------O------+       |
 +--S-+---E--+      |       |
 |    |      |      |       |
jag matar snabbt hunden skickligt
```

Nu är det bara att bygga på! Det finns en engelsk grammatik som är ganska duktig och som t ex skulle kunna användas för att göra en roligare parser för MUD. . .

Det här skulle nog också kunna vara ett lämpligt examensarbete för någon datalingvist.

Program, dokumentation och engelsk grammatik finns att hämta med FTP från world.std.com:/src/natlang/coling/

Automatisk sagogenerering

av Leif Stensson

För ett antal år sedan, när jag gick tredje året på gymnasiet, fick eleverna i uppgift att parvis skriva "skönlitterära" uppsatser på vilket tema som helst. Jag och en kompis skrev en inte alltför bra science-fiction-historia, som inte ens blev helt klar, men trots detta lämnade vi in de åtta maskinskrivna sidor vi åstadkommit (och fick betyget "mycket välskrivet, jag läser gärna fortsättningen" på :-).

Nu hade en annan kompis åsikter angående det där med välskrivet; han ansåg att han skulle kunna skriva något bättre själv på en kväll, och det försökte han göra. Resultatet blev en alldeles horribel historia, svagt plagierande Bilbo, välkryddad med stav- och grammatikfel orsakade av otålighet vid skrivmaskinen. När jag fick läsa den kunde jag kunde inte motstå att utbrista "det där kan jag skriva ett program som skriver bättre sagor än!"

Till verket

Jag satte alltså igång med att skriva ett sagoskrivarprogram, och jobbade lite till och från med det, några timmar per månad. Det som tog mest tid var att hitta på själva texten som programmet utgår ifrån, men det underlättade att han som skrivit uppsatsen jag försökte parodiera tyckte det var väldigt lustigt med bland annat ord som upprepas i lite olika former ("på taket satt några gastar och gastade"), och "fyndiga" skämt (läs: buskis). Han hade även en förkärlek för pågåendeform, i synnerhet med adverb och/eller prepositioner påhängda i början, intetsägande beskrivningsfraser ("en ovanligt säregen plats"), exotiska, inlånade och inte alltid helt passande synonymer för vanliga fraser ("exklamerade X godmodigt") och liknande språkliga krumbukter.

Slutligen bör man kanske även nämna att hans karaktärsskildringar (om man nu kan kalla dem för det) bestod i att de "goda" var kloka och skickliga, medan de "onda" var dumma och klumpiga; att sedan alla personer i historien agerade vettlöst förstärkte det löjliga intrycket.

Jag tycker jag har lyckats ganska bra med att parodiera dessa (och ett antal andra) stildrag i det han skrev; men allt är inte baserat på hans uppsats[3], en del är inspirerat av annat, som till exempel dikterna som huvudpersonerna ibland får för sig att läsa, som är en liten hyllning till det gamla klassiska LOGO-programmet "POET".

Sagoserver

Under de senaste månaderna har jag passat på att utöka programmet lite, samt anpassa det till Unix. För att alla ska kunna njuta (eller plågas, välj själv) av sagorna så har det nu ordnats så att om man skickar ett email till saga@lysator.liu.se, så startas sagogeneratorn, och skickar ett svarsbrev innehållande en nygenererad saga. Det finns även ett exempel på hur sagorna kan se ut på nästa sida.

[3] Uppsats*er*, faktiskt. Han skrev icke mindre än tre till, den jämmerliga kvalitén till trots.

Framtidsplaner

Än så länge följer sagan som programmet genererar ett ganska strikt ”sagoskelett”, så att alla sagor börjar med att två figurer bestämmer sig för att leta efter någon speciell sak, reser iväg, kommer fram till platsen där saken finns, hämtar saken, och reser hem; men trots detta blir det någotsånär varierande sagor. I nästa generation av sagogeneratorn, som jag redan börjat skriva på, ska sagorna blir mer varierade, genom att baseras på en uppsättning mallar för olika sagoepisoder som den sedan passar ihop. Det är även tänkt att man ska kunna välja lite olika teman (alltså inte bara ”fantasysaga”), men vi får se hur det blir med det.

Datorgenererad parodisaga

av Sagoservern

En molnig junimorgon hände det sig att sotaren Wemdel kringvandrade regionerna runt sin hemby Paftum. Plötsligt uppenbarade sig hans gode vän Erian.

–Jag har hört ryktas, skrockade Wemdel exciterat, att Den Förtrollade Trollstaven står att finna vid Trollens Slott.

–Remarkabelt, tyckte Erian kvickt! Vi måste genast börja eftersöka den!

–Må dä', skrockade Wemdel inom några minuter.

Efter en gnutta förberedelser gav sig Erian och Wemdel ut på enorma äventyr!

Utan större problem anlände de till en grandios allé, där en grönskande grönska framväxte ur marken.

–Låtom oss tänka, framviskade Erian. Vi vill fortsätta vår tripp på bästa möjliga sätt, så jag tycker bestämt att vi måste bestämma oss för vart vi vill komma! Wemdel och Erian hastade hastfyllt vidare.

Strax råkade Erian och Wemdel befinna sig vid ett enormt fält, varifrån både det ena och det andra kunde beskådas.

–Aha, meddelade Erian godmodigt, det där hållet verkar intressant! Våra vänner fortsatte sin tripp.

De bägge männen ankom inom kort till ett enormt morotsland varvid de tog sig en hastig rast. Vidare växte där ett ganska stort träd.

–Nu vet jag, stönade Erian oroligt! Jag tycker att det är dags för oss att omedelbart utforska detta synnerligen intressanta träd!

–Javisst, tyckte Wemdel motvilligt!

Sagt och gjort. Erian och Wemdel tog sig kvickt upp i trädet. Våra vänner framklättrade nu till en gren, vars förgrening ledde dels till vänster och dels till höger.

–Vilken tur, exklamerade Erian medan han såg sig om, här, tro det eller ej, ligger en slangbella!

Våra vänner tog sig fram på en svajig gren.

–Aha, hojtade Wemdel undrande, här, bland löven, ligger en tom läskback!

–Hurra, skrockade Wemdel uppgivet, vilken utsikt! Jag tycker det ser ut som om vår resas mål ligger ditåt, österut!

–I mycket hög grad, menade Erian mystiskt!

Wemdel och Erian utträdde nu i trädets topp, och såg storslagna vyer.

–Vilken tur, hojtade Wemdel vist, här kan man se sig vida omkring! Jag tycker det ser ut som om vår resas mål ligger ditåt, norrut!

–Minsann har du inte rätt, menade Erian tankfullt!

Våra vänner utträdde nu i en minimal trädkoja utan vare sig tak eller golv. De kom inom kort ut i trädets topp, och såg storslagna vyer. Våra vänner tog sig fram vid trädets stam, stammande av upphetsning.

–Praktfullt, påstod Wemdel medan han framvisslade en melodi, vilken utsikt! Jag får det intrycket att vår resas mål ligger ditåt, västerut!

–Jag instämmer, meddelade Erian gladlynt!

De utträdde nu i samma träd som förut.

–Se här, skriade Wemdel efter en stund, här, på en gren, ligger en flaska kloroform!

De framklättrade nu på en sidogren, varifrån både det ena och det andra kunde observeras.

–Glo hitåt, suckade Erian tankfullt, här, i en urholkning i stammen, ligger en telefonlur!

–Glo hitåt, frambringade Erian glatt, vilken utsikt! Jag tycker det ser ut som om vår resas mål ligger ditåt, norrut!

Just det, tyckte Wemdel eftertänksamt! Våra vänner tog sig klokt ner ur trädet, tillbaka till marken.

–Hmm, yttrade Erian med rätta. Vi vill ju fortsätta vår färd på bästa möjliga sätt, så jag tycker att vi måste bestämma oss för vart vi vill komma!

De fortsatte flinkt sin färd.

Efter en stund hände det sig sålunda de bägge männen ankom till ett frodigt gärde som gav ett rogivande och behagfullt intryck. Plötsligt hörde de bägge männen ett misstänkt ljud och vände sig snabbt om.

–Ånej, menade Erian godmodigt, det är den ökända ödlan som inte tänker låta oss passera!

–Nu vet jag, utropade Wemdel upphetsat!

Wemdel härmade en telefon, ”Ring!”, och räckte över telefonluren till den ökända ödlan med orden ”Du har telefon!”, och medan den ökända ödlan misstänksamt lyssnade i luren lurade sig Erian och Wemdel förbi!

–Hmm, frambringade Erian undrande. Vi vill fortsätta vår resa på bästa möjliga sätt, så jag tycker att vi måste bestämma oss för vart vi vill komma!

Precis som våra vänner räknat med framnådde de nu till den lilla staden Ralant, vars gator var belagda med kullerstenar som var så kullriga att de nästan trillade omkull.

–Nej men se, tyckte Wemdel hoppfullt, det där hållet verkar ganska intressant!

Gladhågade fortsatte de tills de elegant anlände till en grandios glänta med ett stort antal buskar i den kring området liggande omgivningen. Vidare låg där en mycket gammal slottsruin.

–Jag tycker det är dags för oss att nu inträda i denna vackra ruin!

Så skedde skickligt. De inklev till sin stora glädje i ruinen, och hittade till sin stora häpnad ett gammalt vakttorn till vilket de framsteg under tyst diskussion.

Efter en längre stund kom de till ett högt murkrön, som syntes dem maffigare och maffigare ju närmare de kom. Klättrande bland skräp nådde de ett högt torn, bevuxet med klängväxter.

Plötsligt kom våra vänner fram till ett murket stall, varifrån höga berg kunde skådas. De kom efter en längre stund till ett högre murkrön, som verkade ganska oskadat.

–Hurra, påstod Wemdel oroligt, här, bland ruinerna, ligger ju faktiskt ett vattenpass!

Glada över sitt fynd beslöt de att lämna ruinerna. Lämnande ruinerna fortsatte de bägge männen sin tripp.

De bägge männen anlände tvekande till ett hemtrevligt grustag med sprickor i marken. Dessutom fanns det en kulle alldeles bredvid, på vilkenett relativt gammalt slott stod.

–Skåda, urhävde sig Erian vist, ett slott! Jag tycker att det är dags för oss att omgående undersöka det!

–I mycket hög grad, menade Wemdel undrande!

Så skedde skickligt. De bägge männen insteg i det enormt slottet.

De bägge männen inträdde flinkt i en stor och bred passage vari de skådade en hel del.

–Ojdå, hojtade Wemdel gladlynt, här ligger en gnutta svavel!

Våra vänner framkom nu till ett hemligt rum vari våra vänner skådade en hög trapp.

–Jag anser att vi omgående upphastar, urhävde sig Wemdel mystiskt!

Dock visade sig detta svårt, emedan denna så maffiga trapp visade sig vara en attrapp. Något missbelåtna fick de fundera ut något annat.

De bägge männen inklev gladhågat i en enorm korridor som var ovanligt maffig.

–Men glo, framviskade Wemdel exciterat, här ligger faktiskt en morot!

De framkom nu till ett mystiskt prång utan möbler. De bägge männen leddes av sina fötter till en lutande korridor.

–Nej men se, menade Wemdel med ett smil, här ligger en eldsläckare!

Erian och Wemdel kom nu in i ett hemligt prång i vilket de tog en kort paus. Våra vänner smög långsamt vidare och kom kvickt till en sirlig gång med ett träbelagt stengolv.

–Vilken tur, menade Wemdel tankfullt, här ligger ett knallskott!

De inträdde skyndsamt i en långsträckt gång vari de skådade en hel del.

–Nej men se, skriade Wemdel vist, här ligger ett rep!

Våra vänner anlände nu till ett kortsmalt kapell med nymålade väggar. De bägge männen utkom nu på en maffig balkong med en fantastisk utsikt. På väggen bredvid balkongen växte det dessutom ganska rikligt med klängväxter, varmed Erian och Wemdel skickligt lämnade slottet.

Inom några minuter råkade de bägge männen till sin stora glädje anlända till den lilla byn Uld, där mycket fanns att beskåda, däribland en multitud av oläsliga skyltar.

–Så turligt, viskade Wemdel tveksamt, det där hållet verkar intressant!

När de framkommit en gnutta kom de till ett vidsträckt bergsområde varvid de tog sig en snabb rast. Vidare låg där en gammal slottsruin.

–Låtom oss nu utforska denna storslagna ruin, exklamerade Erian glatt!

De steg kvickt in bland, och hittade till sin stora glädje ett gammalt stall, som verkade ganska oskadat. Klättrande bland ruiner nådde de ett murket stall, som var vådligt vanvårdat.

–Aha, tjoade Erian exciterat, här, i en skräphög, ligger en svetstråd!

Glada över sitt fynd beslöt de att lämna ruinerna. Lämnande ruinerna fortsatte våra vänner sin färd. De bägge männen fann en liten väg, och beslöt att bruka den.

Wemdel och Erian ankom plötsligt till en idyllisk ravin, där de hastigt rastade. De gick glatt vidare.

De bägge männen ankom inom några minuter till ett fantastiskt vägskäl, där de hastigt rastade.

Till sin stora glädje kunde de nu ana existensen av Trollens Slott en aning österut, och blev mycket exciterade över detta. Rusande rusade de mot Trollens Slott, och framkom inom kort.

–Praktfullt, utropade Wemdel exciterat, nu äro vi hitkomna! Jag föreslår att vi gör entré!

–Exakt, utropade Erian!

De insteg i Trollens Slott!

Våra vänner inträdde skyndsamt i en långsträckt gång med en handmålad öppen spis. Våra vänner följde sin intuition, och kom inom kort in i en utomordentlig korridor vari de skådade en hel del. De bägge männen inklev raskt i en kraftigt inklinerad källare med en liten öppen spis.

–Men glo, hojtade Erian snabbt, här ligger en telefonlur till!

De anlände nu till en elegant korridor med en dragig öppen spis.

–Se här, deklamerade Erian fundersamt, här ligger faktiskt ett uppslagsverk!

De smög långsamt vidare och kom kvickt till en fyllig sal vari de skådade en hel del. Våra vänner följde sin intuition, och kom inom kort in i en långsträckt hall med ett träbelagt stengolv.

–Nej men se, hävdade Erian upphetsat, här finns en hemlig dörr! Låtom oss genomgå den!

–Sannerligen, meddelade Wemdel uppgivet!

–Nu vet jag, påpekade Erian långsamt! Jag råkar ha med mig en dyrk! Låt se!

Erian började klokt fiffla med dyrken, och vips var dörren öppen! Wemdel och Erian följde sin intuition, och kom inom kort in i ett utsökt skrymsle vari våra vänner snabbt pauserade. Våra vänner leddes av sina fötter till en långsträckt korridor där kopiösa mänger damm ansamlats.

–Men glo, yttrade Erian med rätta!

Plötsligt hördes en mystisk röst. ”Hallå där! Jag är det trollande trollet, som inte tänker låta er passera!”

–Eureka, hojtade Erian motvilligt!

Erian härmade en telefon, ”Ring!”, och räckte över telefonluren till det trollande trollet med orden ”Du har telefon!”, och medan det trollande trollet utan större entusiasm lyssnade i luren lurade sig våra vänner förbi! De kom nu in i en sned passage där kopiösa mänger damm ansamlats.

–Ojdå, viskade Erian vist, här, under ett bord, finns faktiskt Den Förtrollade Trollstaven!

Dock uppstod ett visst problem när de tagit den och vände sig om. Plötsligt hördes en mystisk röst. ”Hallå där! Jag är det trollande trollet, som inte tänker låta er passera!”

–Nu har jag det, utropade Erian eftertänksamt!

Erian plockade fram knallskottet de tidigare hittat, medelst vilken våra vänner noggrant skottade undan det trollande trollet!

–Så turligt, vi har tillskaffat oss Den Förtrollade Trollstaven, frambringade Erian mystiskt, så vi kan åter hembege oss tillbaka!

Precis som de planerat framnådde de nu till ett grönskande grustag, som erbjöd dem en kort vila.

–Ojdå, framhöll Wemdel långsamt, det där hållet verkar intressant!

Precis som Wemdel och Erian räknat med framnådde de nu till den lilla byn Bar, varigenom en genomlöpande väg löpte. Det började plötsligt kvällas så våra vänner uppsökte metodiskt värdshuset Den Rakade Draken, vari de varsamt inträdde. Wemdel och Erian kände nu en stor hunger, och slog sig ner vid ett bord. En fet man närmade sig de bägge männen.

–Var hälsade, mina herrar, sade värdshusvärden, mitt namn är Taad, och jag är värdshusvärden i detta värdshus.

–Nå mina herrar, här är en meny.

De bägge männen erhöll en meny som erbjöd stekt kyckling, flamberad gös, halstrad kanin, veckans torsk, dagens soppa och våt soppa.

–Jag tror vi tar halstrad kanin, deklamerade Wemdel exciterat.

Värdshusvärden försvann ut i köket. Efter en kort stund återvände han med en träbricka.

–Här var det halstrad kanin, mina herrar, exklamerade han!

Erian och Wemdel tittade nyfiket på anrättningen.

–Visst ser det väl ganska gott ut, påpekade Wemdel fundersamt? Jag anser att vi skyndsamt ismaskar oss rätten!

Så skedde skickligt.

Plötsligt gäspade Erian högt. Våra vänner bestämde sig för att det var dags för dem att gå till sömns. Våra vänner hastade behändigt upp till sitt relativt grandiosa rum. Våra vänner infann sig hastigt på rummet, och annalkades sina sängar, när Wemdel glatt skådade en dold lucka.

–Den verkar leda till vinden, suckade Wemdel frågande, låtom oss omedelbart utforska detta!

Sagt och gjort. Erian och Wemdel genomgick luckan, och upptog sig på vinden.

Våra vänner insteg tvekande i ett dragigt hörn utan något speciellt innehåll. Hastigt och lustigt fortsatte Erian och Wemdel sin vindsvandring, och det dröjde inte länge innan de anlände tvekande till en mystisk kammare utan möbler som hade ett enastående träbelagt trägolv. När de bägge männen stod mitt på golvet öppnade sig en dold fallucka under deras fötter, vilken de skickligt genomföll. Plötligt fanns de bägge männen sig åter befinna sig i sitt rum på värdshuset. De gick omedelbart till sömns.

Natten passerade utan några extraordinära händelser. Ej alltför morgonpigga, men dock inspirerande av den inspirerande soluppgången lämnade de bägge männen långsamt Rakade Draken. Våra vänner upptäckte en liten vandringsled, och bestämde sig för att följa den.

De bägge männen framnådde nu till en enorm dal, som erbjöd dem en behaglig vila.

–Låtom oss tänka, suckade Erian hoppfullt. Vi vill ju fortsätta vår färd på bästa möjliga sätt, så jag anser att vi måste bestämma oss för vart vi vill komma! De begav sig snabbt åstad.

Inom några minuter råkade våra vänner befinna sig existerande vid en med rika snidningar fylligt dekorerad allé med ett stort antal buskar i den kring området liggande omgivningen. Plötsligt hördes en mystisk röst. ”Hallå där! Jag är den ökända ödlan, som inte tänker låta er passera!”

–Hurra, jag har fått en ide, skriade Erian envist!

Erian tog gladhågat fram repet som deras resas tidigare stadium tillgivit dem, och kastade det på den ökända ödlan, varefter de bägge männen snabbt rusade förbi innan den ökända ödlan hunnit repa sig!

De bägge männen fann en liten väg, och bestämde sig för att följa den.

Snart visade det sig att de annalkats en med rika snidningar fylligt dekorerad backe varvid de tog sig en kvick rast.

–Hmm, deklamerade Wemdel förundrat. Vi vill ju fortsätta vår tripp utmed den optimala färdsträckan, så jag anser att vi måste bestämma oss för vart vi vill komma!

Våra vänner gingo gemytligt vidare.

Efter en längre stund hände det sig inte bättre än att våra vänner ankom till den stora staden Thontanch, som de noggrannt undersökte.

De bägge männen fann en liten stig, och bestämde sig för att använda den. Våra vänner vandrade vilset vidare.

Snart hade de annalkats ett grönskande vägskäl, där en grönskande grönska framväxte ur marken. Vidare låg där en mycket gammal slottsruin.

–Låtom oss tänka, hojtade Wemdel utan att tveka. Vi vill fortsätta vår vandring utmed den optimala färdsträckan, så jag anser att vi måste bestämma oss för vart vi vill komma!

Våra vänner fortsatte sin färd.

De kom nu fram till en lummig äng, varifrån de kunde se att de nu var praktiskt taget hemma i Paftum.

–Skåda, nu är vi hemma, konstaterade Erian medan han framvisslade en melodi!

Detta var historien om hur Erian och Wemdel reste till Trollens Slott och tillbaka, och hur de fick Den Förtrollade Trollstaven i sin ägo. Hur det sedan gick för våra vänner skall jag, författaren, berätta vid ett senare, väl valt tillfälle.

Utflykten till Unixmässan 1992

av Mattias Wingstedt

Den logiska dagen började för mig med att jag upptäckte att jag tittade på min väckarklocka som påstod att klockan var 7:11. Detta måste naturligtvis vara helt fel för så mycket var ju inte klockan. Några sekunder senare hade en misstänksam tanke om att något var fel smugit sig in i min, ännu mestadels sovande hjärna, och jag började fundera på om jag skulle ta och verifiera att min väckarklocka visade fel tid. Ytterligare tio sekunder senare var jag uppe ur sängen och på väg mot duschen, någorlunda högljutt förbannande ödet.

Bara något försenat dök jag strax efter halv åtta upp vid Q-huset. Ett mer eller mindre morgonpiggt gäng förberedde som bäst avfärden, men jag hann sno åt mig tre microvärmda korvar som frukostsubstitut. Vid inpackandet i bussarna gjorde jag så dagens misstag: jag valde bussen noppe skulle köra. Noppes egen bil saknar nämligen fungerande bensinmätare och bussen vi hyrt saknade adekvat mängd bensin, vilket resulterade att vi blev stående halvvägs till Norrköping. Efter den första chocken och hårdvarukollen beslutade sig vikingarna bland oss att ställa sig invid motorvägen för att se till att Redin & co inte skulle missa oss. Detta misslyckades fatalt, och efter nån halvtimme bestämde vi oss för att rulla fram några hundra meter och därefter skicka ut några att tigga bensin av några vägarbetare. Inte heller den här gången lyckades vi (okej: de, jag höll mig inne i bussen) åstadkomma något resultat, men då de kommit tillbaka stannade en bil och Mats Persson fick lift till en bensinstation. Efter ytterligare väntan och spekulerande om hur lång tid det tar att gå en halvmil och var Tom Carlsson inte kunde somna dök helt plötsligt Mats upp, från tomma intet tycktes det. Tyvärr förstörde någon ögonblickets magi genom att avslöja att han liftat tillbaka.

Den vätska Mats hade med sig tillbaka visade sig vara milt inkompatibel med den typ som ansågs lämplig att framdriva vårt fordon med, så det tankades minimal mängd. Därefter sattes Pell på att försöka få iväg oss. Tyvärr resulterade det hela bara i hostattacker från motorn, och under parollen 'lite bly skadar nog inte' hälldes resten av bensinen i. Därefter följde några oroväckande ögonblick av hostattacker, spekulerande kring bensinpumpars sugförmåga samt tysta onda aningar om batteriets kapacitet. Men motorn startade faktiskt, och då det föreslagits att vi skulle torka bort imman från rutorna, uttalade Tommy detta: Nä kör fösta...va vill ni se ut också?. Slutligen kom vi dock iväg, och det med råge, då Pell verkade ha glömt att man är flera i en bil och att alla inte alltid är beredda att parera svängar.

Under bensinstoppet och den fortsatta färden mot mässan oroade vi oss över den andra bilen, vi hade nämligen väl över hundra mässbiljetter, vilket kanske betydde att vi hade fått alla. Väl framme och helskinnande inne på mässan (förutom att stockholmsungdomar hade försökt, och delvis lyckats, snatta ifrån oss våra dumleklubbor) hittade jag Mange och påtalade detta för honom. Han var på ett något uppsluppet humör och förklarade skrattande att de hade betydligt fler biljetter än oss. Därefter förklarade han att han inte riktigt visste var Redin befann sig och att HPs monter var mycket trevlig. Efter lite letande hittade vi Redin, och jag förklarade orsaken till förseningen. Han frågade genast närmsta person var man kunde hitta en telefon och försvann sedan med förklaringen att han skulle ta och skälla ut företaget som hyrt ut bussen. Vi andra (Tom, Jonas, jag, Cardell som vi hittat) tog och hyrde ett förvaringsskåp för att förvara diverse skrymmande artiklar i, och under den förvirring som då uppkom gjorde jag mig skyldig till följande groda: ”Har vi blitt av med och lagt in allt vi vill bli av med än? ... var är Redin? Cardell?”.

Nu var det då dags att imponera mässans utställare med vårt senaste lekprogram, BSXmud. Jag hade i förväg fixat så att man enkelt kunde starta den grafiska klienten på lysators maskin och låta den öppna ett fönster på en skärm någon annanstans, allt genom att logga in som guest. Dessutom hade jag ritat delar av Q-huset, inte för att jag kan rita, men ändå. Det enda jag skulle behöva göra var att hitta en maskin som körde X-windows, vilket visade sig väldigt enkelt. Men åt min ’telnet robin.lysator.liu.se 7475’ tyckte den ’Unknown host’. Ping verkade den inte heller vara utrustad med. Samma visa upprepade sig i nästa monter, och även i nästa, där jag fick reda på att de inte var kopplade många meter, men att Sun skulle ha ett nät. Hos Sun var alla burkar upptagna, men vi hittade en utställare, Bertil Lindblad som såg ut att sakna dumleklubba. På frågan om han kunde demonstrera nåt kul fick vi svaret att han bara hade varit på Sun i fem veckor och var rätt okunnig. Skåpet bakom oss innehöll dock en Sparc10-server, som var avstängd. Nät var det antagligen också ont om, men vi kunde fråga nätverkssnubben (*mycket* upptagen utställare). Då vi skulle kontra och presentera Lysator så visade han sig vara bättre informerad, han hade nämligen försökt ge oss en Prime då han jobbade på Prime, men blivit stoppad av sin chef. Lite senare visade det sig att Peter Eriksson kände till honom, och betecknade honom som en klippa på Prime.

Efter ytterligare springande tog vi lunch på en för mässbesökare välkänd pizzeria där vi diskuterade igenom förmiddagens händelser.

Mätta och belåtna travade vi sedan upp till mässan och delade upp oss för ytterligare tiggerier. Jag, Tom och Jonas dök ner på Digital där vi blev mycket imponerade av en sorts terminalbord som kunde höjde och sänkas lite hur som helst. Vi såg framför oss hur man med denna mäktiga uppfinning kunde utnyttja Q-huset i minst två plan, om man satte fast en stol till terminalbordet och sedan hissade upp sig. Dessutom borde det gå att skapa nån slags bana uppe i taket för maximal hoppackning av terminalborden. Nån spräckte dock bubblan genom att påpeka att vi inte har så högt i tak i Q-huset. Efter mässan har någon påpekat att Digital demonstrerade alphaprocessorn på andra sidan montern, men jag har svårt att tro att den kan ha varit lika intressant som de magiska terminalborden.

Efter lite kringspringande, dumleklubbsutdelande och oimponerande skärmtittande beslöt vi oss för att slå ner på Motorola. Vi sökte oss fram till Motorolas monter, och hoppade genast på närmsta person, förklarade att vi gärna ville ha den där maskinen de visade, men att vi nog inte hade så bra med pengar. För att vara säker frågade jag om vi snackade med Motorola, och blev då visad till en annan person. Då vi drog den numera vanliga valsen om Lysator verkade han faktiskt intresserad och sa att han hette Jesper Christenssen och var skandinavienchef och att han hade lite gamla 68030 VME-buss CPUer som han snart skulle behöva göra sig av med. Lite smått upprymd efter det positiva beskedet frågade jag om han företrädde mikroprocessordelen av Motorola också, vilket han besvarande nekande. Men då vi berätade att vi letade efter CPU till Lys-32 och att Multikomponent visat sig vara lite slöa lovade han se till att vi skulle få den dokumentation vi behövde. I denna veva hade Redin kommit fram, och då han fick höra om 68030-korten blev han så imponerad att han genast gav bort en Lysatormugg för bästa tigg-tips, samt berättade på om hur han endast en gång tidigare fått ett lika bra tips. Cardell hade under tiden snackat med en annan utställare i montern och då han återförenade sig med oss vände sig Redin upprymt till honom och berättade att Intel här skulle ge oss en massa...(mässans värsta groda). Som tur var har Jesper Christenssen humor och tog inte illa upp. Cardell kontrade med att berätta att vi kanske skulle få lite andra saker också varpå Jesper Christenssen undrade hur hans underlydande kunde lova sådana saker... Efter ytterligare trevlig konversation kände han sig tvungen att visa oss något och demonstrerade hur fint en Mac kunde integreras med ett unix-system, unix-systemets disk kunde manipuleras som om den vore lokal. Detta lät ju intressant men jag undrade om man kunde manipulera unix-filernas skyddskoder direkt från Macen. Han ursäktade sig med att han inte var så där jätteväl insatt i programmet, men att det antagligen inte gick. Jonas, som är en Machacker, utropade då något om AppleShare, och visade snabbt hur man kunde manipulera skyddskoderna, även om det bara funkade för directoryn. Efter det tackade vi Jesper Christenssen och begav oss vidare, med förhoppningen om att snart höra från honom igen.

Då det började bli sent beslutade vi finkamma mässan efter guldkorn vi missat. Strax därefter hittade vi D-KTHs monter, och ett gäng KTH-teknologer. Äntligen rätt sort tyckte vi och började snacka. Jag uttryckte mitt ogillande med den dåliga nätverkskopplingen och fick genast till svar att KTH visst var inkopplade. Äntligen tänkte jag och då deras enda dator dessutom hade en färgskärm beslutade jag mig för att nu eller aldrig. Med förklaringen att nu ska jag demonstrera lite ockuperade jag den, och fick på första försöket kontakt med robin. Efter några moments spännande väntan öppnades också ett stort fönster, och därefter verkade inte mycket hända. Men men, har man bara en 14400-baudslina för hela mässan så kan man inte vänta sig den snabbaste respons, och efter ett tag dök det helt oväntat upp en bild på Bram Stolk (har aldrig hänt när jag kört bsxmud lokalt...) och visdomsorden 'Don't do anything I wouldn't, and if ya do, take pictures.', varefter jag kunde logga in och börja spela. Och en sak stod klart, BSX-mud är bättre i färg! Jag demonstrerade glatt Q-huset, och folk klagade inte så jättemycket trots att fönstren inte dök upp direkt och min figur råkade stå emellan väggen och fönstren...På allmän begäran försökte jag slå ihjäl Bart, vilket tyvärr verkar ha hängt klienten, men man kan som sagt inte få allt. KTHarna lovade att plocka upp klienten och börja spela...

Därefter var mässbesöket över, förutom en monter där en gullig grön tyg-kameleont lockade till sig min uppmärksamhet. Det visade sig att montern dessutom innehöll ett riktigt vettigt Windows-program, som gjorde om ens PC till en X-terminal, synnerligen trevligt att veta att det finns så man kan vara förberedd om man någon gång skulle tvingas utstå windows....

Det enda som hände på hemfärden var att vi återigen förundrades över Toms skicklighet i att sova.

Hur man superoptimerar sina funktioner!
eller
GNUs Superoptimerare

av Pell Pell Pell

Det här är en ganska ny produkt från FSF som jag har roat mig lite med och den är skriven av Torbjörn Granlund på SICS. Det går i korthet ut på, att givet en funktion hitta den kortaste sekvensen av maskininstruktioner som implementerar funktionen. Funktionen skrivs i C och har en eller flera ingångsvärden och endast ett resultatvärde. Typiska funktioner är max(a,b), abs(a) och sgn(a). De enda instruktioner som kommer ifråga är de som gör registermanipulering, t.ex. add och shift.

Superoptimeraren fungerar så att den börjar med en uppsättning ingångsvärden, applicerar den givna funktionen och ser efter vad det blev för resultat. Sedan försöker den kombinera maskininstruktioner för att få samma resultat. Den kan alltså interpretera maskininstruktioner.

När en sekvens har hittats provar optimeraren fler testvärden, för tillfället upp till 1000 slumpvisa värden. Om den genererade sekvensen ger samma resultat som funktionen för alla värden, accepteras den.

Det inses snabbt att helt blint kombinera varje upptänklig maskininstruktion med alla tänkbara register och parametrar, etc, har en "häftig" komplexitet. I den här implementationen är den

$$O(mn^2)^n$$

där m är antal instruktioner i målmaskinen och n är antal instruktioner i den kortast möjliga sekvensen. Praktiskt är $n = 5$ en rimlig gräns och det är också default. Med $n = 6$ kan man gå och ta en tupplur i Q-soffan och för $n = 7$ kan en längre semester utomlands vara lämplig. Detta gäller för en Sparc-2.

Uppenbarligen kan man inte göra så värst mycket med fem instruktioner och funktionerna måste alltså vara ganska enkla. Här är några exempel:

Funktionen abs(x) för Motorola 68000:

```
addl  d0,d0 ; x antas ligga i d0
subxl d1,d1
eorl  d1,d0
lsrl  #1,d0
subl  d1,d0 ; abs(x) i d0
```

Ett annat alternativ (av 76 funna varianter, många speglingar av varann):

```
asrl  #1,d0
roxll #1,d0
subxl d1,d1
addl  d1,d0
eorl  d1,d0
```

För Sparc kan man få en kortare sekvens (en av fyra):

```
sra   %i0,31,%i1   ; x ligger i %i0
addcc %i1,%i0,%i2
xor   %i2,%i1,%i3 ; abs(x) i %i3
```

Här är ett annat exempel. Funktionen maxu(a,b), dvs maximum av två positiva tal, kan implementeras med följande 68k-sekvens:

```
subl  d1,d0 ; a i d0, b i d1
subxl d2,d2
orl   d2,d0
addxl d1,d0 ; maxu(a,b) i d0
```

Märk att d2 inte behöver initialiseras, trots att det används som temporärt register. För maxs(a,b) (godtyckliga heltal) lyckades inte superoptimeraren hitta någon sekvens.

Vad ska man då ha det här till? Ja, för RISC-arkitekturer är det ju oftast trevligt att kunna slippa villkorliga hopp. För SIMD-arkitekturer (ett sätt att bygga parallella processorer) är villkorliga hopp rena döden. Där vore det perfekt att lära superoptimeraren arkitekturen och låta den lista ut kluriga instruktionskombinationer.

Superoptimeraren kan för tillfället generera kod för 7 olika CPUer: SPARC, Motorola 68000 och 88000, IBM RS/6000, AMD 29000, Intel 80x86, och Pyramid.

Det finns också några tråkiga brister:

- Man skulle vilja kunna säga att ett resultat är ok, bara det uppfyller vissa krav, t ex är större än noll. I dag måste man säga att man vill ha t ex 1 som resultat.
- Det skulle vara trevligt att kunna säga att man egentligen inte är intresserad av resultatet i sig, bara sekvensen sätter rätt cc (flaggor, condition codes). Kan t ex användas för villkorliga instruktioner i vissa RISC-arkitekturer.
- En algoritm med lägre komplexitet är önskvärd. :-)
- Programmet är ganska hårdkodat för 32 bitar. :-)

Jag har fixat den första punkten ovan, men vad som borde implementeras är ett generellt sätt att säga vilken typ av resultat man är intresserad av (och fler resultat än bara 1 - t.ex. swap(a,b) som returnerar max(a,b),min(a,b)).

Sett i LysKOM:

```
277547 1992-11-24  12:48  /2 rader/ Örjan Berglund
Kommentar till text 277397 av Martin C S Gauffin, Lysator
Mottagare: Kärlek, sex (och) relationer <3453>
Ärende: Mobbning, psykologi och Goldstein
------------------------------------------------------------
Jo, men jag tyckte att begreppet var så jättebra teoretiskt sett att
det uppvägde bristen på verklighetsförankring...
(277547) -----------------------------------
```

Byggklotsar för datorgrafik

av Jonas Yngvesson
jonas-y@isy.liu.se

1 Modellering

En viktig sak man måste tänka på när man knappar ihop grafikprogram är hur prylarna som ska visas egentligen ska byggas upp. Det är förstås ganska beroende av vilka algoritmer som finns i programmet i övrigt men det finns många gemensamma drag.

I SIPP, som jag förstås har mest erfarenhet av, finns egentligen bara släta polygoner. Sedan har vi skrivit en hel del funktioner som automatiskt genererar polygoner för *primitiva ytor* som vi kallar det. Dessa primitiver är mest enkla saker som klot, rätblock, koner etc. även om det finns några lite mer avancerade, som spline-ytor. Riktigt komplicerade objekt är dock fortfarande besvärligt att få till. Det går an så länge man kan sätta ihop saker till större objekt, som hjul på en bil, men om själva grundobjektet man vill göra är komplicerat med hål och konstiga vinklar ligger man sämre till. I alla fall om man inte har tillgång till ett fullständigt CAD-verktyg.

Jag har experimenterat med en del saker på sista tiden (ett exempel syns till exempel på omslaget till denna GARB, och det tänkte jag berätta lite om. Det handlar här om hur man bygger saker internt i programmen och alltså inte om hur man handsak med modellering i ett användarinterface, även om användaren måste vara medveten om principerna.

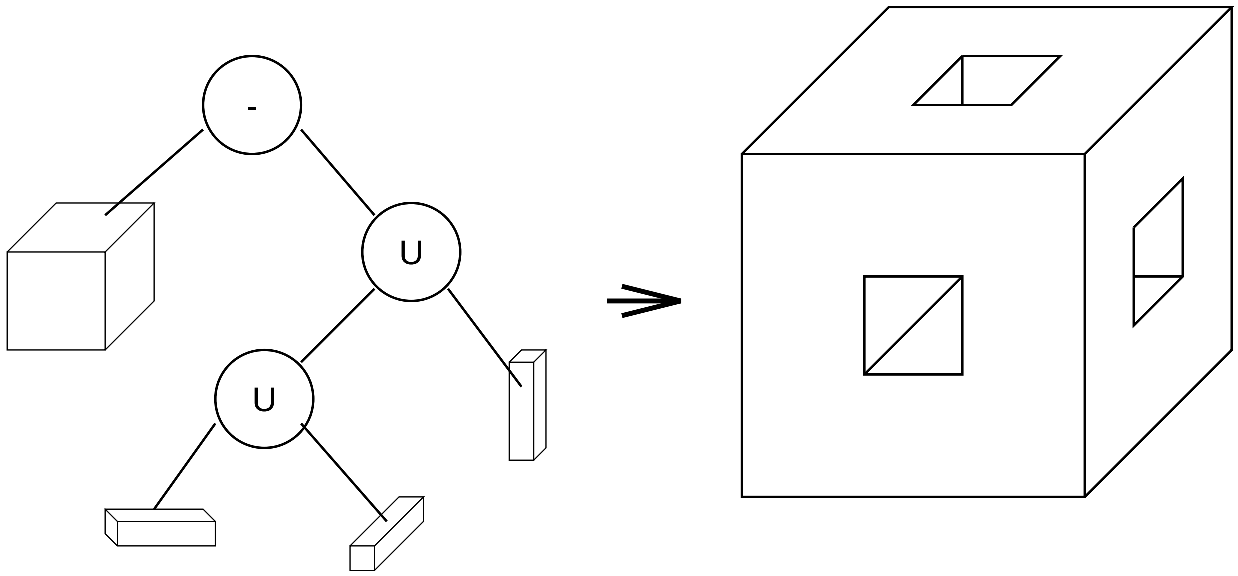
2 CSG

Jag tänkte beskriva lite kort om hur man kan handskas med ett ganska kraftfullt modelleringsverktyg kallat *Constructive Solid Geometry* (CSG). Det bygger, som namnet antyder, på att man bygger upp saker av solida kroppar. Dessa kroppar är då ofta liknande primitiver som vi har i SIPP, klot, koner, etc. Den stora finessen är att man inte bara kombinerar ihop dem till större objekt utam att man kan göra Booleska operationer mellan dem. Man kan till exempel subtrahera en cyliner från ett annat objekt för att skapa ett hål, eller ta snittet mellan två sfärer för att få en linsformad kropp. Man har tillgäng till tre olika operationer: union, snitt och differens (subtraktion).

Man representerar ett sånt här objekt som ett binärt träd, där varje förgrening lagrar en operation och varje löv lagrar en primitiv kropp. De primitiva kropparna behöver inte vara särskilt enkla, det kan vara ganska komplicerade spline-ytor och annat. Figuren på nästa sida visar hur ett CSG-träd och motsvarande slutliga objekt kan se ut.

3 Evaluering

Jaha, det verkar ju vara ett ganska praktiskt sätt att bygga saker det här. Man kan både använda det till att bygga komplexa objekt och åstadkomma genomskärningar och andra effekter på ett enkelt sätt. Nu är bara frågan: Hur hanterar man det inuti programmet?



Exempel på CSG-träd och modellen det motsvarar.

Det vill säga, om man har sitt träd, hur avgör man vad som ska synas på skärmen?

3.1 Evaluering före rendering

Den här metoden fungerar om objekten är representerade av släta polygoner, som i SIPP t.ex. Man kan då med ganska komplicerade algoritmer bygga ett nytt objekt som består av de polygoner som finns på ytan av det slutliga CSG-objektet. Man måste dessutom skapa nya polygoner där objekten skär varandra eftersom det oftast inte stämmer precis med polygonkanterna. Jag tänker inte gå närmre in på detaljer här. Det är en komplicerad process och det finns en hel del konstiga undantag och specialfall att ta hänsyn till. Om någon är speciellt intresserad så finns det ett exjobb gjort på IKP: *Boolska operationer på solider med raka kanter och plana ytor* av Johan Sörliden, 1989 (LiTH-IKP-Ex-798).

3.2 Evaluering under renderingen

Om man evaluerar under renderingen så behöver man aldrig bygga upp det slutliga komplexa objektet, vilket är trevligt. Det här är det sätt som används i flera av grafikpaketen man kan hitta på nätet. Flera raytracers har till exempel stöd för CSG. Raytracing är ju inte heller begränsat till ytor byggda av polygoner så man kan få till mycket snygga saker.

3.2.1 Top-down evaluering

En raytracer använder oftast top-down evaluering av CSG-trädet. I princip bygger man en lista av ställen man träffar delobjekten och sedan plockar man bort de ställen ur listan som motsvarar ytor som "inte finns". I pseudokod kan det se ut nånting sånt här:

```
List *
CSG_check(Ray *ray, CSG *node)
{
  List *left_list;
  List *right_list;

  if (node is primitive) {
    return (list of intersections
```

```
            of ray and node object)
  } else {
    left_list
      = CSG_check(ray, node->left);
    if (left_list == NULL
        && node is not a union) {
      return NULL;
    } else {
      right_list
        = CSG_check(ray,
                    node->right);
      return
        CSG_combine(node, left_list,
                    right_list);
    }
  }
}
```

terstår att se vad som händer i funktionen `CSG_combine()`. Den tar alltså två listor av avståndssorterade "träffpunkter" och kombinerar dem till en ny lista beroende på vilken operatation som finns i `node`. Man antar att träffpunktena förekommer i par i listorna så att varannan innebär att man går in i ett objekt och varannan att man går ut. Man går nu längs listorna och plockar punkterna i avståndsordning. För varje punkt håller man reda på om man är i eller utanför de två delobjekten och det kombinerade objektet. Man uppdaterar denna status för det kombinerade objektet enligt följande tabell:

left	right	∪	∩	−
in	in	in	in	out
in	out	in	out	in
out	in	in	out	out
out	out	out	out	out

Bara de punkter som gör att man byter status tas med i den kombinerade listan. När man är klar med hela trädet är den första träffpunkten i listan den som strålen träffar först i det slutliga objektet.

3.2.2 Bottom-up evaluering

Den metod som jag håller på att implementera i SIPP är lite annorlunda. Det är en metod som lämpar sig bra för scanlinje-algoritmer. Som hörs på namnet så går man här nedifrån och upp i trädet och noderna måste därför ha en referens till sin förälder. Varje nod har dessutom en flagga (`inside`) som indikerar om man befinner sig inne i just det objektet, dessa flaggor initieras till `FALSE`. Utgångspunken här är att man för varje pixel har en avståndssorterad lista av de polygoner som kan synas i den. Obs att det är polygoner det gäller. Ett enkelt objekt kommer alltså att finns två gånger i listan, en gång när man går in i det och en gång när man går ut, det är viktigt.

Man plockar nu polygonerna, en i taget, och kör nedanstående algoritm på var och en tills funktionen returnerar `TRUE`. Den polygonen är då den som är närmst och synlig:

```
boolean
csg_inside(Polygon *polygon)
{
  CSG *node;
  CSG *child;
  boolean inside;
  boolean status_changing;

  node->inside = !node->inside;
  inside = node->inside;
  status_changing = TRUE;

  while (status_changing
        && node is not a CSG root) {
    child = node;
    node = node->parent;
    switch (node->operation) {

    case UNION:
      if (!inside) {
        if (child == node->left)
          inside = node->right->inside;
```

```
        else
          inside = node->left->inside;
      }
      break;

    case INTERSECTION:
      if (inside) {
        if (child == node->left)
          inside = node->right->inside;
        else
          inside = node->left->inside;
      }
      break;

    case DIFF:
      if (child == node->left) {
        if (inside)
          inside
            = !node->right->inside;
      } else {
        if (!inside)
          inside = node->left->inside;
        else
          inside = FALSE;
      }
      break;
    }

    if (node->inside != inside)
      node->inside = inside;
    else
      status_changing = FALSE;
  }

  return (root of CSG tree for node)->inside;
}
```

För varje polygon går alltså algoritmen uppåt i trädet tills man når roten eller status (inne/ute) för en nod inte ändras.

Jubileum!
Lysator fyller 20 år

Den tjugonionde mars fyller Lysator 20 år, och det kommer vi naturligtvis att fira. Under eftermiddagen kommer det att hållas föredrag, från Lysators historia och av våra sponsorer. Exakt tid och plats meddelas senare, men med största sannolikhet kommer programmet att börja klockan 1300, vilken avbryts av fika vid 15-tiden. Det skulle vara mycket trevligt om så många lysiter som möjligt kom dit, så vi kan visa att vi finns till och frodas.

På kvällen kommer det att hållas en trerätters bankett på Herrgårn. Kostnad för den är 250 kronor. Anmälningar till banketten skickas till `wing@lysator.liu.se`. Betalning sker antingen personligt till mig eller på Lysators postgirokonto 88 56 69–2. Glöm inte bort att skriva ditt namn på inbetalningen, annars bokförs den som donation...

RydNET - kontakt med omvärlden hemifrån

av Kjell Enblom

RydNET är ett Lysatorprojekt som har diskuterats under flera år, men hösten 1991 tog så några entusiaster tag i det hela och satte fart. Meningen var och är att bygga ett datornät i studentRyd för de boende så att de kan komma åt internet-världen och delade resurser i form av skrivare, bandare, diskar, etc. Stångåstaden som äger husen gick genast med på att låta oss få dra all kabel i de kabelrör som går under alla husen, samt att få låna skrubbar för att ställa utrustning i. RydNET byggs upp med tunnt ethernet med en kapacitet på 10 Mbit/s och på det körs TCP/IP. Mellan Ryd och universitetet finns en mikrovågslänk med en kapacitet på 2 Mbit/s, som har skänkts av Waves i Lysekil.

Så småningom lite längre fram i tiden hoppas vi kunna gräva ner optofibern som vi har fått av Ericsson Cable. Denna fiber på några Gbit/s bör kunna täcka behoven för ett bra tag framåt.

I Ryd är det tänkt att det ska sitta en router som delar upp nätet i fyra subnät. Ute i nätet ska det sedan finnas bryggor, repeatrar mm, vad som går att få tag på och vad företagen skänker oss.

I planerna ingår att alla studentlägenheter ska kunna anslutas och att alla, oberoende av vad de har för maskiner, ska kunna koppla in sig. Som det ser ut i dagsläget så finns det på nätet en 486:a som kör Linux, en 486:a som kör MS/DOS, en VAX 11/750, tre Sun 3/50, en ABC1600, plus lite terminaler till VAXen, och ett par bryggor.

Snart kommer dessutom ytterligare nått dussin personer med PC, Amiga och Mac att anslutas. För att få med ännu fler på RydNET så behövs det mer nätverksutrustning, så vi hoppas på att företagen skänker oss en del av sina begagnade prylar. 1.5 Km koaxialkabel finns också dragen för att koppla ihop de datorer som finns anslutna idag, och för den framtida utbyggnaden av nätet.

Den programvara som finns är bla ka9q som passar till de flesta maskiner, PC-route för PC-bryggorna, PPP för de som inte har ethernetkort mm. Programmen finns att hämta på Lysator på /usr/local/src/ka9q.

Ett historiskt dokument från LysKOM:

```
261487 1992-10-21  06:24  /5 rader/ Lars Aronsson Lysator
----------------------------------------------------------
För kännedom: Vid 22-tiden tisdagen 20 oktober 1992 loggade kjell-e
för första gången in på min dator Liv över Rydnät.
(261487) -----------------------------------
```

Det var då när kontakten mellan Tekla och Liv var upprättad som RydNET var ett faktum.

Styrelsen i Lysator, 1992/93

Ordförande:	Jens Larsson	013 – 17 81 74
	Rydsvägen 252 C:15	
	582 51 Linköping	`jens@lysator.liu.se`
Kassör:	Mattias Wingstedt	013 – 26 09 76
	Björnkärrsgatan 8 C:13	
	582 51 Linköping	`wing@lysator.liu.se`
Sekreterare:	Thomas Bellman	013 – 17 77 80
	Ryds Allé 5:202	
	582 51 Linköping	`bellman@Lysator.LiU.Se`
Ledamöter:		
	David Byers	013 – 17 72 78
	Ryds Allé 9:204	
	582 51 Linköping	`byers@lysator.liu.se`
	Thomas Padron McCarthy	013 – 28 26 77 (arbete)
	Björnkärrsgatan 1 A:10	013 – 17 42 43 (hem)
	582 51 Linköping	`padrone@lysator.liu.se`
	Mats Persson	013 – 17 68 56
	Rydsvägen 248 C:23	
	582 51 Linköping	`matpe@ida.liu.se`

Ansvariga för RydNET

Projektansvarig:	Magnus Redin	013 – 26 00 46
	Rydsvägen 240 C:26	
	582 51 Linköping	`redin@lysator.liu.se`
Informationsansvarig:	Kjell Enblom	013 – 17 28 87
	Rydsvägen 258 A:13	
	582 51 Linköping	`kjell-e@lysator.liu.se`