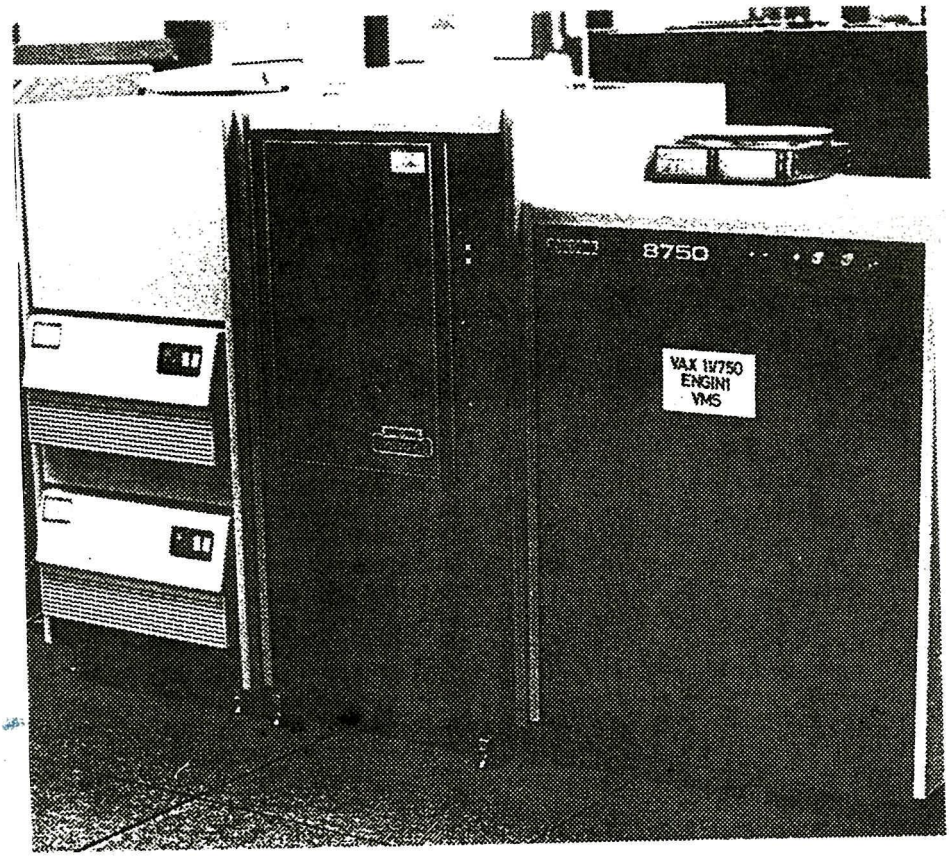
STACKPOINTER

2-1988. Organ för Datorföreningen STACKEN, KTH.



Datorföreningen STACKEN

STACKEN, som bildades våren -78, är datorföreningen på KTH. Tillika kårförening. STACKEN är en ideell förening, där intresse för datorer är den största gemensamma faktorn. Föreningen finns till för att:

- vara en mötesplats för likasinnade.
- ge tillfälle till formell och informell utbildning om tillämpningar, dator- och programteknik.
- tillhandahålla media för utbyte av idéer, kunskap och erfarenheter.
- tillåta närgången undersökning av datorers delar och programmen därtill.
- skapa kontakter med andra datorföreningar.
- vara samlingspunkten för att bilda intressegrupper kring speciella projekt.

Vi har en mängd intressanta verksamheter på gång. Dock hänger det ytterst på den enskilde medlemmen att avgöra vad hon eller han vill göra för föreningen. Under årens lopp har det åstadkommits en hel del. Värt att nämna är :

- samköp av mikrodatorbyggsatser
- diverse kurser och föredrag, bl.a. RMS
- haft visningar, bl.a. en av de första MAC II i Sverige
- studiebesök på intressanta ställen
- hämtat datorer från olika delar av världen
- AMIS, den portabla EMACS-kompatibla editorn för TOPS-10, VMS, NORD, RSTS/E, PRIME,...
- STACKPOINTER, vår tidning
- vi har egen E-mail adress
- en egen datorhall för våra stordatorer
- en egen DEC-10:a (se nedan)

Sedan oktober -85 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.Tyvärr står hon f. n. emedan kylan har packat ihop. Hon står i vår maskinhall "B30", på Brinellvägen 30, gaveln mot Lill-Jansskogen .

Sedan december -87 har vi klubblokal, till höger i valvet en trappa ner, Lindstedtsvägen 5, "LV5".

Är du intresserad av föreningen så är du välkommen till något av våra möten. Ordinarie möten är första torsdagen i varje månad kl. 19 i sal E7 på KTH.

Funderar du på att bli medlem, och går på KTH. Skicka in en lapp med namn, adress och på vilken linje du går. Andra intresserade vill vi veta lite mer om. Vi är lite formella för att undvika att föreningen blir allmän lekstuga.

*** Computers have more fun! * Computare necesse est! ***

I detta nummer

NUCC 88 i Helsingfors

FÖRRA ÅRET träffades de nordiska högskoledatorföreningarna i Linköping. I år är det dags igen. Lördag–söndag 11:e–12:e juni träffas vi igen, i Helsingfors. Vi ger oss av på kvällen fredag den 10:e och kommer hem måndag den 13:e. OtaDATA och Tietokilta/Datagillet står som värdar. Skall Du med?

Vårmötesprotokoll

PROTOKOLL fört vid Datorföreningen STACKENs vårmöte, torsdagen den 25:e februari 1988 kl 19:00 i sal E7 på KTH.

Närvarande:
Eva Albertsson, Henrik Björkman, A-C Borstedt, Tord Beckman, Henning Croona, Mats O Jansson, Carl-Arne Johannesson, Johan Kjellin, Stellan Lagerström, Erik Levlin, Stella Lindblom, Lars S Ljungdahl, Peter Löthberg, Kurt Minnberg, Thord Nilson, Dan Nordstedt, Hans Nordström, Thomas Nyström, Jan Michael Rynning, Peter Svanberg och Anders Westerberg.

§1. Mötets öppnande.
Stellan Lagerström öppnade mötet.

§2. Val av justeringsmän.
Peter Löthberg och Jan Michael Ryning valdes till justeringsmän.

§3. Val av mötesordförande.
Stellan Lagerström valdes till mötesordförande.

§4. Val av mötessekreterare.
Henrik Björkman valdes till mötessekreterare.

§5. Tillkännagivande av röstlängd.
Alla närvarande var röstberättigade.

§6. Fråga om mötets stadgeenliga utlysande.
Mötet ansågs stadgeenligt utlyst.

§7. Fråga om dagordningens godkännande.
Dagordningen godkändes.

§8. Verksamhetsberättelse.
Se bilaga 1.

§9. Revisionsberättelse.
Se bilaga 2.

§10. Balansräkning.
Se bilaga 3.

§11. Ansvarsfrihet för avgående styrelsen.
Styrelsen beviljades ansvarsfrihet.

§12. Fastställande av stadgeändringar från höstmötet.
Ändringarna fastställdes. Begreppet *"ersättare"* avskaffas genom att första meningen i §16 ändras till *"Styrelsen skall bestå av minst fem ledamöter."*, uttrycket *"eller ersättare"* stryks i §18 och ordet *"styrelsemedlem"* ändras till *"styrelseledamot"* i §10 och §19. De valda ersättarna blir styrelseledamöter när ändringen träder i kraft. Meningen *"Tidpunkt för höst/vårmöte fastställs av föreningsmöte."* stryks i §14/15.

§13. Övriga Frågor.
Medlemsvärvning och aktivering av Macintosh-användargruppen diskuterades. Peter Löthberg såg mycket trött ut när Jan Michael Rynning föreslog en PC-användargrupp. Stellan informerade om att huset nog blir försenat.

§14. Mötets avslutande.
Mötet avslutades efter 1 timme, 18 minuter och 40 sekunder.

Vid protokollet

Henrik Björkman, mötessekreterare

Justeras

Peter Löthberg, justeringsman

Justeras

Stellan Lagerström, mötesordförande

Justeras

Jan Michael Rynning, justeringsman

Revisionsberättelse för Datorföreningen Stacken verksamhetsåret 1987

Efter genomgång av räkenskaperna ställs man som vanligt inför några mindre problem:

1. I bland finns kvitto på ett visst belopp men inte faktura eller annan anteckning som beskriver transaktionen. Två exempel kan räcka: Verifikationerna 204 resp 224 redovisar intäkter på dryga 6000:- per styck. Är det för AMIS eller är det för något annat. Detta måste gå att utläsa.

2. Stellan Lagerström har åkt till USA för 6835 kronor på Stackens bekostnad. Varför? Om skälet är behjärtansvärt, vilket antagligen är fallet, varför är Stellan då enligt bokföringen skyldig föreningen 545:-? Var är reseräkningen. Som sagt om resan hade ett ändamål måste väl 545:- vara i snålaste laget för kost och logi. I USA var statliga traktamentet år 1987, 455 kronor per dag!

3. Ulf Asplund har åkt till San Fransisco för 2500 kronor - var är kvittot resp biljetten. Varför kunde han åka för bara 2500 kronor ?

4. Ett antal personer har på Stackens bekostnad ätit och druckit för nästan Tiotusen kronor, 8815:- för att vara exakt. Inom näringslivet är alkohol en inte avdragsbar kostnad, men Stackens gäster har i alla fall druckit alkohol för 1310 kronor! Javisst ja, jag måste tillägga att enligt restaurantnotan var de 41 st så den totala kostnaden för mat och dryck stannade vid 215 kronor per person - hoppas det var gott.

5. Balansräkningen vart tog den vägen?

Efter genomgång av Datorföreningen Stackens räkenskaper för verksamhetsåret 1987 finner revisorerna trots allt att räkenskaperna befunnits i god ordning och att styrelsen med avseende på räkenskaperna kan rekommenderas full ansvarsfrihet.

Stockholm den 24 februari 198?

Lars S. Ljungdahl

Peter Löthberg

På besök i England

av Jan Lien

I slutet på februari skulle jag åka till London några dagar, och jag visste inte riktigt vad jag skulle fylla ut tiden med. Då kom jag att tänka på att det kanske finns någon DEC-utrustning som bara väntar på att bli tiggd till Stacken.

Sagt och gjort, jag började forska i var det fanns dylika monstermaskiner. Först tog jag reda på numret till Digital i England, och blev hänvisad till några andra nummer. Först hamnade jag hos en kille på avdelningen för begagnade system som mekaniskt upprepade att "Vi har ingen information om installerade system..."

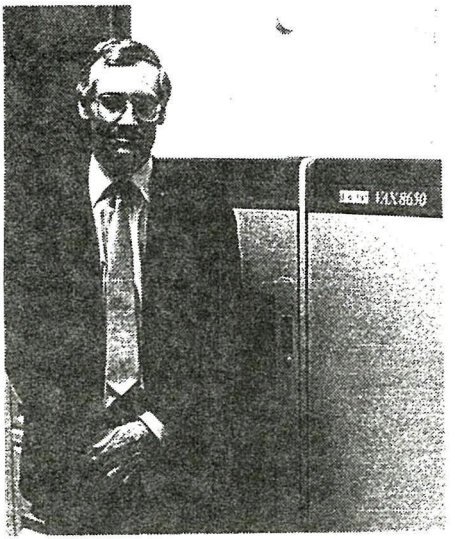
Nåja, jag kan vara lika envis, och försökte ett annat nummer, och blev snabbt och skickligt kopplad till en tjej på någon försäljningsavdelning, och hon frågade ärligt "Vad är DECsystem-20??". Jag insåg att här fanns inget att hämta, bad henne koppla mig till serviceavdelningen, för att kanske hitta någon som skulle veta vad som fanns var. Nu började hon envist fråga "Vad skall du med det till? - Vill du rapportera ett fel?" och så kom vi inte längre...

Jag bara undrar varför USA tillåter att Digitals anställda går samma charmkurs som de ryska gränsvakterna??

Ännu var jag inte helt knäckt, utan började nu leta bland universiteten. Om man tittar i Janus, kanske man kan få veta något. Det som fanns var Essex University, Oxford, och Hatfield Polytecnic. Essex University hittade inte televerkets nummerupplysning, men däremot Oxford och Hatfield. Först ringde jag Oxford, men där verkade ingen så värst överdrivet intresserad (kanske jag hamnade hos fel person?), så jag ringde även Hatfield. Där hamnade jag rätt. De hade en DEC-1091, och en DEC-2020, båda körde TOPS-10. De skulle dessutom skrota sin 1091:a snart, och verkade vänliga, de frågade nämligen om jag ville komma och titta. Givetvis svarade jag ja, utan att egentligen veta var Hatfield ligger, det löser sig säkert senare. Efter en snabb kontakt med Peter för att få veta vad som man skulle visa av Stacken och Kicki, utrustades jag med ett band med Amis på, "Mycket bra skäl att tränga sig in i maskinhallen..." enligt Peter.

När dagen var inne i London, så tog jag tunnelbanan till Kings Cross (det luktade allt lite grillfest ännu) där pendeltåget skulle gå. Min vana trogen missade jag det tåg jag hade tänkt att ta, och hann alltså att se mig omkring ganska noga.

Folk står faktiskt och köar (lång rak kö!) i väntan på att grinden till perrongen skall öppnas. Har jag aldrig sett tidigare. Pendeltåget gick fort, och sista biten tog jag en taxi. Killen jag hade pratat med i tele-



Gordon Spencer-Brand framför en av "sina" VAX 8650.

fon, Gordon Spencer-Brand, visade sig vara chef för Hatfield Computer Centre. Hatfield Polytecnic grundades 1952 genom donationer av flygplanstillverkaren De Havilland (tror jag). Numera är flygsektionen stor, men inte dominerande, Computer Science har vuxit sig stor. Härutöver finns några kurser i ekonomi etc. Skolan har ca 4500 studenter. Samarbetet med flygindustrin finns dock fortfarande kvar. Computer Centre ligger sedan bara några månader tillbaka i en ny byggnad, men man flyttade inte maskinhallen.

Hatfield har idag:
VAX 8650, VMS, SA482 disk array
VAX 8650, UNIX, RA81 disk
VAX 11/785, VMS, RA81 disk
MicroVAX II, MicroVAX 2000,
5 st VAX 11/750, engelska Plessey
DECsystem-1091 768 K, 4 st RP06, 2 st TU77, LP10
DECSYSTEM-2020, 2 st RP06, TU45, LP20

Man började köra TOPS-10 1970, när den första DEC-10 installerades, en KA-10. 1979 ersatte man KA-10 med en KL-10, den som nu körs. Konsolen till KA-10 finns kvar, som souvenir. Nu kör man TOPS-10 7.01, eftersom systemen ändå skall ut så finns inget skäl att uppgradera. DEC-1091 är inbytt mot ett nytt VAX-system, och skall köras till någon gång i

Jeff, ansvarig för programvaran i ANF-nätet.

juni. DEC-2020 skall finnas kvar ett eller två år till, man behöver den nämligen till att kompilera det lokala nätverket. När de började med DEC, så var ju ANF-10 ett naturligt val. Därför utvecklade man ett nätverk baserat på ANF-10, och nu har man (förutom DEC-tiorna) alla VAXar med, oavsett om de kör VMS eller UNIX. Man har skrivit rutiner som pratar med Cambridge ring net, och ringnätet pratar i sin tur ANF-10. Härutöver har man ett stort antal terminalkoncentratorer, baserade på Z80 eller PDP 11/34. En Z80 koncentrator kan supporta 24 terminaler. Alla Z80-koncentratorerna finns med i nätet som noder! Hatfield försöker ersätta sitt ANF-10-baserade nät med ett modernare nät som inte är så teckenorienterat. Man väntar sig att det tar ett tag. Egentligen skulle man vilja göra sig av med sin DEC-2020, men den behövs för att kompilera koden som skall in i PDP 11/34-orna i nätet. Det finns nämligen ingen korskompilator till VAX som ger en färdig binärkod, utan bara kod som förväntar sig att hitta ett OS i PDP 11/34. Lite försiktigt föreslog jag att om de skänkte oss DEC-2020 så kunde de väl få kompilera sitt nät hos oss... Men de verkade inte alls förtjusta över att behöva arbeta via Datapak...

Det hela slutade med att jag fick ett namn på säljaren hos Digital, som nu är på Peters bord. Vi verkade i alla fall kunna få vad som blir över efter att Digital tar sitt byte, dvs:

512 K minne
TU 77 slave
PDP 11/40
PDP 11/45 efter att de plockat de delar de själva vill ha kvar.
RK07 disk
DTR01 (har vi en redan, funkade inte så bra enligt dem)

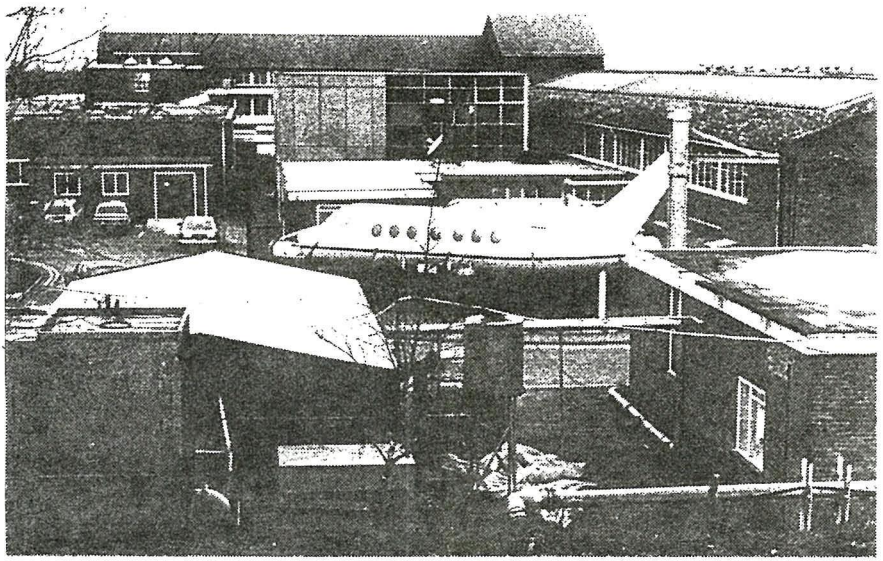
Plus sånt som vi inte vill ha, dvs inte betala frakt för:

kortläsare, golvmodell
BA-10 och 6 st DECtapes, TD-10 control
terminaler, gamla dumma engelska
LP10 printer, golvmodell

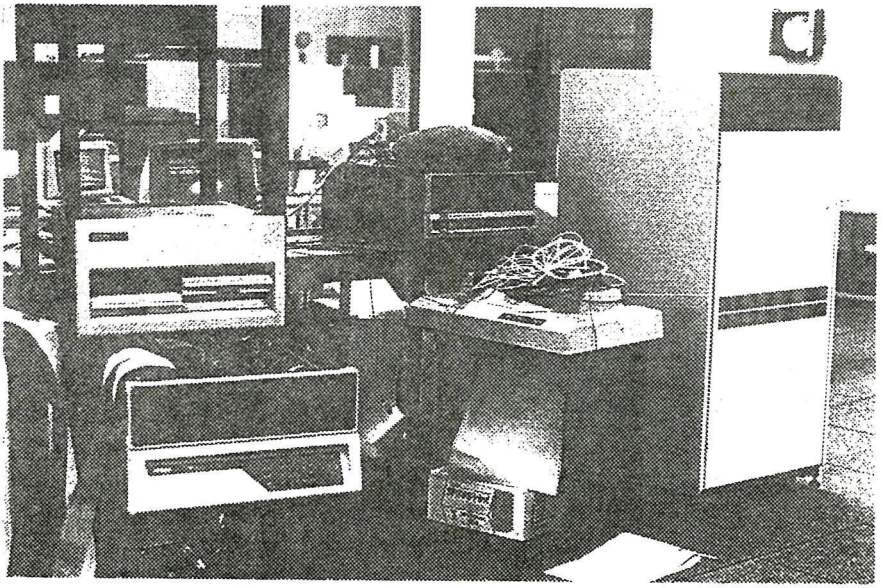
Förutom ett antal bilder, fick jag även namnen på andra universitet med PDP-10-utrustning. Det visade sig att just Essex University hade en DEC-10-fan, så det stället skall vi INTE kontakta innan vi fått den som finns i Hatfield...

Senaste nyheterna är (18 april) att KL-10 skall köras till ca november, och att även andra delar av utrustningen kanske skall bytas in... eller har rektor för skolan sagt att man inte får skänka bort värdefull datautrustning?

Vi får väl se om det blir något...



Flygskrot fanns det lite varstans.



Alldeles utanför datorhallen var uppställningsplatsen.

Lite mera om SuperMail

av Thomas Nyström

Domänadresser

Man har nu börjat använda något som kallas för domänadresser. Detta har kommit till för att användarna ska slippa veta vilken väg ett brev ska skickas för att komma fram. Jag heter till exempel (på KICKI):

THN@KICKI.STACKEN.KTH.SE

Datorns namn är i detta fallet 'KICKI.STACKEN.KTH.SE'. Punkterna använder man som avskiljare mellan de olika orden. När man uttyder denna adress börjar man bakifrån:

SE -> Sverige
KTH -> Kungliga Tekniska Högskolan
STACKEN -> Datorfören. Stacken
KICKI -> Den dator jag finns på.

Finessen med denna typ av adresser är att den lokala mailern (i vårat fall SuperMAIL) inte behöver känna till alla datorer i hela världen (vilket skulle bli *MÅNGA*). Våran SuperMAIL känner bara till vissa domäner och datorer som den kan skicka brev direkt till och alla toppdomäner. När SuperMAIL får en adress att fundera över letar den i sina tabeller och skickar den i rätt riktning. Dags för några exempel:
NIZZE@TUTA.LIU.SE
PEHR@VOLVO.SE
ROLL@AI.AI.MIT.EDU

SuperMAIL kommer här att känna igen 'LIU.SE' och veta att den ska skickas till LISBET i Linköping med protokollet SMTP över DECnet. 'VOLVO.SE' känner inte SuperMAIL till. Däremot vet SuperMAIL att allt som slutar på 'SE' ska den skicka till RTR59B. SuperMAIL vet inte vad som sedan händer med brevet (brevet hamnar så småningom hos ENEA som känner igen 'VOLVO.SE' och kan skicka det rätt). Det sista exemplet är en

adress i USA. SuperMAIL känner igen 'EDU' som en domän som ska gå samma väg (ENEA kommer sedan att skicka den till UUNET.UU.NET, en dator i USA).

Vissa datorer som knyts samman av ett nätverk kan ha namn där nätverksnamnet ingår i adressen. För KICKI gäller att vi också heter 'KICKI.SUNET.SE' (vi är anslutna till SUNET) och 'SESTAK.BITNET' (Bitnet är ett nätverk med huvudsakligen IBM-stordatorer).

Kicki är dessutom knutpunkten i domänen 'STACKEN.KTH.SE', dvs alla brev som slutar med denna domän ska skickas till KICKI. Känner inte KICKI till datorn ifråga så anses det att den inte finns, dvs SuperMAil på KICKI kommer att skicka en rapport där den berättar att datorn inte finns, till avsändaren av brevet. Det går även bra att använda
'THN@STACKEN.KTH.SE'
i stället för
'THN@KICKI.STACKEN.KTH.SE'
eftersom KICKI är knutpunkt.

Nätverk

Jag har redan nämnt två nätverk (SUNET/DECnet och BITNET). Förutom dessa två har KICKI kontakt med ytterligare ett, vi kan här kalla det KTH-Ether. Detta nät är av typen TCP/IP. Nätet kommer ursprungligen från USA och är det protokoll som ARPAnet är uppbyggt på. ARPAnet är det stora datornätverk som knyter samman datorer i USA. Här i Stockholm finns det flera TCP/IP nät. Det finns planer att knyta samman alla dessa nät här i Stockholm med liknande nät på andra ställen i Sverige och eventuellt med ARPAnet (här börjar det bli politik och sådant har jag svårt för...).

SUNET som idag bygger på DECnet kan bara kopplas till datorer från DEC (SUNET är ett svensk nätverk). BITNET till IBM-datorer (BITNET är internationellt). TCP/IP kan anslutas till många olika fabrikat och operativsystem. Man har med hjälp av detta en möjlighet att kunna kommunicera mellan olika datorfabrikat. Alla dessa nät kan förutom brevöverföringar även hantera filöverföringar. DECnet och TCP/IP kan också hantera terminaltrafik.

Förutom dessa nämnda finns det ett internationellt nät till, att nämna: UUCP, detta nät har inte KICKI kontakt med. Detta nät används av datorer som kör operativsystemet UNIX.

Gateway

För att möjliggöra trafik mellan dessa nät så finns det ett antal datorer som enbart är till för att överföra information mellan näten. Här i Stockholm har vi KTHVAX och RTR59B som sköter om brevtrafiken mellan KTH-Ether, SUNET och BITNET för KTH samt ut mot stora världen via ENEA. KTHVAX pratar TCP/IP och BITNET, RTR59B pratar TCP/IP och DECnet. Trafik till och från ENEA överförs från KTHVAX via TTDS till ENEA.

the Mac Scores Virus

by © Howard Upchurch 1988 (via MacUnderground, Sweden)

Introduction

A virus is an organism that attacks and feeds off a host until either the virus or the host dies. A so-called **Scores** virus has spread throughout the Macintosh community. This virus, however, is a nasty piece of software written by a demented individual. Just like a living organism, it reproduces itself and has spread like an epidemic. Rumors (and there are PLENTY!) are that thousands of U.S. Government Macintoshes — including those owned by NASA — are infected, and that the FBI is investigating the outbreak. In addition, Apple, other major corporations, and probably hundreds of thousands of business and private users are infected. This is NOT the MacMag virus, which was relatively benign and was inadvertently spread by Aldus in a few copies of *FreeHand*. It is NOT the nVIR virus, which so far has spread very little, according to published sources. It IS a virus that was purposely designed to spread itself as rapidly as possible. Scores will enter a disk as part of an application. It will spread to the System, then to other applications, some of which will be given to a friend or taken to work, spreading it even further. There is evidence that it can spread through a network. Scores will damage programs, causing unpredictable problems. Its primary intent has not yet been discerned. Don't be the first to discover the evil purpose for which this virus was designed. Get it out of all systems in which it is located, and do it NOW!

Detection

Open the System Folder on all disks in your possession, especially hard disks. Look for two icons representing the Scrapbook File and Note Pad File. The System is infected if BOTH of them are there AND if both icons are generic document icons, i.e., blank dog-eared pages. The System is probably not infected if neither or only one icon is present or if the icons look like Macintoshes, the same icon used for the System and *Finder*. If the disk is infected, do not panic. This document tells how to remove the virus from the System and prevent its recurrence. If the disk is not infected, learn here how to protect yourself and to help someone else remove the virus.

Good News!
Figure 1a

Bad News!
Figure 1b

Discussion of Macintosh Program Structure

Macintosh programs which are used to perform productive tasks are called **applications**. Common applications are *MacWrite*, *MacPaint*, and *Microsoft Word*. Other applications with which everyone is familiar are the *Font/DA Mover*, *HyperCard*, and *TeachText*. Many users do not realize that the *Finder* is also an application.

Items created by applications are called **documents**. A letter created with *MacWrite*, for instance, is a document. There are other items on a Mac like *System* and *General*, which are neither applications nor documents. These items, along with applications and documents, may be termed **files**. Generically, any item on a Macintosh which has an icon is called a file.

Macintosh files are composed of smaller groups of software called **resources** and **data**. Thus any Macintosh file may contain data, resources, or both. An application is comprised primarily of resources, a document primarily of data. Resources with which everyone is familiar are fonts and icons. Others of importance to this discussion are **CODE** and **INITs**. CODE is contained in virtually every application, for it is really the heart of the application itself. CODE is the set of commands which controls all the other resources. An INIT is a set of instructions which is loaded into the Mac's memory when power is turned on and a disk is inserted. INITs are executed in alphabetical order. Common INITs are *Suitcase* and *Pyro*.

Apple has provided an application called the *Resource Editor*, *ResEdit*, or *ResEd* for short. It is a necessary tool for both identification and removal of this virus, but it is quite powerful and beginners are urged to avoid any uses of this program other than those described here.

Analysis of Infected Application

The Scores virus seems to attack only files which have CODE resources, primarily **applications**. Although it is possible for **documents** to contain CODE, no specific examples are known. It should be mentioned that files which have been stored in the *Stuffit* format contain no resources at all, so a file saved or archived in that manner should be impervious to infection IF it was clean when Stuffed.

To observe the infected application, open *ResEd*, and you will see a window like, Figure 2:

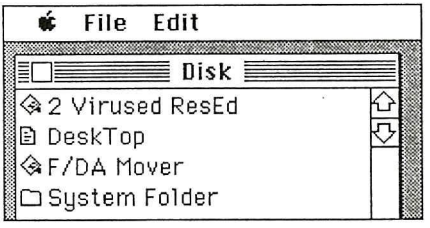


Figure 2

Selecting the infected application by double clicking on its name, in this case *ResEd* itself, we see, Figure 3:

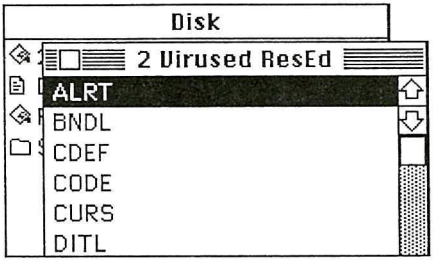


Figure 3

Opening the CODE resource by double-clicking on the name shows, Figure 4:

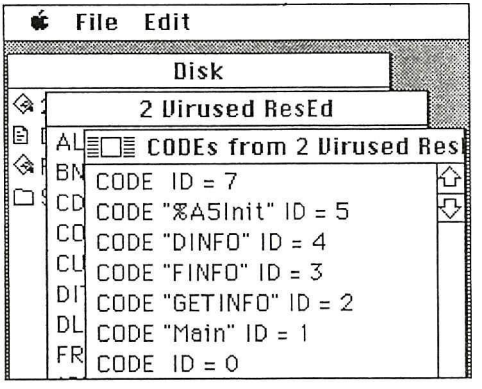


Figure 4

Notice that the first CODE on the list (CODE ID = 7) has an ID which is numbered two higher than the next highest (CODE ID = 5). Selecting this resource by clicking on it once and choosing Get Info from the File menu reveals a size of 7026 bytes, Figure 5:

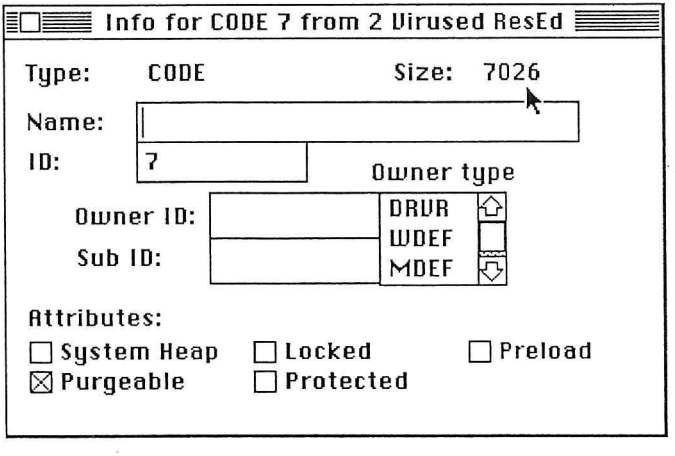


Figure 5

This is the final proof that the application is infected. (An examination of an uninfected copy of *ResEd* would show that CODE 7 was not present.) In addition to this easily detectable change, CODE 0 has been modified and there is at least one other alteration, the details of which are unknown to the author at this time.

Effects of Using an Infected Application

When an infected application is opened, its new CODE commands tell it to add several new pieces of software to the System Folder. Two of these are quite important because they provide the best clue that something is wrong: **Scrapbook File** and **Note Pad File**, as discussed in the Detection section. Other changes the virus makes to the System Folder are less obvious: It adds a **Desktop** file and a file called **Scores**, from which the virus gets its name. These files cannot be observed from the *Finder* because they are invisible. Programs such as *ResEd* and *MacTools* show them to be there, however. The virus also modifies the System itself, adding the following resources: atpl ID 128, DATA ID -4001, and INITs with ID's of 10, 6, and 17. With these new INIT resources in the System, the Mac is figuratively a fused bomb, ready to do damage the next time it is turned on.

Spread of Virus to Uncontaminated Applications

Because these new resources are primarily INITs, they are activated the next time the Mac is started. Once initialized, the virus begins to execute the commands which cause it to spread. As the infected disk is used, the virus continually seeks uncontaminated applications. The present thought is that it searches in a random fashion at an interval of three and a half minutes. At times a disk drive will begin operating when nothing should be happening. This occurs because the virus is writing its code resource to another application. After a long enough period of time, every application on the disk will be infected, apparently whether it has been used or not.

Prevention of Occurrence or Recurrence

CE Software has released into the public domain a utility called **Vaccine**. Vaccine is a "cdev," which means "Control Panel Device." Copies are **free**. Get it from a Disk-of-the-Month (DOM) at a user group meeting or from a telephone communication service such as CompuServe or GEnie. To use it, place the Vaccine icon in the System Folder. Select Control Panel from the Apple menu and you will see "Vaccine" listed right under "General." Close examination will reveal that the name begins with a space before the "V." Leave it that way so it will be the first thing that operates when the Mac is started or reset. Select the Vaccine icon and read the instructions. In case you do not understand them, putting an "X" in the top and bottom boxes is recommended. Be sure to restart the Mac after setting

Vaccine in order to start it working.

To help assure that you have a clean copy of Vaccine, select the Vaccine icon while at the *Finder* (not the Control Panel) and choose Get Info from the File menu. Verify that the size is 11,875 bytes and that the creation date is Saturday, March 19, 1988, at 11:49 PM. We must assure that no one creates a bogus version of this fine work. And thank you, CE Software!

After Vaccine has been installed, look for the following symptoms when using the Mac or opening an application; each is an indication that the virus is in operation:

(1) Vaccine randomly asks for permission to alter a resource.
(2) Opening an application triggers Vaccine.
(3) Opening a resource causes a bomb (usually ID = 02).
(4) Opening an application causes the Mac to hang up.

Do not put a copy of any application on a hard disk until it has been checked for contamination. Do not run a new copy of any program until it has been checked out. Examine any program before uploading it to a Bulletin Board.

Removal of Virus from System

Since the relatively recent discovery of this virus, several programmers are working on developing software which will do any or all of the following: detect the presence of the virus, remove it from the System Folder, detect infected applications, and/or repair the infected applications. As of this writing, however, none are available. What follows is a step-by-step procedure that will enable you to clean up a disk with or without one or more of these utility programs.

First, install the **Vaccine** utility if it is available and reboot the Mac. (**Note:** If you see a bomb, a hangup, or a message from Vaccine when booting, the *Finder* is contaminated. Boot with a clean floppy and replace the *Finder* on the virused disk.) Open *ResEd*. (**Note:** If you see a bomb, a hangup, or a message from Vaccine when trying to open *ResEd*, *ResEd* itself is contaminated. Replace it with a clean copy.) At this point you will see the files at the so-called root level of the disk, resembling Figure 2. Notice the file called DeskTop. This is NOT the bad file. Scroll through the window and open the System Folder by double clicking on its name. You will see a window resembling Figure 6.

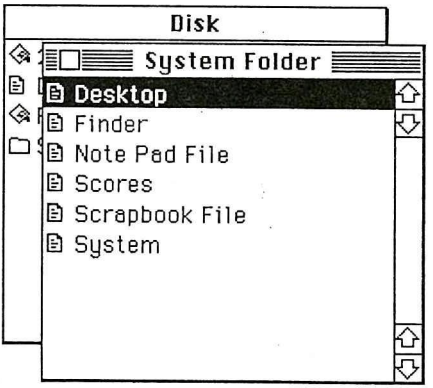


Figure 6

Select the Desktop file by clicking on it one time, then choose Clear from the Edit menu. Do the same thing for the other three infection files, Note Pad File, Scores, and Scrapbook File.

Locate the System and double click on its name to open it. You will observe a window similar to Figure 7.

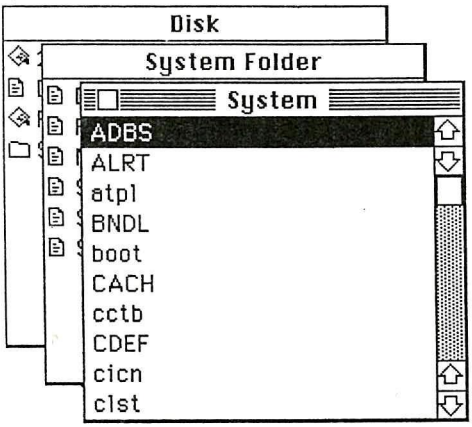


Figure 7

Locate atpl and open it by double clicking. Select atpl ID 128 and Clear it by using the Clear command under the Edit menu. Close atpl and open DATA. Clear DATA ID -4001. Close it and open INIT. Clear ID 10, ID 17, and ID 6. Close all windows except the root level window and save the changes when asked if you wish to.

Important: A virgin System (4.1, at least) from Apple does not contain either resource of the types atpl or DATA, but some programs—*LaserSpeed*, for one—legitimately place them in the System. Remove only the ID numbers listed.

The System is now free of infection, but the work is far from over. When Vaccine has been properly installed on the disk, opening an infected application will cause either a bomb, a message from Vaccine, or the Mac will hang up. In any case, the application should be examined more closely: Use *ResEd* to open the CODE resource of the suspected application. If the top CODE ID is two numbers higher than the next highest, Get Info on it. If the size is 7026, as shown in Figure 5, it is an infected application. Throw it in the trash because it is unusable and will reinstall the virus into the System if it is run with Vaccine off or not installed.

Even if you do not yet have a copy of Vaccine, use *ResEd* to examine every application on your disks. Notice the small icon next to the *Font/DA Mover* name in Figure 2. This icon will help in determining which files on the disk are applications. Check ALL of the applications in the manner described above. It is easy to overlook some of the smaller and perhaps lesser used ones like *Font/DA Mover* and backup programs. Remember, the *Finder* is an application. And an application does not have to have been run for it to be contaminated.

Experiences with this virus over the past four months have shown this to be an effective and relatively simple way to clean a disk. There is nothing wrong with

replacing the System, replacing the System Folder, or re-initializing the hard drive. These, however, are extreme measures and are not considered by the author to be necessary. In any case make sure, with *ResEd*, that all applications put back on the hard drive are clean, especially if Vaccine has not yet been installed, or the whole cycle could begin again.

For more advanced users: After it is felt that all infected applications have been removed and replaced, run *Disk Express*, if available, with the Erase Free Space option turned on. This will cluster the data to the start of the disk and zero out all remaining space. Then use *Fedit*, *MacTools*, or a similar program to search for two strings virtually unique to this virus: **VULT** and **ERIC**. Each string is all caps. If these strings are nowhere on the disk, it is clean. If they are still there, do everything possible to find out which file they are in and remove it from the disk. Repeat this until there is no ERIC or VULT. (The only application so far discovered which contains the VULT string is one called **DD Editor**, and it does not contain ERIC.) Searching a previously infected disk in this manner without having run *Disk Express* first does no good because the infected files were not actually erased when trashed and the remnants are probably still on the disk. In other words, the presence of ERIC and VULT at this stage of the removal process does not mean that the disk is still infected, but their absence DOES mean that the disk is clean.

Removal of Virus from Infected Applications

Unfortunately, at this time there is no known method to repair infected applications, and perhaps there never will be. There is evidence that when the virus attaches itself to an application and inserts the new CODE resource, at least a part of the new CODE is apparently written over some part of the original application software, permanently destroying it. If true, this would account for the many strange effects of the virus because the missing code would be different in each application. There would have to be a separate fix for every application. The safest thing to do is to trash every bad application from the disk and replace it with a known clean copy. If there is no clean copy backed up, save the infected version on a floppy in hopes that a fix will be found.

Comments

Cleaning the virus from one disk will not fix the problem. ALL Macintosh disks must be clean or the problem will be around for a long, long time. And not just YOUR disks: EVERYONE'S disks! After you are familiar with the problem and its solution, share your knowledge. Make as many reproductions of this document as you wish to and give them to anyone with a Mac. It is copyrighted in the hopes that the editorial content will not be altered, but permission is given for the widest possible distribution. Print copies in your club or company newsletter. Visit dealers and see if they are virus-free. Help them

spread the word. Do what you can — it will cost you only a little time, not money.

Why am I taking the time to create this document? I had the virus as early as November of 1987, but dismissed the problem as an offshoot of MultiFinder, due to the fact that the virus struck me just as I had decided to quit using MultiFinder and return to using System 4.1. I spent many hours of work over several weeks figuring it out and ridding myself of its effects. At the time I did not recognize it as a virus, and for that I am very sorry. I should have pounded on Apple's doors relentlessly asking about this problem. Possibly someone there would have recognized it for what it was, early enough to prevent the present massive outbreak of the problem.

I have enjoyed my Mac for well over four years now. I have created three fonts with it, one shareware and two that have actually been published. I have had fun with my Mac, and I have earned money with it. I am a member of two Macintosh clubs and have made many good friends because of this small computer. I can't stand by while some jerk destroys so much of my life. The time has come to repay the Mac community and this is my way.

Help me. One hates to publish a phone number in a document designed for public distribution, but without it you could not relay any important information. Please call only from 8 AM to 8 PM Central time, and only if you have found some information not in this document. Long distance callers, please leave a complete message on the answering machine if it answers, as I cannot afford to return many long distance calls.

Both User Groups of which I am a member have access to AppleLink, a worldwide communications network operated by Apple Computer, so any new information can be relayed directly to the people at Apple who are working on solving this problem. And thanks for any help.

Howard Upchurch
3409 O'Henry Drive
Garland, TX 75042, USA
(214) 272-7826

I have reported information as I have found it. If there are any errors in the above, I apologize but ask not to be held responsible. Some statements may prove false or incomplete as more information comes to light.

Distributed by the Mac Pack and the Dallas Apple Corps for all members of the Macintosh community.
Special thanks to John Cail, Doug Ruddman, Kelly and Cheney Coker, and Steve Schroader for their assistance.

(Någon rapport om detta virus i Sverige, har ännu så länge inte inkommit. Skulle det komma så rapportera gärna till Stacken. Här finns också möjlighet att få Vaccine. *Red.*)

the REAL story of UNIX 5.3 ?

by Rich (rich@devvax....), UUCP Fri Sep 18 17:27:52 1987

[grrrr]

Ok, I'm sick of all this massive confusion about who, what, where, and why UNIX 5.3 for the 80386 is, isn't and should be. Since I have the scoop, I think I'll pass it all on to the world.

In the beginning, there was AT&T.

(You knew it would start that way didn't you ?)

AT&T developed UNIX 5.3 in their labs getting it to run on their line of 3B computers. At this point, they figured that it would be a great idea to put the system on a 80386 chip. Why not ? They were going to end up selling 80386 machines some day soon anyway. So, where is the best place to go to get your OS ported to a chip ? How about the people who made the chip. Well, that's what they did. Intel was contracted to port the 5.3 code to the 80386. Intel, not wanting this massive responsibility for themselves decided to subcontract the work. They chose the best company that they could to do the port because they (Intel) were responsible for delivering back to AT&T a running version of 5.3. This company is Interactive Systems, Inc. in Santa Monica, CA.

Interactive (call them IS for short) went quickly to work porting the code, and in no time at all they had a working version. Not ready to release yet, but working. The way it works is like this: IS would complete a release of the code, be it beta or whatever, and they would release the code back to Intel. Intel would in turn release the code back to AT&T. AT&T would then release the code to those companies which had contracts with them to supply the source to 5.3 as it became available.

So, IS is not just a code porting shop. They also sell UNIX 5.3, but they only sell it to companies. They know very well what kind of crap Microport goes through

trying to support their systems. (Lets face it, a great deal of people out there are buying UNIX for their AT and they don't know crap about crap when it comes to crap. I wouldn't be a customer support person for anything. People calling asking questions like "I put the floppy in the drive without taking the cover off the disk and it won't read it. What should I do ?".) Anyway, since IS sells this system too, they take the code, make some changes to it and market it as two different products. The first is called 386/ix. This is just the stock 5.3 kernel and associated utilities, such as compilers, uucp, sys adm, and so forth. The second is called VP/ix. This is 386/ix with extensions to support the "virtual PC" mechanism. This is the MS-DOG under UNIX option. It utilizes the virtual-86 mode of the 80386.

Now, obviously, since IS is the company doing the port, they are going to be the people who offer the product first right ? Wrong. IS will only sell the system if it is absolutely AT&T SVID certified. And they were, in fact, the first company to have such a product. And to give you a little clue, they only started shipping their 386/ix product about the end of August. What does that mean to those of you who got a similar product from Bell or Microport before then ? That's right. Beta release. Surprised ?

"So, whose product is it that I got when I bought my Bell Tech. machine ?" you're asking yourself. Microport. They are one of the companies that buys the source from AT&T when it becomes available. They want to be the first company in the marketplace to offer a 5.3 for the 80386. So they release code that isn't SVID certified. And if you don't believe it, call them and ask. Call AT&T and ask them what companies currently have a SVID certified 5.3 available. Don't worry, it's a short list.

Bell gives you a UNIX port when you buy their hardware because it's a great marketing ploy. What a lot of people don't know is that all you get is the base system without documentation. Thanks guys.

None of this is to say that any of THESE companies are shifty. As far as I'm concerned, every successful company on the face of the earth is shifty. Caveat emptor. Microport doesn't sell any product called "SVID Certified SystemV/386". It's up to the buyer to find that out. I'm positive that eventually the Microport product will be SVID certified and that just like their SystemV/AT product, it will eventually become a mature, well running, well liked, popular product. The problem is that buyers simply don't beware before they buy something. You've got to watch your butt. How do you think the owners of 1984 Corvettes feel ? That was a new product too.

As for the MS-DOG under UNIX option, Locus Computing of Santa Monica, CA does the DOSMerge stuff for Microport, and Interactive works in conjunction with Phoenix Technologies to do the VP/ix stuff. I have used both. I like VP/ix infinitely better. That is just my

opinion. Take it or leave it.

And as an added note, another company that buys the 5.3 source from AT&T is a little garage shop called Microsoft. Didn't forget about them did you ? This is comp.unix.xenix isn't it ? MS is taking the code, and working with Interactive Systems (they are everywhere, aren't they ?) are producing a 5.3 that will run COFF, Xenix, and DOS executables. Pretty slick, huh ? This product will be called Microsoft UNIX 5.3. Pretty original. Give them marketing people a raise.

Does that clear up anything ? I hope so. If you have any more questions, drop me e-mail. I'm going to regret making that statement. Bye.

How to Post a Fake

by Chuq Von Rospach, UUCP, Sat, 18 Apr 87 10:10:56 PDT

Q: I'm curious: How can you fake a posting without being root? When I post anything to mod.os (er, excuse me, comp.os.research) I'm always listed as the sender. Not wanting to be the bad guy I've never tried to crack it, but I am curious as to the hole that causes the problem.

darrell@beowulf

A: Darrell asks an interesting question, and I might as well let everyone know while I'm at it.

As background, be aware that USENET has a major security hole, in that there is no way for the program rnews to know where a message came from. It has to implicitly trust the information in the header of the message. While uucp does site verification across the net, that information is not passed along through uux to the executed program on the other side. It has to trust the data it gets.

This leads to a trivial hack for creating bogus and untraceable messages. Take an existing message (I borrowed this one from mod.announce):

```
Path: sun!cbosgd!mark
From: mark@cbosgd.ATT.COM (Mark Horton)
Newsgroups: mod.announce,news.announce.important
Subject: mod.announce is being renamed news.announce.important
Message-ID: <3525@cbosgd.ATT.COM>
Date: 13 Apr 87 15:09:20 GMT
Organization: AT&T Bell Laboratories, Columbus, Oh
Lines: 9
Approved: mark@cbosgd.MIS.OH.ATT.COM
Xref: sun mod.announce:24 news.announce.important:1
```

This is your template. Now, change the header lines to fit, and delete the Xref line:

```
Path: cbosdg!mark
From: mark@cbosdg.ATT.COM (Mark Horton)
Newsgroups: mod.announce
Subject: Newgroup renaming is a failure, film at 11.
Message-ID: <3.14159@cbosdg.ATT.COM>
Date: 1 Apr 87 00:00:00 GMT
Organization: The Backbone Cabal, Inc.
Lines: 9
Approved: mark@cbosdg.MIS.OH.ATT.COM
```

Don't worry about the # of lines, inews will be nice enough to adjust it for you. Store that in a file, add the message body to it, and execute:

```
% /bin/rnews < file
```

rnews will read it just as if it had come over the network, and install it. It believes everything you said in the header. When it passes it along, the Path: becomes "sun!cbsogd!mark" and it gets passes along just like a real message. The only place where this is traceable is the Path variable, because you can see that my site is at the beginning of the list of real paths. You can avoid this in a couple of ways if you want to be real sneaky:

• The kremvax syndrome: Instead of having a single address in the path, put in a bunch:

```
Path: kremvax!nsacyber!prarie!wobegon!himom!cbosdg!mark
```

Depending on your ingenuity, you may make it almost impossible to tell where the message joined the net for real.

• Drop out of the loop: Even more fun, rather than execute rnews on YOUR site, execute it on someone else's.

```
% uux - -z ihnp4!rnews < file
```

The Path is now "ihnp4!cbosdg!mark" and your own site is nowhere to be seen. Completely untraceable, unless someone wants to compare uucp's LOGFILE entry times with news 'log' entries and backtrack. Which assumes that they figure out it is happening before they flush the logs. And that they have the time, and care.

That's how you forge messages. And as long as the uucp links exist, there is no way to fix this, because a vital piece of information isn't passed out of uucp.

The possibilities are endless, of course. You can not only post April Fool's messages, but post messages FOR people that they can never prove they didn't post. Completely untraceable. You can change your name, your machine, your religious background, all untraceable. Possibly even skip out on child support, if you find the right control message.

Kids, don't try this at home! These people are paid professionals, and know the risks involved... (grin)

(Next week, how to kick a site off the net with cancel messages!)

[For those of you who never saw the KREMVAX 1984 April Fools' Day hoax, see *ACM SIGSOFT Software Engineering Notes*. July 1984. vol 9 no 4. PGN]

A password-breaking program

by Dean Pentcheff, UUCP Mon 11 May 87 21:24:45 PDT

A few days ago on our university UNIX system (4.3BSD), a friend of mine received the message reprinted below. Very briefly, someone seems to have cracked the passwords in the "passwd" file and sent a piece of warning mail to all the users whose password he cracked. Note that my friend's password was a dictionary word, while mine (uncracked) was a proper name beginning with a capital letter.

> *To: xxxxxx*
> *Subject: A matter of security..*
>
> *Your password: zzzzzzz [correctly stated]*
>
> *As an experiment, and something of an unofficial public service, I*
> *have been experimenting with a password breaking program that was*
> *recently released into the public domain. Since anyone can use this*
> *program now, I thought I'd run it on violet's password file to see*
> *which passwords could be broken. Yours was one of them. If you're*
> *security conscious, or just don't like the idea of your password*
> *being so easily broken, then I would advise that you change it to*
> *a word not found in the english dictionary, or use a combination of*
> *upper and lower case letters. Either of these methods will render*
> *your password fairly invulnerable to attack..*
>
> *Yyyyyyyyy Yyyyyyyy*

The AMIGA virus

by Bill Koester (Commodore Technology), UUCP 13 Nov 87 19:32:05 GMT

When I first got a copy of the Amiga VIRUS I was interested to see how such a program worked. I disassembled the code to a disk file and hand commented it. This article will try to pass on some of the things I have learned through my efforts.

1. - Definition.

The Amiga VIRUS is simply a modification of the boot block of an existing DOS boot disk. Any disk that can be used to boot the Amiga (ie workbench) has a reserved area called the boot block. On an Amiga floppy the bootblock consists of the first two sectors on the disk. Each sector is 512 bytes long so the boot block contains 1024 bytes. When KickStart is bringing up the system the disk in drive 0 is checked to see if it is a valid DOS boot disk.

If it is, the first two sectors on the disk are loaded into memory and executed. The boot block normally contains a small bit of code that loads and initializes the DOS. If not for this BOOT CODE you would never see the initial CLI. The normal BOOT CODE is very small and does nothing but call the DOS initialization. Therefore, on a normal DOS boot disk there is plenty of room left unused in the BOOT BLOCK.

The VIRUS is a replacement for the normal DOS BOOT CODE. In addition to performing the normal DOS startup the VIRUS contains code for displaying the VIRUS message and infecting other disks. Once the machine is booted from an infected disk the VIRUS remains in memory even after a warm start.

Once the VIRUS is memory resident the warm start routine is affected, instead of going through the normal startup the VIRUS checks the boot disk in drive for itself. If the VIRUS in memory sees that the boot block is not infected it copies itself into the boot block overwriting any code that was there before. It is in this manner that the VIRUS propagates from one disk to another. After a certain number of disks have been infected the VIRUS will print a message telling you that Something wonderful has happened.

2. - Dangers.

When the VIRUS infects a disk the existing boot block is overwritten. Since some commercial software packages and especially games store special information in the boot block the VIRUS could damage these disks. When the boot block is written with the VIRUS, any special information is lost forever. If it was your only copy of the game then you are out of

luck and probably quite angry!!

3. - Mechanics.

Here is a more detailed description of what the virus does. This is intended to be used for learning and understanding ONLY!! It is not the authors intention that this description be used to create any new strains of the VIRUS. What may have once been an innocent hack has turned into a destructive pain in the #$@ for many people. Let's not make it any worse!!

a.) Infiltration.

This is the first stage of viral infection. The machine is brought up normally by reading the boot block into memory. When control is transferred to the boot block code, the virus code immediately copies the entire boot block to $7EC00, it then JSR's to the copied code to wedge into the CoolCapture vector. Once wedged in, control returns to the loaded boot block which performs the normal dos initialization. Control is then returned to the system.

b.) Hiding Out.

At this point the system CoolCapture vector has been replaced and points to code within the virus. When control is routed through the CoolCapture vector the virus first checks for the left mouse button, if it is down the virus clears the CoolCapture wedge and returns to the system. If the left mouse button is not pressed the virus replaces the DoIO code with its own version of DoIO and returns to the system.

c.) Spreading.

The code so far has been concerned only with making sure that at any given time the DoIO vector points to virus code. This is where the real action takes place. On every call to DoIO the virus checks the io_Length field of the IOB if this length is equal to 1024 bytes then it could possibly be a request to read the boot block. If the io_Data field and A4 point to the same address then we know we are in the strap code and this is a boot block read request. If this is not a boot block read the normal DoIO vector is executed as if the virus was not installed. If we are reading the boot block we JSR to the old DoIO code to read the boot block and then control returns to us. After reading, the checksum for the virus boot block is compared to the checksum for the block just read in. If they are equal this disk is already infected so just return. If they are not equal a counter is incremented and the copy of the virus at $7EC00 is written to the boot block on the disk. If the counter ANDed with $F is equal to 0 then a rastport and bitmap are constructed and the message is displayed.

d.) Ha Ha.

< Something wonderful has happened >
< Your AMIGA is alive!!! >
< and even better >

< Some of your disks are infected by a VIRUS >
< Another masterpiece of the Mega-Mighty SCA >

4. - Prevention.

How do you protect yourself from the virus?

1) Never warm start the machine, always power down first. (works but not to practical!)

2) Always hold down the left mouse button when rebooting. (Also works, but only because the VIRUS code checks for this special case. Future VIRUS's may not!)

3) Obtain a copy of VCheck1.1 and check all disks before use. If any new virus's appear this program will be updated and released into the public domain. V-Check1.1 was posted to usenet and will also be posted to BIX. (Just like the real thing the best course of action is education and prevention!)

En påse jordnötter!

av Jan Michael Rynning

EFTER att ha använt Desktop Publishing till att sätta STACKPOINTER sedan starten 1978 och TEX sedan 1982, tycker jag Macintorsk är något överreklamerad både som användarvänlig dator och som verktyg för trycksaksframställning.

Till det här numret har vi både använt TEX på DEC-20 och PageMaker på Macintorsk. Kan Du se vilka artiklar som är satta med vilket system? Ledtråd: Var hittar Du de konstiga bokstavsavstånden och de uppochnervända citationstecknen?

Den traditionella påsen jordnötter utlovas i belöning till den som inkommer med den bästa gissningen om vilka artiklar som är satta med vilket system.

Redaktörn

av /hn

Lite före påsk hände det som får svetten att bryta fram på en. Jag fick indikering att det var fel på min hårddisk till Macen.

Min erfarenhet säger att först provas en ominitiering av disken. Till det krävs backup. Men att göra en total backup på disketter var inte lockande. 40 disketter avskräckte iallafall mig. Det blev att se mig om efter alternativ.

Hos en bekant i Upsala (Tack än en gång, Stefan!) hittar jag en Apple tapebackup med tillhörande program. Lånar hem det hela.

Börjar med att göra en volymbackup. Det hela knyter sig pga de felaktiga sektorerna på disken. Gör då en fil-för-fil backup. På ett ställe får jag reda på att det inte går att kopiera en viss tillämpning. Jaha, tänker jag, den får jag väl vara utan t. v. Fortsätter med resten utan missöden.

Initierar om disken. Kontrollerar med ett diskdiagnosprogram. Inget fel på disken. Bra, då skall innehållet återställas. Klockan ett var jag klar. Tyckte iofs att det var lite väl mycket tomutrymme som blev återlämnat. Men ägnade inte någon större energi åt att undersöka det hela. Sov istället.

Nästa kväll tittade jag lite mer engagerat på diskens innehåll. Till min förfäran så ser jag då vad som gjorde det myckna tomutrymmet. Alla tillämpningar och underkataloger som låg på samma gren av katalogstrukturen som den tillämpning som inte gick att läsa, hade också försvunnit. Så hade inte jag uppfattat felmeddelandet!?

Slutsats för min del: Kontrollera innehållet i backupen innan du gör en ominitiering av disken. Backupen kanske inte är vad du tror. Iallafall inte Apples.

Att jag sedan, när jag lämnat tillbaka tapebackupen, missade sista tåget i Upsala är en annan historia.

Någonting som heter HyperCard har kommit till Macen. Nyfiken som jag är så ville jag få något läsbart om det. Hjälpfilen tänkte jag, om jag skriver ut den, blir det säkert något att läsa. Den hjälpfilen kallar jag hjälpfil. 410 skärmsidor lång. En hel liten bok. Väljer jag 8 bilder per sida vid utskriften tar det 2m18s per ark på LaserWriter. Sammanlagt nästan 2h. Så det blev till att låta det komma ut i en nattmangling. Har jag läst alltihopa? Nä, det roliga var väl att få ut det på papper!?

Ni har väl hört att DEC-användare

väljer en Macintosh om de skall använda persondator som terminal. De som använder I*M föredrar en DOS-maskin. De är väl så vana vid krångliga system redan, att det blir för enkelt med en Mac.

En medarbetare hade gjort en stor sammanställning. För att få det enhetligt användes mallar till dokumenten. Det hela resulterade i över 800 dokument. Av olika orsaker hade jag anledning att titta över det hela. Upptäcker då att på varje dokument hängde det med ca 12 KB skräp. Det blir ung 10 MB på disken. Alltså över halva disken är skräp!

Lita inte på en mall. Även om början ser bra ut kan det finnas en svans som är större än mallen.

Ja, det var det hele.

TCP/IP på Kicki m.m.

av Jan Michael Rynning

SVERIGE håller på att kopplas ihop med ett nytt universitetsdatornät, kallat SuperSUNET, som skall kunna transportera alla typer av nätverksprotokoll som går att skicka över Ethernet och för att kunna nyttja det fullt ut har Kicki begåvats med ytterligare en typ av nätverk: TCP/IP.

SuperSUNET.

SuperSUNET är nu i provdrift mellan Stockholm, Göteborg och Lund. Nätet består av fasta 64 Kbit/s-förbindelser med fjärrbryggor, routers för TCP/IP och DECnet, samt filtrerande bryggor för övriga protokoll.

TCP/IP på Kicki.

TELNET används för terminalkommunikation och FTP för filöverföring. HELP IMPCOM och HELP FTP för mer info. MAIL går automatiskt över rätt nät.

Dokumentation.

På RFC: på Kicki finns de flesta av RFC-standarderna. SYS:HOSTS.TXT är den svenska hosttabellen i RFC952-format. DOC:TCPIP.DOC innehåller Charles L. Hedricks introduktion, som vi publicerade i förra numret av STACKPOINTER.

Lee Schneider

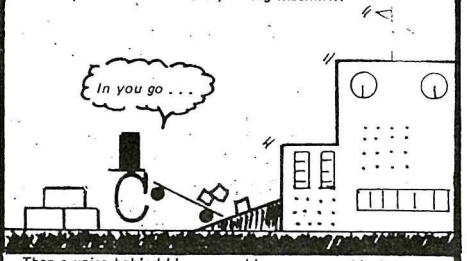
Todd Voros

FORTRAN MAN

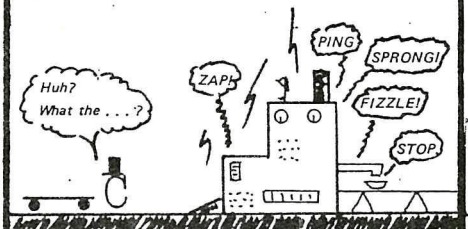
Despair not Fortran Man fans, deep in the heartland of the USA the fertile minds of Lee and Todd are churning, churning, churning . . . . . F-Man series.

The Dragon

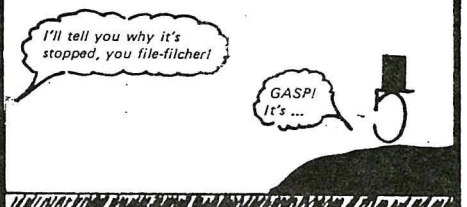
But then, before he can do anything, Fortran Man is dumped into the file compressing machine!!

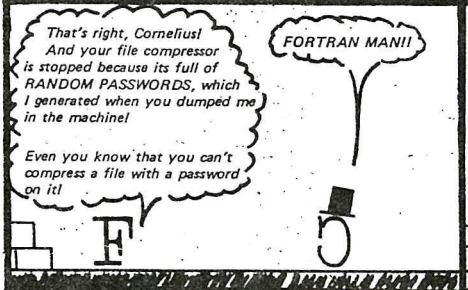


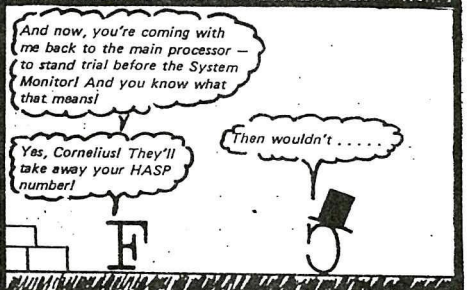
For a moment, nothing happens . . . . . but then suddenly . . . . . .



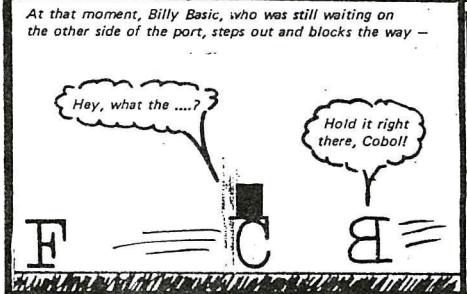
Then a voice behind him causes him to turn suddenly . . . . .



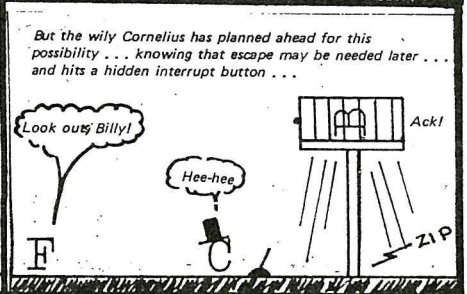




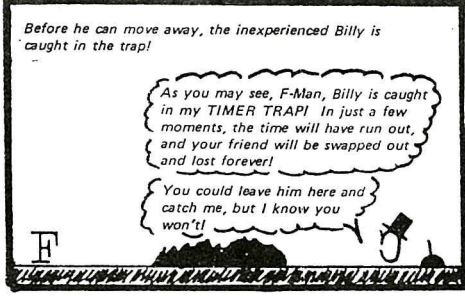
At that moment, Billy Basic, who was still waiting on the other side of the port, steps out and blocks the way —

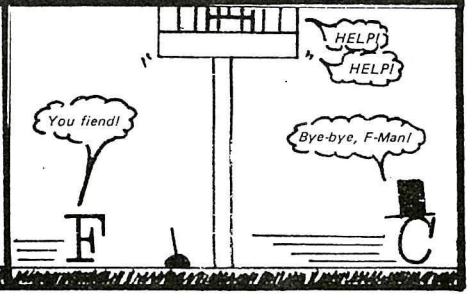


But the wily Cornelius has planned ahead for this possibility . . . knowing that escape may be needed later . . . and hits a hidden interrupt button . . .



Before he can move away, the inexperienced Billy is caught in the trap!





GNU News

by Richard M. Stallman (rms@wheaties)

Here is a summary of news from the GNU project.

Some parts of BSD are becoming free.

After years of urging from us and others, the people who maintain Berkeley Unix have decided to release various parts of it (those which don't contain AT&T code) separately as free software. This includes substantial programs which we hope to use in GNU, such as TCP/IP support and the C-shell.

Berkeley and GNU project cooperating.

The next release of Berkeley Unix may contain Make, AWK and SH from the GNU project instead of those from Unix.

The reason is that they would like to have improvements in these programs like those in system V.3; but they find the new restrictions on V.3 licenses unpalatable. Both we and they hope they never get a V.3 license. We may help them avoid it by providing alternative software.

GNU Make already supports the system V features; a volunteer is now writing the

extensions for Gawk.

Shell disappointment.

For a year and a half, the GNU shell was "just about done". The author made repeated promises to deliver what he had done, and never kept them.

Finally I could no longer believe he would ever deliver anything.

So Foundation staff member Brian Fox is now implementing an imitation of the Bourne shell. Once it is done, we will extend it with the features of the Korn shell, thus coming to Berkeley's aid.

We may use Sprite, or Sprite file system with MACH.

We still hope to use the MACH kernel from CMU when it becomes free, after the parts of Berkeley Unix which currently form part of it have been replaced as planned.

The MACH people say that in a month or two certain new features (call-outs from the kernel to user code) should be ready that will enable us to start working on replacing some of these parts with new code.

One thing we are considering is adapting the file system from Berkeley's Sprite kernel for use in MACH. This file system was designed from the beginning to work in a distributed manner. The file system is the largest part of MACH that needs replacement, now that the Berkeley TCP/IP code, also used in MACH, has been declared free.

GNU Make is done.

The GNU version of Make is now ready, and will be distributed soon. It features conditionals, pattern rules, and indirect search for implicit rules, and built-in functions for text processing.

Here is how a GNU Makefile can say that the file 'foo' is linked from the object files of all C source files in the current directory:

```
objects=$(subst .c,.o,$(wildcard *.c))
foo: $(objects)
    $(CC) -o foo $(objects) $(LDFLAGS)
```

Most of libraries are done.

Roland McGrath, who contributed a great deal to GNU Make, has a nearly complete set of ANSI C library functions. We hope they will be ready some time this spring. These join the GNU malloc, regexp and termcap libraries that have existed for some time.

Meanwhile, Steve Moshier has contributed a full series of mathematical library functions.

Gprof replacement done.

Foundation staffer Jay Fenlason has recently completed a profiler to go with GNU C, compatible with 'gprof' from Berkeley Unix. I hope it will be distributed with GNU C soon.

GNU mailer being done.

Landon Noll and Ronald Karr of Amdahl are writing a mail queueing and delivery system, called Smail. This project will be a supported part of Amdahl's UTS system and it will be available on exactly the same terms as GNU Emacs!

We may use this mailer for the GNU system, or another mailer that Rayan Zachariasen is writing, whichever turns out best.

Our first large donation.

Software Research Associates, a Japanese software house, has donated $10,000 to the GNU project. In addition they plan to send us a Sun-like SONY workstation and lend us a staff programmer for 6 months.

This represents the influence of Kouichi Kishida, who organized the Japanese Sigma project (to stimulate Unix competence in Japan), only to conclude later that the project had gone astray and that a "grass roots movement" was needed instead. We hope to be this movement.

Looking for a tech writer.

We are trying to hire a technical writer, but so far we have not found anyone suitable. It seems that tech writers are not as likely as programmers to accept a pay cut to work for our cause. We still have a couple of candidates who are possible, and we're still looking.

Termcap Manual.

We are now publishing the first thorough manual for Termcap, entitled "Twice as much as you ever wanted to know about Termcap".

Ghostscript status.

Ghostscript, the free Postscript for GNU, will with luck be finished (except for bugs) in March. Therefore, it might be ready for us to distribute a few months later.

Emacs version 18.50.

To be available in a few weeks, will fix many bugs and add support for the 80386, the Sun 4, the Convex, the IRIS 4d and the HP 9000 series 800; also support for system V.3.

GDB can read COFF format.

COFF is the hairy executable file format used on system V. Recently Dave Johnson of Brown University contributed support for reading COFF files in GDB, the GNU debugger. This code will appear in GDB 2.5, accompanying Emacs 18.50.

As a result, it should now be possible to use GDB on system V without a large amount of work.

In general, support for COFF isn't important for the GNU project, since we are going to use the BSD object file format in GNU. Everything said below about VMS applies to COFF support as well.

GDB and GNU C support the 32000.

GNU C has now compiled itself correctly on the Sequent 32000 system. The port was done by Michael Tiemann of MCC, who says it is more reliable than Sequent's compiler and yields a 40% speedup for several programs including a Prolog interpreter.

Support for the 32000 is now released in GNU C version 1.17, along with existing support for the 68000 series and the Vax.

GDB support for the 32000 is in GDB 2.5, to appear with Emacs 18.50.

GNU C ports in progress.

People are working on porting GNU C to the IBM 370, to the IBM RT/PC, and to the 80386. The 80386 is the easiest; there is little doubt that this port will be available in a few months at most. The other machines have more troublesome architectural differences and it isn't yet certain whether GNU C can handle them fully without significant new features.

G++, the GNU C++ compiler.

Michael Tiemann of MCC has written a C++ compiler as an extension of GNU C. This is the first compiler that compiles C++ directly instead of preprocessing it into C.

G++ is now being tested at several sites. Michael believes it is as reliable as AT&T's C++ preprocessor, but this still leaves a long way to go before it is a solid product.

G++ comes with GDB+, a version of GDB that supports C++ class operations in its expression evaluator.

GNU C is becoming reliable.

GNU C bug reports are becoming less frequent, suggesting that it is approaching a state of reliability. People are still reporting bugs, but they also say they think there are fewer bugs than in commercial compilers.

John Gilmore is now compiling all of BSD Unix with GNU C. He has found several bugs, but not a great number for such a large amount of code never before compiled.

GNU C for VMS.

Fed up with the deficiencies of the VMS C compiler, David Kashtan from SRI decided to spend a couple of weeks and make GNU C run on VMS. After making considerable changes to satisfy the VMS

C compiler, he got it running and was able to take most of the changes out.

We hope to deliver VMS support in GNU C version 1.19 or 1.20, but we can't be certain when he will finish merging it and deliver it. (Uh oh, I hope it's not going to take a year...)

When VMS support is delivered to us, the usual GNU C sources will contain everything needed for it, but you can't compile them with the standard VMS C compiler due to its various incompatibilities and deficiencies. You will need a binary of GNU C. We plan to offer mag tapes with VMS backup savesets containing binaries as well as sources.

Other GNU programs currently working on VMS include GNU Emacs and Bison.

Please don't ask us to devote effort to additional VMS support, because it is peripheral to the GNU project. We merge in and support VMS ports that users do, because it is hard to refuse to pass on work that other people have done. But even when the changes are clean, this drains considerable effort from our real goal, which is to produce a complete integrated system. (When they aren't clean, we summon up the courage to ignore them.) Merging VMS GNU Emacs and reorganizing the changes to ease future maintenance consumed several weeks even though the "real work" was done by others. I hope we have learned not to let this happen again.

How NOT to find out when a program becomes available.

I have dared to make a few predictions about when certain programs will be ready for distribution. Now, after the fact, I hope that wasn't make a mistake.

Creating a given program often takes much longer than expected. If this happens, many of you who are eager to use it might have the idea of phoning or writing to ask me whether the program is available yet. Or, even worse, to describe which parts are completed, or which machines it will be ported to, or how easy it will be to port to certain machines, or which features are going to be implemented.

Some of these questions have no answers. When a program isn't finished, we don't know which features we will add while finishing it. We don't make plans to port programs because our usual policy is to leave porting up to you. This is a way of recruiting wider participation.

Responding to these inquiries would divert time and energy and slow the completion of the program, causing even more inquiries. The amount of time left to devote to programming could approach zero.

Only you can save the GNU project from this absurd fate. Please exert your will power, be patient, and wait for us to announce that programs are available. If you would like to see more GNU software

appear faster, the best thing to do is to volunteer a significant amount of your time.

The best way to find out quickly when a piece of GNU software is released for general on-line distribution is to arrange to receive the info-gnu mailing list on which our announcements are made. To receive info-gnu, send a request to:

work on reliability.

When we expect true reliability to take many months to achieve, we may offer a beta-test tape at an earlier date, as we are doing with GNU C. We announce these tapes in the usual ways.

If you can't receive Usenet or Internet mail and you are planning to order a tape

```
ucbvax!prep.ai.mit.edu!info-gnu-request  (usenet)
info-gnu-request@prep.ai.mit.edu         (internet)
```

Some specific programs such as Emacs, GCC, GDB, GNU Chess and G++ have specific mailing lists of their own on which new versions are announced. To be on these lists, just mention in your request which programs you are interested in.

Tape distribution normally starts some weeks later than on-line distribution, either because we demand a higher standard of reliability for users who cannot easily get upgrades, or because we are hopelessly confused and can't get our act together. We generally make another electronic announcemet when tape distribution starts. You can order tapes in advance if you wish, if you understand that delivery may be delayed for many weeks while we

of a specific program, you can call the Foundation's phone to find out whether the program is yet available for tape distribution. Our phone number is (617) 876-3296. Opus might agree to send you a letter when the program is ready, if she feels she has time left after her other responsibilities.

If you are planning to get a copy via someone else, please ask that person to inform you when you can do it. This will spread the burden.

If you are truly desperate for a prognostication, you might try consulting your local fortune-tellers. They don't know any less than we about the future.

STACKPOINTER

STACKPOINTER är husblad för datorföreningen STACKEN på KTH. STACKPOINTER utkommer när material i tillräcklig mängd finns, förhoppningsvis 4-5 gånger per år.

Redaktör: Hans Nordström
I redaktionen: Jan Michael Rynning
Ansvarig utgivare: Mats O Jansson

Färdigställd: 1988-05-24

Datorföreningen STACKEN

Post till föreningen skickas till adressen eller läggs i postfacket på NADA.

Datorföreningen STACKEN
c/o NADA
KTH
100 44 STOCKHOLM

Klubblokal: Lindstedtsvägen 5
Datorhall: Brinellvägen 32
Telefon: 08-791 87 97

Postgiro 433 01 15-9
Bankgiro 344-3595

Medlemsavgift för 1988 är SEK 88.

Ordf: Stellan Lagerström, *stellanl@lne.kth.se*, 08-790 78 14 (arb) 08-663 68 25 (hem)
Sekr: Olle Betzén, *betzen@lne.kth.se*, 0758-318 48 (hem)
Kassör: Henning Croona, *hcroona@stacken.kth.se*, 08-799 82 80 (arb) 08-739 17 40 (hem)
Hexm: Henrik Björkman, *bjorkman@lne.kth.se*, 08-30 89 01 (hem)
Redax: Hans Nordström, *sib@stacken.kth.se*, 0760-405 69 (modem), 08-797 80 13 (arb)
Mats O Jansson, *moj@stacken.kth.se*, 08-34 93 20 (arb) 08-711 70 37 (hem)
Thord Nilsson, *thordn@stacken.kth.se*, 08-712 00 90 (arb) 08-749 21 92 (hem)

Elpost: *stacken@stacken.kth.se*
UUCP alternativt: *...{uunet,mcvax,...}!stacken.kth.se!stacken*
ARPA alternativt: *stacken%stacken.kth.se@uunet.uu.net*
BITNET/EARN alternativt: *STACKEN@SESTAK*
SUNET DECnet alternativt: *KICKI::STACKEN*
PSI alternativt: *PSI%0240200101905::KICKI::STACKEN*

The Ten Commandments for C Programmers

1. Thou shalt run *lint* frequently and study its pronouncements with care, for verily its perception and judgement oft exceed thine.

2. Thou shalt not follow the NULL pointer, for chaos and madness await thee at its end.

3. Thou shalt cast all function arguments to the expected type if they are not of that type already, even when thou art convinced that this is unnecessary, lest they take cruel vengeance upon thee when thou least expect it.

4. If thy header files fail to declare the return types of thy library functions, thou shalt declare them thyself with the most meticulous care, lest grievous harm befall thy program.

5. Thou shalt check the array bounds of all strings (indeed, all arrays), for surely where thou typest "foo" someone someday shall type "supercalifragilisticexpialidocious".

6. If a function be advertised to return an error code in the event of difficulties, thou shalt check for that code, yea, even though the checks triple the size of thy code and produce aches in thy typing fingers, for if thou thinkest "it cannot happen to me", the gods shall surely punish thee for thy arrogance.

7. Thou shalt study thy libraries and strive not to re-invent them without cause, that thy code may be short and readable and thy days pleasant and productive.

8. Thou shalt make thy program's purpose and structure clear to thy fellow man by using the One True Brace Style, even if thou likest it not, for thy creativity is better used in solving problems than in creating beautiful new impediments to understanding.

9. Thy external identifiers shall be unique in the first six characters, though this harsh discipline be irksome and the years of its necessity stretch before thee seemingly without end, lest thou tear thy hair out and go mad on that fateful day when thou desirest to make thy program run on an old system.

10. Thou shalt foreswear, renounce, and abjure the vile heresy which claimeth that "All the world's a VAX", and have no commerce with the benighted heathens who cling to this barbarous belief, that the days of thy program may be long even though the days of thy current machine be short.

Henry Spencer