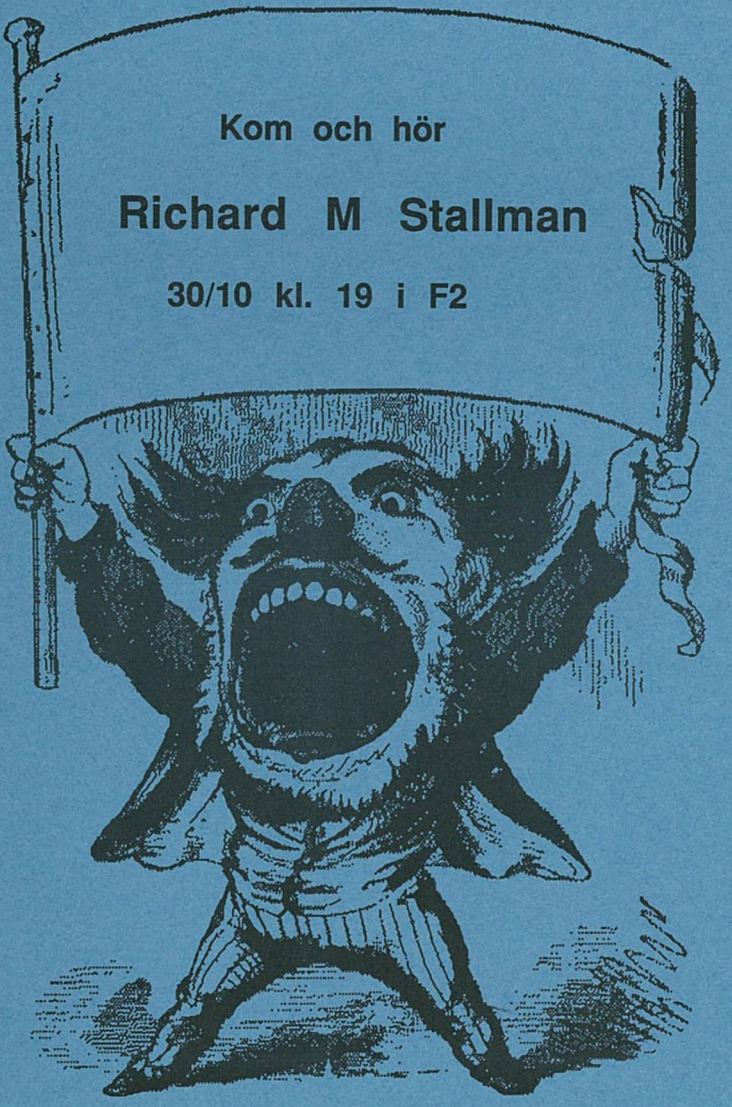
STACKPOINTER

4-1986. Organ för Datorföreningen STACKEN, KTH.



Datorföreningen STACKEN

STACKEN är datorföreningen på KTH. Vi har en mängd intressanta verksamheter på gång. Dock hänger det ytterst på den enskilde medlemmen att avgöra vad han eller hon vill göra för föreningen. STACKEN är en idéell förening, där intresse för datorer är den gemensamma faktorn. Sedan föreningen grundades 1978 har vi (bland annat) åstadkommit:

- samköp av mikrodatorbyggsatser
- diverse kurser och föredrag
- studiebesök på intressanta ställen
- AMIS, den portabla EMACS-kompatibla editorn för TOPS-10, VMS, PRIME, NORD, ...
- STACKPOINTER, föreningens tidning
- en egen datorhall för våra stordatorer
- en egen DEC-10:a (se nedan)

Vi har sedan ett år tillbaka en egen DEC-10, som vi installerat, felsökt och kör på. Det är en gammal modell med KA10-processor. Vi kallar henne KATIA. Hon står i vår nya klubblokal och maskinhall "B30", på Brinellvägen 30 (V-sektionen) på gaveln mot Lill-Jansskogen (där vi har en egen ingång). En våning ovanför finns en hörsal, där vi håller till vid större möten.

Ordinarie möten är första torsdag i varje månad kl 19 i antingen lokalen eller hörsalen. Är Du intresserad av föreningen, är Du välkommen till något av våra möten. Vill Du sedan bli medlem, lämnar Du in en skriftlig ansökan till styrelsen eller skickar den till vår postadress:

Datorföreningen STACKEN
c/o NADA
KTH
100 44 STOCKHOLM

STACKPOINTER

STACKPOINTER är organ för Datorföreningen STACKEN. STACKPOINTER utkommer när material i tillräcklig mängd finns, 4–5 gånger per år.

Redaktör: Hans Nordström.
I redaktionen: Jan Michael Rynning.
Ansvarig utgivare: Mats O Jansson.
Färdigställd: 1986-10-10.

Stack-32, datahårdvara

Idéer och ambitioner runt en 32-bitars dator, gjord för UNIX och GNU. En av grundidéerna runt den här maskinen är att göra den så billig att den i praktiken kan bli var och en av de intresserade STACKEN-medlemmarnas egendom, så en hel del prutningar har gjorts i designen för att få den så billig och enkel som möjligt. Inte heller har så mycket arbete lagts ner på suboptimering. Detta för att ge möjlighet för STACKENs experter på området att göra en storstilad insats.

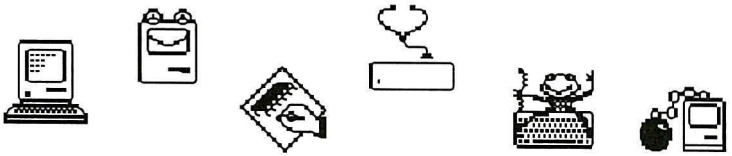
Huvudmaskin är National Semidestructor 32332. Den har en trevlig finess, man kan variera bussbredden dynamisk. Nackdelarna är att den i sedvanlig NS-anda inte fungerar enligt specifikationen. Förutom CPU har maskinen memory management, 512 bytes sidor, floating point, interruptkontroll, SCSI och ethernethårdvara. Minnet är 2 Mbyte ECC med 256 k-chippar och 8 Mbyte med 1 M-chippar. ECC-metoden är check och vid correctable error får CPU:n vänta samt ombesörja återskrivningen av det korrigerade data.

Slavmaskin är en Z80, utrustad med 4 asynkrona och 2 synkrona portar, floppykontroller, kalenderklocka, boot-PROM etc.

Hela datorn ryms på ett enda kort. Den enda expansionsmöjligheten som finns är en kontakt benämd i-bus. I första hand är den tänkt för anslutning av häffans grafikprocessor och en buskonverter mot VME. Från Z80 finns möjlighet att få ut en traditionell Z80-buss. Detta för att man skall kunna koppla in diverse specialkort.

Om någon orkar vore det roligt med en i-bus till IBM PC-bus-konverter, så att man kan använd sådana pereferikort. De är ju trots allt billiga.

Peter Löthberg



Datorer och Musik

Det har väl knappt undgått någon, den snabba utvecklingen inom musikvärlden. Både musikaliskt och ljudmässigt har en hel del hänt de senaste 30 åren. Intressant nog är detta tidsperspektiv detsamma som den kommersiella användningen av datorer. Och vart vill jag nu komma med detta? Jo:

För att börja med datorerna:

De som kunde arbeta med datorer i musikskapandet kunde för 30 år sedan lätt räknas på ena handens finger. Datorer var som ni alla vet, stora, dyra och hyfsat slöa. Trots dessa besvär fick ett antal entusiaster/kompositörer möjlighet att göra musik genom datorerna. I början var musiken uteslutande av s k avantgardistisk form, d v s itererat navelskådande. Sverige hade bl a inrättat EMS, Elektronmusikstudion, på Kungsgatan 8, där några dammiga typer gjorde de första musikaliska batchjobben. Inträdet var (och fortfarande är) i denna klubb lika lätt som det berömda nålsögat ni vet. Jaja, tiden gick och datorutvecklingen med den. I takt med att komponenter blev billigare kunde de första dedikerade maskinerna för syntetisk musik göras, 'synthesizer' på engelska. Från och med de första små 8-bitsprocessorerna, främst Z80, kunde nu också priset på dessa maskiner bli överkomligt för en fattig man/kvinna. Vissa specialiserade ljudchip började också göras, de första enkla med bara en kanal. Så småningom växte dessa till sig och blev hela 'paket' med DA/AD-omvandlare, flera kanaler, filter m m.

Syntarna:

Konsten att framställa ljud på syntetisk väg är äldre än man kanske kan tro. Redan på 30-talet gjordes en hel del försök att m h a oscillatorer och kristaller framställa vågformer för att efterlikna vissa instrument. Det egentliga genombrottet kom dock inte förrän rockmusiken hade börjat tränga igenom på allvar och blivit etablerad. I takt med att musiker började söka efter nya uttrycksformer blev efterfrågan efter syntetiska ljud större. De första effekterna som användes var olika pedaler som t ex 'Fuzz' och 'Wha wha'. I slutet av 60-talet kom en fantastisk maskin som kunde spelas från ett enkelt keyboard. Maskinen hette EMS (ej att förväxla med ovanstående) och bestod av 3 oscillatorer varav en var avsedd för modulering i lågfrekvens. Själv var jag sexton när jag fick syn på denna skapelse för första gången och jag minns den tystnad och stämning av andakt i butiken (Musikbörsen) som blev när maskinen skulle demonstreras. På maskinen fanns ett korskopp-

lingsbord, där de olika funktionerna kopplades ihop med små nålar. Synten kunde antingen köras som den var eller styras med ett plastigt keyboard med påmålade (!) gula och svarta tangenter.

Nåväl, musiken frodades och med den kom fler och fler syntar, den ena häftigare än den andra. I slutet av 70-talet fanns det riktiga monster ute, med massor av oscillatorer, filter och finesser. Detta medförde vissa problem, eftersom knappmängden gjorde att man mer eller mindre fick fotografera av synten för att senare kunna rekonstruera samma ljud. Lösningen på detta blev maskiner med funktioner som kunde 'minnas' en ljudinställning och med en enkel knapptryckning ta fram det senaste ljudet. Den mest omtalade synten med denna möjlighet var Prophet 5, en amerikansk analog synt, polyfon med fem 'röster'.

Så till dagsläget:

Idag har datorer gjort ett otroligt stort insteg i musikvärlden. Vart man än kommer hittar man dem. I syntar, studios, instrument, effekter m m. Idag kan hela kedjan från mikrofonen till din högtalare hemma vara digital. Det enda analoga i kedjan är instrumentet (om det är ett analogt sådant ...) och signalen från förstärkaren till högtalarna. Resten är 0 och 1.

Musikeran kan idag klassas som tiden EM, alltså tiden efter MIDI. MIDI är en standard för överföring av signaler mellan syntesizers som har kommit till stånd runt 1982. Denna standard är dock inte begränsad till syntar, utan används också i mixerbord, gitarrer, ekon, m m.

Hur det går till idag kan visas med en inspelningssituation:

Ett rockband kommer in i studion. Med sig har de endast några akustiska instrument. Teknikern förbereder inspelningen genom att formattera en diskett för mixerbordets dator, där all kanal- och tidsinformation kommer att lagras. Digitala ekon från efterklangsenheter ställs in på defaultvärden. Producenten har kanske önskemål om vissa efterklangkarakteristika, och dessa programmeras in för kommande bruk.

Så till första tagningen:

Basisten kopplar in den elektriska basen i mixerbordet. En speciell kabel kopplas från basens MIDI-utgång till en polyfon synt, vilken i sin tur skickar en stereosignal till mixerbordet. En digital trummaskin kopplas in till bordet, men den analoga signalen spelas inte in, utan endast klocksignalen från MIDI-utgången används. Tejpen körs igång, och efter lite trixande med ljud är bas och trummor klara. Basisten är nu ledig för resten av dagen och klaviaturfolket ropas ner till studion.

Eftersom klaviaturspelaren redan visste vad som skulle göras har han spelat in hela huvudsekvensen på en sequenser, och denna kopplas nu in på mixerbordet. Den inspelade klocksignalen från trummaskinen triggar sequensern, som i sin tur styr alla inkopplade syntar. Fortfarande är endast basistens analoga signal och klocksignalen inspelad. I högtalarna hörs dock en låt med bas, trummor, stråkar och lite effektljud via studions ekoenhet. Sångaren ropas in och blir den ende som inte kan arbeta i kontrollrummet. Huvudstämma och körstämmor spelas in som vanligt. Gitarristen lägger på ett solo via det förprogrammerade ekot, i detta fall ett 'växande' eko. (Tänk dig att du står i en hangar och ropar. Ett eko bildas och dör sakta bort. I detta fall ropar du, ekot växer och når till slut samma styrka som vid ropet, där det tvärt huggs av.)

Låten mixas:

Musikerna kastas ut och ensam sitter teknikern, producenten och en inkallad programmerare kvar. Nivåerna från de olika kanalerna läggs upp och klocksignalen används återigen för att trigga trummaskin, sequenser och syntar.Teknikern och producenten går igenom ljudbilden och bestämmer vad som skall göras. I detta fall vill producenten ha trummorna spelade på ett annat sätt. Trumprogrammeraren sätter sig ner vid trummaskinen och lägger in ett helt nytt komp. Timingen av detta är inget problem eftersom det hela klockas av den inspelade klocksignalen. Vidare var det här med kyrkklockor i låten inte så lyckat tycker man, så producenten går fram till sequensern och ändrar kanalinformationen till synten. I stället hörs nu program 45 från synten, en vacker grisröst som blivit digitalt inspelad. Nya program körs in i ekoenheterna och ljudet mixas med eko till en behaglig nivå. På mixerbordet programmeras de kanaler in som används. Ibland måste vissa instrument höjas, ibland sänkas. Dessa förändringar och andra såsom t ex filtreringar programmeras in och sparas på mixerbordets diskett. Den digitala mastermaskinen kopplas in och låten mixas ner på två kanaler.

Framtiden:

I dag kan en blandad digital/analog utrustning köpas för c:a 80.000:-. För detta får man 16 kanaler (analogt), digitala ekon och effekter, mikrofoner, förstärkare m m. Det är bara att börja spela in.

I morgon kommer du att kunna köpa en helt digital utrustning för 50.000:-. För det får du 24 brusfria digitala kanaler som hanteras på hårddisk, ett delvis digitalt mixerbord, dator som m h a MIDI kontrollerar hela paketet inklusive instrument, mixerbord, bandspelare och effekter.

Bäva mände Inspelningsindustrin!

Varför betala 400:-/tim där, när du och några kompisar kan dela ett helt OK paket och köra hemma istället?

För dig som är intresserad av allt detta rekommenderas läsning av:

International Musician (engelsk, c:a 33:-).

Showtime.

MIDI Specifications (fås genom Roland, Yamaha, Casio, Sequential m fl).

Några intressanta syntar m m:

Casio CZ 101: Digital MIDI-synt, c:a 3.500:-.

Yamaha DX21: d:o.

Yamaha DX7: Kraftfull synt, MIDI, c:a 12.000:-.

Stepp DG1: Gitarrsynt, MIDI, c:a 40.000:-.

Alesis MIDIverb: Digitalt eko (rymdklang), c:a 40.000:-.

Commodore Amiga: 68000-baserad dator med mycket kraftfullt ljudchip, c:a 15.000:- (chipet i klass med dyr synt!).

Atari ST: 68000-baserad dator med hyfsat ljudchip. Inbyggd MIDI in/ut. Till denna maskin finns den i marknaden i särklass bästa MIDI-mjukvaran. C:a 11.000:-+3.000:-.

Christer Lindström

Redaktörn

Häromdagen hade vi bekymmer med en av firmans hyrda ledningar. Televerket underrättades. Efter ett tag stod en televerkare i receptionen. Han blir visad till modemet. Då får jag veta att han kom helt opreparerad till oss. Så m h a vår fax-apparat rekvirerarar han fram en ritning hur ledningen är dragen. Mäter sig till att signalen ligger under detekteringsnivån för modemet. Han tar bort dämpstrappen och ledningen kommer igång. Justerar nivån på stationen. Jag stoppar tillbaka strappen. Och med detta trodde jag det hela var över. Men nej. Nästa morgon ringer televerkaren och frågar hur det går med linjen! Det är inte ofta man träffar på så serviceinriktade tekniker. Heder åt den televerkaren.

För er som har kontakter med USA och behöver paket därifrån eller dit. Nu har det hänt något som kan underlätta det hela. UPS (United Parcel Service), en i USA inte okänd firma, har numera öppnat ett kontor i Sverige. Så den som har behov av snabbare transport än vad postverket kan erbjuda, här är säkert ett bra alternativ.

Ja, det var det hele.

/hn

Schack på datorer, på riktigt

The king (a chess program) is dead, long live the king (a chess machine).

Så börjar en artikel i Scientific Americans februarinummer. Där berättas om den stora årliga datorschackturneringen på Radison Hotel i Denver, Colorado, den 15 oktober 1985. Det som avgörs är NACCC, North American Chess Computer Championship. Det är årsmöte i The Association for Computing Machinery. Tekniker och programmerare står kring borden bakom en avspärrning och samtalar, skämtar med varandra eller väntar spänt på maskinernas nästa drag. 10 stora namn tävlar om titeln: AWAIT, BEBE, CHAOS, CRAY BLITZ, HITECH, INTELLEGENT SOFTWARE, LACHEX, OSTRICH, PHOENIX, och SPOC. Tre stora namn saknas dock: BELLE, CHESS 4.7, och NUCHESS.

Det mesta av intresset fokuseras på kampen mellan CRAY BLITZ och HITECH. På CRAY BLITZ sida om bordet sitter Robert Hyatt och Albert Gower, från University of Southern Mississippi och Harry Nelson från LLL. Mitt emot dem sitter Hans Berliner och Murray Campbell från Carnegie-Mellon University. Berliner är både schackrådgivare och programmerare i HITECH-laget.

Tvärtemot US Championships, där dödens tystnad råder, är denna turnering fylld av konversation, skratt, knatter från tangentbord och ständiga kommentarer från kommentatorn Miichael Valvo, datorkonsult och internationell schackmästare från Sedona, Arizona. "Ett svagt drag av Svart. Kungen är fortfarande för exponerad och bönderna på c5 och c6 forsätter att hämma försvaret". Intill säger en medlem i CRAY BLITZ lag "det var konstigt, jag trodde den skulle spela kungen till f3", rätt ut i luften. En internationell mästare kan fortfarande hitta fel i datorschack och program kan fortfarande förvåna sina skapare.

Vid midnatt är allt över. De flesta matcherna är avgjorda och experterna rankar HITECH som vinnare. Laget CRAY BLITZ ber Valvo om tillåtelse att ge upp. Han föreslår två drag till: om CRAY BLITZ läge inte är bättre då får laget ge upp. Läget är inte bättre och de ger upp. HITECH är North American Champion och de facto datorschackets kung. Även om CRAY BLITZ är officiell världsmästare (det vann titeln 1983 och behöver inte försvara den förrän i juni) är HITECHs seger, tillsammans med tre andra segrar imponerande. HITECH är alldeles säkert världens starkaste schackdator.

Leenden och konversation. Spelade frånvaron av BELLE, CHESS 4.7 och

NUCHESS någon roll? "Det hade varit trevligt om de hade varit med" säger en organisatör "men jag tror inte att resultatet hade blivit mycket annorlunda". Han fortsätter och betonar att när det gäller program och maskiner är det ingen egentlig skillnad mellan North American- och Världsmästerskapet (i schack). Samtalet går över till Kasparov och Karpov och vidare till teori. "Jag skojar inte" säger en deltagare som tydligen vet en del, "ett 20-plyprogram som bara tittar på pjäserna kan slå vilken mästare som helst". Man argumenterar lite men efter ett par minuter är rummet tomt. The North American Championship är över.

Påståendet om 20-plyprogrammet är intressant. Schack kan representeras som ett stort träd, bestående av noder och linjer. Visualisera trädet upp och ner så att rot-noden är överst. Varje nod representerar en möjlig ställning, nämligen ett schackbräde på vilket pjäserna har kommit till sina positioner genom spel enligt reglerna. En nod förenas med en efterföljande nod genom en linje när en föflyttning av en pjäs genererar en förändring av det förra brädets utseende till det senare. En schackmatch kan alltid identifieras som en speciell väg genom schackträdet, från rot-noden (vid vilken inga drag har gjorts) ned genom hela trädet till någon nod där, som regel, få pjäser finns kvar och en spelare har mattats eller tvingats ge upp.

Ett schackprogram försöker utforska så lite av spelträdet som möjligt. Från den aktuella noden undersöker det alla följande bräden (ply 1), alla följande på de följande (ply 2) och så vidare. Medeldjupet för dessa undersökningar kallas dess lookahead. Detta är ett mått på större delen av vad som innefattas i ett schackprograms intelligens. Den mindre delen består i hur programmet behandlar och undersöker brädena som ligger vid horisonten för dess lookahead. Det analyserar brädena och vart och ett ges ett numeriskt värde. Värdet reflekterar hur stark önskan är att uppnå den positionen. Genom en process som kallas minimax fås några åsatta värden att bubbla upp genom trädet och nå ply 1. Den nod som får högst värde visar vilket drag som skall göras.

Det finns ett intressant förhållande mellan de båda delarna av programmets intellekt: ju bättre dess utvärderingsförmåga är, ju mindre behöver det söka igenom spelträdet. Hade det haft den perfekta utvärderingsförmågan hade det aldrig behövt söka djupare än en ply. Å andra sidan måste ett program med svagare förmåga söka djupare om det skall kunna spela effektivt. Hur djupt måste en sökning som ser bara till pjäsernas positioner gå för att vara effektivt mot en stormästare? Räcker 20-ply?

Titeln stormästare delas ut av Federation International des Echecs till spelare som utmärker sig i internationellt spel. (Federationen har uteslutit datorer från sina bedömanden.) Stor-

mästare har vanligen poängsummor i storleksordningen över 2400, nivån hos en seniormästare. Fram till NACCC hade HITECH gått 21 matcher i mänskliga turneringar och tillskansat sig 2233 poäng. Detta gjorde den till den bästa schackdatorn i världen. Berliner, som hade 2443 poäng vid tiden då han själv tävlade, påstod att HITECH's summa ökar med i medeltal 8 poäng per match vid nationella turneringar. Törs man anta att maskinen om bara 14 matcher har gått om sin skapare?

Allt detta väcker frågan om hur bra schackspelande datorer kan komma att bli. Kommer en dator någonsin att kunna slå en världsmästare? David Levy, författare, f d spelare, har slagit vad flera gånger om saken. 1968 slog Levy vad med John McCarty vid Stanford Unversity om 500 pund att ingen dator skulle lyckas slå honom inom de nästa 10 åren. Levy kunde hämta sina pengar i augusti 1978, vid The Canadian National Exhibition i Toronto. Där slog han lekande lätt CHESS 4.5, ett program skrivet vid Northwestern University. Vadet förnyades, nu med 6000 dollar och skulle gälla för sex år. I april 1984 spelade Levy en telefonmatch från London med CRAY BLITZ. Han vann igen.

Levys vinst gjorde honom så modig att han slog följande 100.000-pundsvad i Denver: inom 10 år från nu kommer varje utmanande dator att ha slagits av en mänsklig spelare, utvald av Levy. Om Levy stöter på en utmanare kommer det inte bara att bli ett program utan säkerligen en specialiserad dator. Hittills har det inte funnits några utmanare.

De två toppfinalisterna i North American-turneringen, HITECH och BEBE var sådana specialiserade schackmaskiner. Det är intressant att notera att Levys eget bidrag kallat INTELLIGENT SOFTWEARE, kom på tredje plats. Det kördes på en Apple IIe som inte innehöll något mera sofistikerat än ett acceleratorkort med speciella kretsar som fördubblade maskinens hastighet. Kanske hade Levy utvecklat en överlägsen utvärderingsalgoritm.

Specialhårdvara

Schackkännarna vid turneringen ansåg att det bästa spelet hade rått mellan CRAY BLITZ och BEBE, en ren privatprodukt. BEBE är inte bara ett program, utan en schackmaskin. Matchen var betydelsefull inte bara för att den var den mest intressanta i turneringen, utan också för att det var första gången CRAY BLITZ hade förlorat på tre år. CRAY BLITZ körs på en Cray X MP48. Berömd för sin hastighet som multiprocessor är Cray icke desto mindre en general-purpouse-dator och inte en schackmaskin. BEBE, vars kretsar är specialtillverkade för schack var uppenbarligen bättre än Cray-CRAY BLITZ i ovannämda match.

HITECH är på ett sätt ytterligare specialiserad. När Carnegie-Mellon

University var Carnegie Institute of Technology, utvecklades där ett schackprogram som kallades TECH. Namnet HITECH innebär att forskarna återupplivade TECH i VLSI-form, med parallellprocessing. I HITECH-maskinen kombineras en Sun-dator med en speciell processor som Berliner kallar "the searcher". Sun-maskinen kör tre program: ett användarinterface, en task controller och ett orakel. Oraklet står för vad schackexperter kallar "boken". Det är en stor katalog över olika öppningar och variationer som mänskliga spelare känner till. Oraklets databas innehåller också mycket annan schackkännedom och kan enkelt utvidgas. När searchern utvärderar spelets möjligheter vid ett givet tillfälle använder den sig av information som är relevant vid det tillfället, vilken den laddat ned från oraklet.

Sökaren själv består av en mikroprocessor och flera andra hårdvarumoduler som genererar drag, utvärderar drag, söker efter upprepade drag och så vidare. Mikroprocessorn koordinerar deras aktiviteter. Draggeneratorn består av 64 VLSI-kretsar, en för varje ruta på schackbrädet. Varje chip undersöker hela brädet för att avgöra om någon pjäs kan flyttas till dess område. Den utvärderar det bästa draget med hjälp av standardkriterier, såsom t ex möjligheten att ta eller kontrollera mittfältet. På samma gång gör alla de 63 andra chiparna samma sak. Om det finns 10 pjäser på brädet gör den här arkitekturen att möjliga drag genereras 10 gånger fortare, om alla andra faktorer är lika.

Utvärderingen av drag måste hålla takt med genereringen. En första utvärdering klaras av av generatorn själv. Den innehåller en slags övervakare som dömer mellan de 64 chiparna. Varje chip beräknar ett tal som symboliserar dragets styrka och skickar det till övervakaren. De olika talen är som rop på uppmärksamhet. Övervakaren listar dem i ordning efter hur högt de ropar.

HITECH börjar sedan söka igenom spelträdet och följer listan som den fått av övervakaren. Det andra steget i utvärderingen görs av utvärderingsmodulen, vid varje ny position (nod) i spelträdet. Med hjälp av schackkännedom den laddat in från oraklet utvärderar modulen varje bräde (nod) antingen det befinner sig vid lookahead-gränsen eller ej. Så fungerar parallellprocessing, det extra arbetet kostar ingen extra tid. Sun-maskinens task controller talar om för serchern hur djupt den skall söka i trädet, och, när den har sökt färdigt, om den skall söka ännu djupare. På detta sätt klarar HITECH att hålla en medel-lookahead på 8 ply, men kan då och då söka så djupt som 14 ply. Det här kan verka vara ganska långt från de 20 ply som kan behövas för att slå en stormästare. Å andra sidan kan parallellprocessingen och det sofistikerade användandet av schackkännedom vid sökningarna kompensera för det relativt grunda sökandet. I vilket fall som helst, när

det är dags för nästa World Computer Chess Championship i juni i Köln, kan HITECH vara oövervinnerlig.

Deltagare i 1985 North American Computer Chess Championship. Observera att tredje plats innehölls av en liten Apple II, i kamp med Cray-giganterna. Appe-tvåan behöver tydligen bara söka igenom 500 bräden per sekund och klarar sig ändå alldeles utmärkt. Dess algoritmer är tydligen helt annorlunda än alla de andras! (Intelligent software kanske betyder nåt?). Observera också att DEC placerade sig ganska långt ner.

gm js

Program	ursprung	dator	språk	bräden/ sekund	look-ahead	place-ring
AWIT	University of Alberta	Amdahl 5860	Algol W	10	3-ply	
BEBE	SYS-10 Inc. Illinois	Special-bygge	Assembler	20,000	7-ply	2
CHAOS	University of Michigan	Amdahl 5860	FORTRAN	70	4-ply	
CRAY BLITZ	University of Miss.	Cray X-MP 48	FORTRAN/ assembler	100,000	8-ply	
HITECH	Carnegie-Mellon	Sun med x. VLSI	C	175,000	8-ply	1
INTELL. SOFTWARE	Intelli-gent softw.	Apple IIe acceler.	Assembler	500	7-ply	3
LACHEX	Los Alamos Nat. Lab.	Cray X-MP 48	FORTRAN/ assembler	50,000	7-ply	
OSTRICH	McGill University	7×NOVA+ ECLIPSE	Assembler	1,200	6-ply	
PHOENIX	University of Alberta	X×VAX 780 +10×SUN	C	540	6-ply	
SPOC	SDI/Cypress Software	IBM PC	Assembler	300	5-ply	

Danmark

Så var det dags igen då. Denna gång fanns maskinen att hämta i Danmark, vilket jag ansåg var såpass nära att vi på någon dag skulle kunna åka dit och hämta den själva. Attans, som vanligt strular det till sig i kvadrat. Det första var att Daimi (instutionen heter så som hade den) bestämde sig för att köra 10 dagar extra. Så jag fick riva upp bokningar och annat jag för en gångs skull lyckats göra i tid. Jag försökte låna en lastbil av Volvo, där jag först fick positiva besked, men när den ansvarige chefen väl fick ärendet var det stopp. Slutsats: Volvo är inte intresserade av att få datorkunniga civilingengörer, vilket ni kan lägga på minnet. Det vart till sist så att vi fick hyra av Karlbergs till sedvanlig KTH-rabatt (Kårspexet och vi är visst de enda).

Nå till sist spikades en dag, torsdagen den 28/8 skulle den hämtas. Allt verkade frid och fröjd, lastbil fixad, färjebiljetter bokade (att åka med en lastbil är lite extra), allt VERKADE lugnt. På tisdag ringer de från Århus och berättar, "Vi har ingen exportlicens eller danskt utförseltillstånd". Totalmörker, exportlicens tar 6–8 veckor att få och vi skulle åka dan därpå, problem. Några telefonsamtal till amerikanska handelsattachen, får ett telefonnummer till nån jourservice i Washington, där jag pratar med flera tjänstemän och till sist hittar jag en som förstår vårt problem. Det visar sig att en paragraf behandlande gods av lågt värde (under 1000 USD) säger att man då inte behöver exportlicens till vissa länder, däribland Sverige. Sen gällde det att få danska handelsdepartementet och andra att förstå att denna paragraf fanns och var tillämplig i detta fall. Universitet i Århus hade ingen rutin på danska exporter, så Hans Nordström ordnade genom sina kontakter fram en dansk speditör som med kurirexpress ordnade fram alla dokumnet som behövdes. De anlände till Århus kl 14 på torsdagen, då vi nästan hade lastat klart.

Tillbaka i tiden. Efter en klart stressad tisdag trodde jag att allt var solklart. Onsdag morgon hämtar vi lastbil, åker till B30, där vi plockar upp de 4 skåp som danskarna ville ha istället för de vi fick. Efter sedvanligt strulande var vi äntligen på väg mot Varberg, för därifrån skulle färjan gå. Körningen till Varberg gick utan missöden. Vi anländer dit ca kl 17, d v s i god tid. Ett besök på tullkontoret informerar oss om de lokala rutinerna. På dieselbilar i Sverige har man en s k kilometerräknare, som skall stämplas vid in- och utfart. Vi fick ett stämpelkort, men det var för brett, så Henrik rev av det till lämplig storlek, tullaren vart rosenrasande och tyckte saker om att vi borde banne mej veta hur blanketten skulle se ut osv. Det var fel stämpelkort, vi fick

ett nytt som passade, sen var alla glada. De fyra skåpen deklarerades till 25 kronor styck, och exporterades ut ur landet, d v s vi har moms att få tillbaka

Små lastbilar (11 meter) som våran utan släp fick åka ombord nästan sist och färjan var försenad. Sen dundrar det på färjor. Framme i Grenå klockan 01.22 utkörning av lastbilarna, som var packade som sardiner med bara någon centimeter emellan. Lastbilschaffis är lixom inget jobb för oss datadårar. Nåväl denna gången gick det ju bra ändå. Danska tullen var förstående och ville inte befatta sig med de 4.34 danska kronorna vi skulle betalat i tull utan tyckte: "Försvinn nu, vi har ingenting sett".

Från Grenå till Århus var det 8 mil att åka (vi var nu lätt trötta) och i Danmark får man bara köra 80 km/tim. Vår tanke var att ta in på ett hotell i Århus och sova fram till 11–12 och sen lasta, men det sket sig. Något hotell hade vi ju inte bokat. Århus är ju Danmarks andra stad, så jag trodde det inte skulle vålla några problem att hitta hotell utan att boka. Luring igen.

Klockan 03.30 efter x hotell gav vi upp. Vi hade ju trots allt en stor lastbil och ett ansenligt lager med filtar och täcken, så vi kör lite utanför staden och parkerar, i tron om att få sova. Otur igen, först börjar det hagla, alla kan gissa hur det låter mot taket på en glasfiberkaross, helt tom inuti (jfr bastrumma). När regnet taget slut var det tyst och jag sov bergis två timmar, innan en dansk trädsgårdsarbetare utrustad med gräsklippare började köra runt-runt i bästa indianvästernstil.

Då gav vi upp, letade reda på universitetet och hittade ett rum benämt "PDP-10 Maskinstue", men kl 9 på morgonen var ingen där. Efter nån timme kom det någon som tyckte "Er dere svenskerne? Jeg såg lastebilen uten." En kuslig repris av min resa till UCI. Så småningom kom det fler danskar och dessa kokade kaffe, lycka.

Det visade sig att folket runt maskinen var ett gäng kunniga och glada, arbetsvilliga typer. Efter några minuters genomgång om hur jag ville maskinen skulle monteras ned, fördelades arbetet och sen gick det med farlig fart. En god dansk mittpådagenmåltid och vi var i det närmaste klara, väntade bara på exportpappren. Dessa kom fram vid tvådraget, helt enligt planerna. Danskarna imponerade på mig, men de pratar konstigt. Till och med en ramp hade de att rulla ut den på, tittut Johnny.

En välbehövlig dusch, nya kläder och vi reste tillbaka till Grenå. Mat instoppades och klockan var 19. Färjan skulle gå 01.00, så jag lade mig att sova på en soffa i vänthallen.

Efter uppvaknandet var det dags för den danska TULLEN: "Skruttet datamaskin 2000 kg, mutter, mutter, mutter, muttter, skruttet?, mutter", sedan kliade det ordentligt i stämpelarmen, och det

sa slammer. "Mutter", så fick jag tillbaka papperen och det var klart.

Sardin-inpackning i lastbilen igen, och en hytt med säng, zzzzzz i fem timmar. Sen var det urlastning av färjan, tullen, kilometerstämplingen, samma tullare som när vi åkte åt andra hållet, så han tyckte: "DU SKALL HA DEN HÄR". Vi styrde kosan från Varberg till Göteborg och Medicindata, där det stod mer saker att hämta.

Instutionens chef Irmin var först på plan, vänligheten och totalservicen personifierad. Vi fick kaffe, hotellrum bokade och en utfrågning om allt verkligen var bra. Inventeringen av extra saker hos Medicindata gav att de hade mer saker än vi kunde ta med oss. JMR hade önskat sig RP02, de blinkar och skramlar effektfullt. Sen fanns det RP03, TU10, diverse kontrollers och en DL10 som vi ville ha till Katia, för att kunna göra ANF10-anslutning.

Lämplig prioritering med hänsyn taget till användbarhet och lastbarhet gjordes, 8 RP03/02 lastades (vi hade 2 redan från Danmark). Sen var det FULLT, bilens stötdämpare gjorde mindre nytta. En repa gjordes i deras fina barack med baklyften, men den skall ändå rivas om något år.

Vi tackade för oss, drog till CD på Chalmers, där vi fick vet att de skulle ha kräftskiva och vi som var så trötta. Dusch på hotell och en snabbvisit på deras fest, sen sova. Vid middagstid på lördag startade vi mot Stockholm och anlände lyckligt vid 17-tiden. Urlastning i CCCC, inlastning igen och iväg till B30, urlastning, osv.

Maskinen var lätt tjurig och alla kraftaggregat var feljusterade. Efter flera dagars skruvande på timrar, pottar och byte av kort, rengöring av glappkontakt osv gick CPU:n. Att sedan periferiutrustningen bestod av det samlade "bottenskrapet" av undanställda och klassade som besvärliga skåp gjorde ju inte saken lättare. Tror vi höll på i tre veckor, till och från, innan ett fungerande system uppstod ur spillrorna.

Peter Löthberg

$1.00

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Contents

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Gnu's Zoo

First and foremost there's our porcupine Richard M. Stallman. The last of the true hackers and founder of project GNU.

Secondly there's Leonard H. Tower, Gnu's teddy bear. Len is Gnu's first and so far only paid full time employee.

Gnu's Hawk, Robert Chassell, is the world's only generous treasurer.

Gnu has two wise old night owls, Professor Hal Abelson and Professor Gerald Sussman. They are advisors and round out FSF's board of directors.

Amoung our volunteer hackers there's Dean L. Elsner, our world hopping platypus (I originally called him a kangaroo but he insists he's a platypus). In case you haven't guessed, Dean comes from Australia. Dean is writing Gnu's assembler.

Another Australian, Richard Mlynarik, is acting as Gnu's Emacs Guru. I'll try calling him our kangaroo and see what happens.

Eric Albert walked in off the street on January 24. So far, he's sped up the GNU LD command to be faster than UNIX's (it was much slower), and is now fixing some bugs in it. After that, he'll be working on removing fixed-length limits from GNU CPP, and also speeding it up. Eric claims he's Gnu's humuhumunukunukuapuaa, the current state fish of Hawaii. And we're happy to have the help of such a rare fish.

There is also Paul Rubin on the West coast. Gnu's spider, Paul weaves Gnu Emacs reference cards and produces nifty covers for the new version of the Gnu Emacs manual.

Me? My name's Jerry Puzo. I answer the mail and send out tapes. It explains a lot to say I'm Gnu's turtle.

end

G N U ' S B U L L E T I N

Editor: Jerome E. Puzo

end

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

What is GNU Emacs and do you want a copy?

by Richard M. Stallman

GNU Emacs is a new implementation of the Emacs text editor. (Recently text editors have been called "word processors" among microcomputer users.)

Emacs is a kind of architecture for text editors, in which most editing commands are written in an interpreted language (usually Lisp) so that the user can write new editing commands as he goes. This allows Emacs to have editing commands that are more powerful or more adapted to individual uses than other kinds of editors.

Any particular editing command could be written in C, but with Lisp it is much easier for users to change the editing commands or to implement new editing commands. Users can also exchange their adaptations and extensions of Emacs. The result is a library of extensions that continues to grow.

GNU Emacs boasts an especially clean Lisp system for writing editing commands, and an already large library of extensions.

GNU Emacs is written in C, designed for a Unix or Unix-like kernel. It includes its own Lisp interpreter which is used to execute the portion of the editor that is written in Lisp.

It is a fairly large program, around 525k on vaxes or 68000s, to which must be added space for the files you are editing, undo buffers, Lisp libraries loaded, and Lisp data such as recently killed text, etc. This is not really a problem on a timeshared machine because most of that 525k is shared, but on a personal computer there may be nobody to share with. Thus, GNU Emacs probably could not be used on an IBM PC clone for lack of memory, unless you want to implement virtual memory in software within Emacs itself. Perhaps on an 80286 with 1 meg of memory you can win using their memory management.

In general, a 32-bit machine with either a meg of real memory or virtual memory can probably run GNU Emacs, as long as a suitable Unix system call environment is provided, simulated or imitated.

end

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Version 17 of Gnu Emacs comes with its own doctor

* Gnu Emacs version 17 is now available. See the article HOW TO GET GNU EMACS and our Order Form elsewhere in this bulletin.

* Gnu Emacs 17 works on system V. Even subshells work.

* The online Emacs manual is available through the info command.

* Outline mode now allows the user to selectively hide or display the subtree of an item.

* TeX and Nroff editing modes have been added.

* C editing mode has been made smarter. It now understands how to indent else clauses.

* Consistency between modes has been improved by assigning some commands to different keys.

* Toys. To the disassociated press has been added:

hanoi, the (slightly) animated puzzle solver,
yow, a Zippy saying producer, and
doctor, the infamous psychiatrist.

The folks on net.emacs have sent a suggestion for yowza which lets you watch the doctor respond to yow.

end

H O W T O G E T G N U E M A C S

All software and publications are distributed with a permission to copy and redistribute. The easiest way to get a copy of GNU Emacs is from someone else who has it. You need not ask for permission; just copy it.

If you have access to the Internet, you can get the latest distribution version of GNU Emacs from host: 'prep.ai.mit.edu'
For more info read: '/u2/emacs/GETTING.GNU.SOFTWARE' on said host.

If you cannot get a copy in any of these ways, you can order one from the Free Software Foundation. Please consult the accompanying Order Form for prices and details.

Although Emacs itself is free, our distribution service is not.
The income from distribution fees goes to support the foundations's purpose: the development of more free software to distribute just like GNU Emacs.

Currently, all software is available for UNIX 4.2 BSD on 1600 bpi tar tape. It runs on VAX computers, as well as several 68XXX and 32XXX machines. Contact FSF regarding suitability of your computer system. We encourage porting to other machines.

end

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Status of GNU Emacs on Various Machines and Systems.

Systems:
For each type of system, the name of the appropriate s- header file is given.

Berkeley 4.1 (s-bsd4.1.h)
Some conditionals have been provided for 4.1, but I do not know for certain that they work as merged in.

Berkeley 4.2 (s-bsd4.2.h) Works on several machines.

Berkeley 4.3 (s-bsd4.3.h) Works, on Vaxes at least.

Ultrix This is another name for Berkeley 4.2.

Uniplus 5.2 (s-uniplus5.2.h) Works, on Dual machines at least.

System V rel 0 (s-usg5.0.h) Close to working, on Vaxes.
A couple of bugs remain.

System V rel 2 (s-usg5.2.h)
Works, on Stride, TI/LMI Nu and HP 9000s200 machines; but in each case the basic system V has been enhanced somewhat. How Emacs works on a vanilla system V (if you can find one) is not clear.

The s- file for the HP machine is s-hpux.h, not s-usg5.2.h.

System V rel 2.2 (s-usg5.2.2.h)
In 5.2.2 AT&T undid, incompatibly, their previous incompatible change to the way the nlist library is called. A different s- file is used to enable the other interface.

Machines:
For each type of machine, the names of the m- and s- header files are given.

Apollo running Domain (m-apollo.h; s-bsd4.2.h)

Currently has a bug: exhausts pure Lisp code space while building Emacs. This is probably one trivial error, but someone with an Apollo will have to find it.

Once that bug is fixed, one problem will remain permanently. It is impossible to dump Emacs; the standard Lisp code must be loaded each time Emacs is started. This is a limitation of their operating system. In other respects the system appears to be Berkeley 4.2, and Emacs is told that it is running under 4.2.

AT&T 7300 running System V
This port has been done but I have not received the diffs yet.

Celerity
17.36 has been ported, but I have not seen the port yet.

cont

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Status of GNU Emacs on Various Machines and Systems con't

Dual running System V (m-dual.h; s-usg5.2.h)
As of 17.46, this works except for a few changes needed in unexec.c.

Dual running Uniplus (m-dual.h; s-unipl5.2.h) Works.

Gould

Previous versions ran into trouble with their failure to support alloca. Now that there is a portable alloca supplied with Emacs, it should not be very hard to do this port.

HP 9000s200 (m-hp200.h; s-hpux.h) Works. This machine is a 68020.

Megatest (m-mega68.h; s-bsd4.2.h)
Emacs 15 worked; do not have any reports about Emacs 16 or 17 but any new bugs are probably not difficult.

Nu (TI or LMI) (m-nu.h; s-usg5.2.h) Nearly working; a few bugs remain.

Pyramid (m-pyramid.h; s-bsd4.2.h) Works.

Sequent Balance (m-sequent.h; s-bsd4.2.h)
Emacs 17.48 works in their system version 2.0.
Emacs has not been tried on their system version 1.3.

Stride (m-stride.h; s-usg5.2.h)
Works, though has not been tested for long. Note, however, that this was on a Unix version not yet released by Stride. It is probably also possible to run on Stride's 5.1 system but changes in the s- file are probably needed.

Sun (m-sun.h, m-sun2.h, m-sun3.h; s-bsd4.2.h)
There are three m- files for different models of Sun.
All use Berkeley 4.2. Emacs 17 has run on all of them.

Tahoe (m-tahoe.h; s-bsd4.2.h) Works.

Tektronix(?) 16000 box (m-16000.h; s-bsd4.2.h)
Emacs 15 worked; no reports since then.

Vax running Berkeley Unix (m-vax.h; s-bsd4.1.h or s-bsd4.2.h or s-bsd4.3.h)
Works for certain under 4.2 or 4.3; probably a few bugs to fix for 4.1. Note tha "ultrix" is essentially 4.2; use s-bsd4.2.h.

Vax running System V rel 0 (m-vax.h; s-usg5.0.h) Still has a couple of bugs.

Vax running VMS Port nearly completed.

end

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

A Sample .emacs File

```
; Robert J. Chassell 6 December '85 simplified 9 January '86 Jerome E. Puzo

; This is a sample .emacs file for GNU Emacs on a Vax running BSD 4.2 Unix.
; Lines that begin with a semi-colon are comments not executed by Emacs.

; TEXT MODE AND AUTO-FILL-MODE

; The next two commands put Emacs into text mode and auto-fill-mode
; when Emacs starts.  They are designed for writers who want to start
; writing prose rather than code.
; A programmer might want to enter Lisp mode or C mode.

(setq default-major-mode 'text-mode)
(setq text-mode-hook 'turn-on-auto-fill)

; Sample KEY BINDINGS for a Z-29 terminal

; These functions show how to bind keys to commands.
; The keyboard commands continue to work: for example, you can go
; forward by word either with the right arrow key or with <esc f>.
; If you do not know what meta sequence a function key returns,
; you can use the 'describe key' function: type control-h k and then
; the key.  Emacs will tell you the meta sequence and any commands
; to which the key is bound.
; note: \e indicates the esc character

(global-set-key "\eT" 'backward-kill-word) ; function key F2
(global-set-key "\eU" 'kill-word)          ; function key F3
(global-set-key "\eD" 'backward-word)      ; function key left-arrow
(global-set-key "\eC" 'forward-word)       ; function key right-arrow
(global-set-key "\eB" 'scroll-up)          ; function key up-arrow
(global-set-key "\eA" 'scroll-down)        ; function key down-arrow
(global-set-key "\eJ" 'forward-sentence)   ; function key erase-key
(global-set-key "\eH" 'backward-sentence)  ; function key home-key
(global-set-key "\eP" 'goto-line)          ; function key F6

; Example of how to specify control key:
; to redefine control-y to go to the start of the line (like control-a)
; (global-set-key "\C-y" 'beginning-of-line)

; Example of how to cancel a key binding:
; (global-unset-key "\C-y)

; UPDATING EMACS

; After writing a function in your .emacs file, you can send the
; changed information to the rest of emacs by entering meta-control-x

; This command finds the function around or following the point.
; As soon as you do this, you can begin to use your new function.

                                *end*
```

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

What is the Free Software Foundation?

by Richard M. Stallman

The Free Software Foundation is dedicated to eliminating restrictions on copying, redistribution, understanding and modification of software.

The word "free" in our name does not refer to price; it refers to freedom. First, the freedom to copy a program and redistribute it to your neighbors, so that they can use it as well as you. Second, the freedom to change a program, so that you can control it instead of it controlling you; for this, the source code must be made available to you.

The Foundation works to give you these freedoms by developing free compatible replacements for proprietary software. Specifically, we are putting together a complete, integrated software system "GNU" that is upward-compatible with Unix. When it is released, everyone will be permitted to copy it and distribute it to others; in addition, it will be distributed with source code, so you will be able to learn about operating systems by reading it, to port it to your own machine, to improve it, and to exchange the changes with others.

There are already organizations that distribute free CPM and MSDOS software. The Free Software Foundation is doing something different.

1. The other organizations exist primarily for distribution; they distribute whatever happens to be available. We hope to provide a complete integrated free system that will eliminate the need for any proprietary software.

2. One consequence is that we are now interested only in software that fits well into the context of the GNU system. Distributing free MSDOS or Macintosh software is a useful activity, but it is not part of our game plan.

3. Another consequence is that we will actively attempt to improve and extend the software we distribute, as fast as our manpower permits. For this reason, we will always be seeking donations of money, computer equipment or time, labor, and source code to improve the GNU system.

4. In fact, our primary purpose is this software development effort; distribution is just an adjunct which also brings in some money. We think that the users will do most of the distribution on their own, without needing or wanting our help.

cont

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

What is the Free Software Foundation? con't

Why a Unix-Like System?

It is necessary to be compatible with some widely used system to give our system an immediate base of trained users who could switch to it easily and an immediate base of application software that can run on it. (Eventually we will provide free replacements for proprietary application software as well, but that is some years in the future.)

We chose Unix because it is a fairly clean design which is already known to be portable, yet whose popularity is still rising. The disadvantages of Unix seem to be things we can fix without removing what is good in Unix.

Why not imitate MSDOS or CPM? They are more widely used, true, but they are also very weak systems, designed for tiny machines. Unix is much more powerful and interesting. When a system takes years to implement, it is important to write it for the machines that will become available in the future; not to let it be limited by the capabilities of the machines that are in widest use at the moment but will be obsolete when the new system is finished.

Why not aim for a new, more advanced system, such as a Lisp Machine? Mainly because that is still more of a research effort; there is a sizeable chance that the wrong choices will be made and the system will turn out not very good. In addition, such systems are often tied to special hardware. Being tied to one manufacturer's machine would make it hard to remain independent of that manufacturer and get broad community support.

end

February 1986 G N U ' S B U L L E T I N Volume 1 No 1

Gnu Status
by Richard M. Stallman

1. GNU Emacs.

GNU Emacs is in wide use on several kinds of 4.2 systems. Support for some versions of system V now exists, and VMS support is expected now in a few weeks. There is now an Info-style reference manual also.

Berkeley is going to include GNU Emacs on the 4.3 distribution, and DEC has also expressed an interest in distributing it with Unix systems.

2. gsh, the GNU imitation C shell.

This is being tested at a few sites. Wider distribution is expected soon.

3. Kernel.

I am planning to use a remote procedure call kernel called TRIX, developed at MIT, as the GNU kernel. It runs, and supports basic Unix compatibility, but needs a lot of new features. Its authors have decided to distribute it free. It was developed on an obscure, expensive 68000 box designed years ago at MIT.

4. C compiler

Although I have a portable C and Pascal compiler, it has a serious drawback: it is a very large program, and intrinsically cannot be made smaller. It is also very hard to bootstrap.

The problem is that most of the compiler is written in Pastel, a super-hairy extended Pascal, and it is also the sole compiler for that language. To make it smaller, we must eliminate the hair needed to compile Pastel; then we will not be able to compile Pastel, so it must all be rewritten into C.

Len Tower, the sole full-time GNU staff person, is working on this, with one or two assistants. He can certainly use more, but they must be in Cambridge or else be able to communicate on the Internet.

5. Documentation system.

I now have a truly compatible pair of programs which can convert a file of texinfo format documentation into either a printed manual or an Info file.

Documentation files are needed for many utilities.

con't

February 1986 G N U ' S B U L L E T I N Volume 1 No. 1

Gnu's Status Con't

6. Other utilities.

'diff', 'tar' and 'find' are being written. 'ls', with full 4.2 and system V features, is finished. 'make', with full 4.2 features, is also finished. 'lex' is supposedly finished and to be sent soon.

A mostly-machine-independent assembler is mostly finished.

I have started writing a debugger, somewhat along the lines of dbx. It can now read dbx symbol tables and evaluate C expressions with respect to a core dump.

7. Free Software Foundation.

This foundation exists for two purposes: to accept gifts to support GNU development, and to carry out distribution. It was incorporated at the beginning of October, and we applied for a tax examption in late December.

Its address is

Free Software Foundation, Inc.
1000 Mass Ave
Cambridge, MA 02138

and its phone number is (617) 876-3296.

According to our incorporation papers:

"The corporation is formed for literary, educational and charitable purposes with the special purpose of

i) encouraging, fostering, and promoting the free exchange of computer software and information related to computers and other technology.
ii) distributing and disseminating software and information related to computers and other technology; and
iii) increasing the public's access to computers and high technology devices.

con't

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Gnu's Status Con't

8. Service directory.

The foundation now maintains a Service Directory; a list of people who offer service to individual users of GNU Emacs and, eventually, all parts of the GNU system. Service can be answering questions for new users, customizing programs, porting to new systems, or anything else.

9. Porting.

It is too early to inquire about porting GNU (except GNU Emacs). First, we have to finish it.

10. Possible target machines.

GNU will require a cpu that uses 32-bit addresses and integers and addresses to the 8-bit byte. 1 meg of core should be enough, though 2 meg would probably make a noticeable improvement in performance. Running much of the system in 1/2 meg may be possible, but certainly not GNU Emacs. I do not expect that virtual memory will be required, but it is VERY desirable in any case.

GNU Emacs requires at least a meg of memory in the system, either physical or virtual.

A hard disk will be essential; at least 20 meg will be needed to hold the system plus the source code plus the manual plus swapping space. Plus more space for the user's files, of course. I'd recommend 80meg for a personal GNU system.

This is not to say that it will be impossible to adapt some or all of GNU for other kinds of machines; but it may be difficult, and I don't consider it part of my job to try to reduce that difficulty.

I have nothing to say about any specific models of microcomputer, as I do not follow hardware products.

end

February 1986 G N U ' S B U L L E T I N Volume 1 No. 1

Some Arguments for Gnu's Goals

by Richard. M. Stallman

Once GNU is written, everyone will be able to obtain good system software free, just like air.

This means much more than just saving everyone the price of a license. It means that much wasteful duplication of system programming effort will be avoided. This effort can go instead into advancing the state of the art.

Complete system sources will be available to everyone. As a result, a user who needs changes in the system will always be free to make them himself, or hire any available programmer or company to make them for him. Users will no longer be at the mercy of one programmer or company which owns the sources and is in sole position to make changes.

Schools will be able to provide a much more educational environment by encouraging all students to study and improve the system code. Harvard's computer lab used to have the policy that no program could be installed on the system if its sources were not on public display, and upheld it by actually refusing to install certain programs. I was very much inspired by this.

Finally, the overhead of considering who owns the system software and what one is or is not entitled to do with it will be lifted.

"So, how could programmers make a living?"

There are plenty of ways that programmers could make a living without selling the right to use a program. This way is customary now because it brings programmers and businessmen the most money, not because it is the only way to make a living. It is easy to find other ways if you want to find them. Here are a number of examples.

A manufacturer introducing a new computer will pay for the porting of operating systems onto the new hardware.

The sale of teaching, hand-holding and maintenance services could also employ programmers.

People with new ideas could distribute programs as freeware, asking for donations from satisfied users, or selling hand-holding services. I have met people who are already working this way successfully.

Users with related needs can form users' groups, and pay dues. A group would contract with programming companies to write programs that the group's members would like to use.

end

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Wish List

There are various thing which project GNU and the Free Software Foundation can do with the donation of:

* Money

* A modular, customizable, optimizing, free or public domain C compiler with source.

* Money. Salary for two more full time programers.

* Equipment to keep them busy on. Or a 68xxx or 32xxx based system with one meg or more of memory and 80meg of disk storage would do.

* Money

* Office space of our own.

* Money

* Dedicated people, with C and Unix knowledge, especially those with a local (Cambridge and surrounds) address. We have utilities for programmers to program. We have documentation for dedicated people to write.

* Money

end

February 1986 G N U ' S B U L L E T I N Volume 1 No.1

Free Software Foundation Order Form
February 6, 1986

All software and publications are distributed with a permission to copy and redistribute.

Quantity	Price	Item
_____	$160	GNU Emacs source code, on a 1600bpi industry standard mag tape in tar format. The tape also contains MIT Scheme (a dialect of Lisp), hack (a rogue-like game) and bison (a compatible replacement for yacc).
_____	$15	GNU Emacs manual. This includes a reference card.

Thus, a tape and one manual come to $165.

_____	$60	Box of six GNU Emacs manuals, shipped book rate.
_____	$1	GNU Emacs reference card. Or:
_____	$6	One dozen GNU Emacs reference cards.

Shipping outside North America is normally by surface mail. For air mail delivery, please add $15 per tape or manual, $1 for an individual reference card, or 50 cents per card in quantity twelve or more.

Prices are subject to change without notice. Massachusetts residents please add 5% sales tax to all prices.

_____ Total paid

Orders are filled upon receipt of check or money order. We do not have the staff to handle the billing of unpaid orders. Please help keep our lives simple by including your payment with your order.

Make checks payable to Free Software Foundation. Mail orders to:

Free Software Foundation, Inc.
1000 Mass Ave
Cambridge, MA 02138

All software from the Free Software Foundation is provided on an "as is" basis, with no warranty of any kind.

February 1986 G N U ' S B U L L E T I N Volume 1 No 1

Thank Gnus

The Free Software Foundation would like to send special thank gnus to the following:

Thanks to Micheal Zelyony. Mike answered the mail and sent out manuals and publicity for the FSF from September to November of 1985. As the one who has taken over his job I can appreciate the size of his contribution.

Thanks to Ed Zimmer. Ed's philanthropy has given the FSF the salary for one full time programmer.

Thanks to Lisp Machine, Inc. LMI has generously provided office space, computer resources and a mailing address for FSF.

Thanks to Jerry Pournelle. Jerry mentioned us in his BYTE column. We have received over one hundred responses so far. Ninety percent of Jerry's readers take what he says literally. One or two single dollar bills seem to fall out of each letter I open.

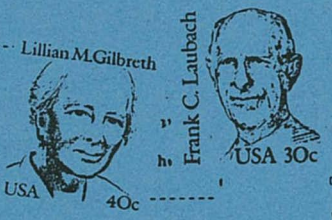
Thanks to all those who have contributed source code.

Thanks to those who sent money and offered help. Thanks also to those who support us by ordering Emacs manuals and distribution tapes.

The creation of this bulletin is our way of thanking all who have expressed interest in what we are doing.

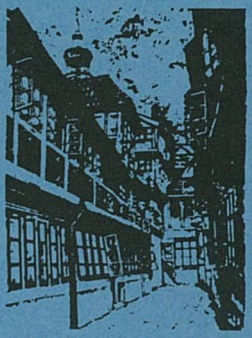
end

Free Software Foundation, Inc.
1000 Mass Ave
Cambridge, MA 02138



Royal Institute of Technology
Department of Numerical Analys
and Computing Science
Matti Rendahl
S-100 44 Stockholm 70, Sweden

Rezept



Galerie-Stuben
Inh. W. Sicl

Krauenkamp 10
2000 Hamburg 11
Telefon 36 58 00

De Hamborger „Suurkruutsupp"

Twee Liter klore Bulljong
Een Pund Suurkruut (affkokt!)
Een un een halbes Pund Backobst
(mit Plumm'n ut Kalifornien!)
Twee Löpel vull Honig
Een beeten Körri
Dat Ganze mischen, twintig Minuten
koken un denn op'n Toller!
Bobenop een Löpel stief slogene „Sahne"
(datt sütt scheun ut!)

Die Hamburger „Sauerkraut-Suppe"

2 Liter klare Fleischbrühe,
1 Pfund gekochtes Sauerkraut,
1½ Pfund Backobst,
2 Löffel Honig, etwas Curry.
20 Minuten aufkochen
und im Teller servieren.
Das Ganze wird mit
einer Sahnehaube gekrönt.

Dat smeckt!!