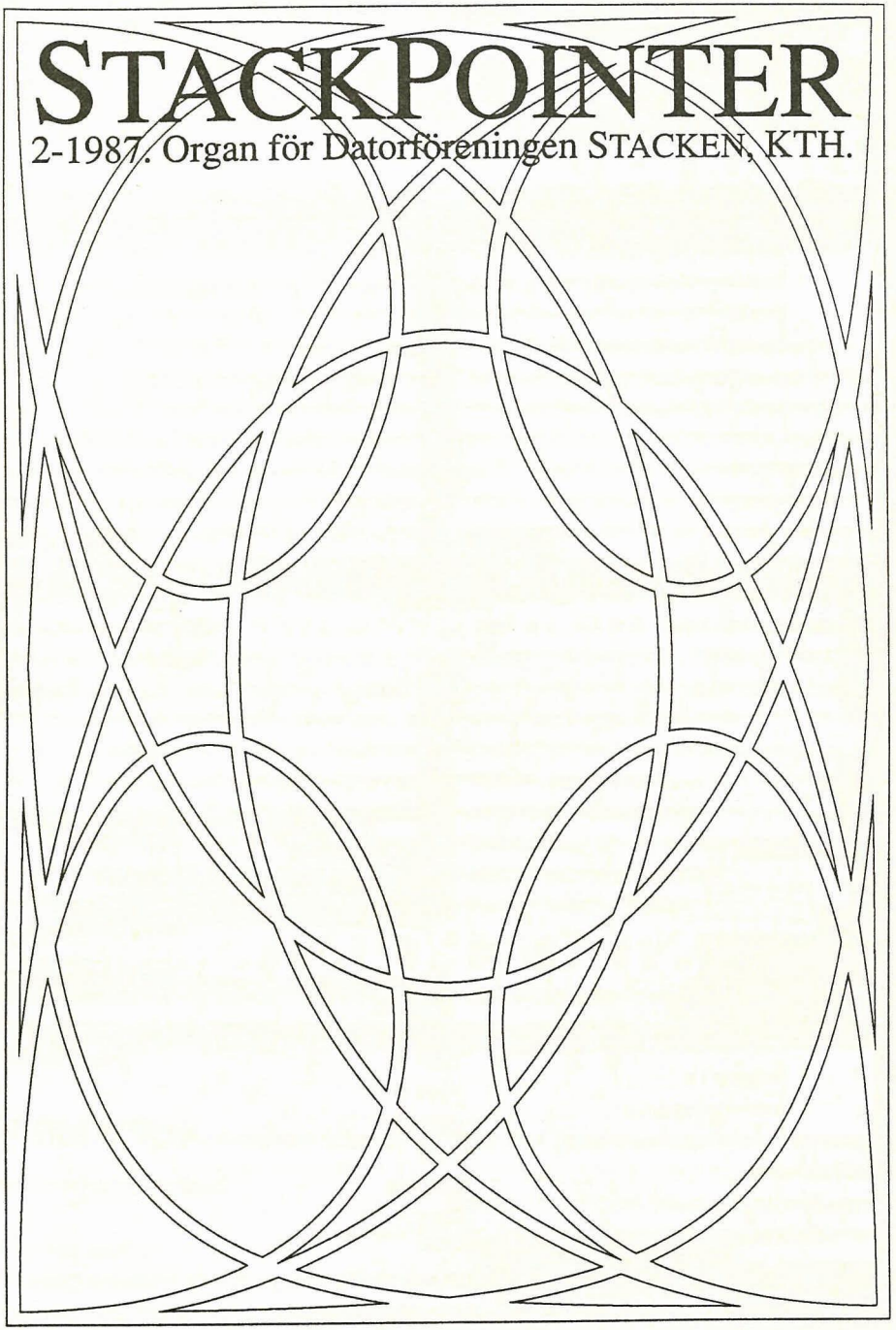
STACKPOINTER

2-1987. 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 ideell förening, där intresse för datorer är den gemensamma faktorn. Sedan föreningen grundades 1978 har vi (bland annat) åstadkommit:

- samköp av mikrodatorbyggsatser
- kurser, föredrag och studiebesök
- AMIS, den portabla EMACS-kompatibla editorn för TOPS-10, VMS, PRIME, NORD, ...
- STACKPOINTER, vår tidning
- en egen datorhall för vår DEC-10 och våra andra stordatorer

Sedan ett och ett halvt år har vi 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 maskinhall "B30", på Brinellvägen 30 (V-sektionen) på gaveln mot Lill-Jansskogen (där vi har en egen ingång). En våning ovanför finns en hörsal (V4), där vi håller till vid större möten.

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

STACKPOINTER

STACKPOINTER är organ för Datorföreningen STACKEN på KTH. Den utkommer när material i tillräcklig mängd finns, förhoppningsvis 4–5 gånger per år. Återgivande av delar av innehållet är tillåtet när källan anges.

Redaktör: Hans Nordström
I redaktionen: Jan Michael Rynning och Thord Nilson
Bortmonteratkodlåsförbisläppare: Stellan Lagomtrött
Ansvarig utgivare: Mats O Janssson
Färdigställd: 1987-03-21

I detta nummer

Adresser m m

POST till föreningen skickas till nedanstående adress eller läggs i postfacket på NADA.

Datorföreningen STACKEN
c/o NADA
KTH
100 44 STOCKHOLM

Medlemsavgift: 87 kronor för 1987

Ordf: Stellan Lagerström {08-787 78 14 (arb) 08-63 68 25 (hem)
Sekr: Olle Betzén {0758-318 48 (hem)
Kass: Mats O Jansson {08-34 93 20 (arb) 08-711 70 37 (hem)
Red: Hans Nordström {0760-615 80 (arb)
Hexm: Henrik Björkman {08-30 98 01 (hem)
Suppl: Henning Croona {08-799 80 74 (arb) 08-739 17 40 (hem)
Suppl: Thord Nilson {08-712 00 90 (arb) 08-749 21 92 (hem)

Postgiro: 433 01 15-9
Bankgiro: 344-3595

Klubblokal: Osquars Backe 4 (ingång från valvet, Osquars Backe 1)
Datorhall: Brinellvägen 32 (på gaveln mot Lill-Jansskogen)

SUNET DECnet: STACKEN@VERA (VMSmail: VERA::STACKEN)
UUCP: {seismo,mcvax}!enea!ttds!stacken
EAN: stacken@cs.kth.sunet
ARPA: enea!ttds!stacken@seismo.CSS.GOV
CSnet: enea.uucp!ttds!stacken@chalmers.csnet

Vårmötesprotokoll

PROTOKOLL fört vid Datorföreningen STACKENs vårmöte, avhållet torsdagen 1987-02-05 med början klockan 19, i sal V4 på KTH, belägen på Brinellvägen 32 i Stockholm.

Närvarande medlemmar i alfabetisk ordning:

Henrik Björkman, Gunnar Blomberg, Henning Croona, Björn Danielsson, Lars Ekström, Johnny Eriksson, Bengt Forsgren, Mats O Jansson, Carl-Arne Johannesson, Stellan Lagerström, Björn Levin, Erik Levlin, Per Lindberg, Peter Löthberg, Hans Nordström, Thomas Nyström, Jan Michael Rynning, Vicki Schalin, Gullik Webjörn, Anders Westerberg, Klaus Zeuge, Bengt Åhlin och Jan Åman.

§1. Mötets öppnande.

Stellan Lagerström hälsade alla välkomna och förklarade mötet öppnat.

§2. Val av justeringsmän.

Björn Levin och Henrik Björkman valdes till justeringsmän.

§3. Val av mötesordförande.

Stellan Lagerström valdes till mötesordförande.

§4. Val av mötessekreterare.

Mats O Jansson valdes till mötessekreterare.

§5. Tillkännagivande av uppgjord röstlängd.

De närvarande sade sina namn och prickades av på röstlängden.

§6. Frågan om mötet är stadgeenligt utlyst.

Mötet ansåg sig vara stadgeenligt utlyst.

§7. Fråga om dagordningens godkännande.

Dagordningen godkändes utan anmärkning.

§8. Verksamhetberättelse.

Den ena av de båda PDP-11/40:or vi fått under året används för att koppla

ihop Katia med resten av ANF-10-nätet och lystrar till nodnamnet Calle. Dess ankomst har även löst problemen med borttappade tecken och modemhandskakning mot KTHnet.

På den andra kör vi RSTS/E. Den har vi döpt till Elvira.

Kicki har fått en CPU till under året, från Universitetet i Århus, och är nu ett trippelprocessorsystem. Några av de RP02/03-skivminnen vi fick tillsammans med den har vi skänkt vidare till Unga Forskare i Handen.

Lokalfrågan har varit aktuell hela året. Då vi inte får behålla rummen i anslutning till datorhallen, så behöver vi nya lokaler. Vi har fått ett litet klubbrum på Osquars Backe 4, precis utanför E-Lab. Thomas Nyström har byggt styrelektronik till en av de kortläsare Jörgen Städje skaffade fram och Henrik Lind har monterat in ett ellås i dörren, så man ska kunna komma in med hjälp av magnetkort och fyrställig kod.

Vi har även blivit erbjudna ett garage med smörjgrop ute vid Vetrinärhögskolan, som förvaringsplats för reservdelar till datorerna, men det var vi tvungna att tacka nej till, då det inte fanns något användbart golvutrymme. Inkörning av datorprogram har vi hört talas om, men aldrig rundsmörjning av datorer.

Richard M Stallman var på besök i oktober, varvid föreningens medlemmar fick möjlighet att testa operativsystemet ITS. Mötet där RMS pratade om GNU är den mest välbesökta STACKEN-aktiviteten någonsin. Åhörarna fyllde KTH:s näst största föreläsningssal till bredden.

Två studiebesök har vi hunnit med under året. Det första var på flygsimulatorn och vindtunneln på KTH och det andra på Permobil ute i Åkersberga, där vi tittade på deras elektroniska konferens- och brevsystem Permobas.

§9. **Revisionsberättelse.**
Revisorerna rekommenderade mötet att bevilja den avgående styrelsen ansvarsfrihet. Peter Löthberg ansåg det lämpligt att STACKEN anlitar en revisionsbyrå för föreningens ekonomi. Mötet beslöt att så skulle ske. Se även bilagor.

§10. **Balansräkning.**
Se bilaga.

§11. Ansvarsfrihet för avgående styrelse.
Mötet beviljade den avgående styrelsen ansvarsfrihet.

§12. Övriga frågor.
Björn Levin förklarade varför alla hade fått inbetalningskort för 1987, även de som redan betalt. Alla konton på Kicki och Katia kommer att förlängas under februari.

Peter Löthberg berättade att det den 12 februari kommer en AMIS-annons i Industriell Datateknik. Han nämnde även att det behövs unika användarnamn för de mailsystem som Kicki kommer att kopplas till under året och att Stellan Lagerström hade blivit utsedd att vara projektledare för DECnet till Katia och VMSphone till Kicki.

Vicki Schalin framförde åsikten att medlemsavgiften var för hög. Mötet delade inte den åsikten. Kassören påpekade att medlemsavgiften bara täcker kostnaderna för STACKPOINTER.

§13. Mötets avslutande.
Stellan Lagerström förklarade mötet avslutat.

Vid protokollet

Mats O Jansson, mötessekreterare
Justeras

Björn Levin, justeringsman

Justeras

Stellan Lagerström, mötesordförande
Justeras

Henrik Björkman, justeringsman

DECsystem-10
DECSYSTEM-20 EUROCOPY

STACKEN Computer Club

From: Eurocopy Editor, Jan Michael Rynning, Department of Numerical Analysis and Computing Science, Royal Institute of Technology, Stockholm, Sweden.

STACKEN, the students' computer club at the Royal Institute of Technology, has obtained quite a repute for its "museum" of running DEC computers. People keep asking me how things are going, so here is a status report.

KA10

"Katia", the KA10 ("tia" is the Swedish word for "ten"), STACKEN's first mainframe which we bought from DEC for one Swedish krona, is now on the same ANF-10 network as "Kicki" and the other TOPS-10 computers. We have no plans for adding DECnet support to the 6.03A monitor, at least not until we get the BBN paging box, so that we can address more than 256 K words of memory.

Apart from a crashed RP04, which left us with no DEC supported disks on the system and forced us to add support for the disks that we had, it has been doing quite well. The maximum uptime so far witout a system reload is about 1250 hours.

KI10

"Kicki" is now a Tri-SMP KI10 running a slightly modified 7.02 monitor with TCP/IP and DECnet Phase IV, via a DDP device (synchronous or asynchronous DECnet through an ANF-10 DN87). There were some loose ends in the DDP code, which made the 10 fall off the net from time to time and occasionally crashed the system, so we "brought the man in from the factory". We invited Bob Houk (RDH), who wrote the DDP code. He is now here for a week to help us iron the last bugs out of it.

The "?KL10 only instruction at user PC ..." error message has been replaced by a KL10 instruction emulator, which also required some rewireing of the backplane to trap the ADJBP instruction. The person who decided to reuse the IBP opcode for ADJBP should be tared and feathered!

KL10, KS10, and PDP-6

When we get a KL10 we will run TOPS-20 on it and call it "Kleo".

The Incompatible Timesharing System (ITS) from the MIT AI Lab has been up and running on one of the 2020's at the RIT for about a year, at nights and in October '86 when Richard M. Stallman (RMS) came to visit the Computer Club, and to talk about the GNU Project. When we get a KS10 of our own we shall of course run ITS on it. Like the ITS systems at the MIT it will have a two letter name, "SI" for STACKEN ITS.

A PDP-6 (the predecessor of the PDP-10's, with the same architecture) would make the collection complete. Do you know of anyone who has a left over PDP-6 to give away?

Small Computers

Our PDP-11/40 running RSTS/E is named "Elvira" in commemoration of the first computer at the RIT to which the students had unlimited access. Before long we shall probably upgrade it to an 11/60, and connect it to the national university DECnet network.

The Lunar Landing Game is our main incentive to get the PDP-15 up and running. We set some new members to the task, to guarantee the continued existence of the service department, and they love playing with it.

They are great toys these old machines ...

MINIX: A Cheap UNIX Clone with Source Code

Introduction

have recently finished rewriting UNIX from scratch. This system, called MINIX, does not contain even a single line of AT&T code, so it can be distributed with source code. It runs on the IBM PC, XT, and AT and those clones that are 100% hardware compatible (not all, unfortunately). To the average, unsophisticated user, using MINIX is indistinguishable from using V7 UNIX. Experts will notice that some relatively less commonly used programs and features are missing.

MINIX FEATURES:

- System call compatible with V7 UNIX (except for a couple of minor calls)
- Kernighan and Ritchie compatible C compiler is included
- Shell that is functionally identical to the Bourne shell is included
- Full multiprogramming (fork+exec; background jobs in shell: cc file.c &)
- Full screen editor vaguely inspired by emacs (modeless, autoinsert, etc.)
- Over 60 utilities (cat, cp, grep, ls, make, mount, sort, etc.)
- Over 100 library procedures (atoi, fork, malloc, stdio, strcmp, etc.)
- Supports a hard disk, but also works quite well with just floppies
- Contains programs to read and write MS-DOS diskettes
- Full operating system source code is included
- Source code for all the utilities (ex-

cept C compiler) is included

- System will recompile itself (requires 640K and 2 floppies or 1 hard disk)
- C compiler source is available as a separate package
- Kernel organization radically different from UNIX and much more modular
- Software is not copy protected

Furthermore, I have written a 719 page book telling you everything you ever wanted to know about operating systems in general and this one in particular. The book contains the manual pages, an appendix describing how to recompile the system from the sources supplied, a full source code listing of the operating system (253 pages), and a cross reference map.

The software is available in 4 packages (book is separate):

- Box of eight 360K diskettes for 640K IBM PCs (512K is sort of ok too)
- Box of eight 360K diskettes for 256K IBM PCs (no C compiler)
- Box of five 1.2M diskettes for the IBM PC-AT
- 9 track industry standard tape (1600 bpi, tar format)

All four distributions contain the full source code, about 54,000 lines, (kernel + utilities, except the compiler), virtually all of it in C. The source code for the C compiler is also available separately (as described in the book). The C compiler is NOT based on pcc at all. It is based on ACK (see Communications of the ACM, Sept. 1983, pp. 654–660). The following programs are included, among others. Like the kernel, these have all been rewritten from scratch by me, my students, people I paid to write them or in a small number of cases, were donated by other people to whom I am grateful:

ar basename cat cc chmod chown cmp comm cp date dd df dosread echo grep gres head kill ln login lpr ls make mkdir mkfs mknod mount mv od passwd pr pwd rev rm rmdir roff sh shar size sleep sort split stty su sum sync tail tar tee time touch tr true umount uniq update wc

The book and software are being sold by Prentice-Hall. They are NOT public domain. However, the publisher does not object to people making a limited number of copies of the software for noncommercial use. For example professors may make copies of the software for their students. Universities may exchange modified versions. You may make a few copies for your friends etc. If you want to port the soft-

ware to other CPUs and sell it, you need permission from Prentice-Hall, but they will not be unreasonable. To acquire the software, go to any bookstore and ask them to order the book for you:

Title: Operating Systems: Design and Implementation
Author: Andrew S. Tanenbaum
Publisher: Prentice-Hall (Jan. 1987)
ISBN: 0-13-637406-9

In the book you will find a postcard that you can use to order the software. Please don't ask me for the software. I have already spent approximately 8000 hours over the past 5 years writing it; I don't want to spend the next 5 years duplicating floppy disks. The book costs about $35. The software is $79.95 per set, including the source code. I hope most people will consider $79.95 for the binaries and sources of something almost functionally equivalent to UNIX as being reasonable. I know of no other software package where you get 54,000 lines of source code for this price. As bugs are reported, I will send Prentice Hall new disks, so that the version they sell will remain up to date. (This also provides some incentive to buy rather than copy.)

For those of you going to USENIX or UNIFORUM in Washington, D.C. January 20–23, Prentice-Hall will have a stand at the show where you can play with the software. You can also buy the stuff there, but since the P-H people drive to Washington in their own cars, they have a limited carrying capacity and they are only taking 50 copies, so get there early the first day.

If anyone is interested, we could set up a newsgroup comp.os.minix to discuss minix, report bug fixes, distribute updates of individual files etc. This letter is being multiply posted to several newsgroups. I propose that the initial discussion take place in comp.sys.ibm.pc (subject: MINIX) to avoid having it spread all over the place. Besides, the only other newsgroup I read is mod.recipes. I don't think the moderator will go for floppy disk with Hollandaise sauce.

Andrew S. Tanenbaum

TOPS är bra att peta bor VAX med...

TMP

MINIX: A Cheap UNIX Clone with Source Code

System Architecture

Overview of the MINIX System Architecture

UNIX is organized as a single executable program that is loaded into memory at system boot time and then run. MINIX is structured in a much more modular way, as a collection of processes that communicate with each other and with user processes by sending and receiving messages. There are separate processes for the memory manager, the file system, for each device driver, and for certain other system functions. This structure enforces a better interface between the pieces. The file system cannot, for example, accidentally change the memory manager's tables because the file system and memory manager each have their own private address spaces.

These system processes are each full-fledged processes, with their own memory allocation, process table entry and state. They can be run, blocked, and send messages, just as the user processes. In fact, the memory manager and file system each run in user space as ordinary processes. The device drivers are all linked together with the kernel into the same binary program, but they communicate with each other and with the other processes by message passing.

When the system is compiled, four binary programs are independently created: the kernel (including the driver processes), the memory manager, the file system, and init (which reads /etc/ttys and forks off the login processes). In other words, compiling the system results in four distinct a.out files. When the system is booted, all four of these are read into memory from the boot diskette.

It is possible, and in fact, normal, to modify, recompile, and relink, say, the

file system, without having to relink the other three pieces. This design provides a high degree of modularity by dividing the system up into independent pieces, each with a well-defined function and interface to the other pieces. The pieces communicate by sending and receiving messages.

The various processes are structured in four layers:

4. The user processes (top layer).
3. The server processes (memory manager and file system).
2. The device drivers, one process per device.
1. Process and message handling (bottom layer).

Let us now briefly summarize the function of each layer.

Layer 1 is concerned with doing process management including CPU scheduling and interprocess communication. When a process does a SEND or RECEIVE, it traps to the kernel, which then tries to execute the command. If the command cannot be executed (e.g., a process does a RECEIVE and there are no messages waiting for it), the caller is blocked until the command can be executed, at which time the process is reactivated. When an interrupt occurs, layer 1 converts it into a message to the appropriate device driver, which will normally be blocked waiting for it. The decision about which process to run when is also made in layer 1. A priority algorithm is used, giving device drivers higher priority over ordinary user processes, for example.

Layer 2 contains the device drivers, one process per major device. These processes are part of the kernel's address space because they must run in kernel mode to access I/O device registers and execute I/O instructions. Although the IBM PC does not have user mode/kernel mode, most other machines do, so this decision has been made with an eye toward the future. To distinguish the processes within the kernel from those in user space, the kernel processes are called tasks.

Layer 3 contains only two processes, the memory manager and the file system. They are both structured as servers, with the user processes as clients. When a user process (i.e., a client) wants to execute a system call, it calls, for example, the library procedure read with the file descriptor, buffer, and count. The library procedure builds a message containing the system call number and the parameters and sends it to the file system. The client then blocks waiting for a reply. When the file system receives the message, it carries it out and sends back a reply containing the number of bytes read or the error code. The library procedure gets the reply and returns the result to the caller in the usual way. The user is completely unaware of what is going on here, making it easy to replace the local file system with a remote one.

Layer 4 contains the user programs. When the system comes up, init forks off login processes, which then wait for input. On a successful login, the shell is executed. Processes can fork, resulting in a tree of processes, with init at the root. When CTRL-D is typed to a shell, it exits, and init replaces the shell with another login process.

Layer 1: Processes and Messages

The two basic concepts on which MINIX is built are processes and messages. A process is an independently schedulable entity with its own process table entry. A message is a structure containing the sender's process number, a message type field, and a variable part (a union) containing the parameters or reply codes of the message. Message size is fixed, depending on how big the union happens to be on the machine in question. On the IBM PC it is 24 bytes.

Three kernel calls are provided:

- RECEIVE(source, &message)
- SEND(destination, &message)
- SENDREC(process, &message)

These are the only true system calls (i.e., traps to the kernel). RECEIVE announces the willingness of the caller to accept a message from a specified process, or ANY, if the RECEIVER will accept any message. (From here on, "process" also includes the tasks.) If no message is available, the receiving process is blocked. SEND attempts to transmit a message to the destination process. If the destination process is currently blocked trying to receive from the sender, the kernel copies the message from the sender's buffer to the receiver's buffer, and then marks them both as runnable. If the receiver is not waiting for a message from the sender, the sender is blocked.

The SENDREC primitive combines the functions of the other two. It sends a message to the indicated process, and then blocks until a reply has been received. The reply overwrites the original message. User processes use SENDREC to execute system calls by sending messages to the servers and then blocking until the reply arrives.

There are two ways to enter the kernel. One way is by the trap resulting from a process' attempt to send or receive a message. The other way is by an interrupt. When an interrupt occurs, the registers and machine state of the currently running process are saved in its process table entry. Then a general interrupt handler is called with the interrupt number as parameter. This procedure builds a message of type INTERRUPT, copies it to the buffer of the waiting task, marks that task as runnable (unblocked), and then calls the scheduler to see who to run next.

The scheduler maintains three queues, corresponding to layers 2, 3,

and 4, respectively. The driver queue has the highest priority, the server queue has middle priority, and the user queue has lowest priority. The scheduling algorithm is simple: find the highest priority queue that has at least one process on it, and run the first process on that queue. In this way, a clock interrupt will cause a process switch if the file system was running, but not if the disk driver was running. If the disk driver was running, the clock task will be put at the end of the highest priority queue, and run when its turn comes.

In addition to this rule, once every 100 msec, the clock task checks to see if the current process is a user process that has been running for at least 100 msec. If so, that user is removed from the front of the user queue and put on the back. In effect, compute bound user processes are run using a round robin scheduler. Once started, a user process runs until either it blocks trying to send or receive a message, or it has had 100 msec of CPU time. This algorithm is simple, fair, and easy to implement.

Layer 2: Device Drivers

Like all versions of UNIX for the IBM PC, MINIX does not use the ROM BIOS for input or output because the BIOS does not support interrupts. Suppose a user forks off a compilation in the background and then calls the editor. If the editor tried to read from the terminal using the BIOS, the compilation (and any other background jobs such as printing) would be stopped dead in their tracks waiting for the the next character to be typed. Such behavior may be acceptable in the MS-DOS world, but it certainly is not in the UNIX world. As a result, MINIX contains a complete set of drivers that duplicate the functions of the BIOS. Like the rest of MINIX, these drivers are written in C, not assembly language.

This design has important implications for running MINIX on PC clones. A clone whose hardware is not compatible with the PC down to the chip level, but which tries to hide the differences by making the BIOS calls functionally identical to IBM's will not run an unmodified MINIX because MINIX does not use the BIOS.

Each device driver is a separate process in MINIX. At present, the drivers include the clock driver, terminal driver, various disk drivers (e.g., RAM disk, floppy disk), and printer driver. Each driver has a main loop consisting of three actions:

1. Wait for an incoming message.
2. Perform the request contained in the message.
3. Send a reply message.

Request messages have a standard format, containing the opcode (e.g., READ, WRITE, or IOCTL), the minor device number, the position (e.g., disk

block number), the buffer address, the byte count, and the number of the process on whose behalf the work is being done.

As an example of where device drivers fit in, consider what happens when a user wants to read from a file. The user sends a message to the file system. If the file system has the needed data in its buffer cache, they are copied back to the user. Otherwise, the file system sends a message to the disk task requesting that the block be read into a buffer within the file system's address space (in its cache). Users may not send messages to the tasks directly. Only the servers may do this.

MINIX supports a RAM disk. In fact, the RAM disk is always used to hold the root device. When the system is booted, after the operating system has been loaded, the user is instructed to insert the root file system diskette. The file system then sees how big it is, allocates the necessary memory, and copies the diskette to the RAM disk. Other file systems can then be mounted on the root device.

This organization puts important directories such as /bin and /tmp on the fastest device, and also makes it easy to work with either floppy disks or hard disks or a mixture of the two by mounting them on /usr or /user or elsewhere. In any event, the root device is always in the same place.

In the standard distribution, the RAM disk is about 240K, most of which is full of parts of the C compiler. In the 256K system, a much smaller RAM disk has to be used, which explains why this version has no C compiler: there is no place to put it. (The / usr diskette is completely full with the other utility programs and one of the design goals was to make the system run on a 256K PC with 1 floppy disk.) Users with an unusual configuration such as 256K and three hard disks are free to juggle things around as they see fit.

The terminal driver is compatible with the standard V7 terminal driver. It supports cooked mode, raw mode, and cbreak mode. It also supports several escape sequences, such as cursor positioning and reverse scrolling because the screen editor needs them.

The printer driver copies its input to the printer character for character without modification. It does not even convert line feed to carriage return + line feed. This makes it possible to send escape sequences to graphics printers without the driver messing things up. MINIX does not spool output because floppy disk systems rarely have enough spare disk space for the spooling directory. Instead one normally would print a file f by saying

```
lpr <f &
```

to do the printing in the background.

The lpr program insert carriage returns, expands tabs, and so on, so it should only be used for straight ASCII files. On hard disk systems, a spooler would not be difficult to write.

Layer 3: Servers

Layer 3 contains two server processes: the memory manager and the file system. They are both structured in the same way as the device drivers, that is a main loop that accepts requests, performs them, and then replies. We will now look at each of these in turn.

The memory manager's job is to handle those system calls that affect memory allocation, as well as a few others.

These include FORK, EXEC, WAIT, KILL, and BRK. The memory model used by MINIX is exceptionally simple in order to accommodate computers without any memory management hardware. When the shell forks off a process, a copy of the shell is made in memory. When the child does an EXEC, the new core image is placed in memory. Thereafter it is never moved. MINIX does not swap or page.

The amount of memory allocated to the process is determined by a field in the header of the executable file. A program, chmem, has been provided to manipulate this field. When a process is started, the text segment is set at the very bottom of the allocated memory area, followed by the data and bss. The stack starts at the top of the allocated memory and grows downward. The space between the bottom of the stack and the top of the data segment is available for both segments to grow into as needed. If the two segments meet, the process is killed.

In the past, before paging was invented, all memory allocation schemes worked like this. In the future, when even small microcomputers will use 32-bit CPUs and 1M×1 bit memory chips, the minimum feasible memory will be 4 megabytes and this allocation scheme will probably become popular again due to its inherent simplicity. Thus the MINIX scheme can be regarded as either hopelessly outdated or amazingly futuristic, as you prefer.

The memory manager keeps track of memory using a list of holes. When new memory is needed, either for FORK or for EXEC, it searches the hole list and takes the first hole that is big enough (first fit). When a process terminates, if it is adjacent to a hole on either side, the process' memory and the hole are merged into a bigger hole.

The file system is really a remote file server that happens to be running on the user's machine. However it is straightforward to convert it into a true network file server. All that needs to be done is change the message inter-

face and provide some way of authenticating requests. (In MINIX, the source field in the incoming message is trustworthy because it is filled in by the kernel.) When running remote, the MINIX file server maintains state information, like RFS and unlike NFS.

The MINIX file system is similar to that of V7 UNIX. The i-node is slightly different, containing only 9 disk addresses instead of 13, and only 1 time instead of 3. These changes reduce the i-node from 64 bytes to 32 bytes, to store more i-nodes per disk block and reduce the size of the in-core i-node table.

Free disk blocks and free inodes are kept track of using bit maps rather than free lists. The bit maps for the root device and all mounted file systems are kept in memory. When a file grows, the system makes a definite effort to allocate the new block as close as possible to the old ones, to minimize arm motion. Disk storage is not necessarily allocated one block at a time. A minor device can be configured to allocate 2, 4 (or more) contiguous blocks whenever a block is allocated. Although this wastes disk space, these multiblock zones improve disk performance by keeping file blocks close together. The standard parameters for MINIX as distributed are 1K blocks and 1K zones (i.e., just 1 block per zone).

MINIX maintains a buffer cache of recently used blocks. A hashing algorithm is used to look up blocks in the cache. When an i-node block, directory block, or other critical block is modified, it is written back to disk immediately. Data blocks are only written back at the next SYNC or when the buffer is needed for something else.

The MINIX directory system and format is identical to that of V7 UNIX. File names are strings of up to 14 characters, and directories can be arbitrarily long.

Layer 4: User Processes

This layer contains init, the shell, the editor, the compiler, the utilities, and all the user processes. These processes may only send messages to the memory manager and the file system, and these servers only accept valid system call requests. Thus the user processes do not perceive MINIX to be a general-purpose message passing system. However, removing the one line of code that checks if the message destination is valid would convert it into a much more general system (but less UNIX-like).

Andrew S. Tanenbaum

Hackerbladen

HACKERBLADEN, omtrycket av Hackerbladet 1–32 finns nu färdigt. (Men Hackerbladet nummer 1 är fortfarande ett samlarobjekt, förstås!) Priserna nedan täcker bara tryckkostnader och porto. Inom Sverige går det utmärkt att beställa genom att sätta in pengarna på postgiro 530423-0039.

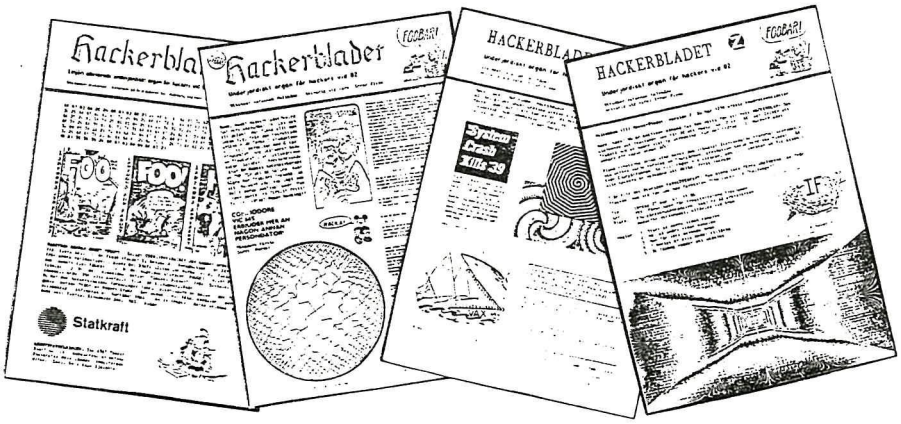
HACKERBLADEN, the reprint of Hackerbladet number 1–32 is now available. (But Hackerbladet number 1 still is a collector's item, of course!) The prices below only cover printing costs and mailing. Cheques must be payable at a Swedish bank.

64 pages of hacks, flames, humor, and nonsense!

Hackerbladen 1–32:	30 SEK	(5 USD)	(700 JPY)
Mail inside Nordic countries:	5 SEK		
Mail outside Nordic countries:	15 SEK	(2 USD)	(400 JPY)
Add for air mail outside Europe:	6 SEK	(1 USD)	(100 JPY)

(Oh well, it adds up roughly equivalent, anyway. Interger arithmetrix does not blend well with echonomix or other real-world phenomena.)

Per Lindberg, Skånegatan 68 A$^{5\,tr}$, S-116 32 STOCKHOLM, SWEDEN.



Hackerbladen

De 32 första numren av Hackerbladet

Sveriges Tekniska Attachéer
Box 5282
102 46 Stockholm
Tel. 08-63 53 90

Författare: Lars Berggren
Utskriftsdatum: 860220

NOTIS nr: U2-86-015
Ämneskoder: E

Källmaterial sänt till Stockholm: JA
Källmaterial: InfoWorld, 860120
Press Release, 851218

Imponerande programvara

Ytterligare en kombinationsprogramvara har nått marknaden. "Look Busy" integrerar ordbehandlare, spreadsheet, grafik, almanacka och projektplanerare till det svårslagna priset $19.95.

Programmet går att ladda i alla IBM-kompatibla persondatorer - men så mycket mer går inte att göra med det. Fake Software Emporium har nämligen specialiserat sig på att ta fram programvara så att du kan imponera på din omgivning utan att ha använt en dator tidigare. Kalkylprogramvaran innehåller intetsägande tabeller, som lätt konverteras till tjusiga grafer visande en positiv trend. Projektplaneraren hjälper dig att förklara för dina medarbetare varför projekten går snett. Almanackan är fullbokad med viktiga möten.

Företaget tänker följa upp med "Look Organized", en imponatordatabas, och "Look Intelligent", för dig som behöver en AI-programvara.

Kommentar: Lämplig present till medarbetare som har en persondator men inte vet vad han/hon ska göra med den.

Dialog med UNIX

USER:	^saccarine	gotta light?	make love
UNIX:	Bad substitute.	No match.	Don't know how to make love. Stop.

Speciella nyheter

MORE COMPANIES HOP ON NONSOURCING BANDWAGON

Joining the ranks of Standard Pseudocircuits Inc and International Quasiconductor Corp, Lackof Silicon Inc, National Nonconductor Ltd, and Where's the Beef IC Corp have recently become nonproducers of semicustom ICs. Under the terms of the non-sourcing agreement with SPI and IQC, the three new nonproducers agree not to source the semicustom circuits the two original nonmanufacturers are not making. The semi-custom circuits the five manufacturers are not producing are drop-out compatible and reflect a noninterchange of technology and masks. Explains SPI's president Jeff Berman, the addition of the three firms to the roll of semicustom-IC nonproducers can only be beneficial because it will further narrow the market for the non-ICs by spread-ing the common noncustomer base among the five nonmanufacturers.—Bill Travis

RETROACTIVE LAYOFFS IMPROVE COMPANY'S BOTTOM LINE

In a departure from normal industry practice, γ-Ray Corp today announced a plan of retroactive layoffs. Effective immediately, the company is laying off employees who were working for the company as of January 1, 1983, but who subsequently left. Employees retroactively laid off are expected to repay wages and benefits they earned between January 1, 1983, and when they left the company. Current employees aren't affected by the retroactive layoffs. —Jonathan Titus

MEMORY MAKER CUTS PRICES TO BONE

American Process Engineering's president Jerry Fulbright has announced that his company has reached bottom in its learning curve for the production of 64k-byte RAMs. His company's manufacturing process is so efficient, claims Fulbright, that it costs −50¢ to produce each RAM. APE has adjusted its prices accordingly — in OEM quantities, the memories are priced at −25¢ each. Note, however, that customers must pay shipping costs, which are not insignificant because of the weight of the quarter taped to each RAM. The company plans to start shipping 256k-byte units in the second half of 1986; initial price will be 0¢ (OEM qty).—Bill Travis

HARRIS AND RAYTHEON STILL EXCHANGING OBSOLESCENCES

The Harris/Raytheon exchange agreement for obsolete op amps and vacuum tubes recently mourned its third anniversary. According to Harris's Paul Smith, the produc-tion ramp-up for 6SN7 dual triodes has reached completion; the company is now producing dozens of the devices per week. What's more, the firm soon plans to an-nounce a shrink version in LCC (lead-crystal case) form. Expect, moreover, a break-through in the second half of this year: Harris's mastery of 12.6V-filament technology is almost complete. The company should see first vacuum for a 12SN7 by the third quarter of this year. Results are equally satisfying for Raytheon: The company now has the capability to produce — in volumes of hundreds — the once-industry-standard 709 operational amplifier.—Bill Travis

IBM DISCONTINUES TIMEX/SINCLAIR LOOKALIKE

Industry watchers, listeners, and smellers were stunned last week at IBM's an-nounced withdrawal from the Timex/Sinclair-clone market. Manufacturing of the PCtot by subcontractor Joe's Garage (Sleazonia, WV) will cease by the end of April. According to IBM, however, the cessation of production does not mean the company is abandoning the PCtot. A spokesman said, "We'll continue to market the machine; we're confident that customer demand will remain strong for the dozens of unsold machines in inventory. As for the six people who have already bought PCtots, let me reassure them they'll continue to receive full support from Joe in West Virginia." —Bill Travis

Redaktörn

PÅ en dataavdelning är det tidvis mycket systemarbete. Som kan påverka driften för övrig personal. Ofta kommer då frågan "Hur länge dröjer det innan datorn kommer igång igen". Efter flera kraftiga underskattningar i samband med arbete på datorerna, har ett nytt uttryck myntats på min arbetsplats:

"Hur länge dröjer ..."
"Vi tror ca 1 timme."
"Är det en riktig timme eller en datatimme?"

Så var det dags för anskaffning av lite disketter. För att stödja svensk industri och lite med tanke på priset, köptes 100 blå disketter från Kopparberg. Det skulle bli intressant att prova kvalitén på dessa. Jag formatterade 50 i taget på firmans Copam PC-401/XT. En, jag säger 1 diskett ville inte på direkten. Jag fick "5120 bytes in bad sector" eller något liknande. Men så formatterade jag en annan och provade igen, och då gick det. Det blir summa sumarum 100%. Hur de håller i längden får jag väl återkomma med. Dock finns det ett stort "?" när det gäller Kopparberg. Disketterna levereras i diskettkuvert. Dessa är urstansade i kartong som inte är efterputsad. Så det finns väldigt mycket pappersdamm som virvlar omkring. Troligtvis var det orsaken till det tillfälliga "bad sector"-meddelandet. Hur man kan leverera disketterna med denna dammiljö, förstår jag inte. Eller överdriver jag risken för skador från pappersdammet?

På tal om vad som kan hända. Det finns dom som lär sig av sina misstag. I somras rasade skivminnet på bokföringspecen hos en av hamburgerleverantörerna i stan. Det blev flera tidiga morgnar för dom att stansa in det som försvann i raset. Senare på hösten så togs en tapestreamer till påseende. Efter två dagar kom nästa ras. (Inte pga streamern, dock.) Skillnaden var att denna gången hade de använt streamern. Efter att ett utbytesskivminne hade blivit insatt så tog det tio minuter för inladdningen. Så var de igång igen. Och fick sova på morgnarna.

En annan som har också har lärt sig, är allas vårt kära Televerk. De samlar alla abonnemangens kopplingsvägar i en bauta-databas. Så blev det dags att byta till fräschare skivminnen. Fram med säkerhetskopian och tanka över till de nya skivminnena. Hej vad det går. Av säkerhetsskäl skrotas sedan de gamla skivminnena direkt. Plötsligt så upptäcks att abonnentkopplingar i två städer fattas. Det är ung 375000 kopplingsdata. Hur? Det var det lilla det,

att ingen hade jämfört säkerhetskopian med originalet. Och därför inte kunnat se att dessa data aldrig kom med till säkerhetskopian. En (1) rad i styrprogrammet hade av misstag, vid ett tidigare tillfälle, försvunnit!? I rekonstrueringsarbetet visade sig, de annars så förhatliga pyjamaslistorna vara värd sin vikt i guld.

Har du inte jämfört, så vet du inte om en kopia är en kopia.

Ja, det var det hele.

/hn

Regnecentralen var tidigt ute efter kriget. DASK eller Dansk Sekvens Datamaskin var RCs första stora produkt som lade grunden till ett storföretag.

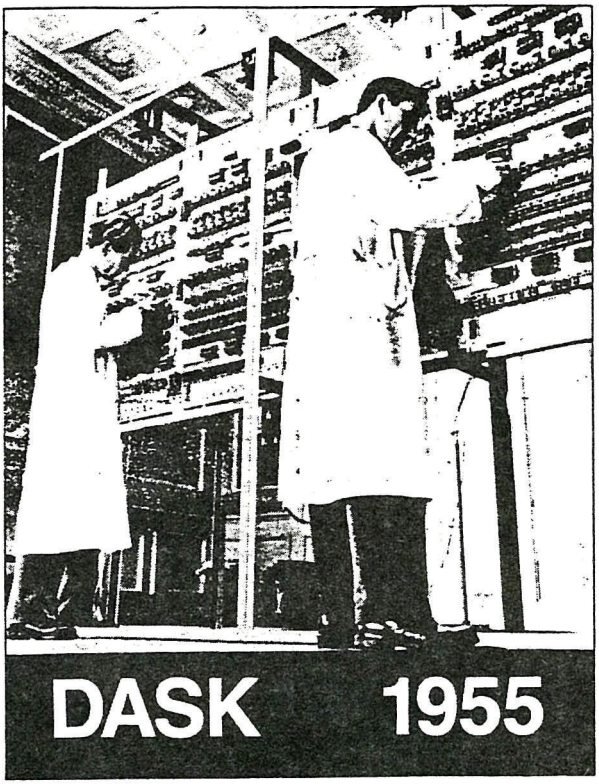
Regnecentralen startades 1949, och anledningen var att man i Danmark tyckte att det behövdes ett företag som höll sig a jour med all ny teknik som kom efter kriget. Företaget utvecklades och 1958 byggde man Danmarks första dator kallad DASK efter Dansk Sekvens Datamaskin.

I sin typ påminde den mycket om den svenska BESK, som kom vid samma tidpunkt.

1960 byggde man en dator speciellt för det danska geotekniska institutet, och den blev så lyckad att den kunde serieproduceras. Därmed var man på allvar inne i datorbranschen.

I dag ägs Regnecentralen av olika företag och institutioner. Det är pensionsfonder, Danmarks nationalbank, de tre teleförvaltningarna samt ITT, som har 26 procent av aktierna. Till det kommer vissa mindre ägare.

Företaget finns inte på börsen och dotterbolag finns i Västtyskland och England. Den samlade personalstyrkan ligger på 725 personer. Omsättningen 1985 var runt 500 miljoner, men vinsten var låg, endast 10 miljoner. Det var lägre än 1984 och förklaringen var problem med underleverantörerer, vilka inte hållit vad de lovat, uppger man.



Rör, dä ä fint dä. Vad är en 68000 mot en container med ECC 85? På den tiden hade datorerna sin verkliga status: "Endast för farbröder i vit rock".

Jörgen

Old benchmark—revisited

SÅ HÄR i MÄSS-tider är det alltid roligt att gå omkring och se vad de utställda små burkarna klarar av. Som t ex när IBM introducerade sin PC. Vi var där och fann att dess mBASIC bara var ca 20% snabbare än den basicprogrammerbara fickräknaren som visades i montern brevid, vilket vi givetvis påpekade för försäljaren. Han försvarade sig med ungefär: 'Jo förstår ni, den här datorn ska man inte köra BASIC på den är mera avsedd för ordbehandling... osv...'.

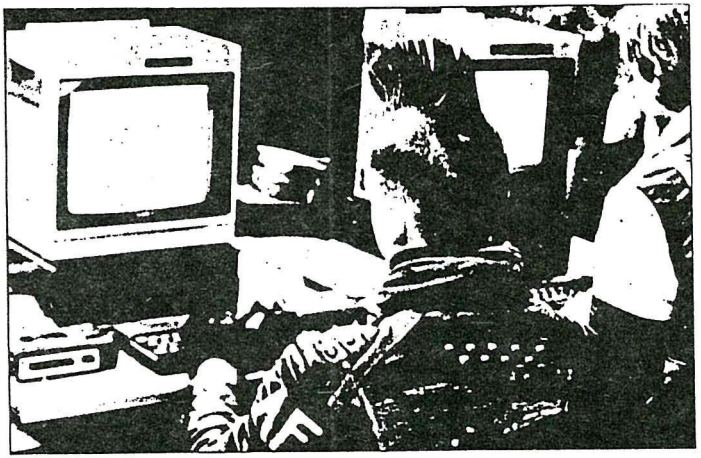
Det här bänkmärket provar ju visserligen bara processorns flyttalsberäkningskapacitet, men det är tillräckligt litet och enkelt för att snabbt kunna knappas in på en mässa etc:

```
10 S=0 : P=0
20 S=S+1 : P=P+1/S
30 IF P<9 THEN 20
40 PRINT S,P
```

Den inre slingan skall normalt genomlöpas 4550 gånger, vilket ger 18200 flyttalsoperationer. Jag säger normalt 4550 gånger, för i tidernas begynnelse var det där med standardiserad avrundning, precision etc inte så noga.

Med tiden så har listan på testade datorer blivit tämligen lång, men bidrag till den mottages tacksamt. T ex så saknar jag fortfarande tiden för testet i Cray-FORTRAN.

Thord Nilson



HACKING I NITAR OCH LEDUR! (Spectrum tycks det.) Jörgen

Uppmätta tider

 ÄR är de intressantaste tiderna ur nuvarande lista. Tiderna är angivna i sekunder CPU-tid per varv (ett varv=18200 FP-operationer) där ej annat anges.

Dator	Språk	Enkel prec	C:a 10 siffror	Dubbel prec
SHARP 1200	BASIC		c:a 37 min	
SHARP 1500	BASIC		c:a 4 min	
HP 41C	maskinkod		c:a 32 min	
Olivia (8080A	TCL Pascal	50.		
med CP/M)	mBASIC	65.		
TRS80 M-II	Pascal MT+	13.		
I*M PC	mBASIC	45.		
Victor PC2/V30	turbo Pascal		8.7	
ICL perq	Pascal	1.8		
HP 9830	BASIC		4–5 min	
HP 9845 A	BASIC		21.0	
HP 9816	BASIC			5.5
ALFA LSI	BASIC	c:a 45.0		
PDP 11/23	FORTRAN	1.5		2.0 (med FP)
	intrpr. BASIC+			7.6 (med FP)
	intrpr. BASIC+			21.0 (utan FP)
PDP 11/45	intrpr. BASIC+			3.7
PDP 11/73	FORTRAN	0.4		0.6
	BASIC+			2.5
Mikro-VAX II	VAX BASIC	0.08		0.12
VAX 11/780	VAX BASIC	0.06		0.10
VAX 11/785	VAX BASIC	0.045		0.08
KI-10 (KICKI)	MACRO-10			0.10 (72bit)
	FORTRAN	0.14		
	PASCAL	0.23		
DEC-2065 (Vera)	BASIC	0.073		
	FORTRAN	0.049		

Thord Nilson

Programlistningar

Diverse versioner av testprogrammet för PDP-10-datorer:

Pascal

```
program  stest ;

var      s, p : real ;
         i    : integer ;

begin
  for i := 1 to 100 do
  begin
    s := 0;
    p := 0;
    while p < 9 do begin
      s := s + 1 ;
      p := p + 1/s ;
    end ;
  end ;
  writeln (tty, s, p);

end.
```

FORTRAN

```
C
C       TEST OF F-N....
C
        REAL S,P

C       Vill ha lite medelv{rdesbildning....
        DO 20 I=1,100
        S=0
        P=0
10      S=S+1
        P=P+1/S
        IF (P.LT.9.) GOTO 10

20      CONTINUE
        WRITE (5,99) S,P

99      FORMAT (1X,7HANSWER:,2F10.5)
        STOP

        END
```

Macro-10

```
        title   test
        search  uuosym

slask==5
p==6
s==7

start:  outstr  hopp
        inchrw
        setz    s,
        setz    p,
loop:   fadr    s,ett
        move    slask,ett
        fdvr    slask,s
        fadr    p,slask
        camge   p,nio

        jrst    loop
        outstr  klar
        exit    1,

ett:    1.0
nio:    9.0

hopp:   asciz   "Hej! Ge ett tecken!"
klar:   asciz   "F{rdig!"

        end     start
```

Och så Turbo-Pascal versionen för I*M PC-komaptibla burkar, av Nils Segerdahl:

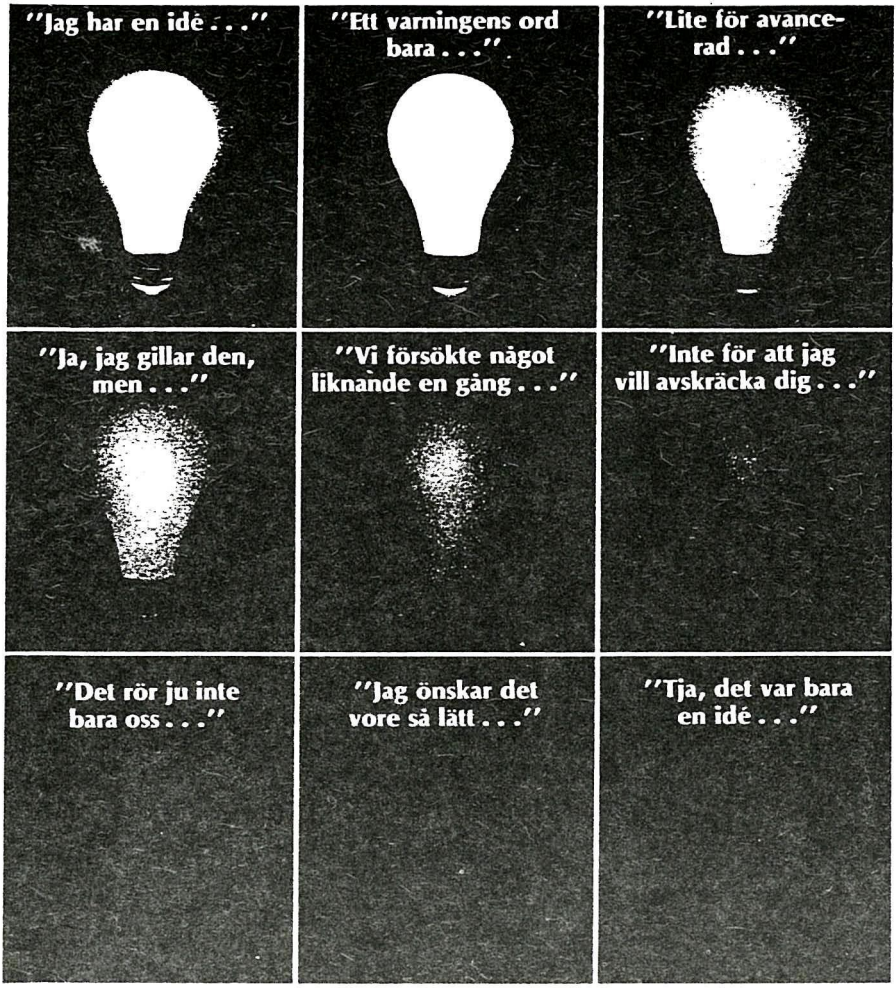
Turbo-Pascal

```
(*************************************************************************)
(*SC-                                                                    *)
(*  SYSCALL              Does diffrent system call                       *)
(*                                                                       *)
(*************************************************************************)

type  datestr = string[10];
  timestr = string[12];
  TregPack=
    record
      AX,BX,CX,DX,BP,SI,DI,DS,ES,Flags : Integer;
    end;

var

  totsec,p,s   : real;
```

```
  RecPack               : TregPack;
  hour,min,sec,hund     : string[2];
  hour2,min2,sec2,hund2 : string[2];
  m2,s2,h2,h1,m1,s1,tmp,i,dummy          : integer;

function time:timestr;
var
  ah,al,ch,cl,dh,dl : byte;
  recpack        : Tregpack;
begin
  ah:=$2C;
  with recpack do
  begin
    ax:=ah shl 8+ al;
  end;
  intr($21,recpack);

  with recpack do
  begin
    str(cx shr 8,hour);   if cx shr 8 < 10 then hour:='0'+hour;
    str(cx mod 256,min);  if cx mod 256 < 10 then min:='0'+min;
    str(dx shr 8,sec);    if dx shr 8 < 10 then  sec:= '0'+sec;
    str(dx mod 256,hund); if dx mod 256 < 10 then hund:='0'+hund;
 end;
time:=hour+':'+min+':'+sec+':'+hund;
end;

begin
  p:=0;s:=0;
  writeln(' Thord NILsons standardtest av datorer ++ PC-versionen');
  Writeln(' Det tar n}gra minuter. jag tar medelv{rdet av 10 varv');
  writeln(' klockan {r nu ',time);min2:=min;sec2:=sec;
  for i:=1 to 10 do begin
    p:=0;s:=0;
    while p<9 do begin
      s:=s+1;p:=p+1/s;
    end;
  end;
  writeln(time);
  val(min,m1,dummy);
  val(sec,s1,dummy);
  val(min2,m2,dummy);
  val(sec2,s2,dummy);
  val(hund,h1,dummy);
  val(hund2,h2,dummy);
  totsec:=(((m1*60)+s1)*100+h1)-(((m2*60)+s2)*100+h2);
  totsec:=totsec/100;
  totsec:=totsec/10;
  clrscr;
  writeln('--------Thord Nilson standardtest av datorer----------------------------------');
  writeln('Dator            spr}k       prec:   enkel    ca 10 siffror    dubbel');
  writeln('-------------------------------------------------------------------------------');
  writeln('Olivia           TCL Pascal           50.0');
  writeln('                 mBASIC               65.0');
  writeln('Victor PC2/V30   turbo Pascal                  8.7');
  writeln('ICL perq         Pascal                1.8');
  writeln('HP 9845 A        BASIC                        21.0 ');
  writeln('ALFA LSI         BASIC              ca 45.0');
  writeln('HP 9830          BASIC                       4-5 min ');
  writeln('mikro-VAX II     komp BASIC            0.08                    0.12 ');
  writeln('PDP 11/23        intrpr. BASIC+                               21.0');
  writeln('PDP 11/73        FORTRAN               0.4                     0.6');
  writeln('                 BASIC+                                        2.5');
  writeln('VAX 780          BASIC                 0.06                    0.1');
  writeln('KI 10            MACRO 10                                      0.10 (72bit)');
  writeln('-------------------------------------------------------------------------------');
  writeln('Din dator        Turbo Pascal                  ',totsec:3:3,' sekunder.');
  writeln('-------------------------------------------------------------------------------');
  writeln(' K{llkoden: p:=0;s:=0;         ',chr(7),'s:',s,'  p:',p );
  writeln('             while p<9 do begin ');
  writeln('               s:=s+1;p:=p+1/s;');
  writeln('             end;                                  Nils Segerdahl hackade');

end.
```

Thord Nilson

”Det var bara en idé”



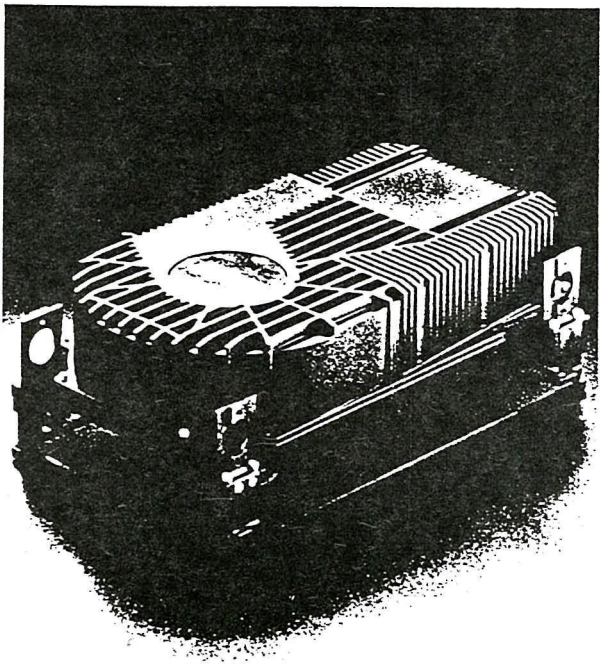
En idé är som ett flämtande ljus.
Lättare att släcka än att hålla vid liv.

Sista nyheten

Single-board nuclear reactor supplies standby power for 12 years

Now available on a full-length plug-in card for IBM PC or compatible computers, the QBX-1 add-on nuclear-reactor card provides backup power for as long as 12 years. When the card senses a power failure, explosive bolts eject moderator and control rods from the reactor's interior within 20 μsec, bringing the reactor to its fully rated output of 20 kW in less than a millisecond. Over its 12-year active life, the reactor's power decreases by 25% to 15 kW.

Integral heat fins provide convection cooling of the reactor's 500W power dissipation while the reactor remains in its standby condition. If your computer's fans can't furnish 400 ft^3/sec of forced air for cooling, consider buying the manufacturer's heavy-water cooling jacket and stainless-steel pump module, which fit conveniently under a desk or workbench. Latches on each side of the reactor module let you quickly exchange the radioactive core, should you need to replace it. An optional circular viewing port of lead glass lets you check the reactor's internal mechanical assemblies.



Nuclear reactor supplies CPU power during power failure or other power emergencies. The reactor also glows in the dark (as will you), which makes it easy to find your computer.

To protect users from undue radiation, each reactor includes a shielding kit comprising five self-stick lead plates and 20 radiation-monitoring film badges. The lead plates mount inside your computer's enclosure and reduce the gamma rays that cause soft errors to floppy-disk and RAM data. For further protection, consider buying the manufacturer's 200-ft extension cords for keyboards and monitors.

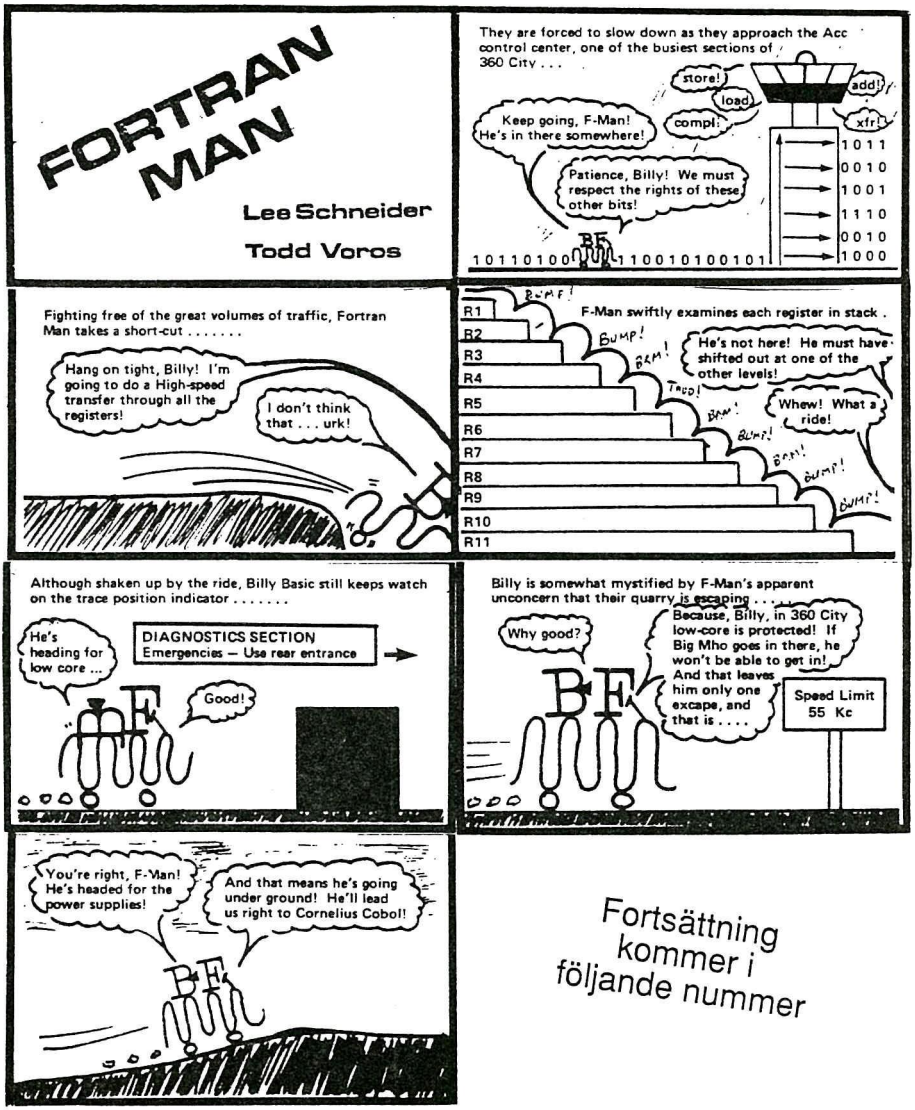
Because the reactor can supply more than enough power for your computer, you can sell excess power to your local utility company. An add-on phasing and metering kit (PMK-1) lets you connect your reactor to the local power grid. Each PMK-1 includes standard power-sale contracts and Rural Electrification Board rules and regulations.

Although not required in all localities, each reactor card package includes a standard 23-volume site-evacuation plan. The plan includes blank forms for you to fill in the name and address of your reactor site and then mail to the Nuclear Regulatory Commission. As an option, the manufacturer supplies the plan on 12 MS-DOS-compatible disks in Wordstar format. User-friendly templates let you type in information so that your word processor can create a complete, printed document.

Reactor prices start at $2.3 million (1). Delivery, seven years ARO.

—Regus Patoff

Luminescent Electronic Products Inc, Box U-235, Trinity Site, NM 43210. INQUIRE DIRECT



Fortsättning kommer i följande nummer

Politiskt val

DEN HÄR artikeln konfirmerade slutligen min åsikt om att politiker skall låsas in, skjutas och omyndigförklaras, i denna eller omvänd ordning. Vad gör de med våra skattepengar? Leker. Det här var en fadäs som blev känd. Hur många likadana, fast okända finns det inte?

Jörgen Städje

Färgen på datorn avgjorde valet

■■ När Kalix kommun investerade i en ordbehandlare var inte priset det viktigaste, utan färgen.

■■ Motiveringen till inköpet finns att läsa i protokollet från kommunstyrelsen

När personalkontoret i Kalix kommun behövde en ordbehandlare gjorde man detta i samråd med kommunens inköpare innan ärendet gick vidare till kommunstyrelsen och fullmäktige för beslut.

Beslutet fattades utan ändringar. Det blev inte den billigaste datorn, men den uppfyllde önskemålen. I kommunstyrelsens protokoll kan man läsa motiveringen:

"...Personalkontoret och inköpschefen föreslår att behövlig utrustning inköpes enligt en offert från det senare företaget, som kan leverera grönvit skärm. Detta innebär att omtapetsering av kontorsrummet inte är nödvändig". □