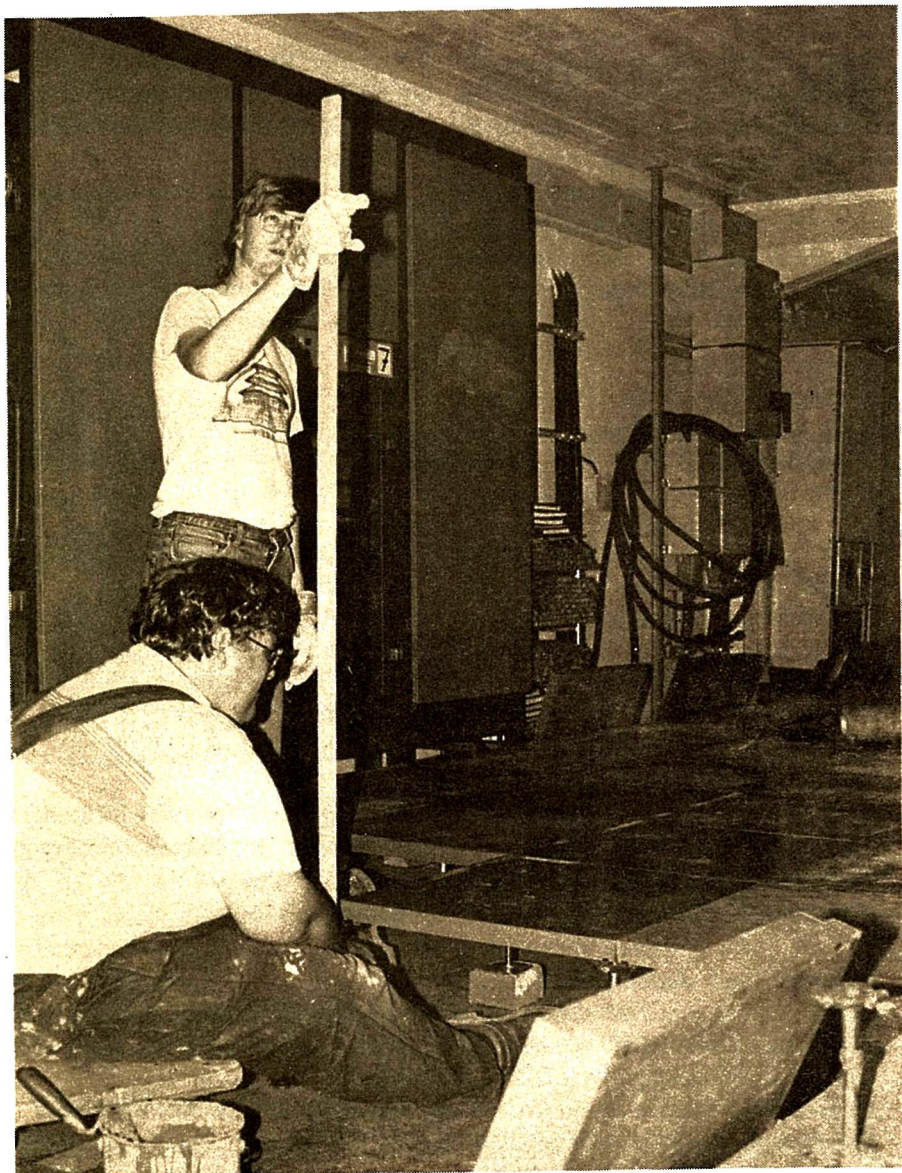


# STACK POINTER

4-1989. Organ för Datorföreningen STACKEN, KTH.



# STACKPOINTER



STACKPOINTER är organ för Datorföreningen STACKEN på KTH. Vi tänkte försöka oss på en mer regelbunden utgivning under 1989: Manusstopp och redaktionsmöte helgen efter månadmöte i jämn månad. Citera gärna, men kom ihåg att ange källan och tillskicka gärna föreningen ett exemplar.

Redaktör: Hans Nordström.  
I redaktionen: Jan Michael Rynning.  
Foto: Per Eriksson och Jan Michael Rynning.

Ansvarig utgivare: Mats O Jansson.

Färdigställd: 5:e november 1989.  
Nästa manusstopp och redaktionsmöte:  
11:e-12:e november 1989.

## Datorföreningen STACKEN



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

Datorföreningen STACKEN  
c/o NADA  
KTH  
100 44 STOCKHOLM

Telefon: 08-791 87 97  
Klubblokal: Lindstedsvägen 5  
Datorhall: Teknikringen 22

Ordf: Stellan Lagerström {08-790 78 14 (arb)  
08-46 93 93 (hem)  
Skr: Olle Betzén {0758-318 48 (hem)  
Kass: Henning Croona {08-799 82 80 (arb)  
08-739 17 40 (hem)  
Red: Hans Nordström {08-797 80 13 (arb)  
0760-405 69 (modem)  
Hexm: Johnny Eriksson {08-790 65 17 (arb)  
08-27 24 30 (hem)  
Övr: Mats O Jansson {08-34 93 20 (arb)  
08-711 70 37 (hem)  
Övr: Thord Nilson {08-712 00 90 (arb)  
08-749 21 92 (hem)

Medlemsavgift för 1989: 89 kronor för  
studerande, 189 kronor för övriga  
Postgiro: 433 01 15-9  
Bankgiro: 344-3595

Electronic mail: *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*  
NORDUNET DECnet alternativt: *KICKI::STACKEN* eller *60456::STACKEN*  
PSI alternativt: *PSI%0240200101905::KICKI::STACKEN*

# I detta nummer

**F-N man** ..... 4  
**PDI 3**..... 5

**Datorhallsbygget** ..... 6  
**GNU's Bulletin** ..... 8

## Kontostrul på Kicki

*av Jan Michael Rynning*



ATABASEN över vilka konton som finns på Kicki har gått sönder och vi vart tvungna att hämta tillbaka en gammal version av den från en säkerhetskopia från den 24:e februari. Om Du har bytt lösenord

efter det, försök med Ditt gamla, om Du kommer ihåg det. Om Du har fått konto efter det datumet eller inte kommer ihåg Ditt gamla lösenord, tala med någon i styrelsen.

## Nya modemnummer

*av Jan Michael Rynning*



ODEMEN på KTH har fått nya telefonnummer. En privatperson, som hade ett telefonnummer som liknade ett av de gamla modemnumren, besvärades av att dygnet runt bli uppringd av folk som ville prata med datorn eller bara gav

honom en tjutande modemsignal i örat. De gamla numren fungerar parallellt med de nya under en övergångsperiod. De nya numren är:

300/300, 1200/75:	08-14 97 30.
1200/1200, 2400/2400:	08-14 96 80.

*Framsidan: Thomas Nyström lägger golv och Stellan Lagerström mäter höjden.  
 Baksidan: Till och med kylskåpet står på lastpall, så det går lätt att flytta på.*



Lee Schneider

Todd Voros

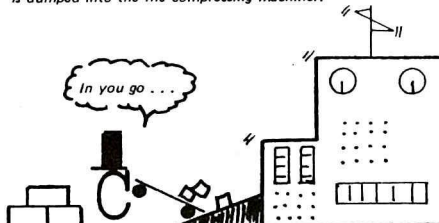
# FORTMAN

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

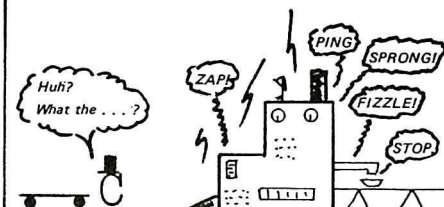
The Dragon

22

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



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



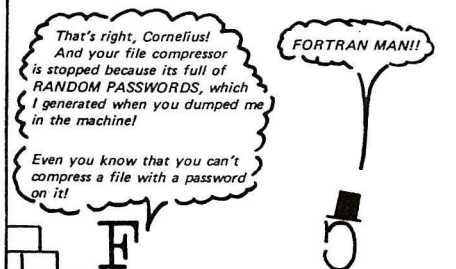
Then a voice behind him causes him to turn suddenly . . . .



That's right, Cornelius!  
And your file compressor  
is stopped because its full of  
RANDOM PASSWORDS, which  
I generated when you dumped me  
in the machine!

Even you know that you can't  
compress a file with a password  
on it!

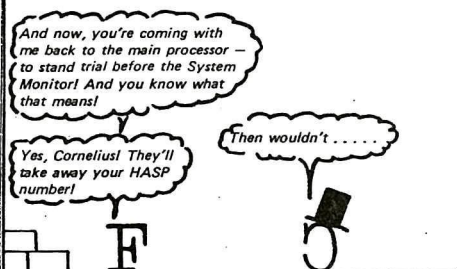
FORTMAN MAN!!



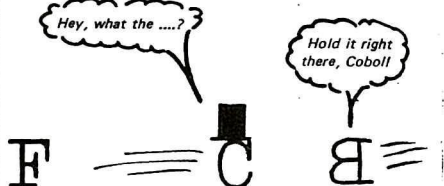
And now, you're coming with  
me back to the main processor —  
to stand trial before the System  
Monitor! And you know what  
that means!

Yes, Cornelius! They'll  
take away your HASP  
number!

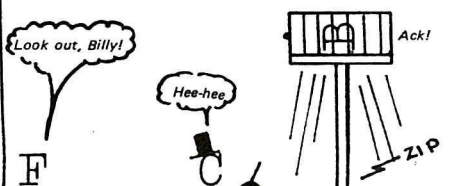
Then wouldn't . . . . .



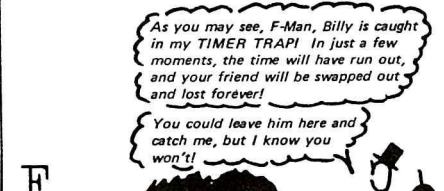
At that moment, Billy Basic, who was still waiting on  
the other side of the port, steps out and blocks the way —



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!



To be  
fortsatt



# Recept på PeanutBlaster

- 1 liten kylfläns.
- 1 STOOR kylfläns. (Kylflänsarna står för > 90% av vikten.)
- 1 st 6 A 220 V likriktarbrygga.
- 2 st 220  $\mu$ F 400 V el-lytar.
- 6 st IRFP450, 500 V 8 A (kont), 14 A (peak) transistor.
- 3 st drivsteg till dessa.
- 1 st CPU 8098 som fipplar med utspänningen. (CPU:n genererar 3 st PWM-mönster som via effekttreisorna matar motorn direkt.)
- 1 st nätaggregat till CPU, logik och drivsteg.
- 1 st knappsats för att kunna välja fasföljd och utfrekvens (stilla, 200 Hz, 250, 300, 350, 400, 450 och 500 Hz).

Ovanstående blandas och formas till en frekvensomformarburk som omvandlar 220 V enfas 50 Hz till 0–220 V trefas 0–500 Hz, max toppström 12 A rms/fas.

- 1800 E-transformatorplåtar, totalt ca 5 kg, till 2 st kammar 1.60 meter långa, med 81 poler var.
- 4000 varv koppartråd,  $\varnothing$  0.6 mm, bildades de 81 spolarna, totalt ca 1.8 kg koppartråd.
- 1.8 meter bräda, till uppfästning.
- X antal rullar kontorstejp och en försvarlig del kopplingstråd.
- PeanutCarriers, den del som rör sig.
- Fanns i tre smaker:
  - Ren aluminum.
  - Aluminium + järn.
  - Ren koppar.

Blanda rätt och det bildas en linjär asynkronmotor.

Till dekoration behövs säkringar, lim-logik, kontakter, kretskort, skruvar, mm.

Visning och provsmakning av

## PDI 3

torsdagen 2 november


i STACKEN-lokalen

efter mötet!

SAFA

# Datorhallsbygget

av Jan Michael Rynning

 ATORGOLVET är inlagt och uppmätt med laser. Rampen och det lilla rummet bredvid är färdigbyggda. Hälften av lysrören och en del av akustikplattorna är uppsatta i

*Henrik Björkman och Per Eriksson sätter ihop underredet till rampen.*



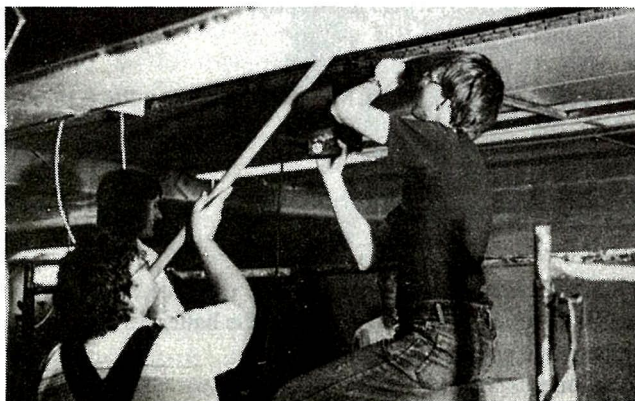
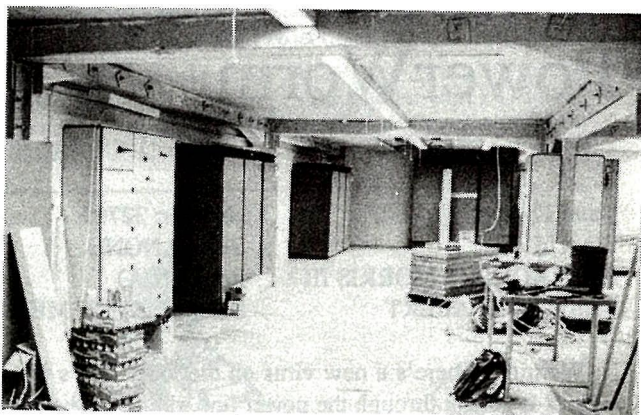
taket när jag skriver detta. Thomas Nyström har satt ihop ett elskåp, med kontaktor och säkringar, samt styrning, så man kan tända och släcka från båda ingångarna. En del av lysrören sitter på en särskild nödströmskrets, så det inte blir beckmörkt i hallen om det blir strömbrott.

Jan Lien har tigt ihop lite av varje. Från Vin- & Spritcentralen har vi bl. a. fått en UPS-anläggning, som vi tänker använda till nödbelysningen och några av datorerna. Stockholm Energi har lovat hjälpa till med distributionen till elskåpen, med materiel från någon anläggning de skall montera ner. Ericsson har gett oss kabel.

Elektroniken som skall styra kylanläggningarna har Thord Nilson lovat bygga.

Förutom taket återstår således att bygga distributionen ut till elskåpen, lägga ner kablar och uttag under golvet för datorerna, samt, sist men inte minst, att få igång datorerna.

*1/3 av datorhal-  
len med 1/3 av  
lysrören uppsatta.*



*Stellan Lager-  
ström borrar hål  
för lysrören och  
upphängningen  
av akustikplat-  
torna. Thomas  
Nyström suger  
upp borrdammet.*



*Henning Croona  
mäter avståndet  
mellan armaturerna.*



# PowerWorm

**Date:** 11-31-88 (24:60)

**Number:** 32769

**To:** ALL

**Refer#:** NONE

**From:** ROBERT MORRIS III

**Read:** (N/A)

**Subj:** VIRUS ALERT

**Status:** PUBLIC MESSAGE

**Warning:** There's a new virus on the loose that's worse than anything I've seen before! It gets in through the power line, riding on the powerline 60 Hz subcarrier. It works by changing the serial port pinouts, and by reversing the direction one's disks spin. Over 300,000 systems have been hit by it here in Murphy, West Dakota alone! And that's just in the last twelve minutes.

It attacks DOS, Unix, TOPS-20, Apple II, VMS, MVS, Multics, Mac, RSX-11, ITS, TRS-80, and VHS systems.

To prevent the spread of this dastardly worm:

- 1) Don't use the powerline.
- 2) Don't use batteries either, since there are rumors that this virus has invaded most major battery plants and is infecting the positive poles of the batteries. (You might try hooking up just the negative pole.)
- 3) Don't upload or download files.
- 4) Don't store files on floppy disks or hard disks.
- 5) Don't read messages. Not even this one!
- 6) Don't use serial ports, modems, or phone lines.
- 7) Don't use keyboards, screens, or printers.
- 8) Don't use switches, CPUs, memories, microprocessors, or mainframes.
- 9) Don't use electric lights, electric or gas heat or airconditioning, running water, writing, fire, clothing, or the wheel.

I'm sure if we are all careful to follow these 9 easy steps, this virus can be eradicated, and the precious electronic fluids of our computers can be kept pure.

RTM III

The GNU's Bulletin is the semi-annual newsletter of the Free Software Foundation, bringing you news about the GNU Project.

Free Software Foundation, Inc.  
675 Massachusetts Avenue  
Cambridge, MA 02139 USA

Telephone: (617) 876-3296  
Electronic mail: [gnu@prep.ai.mit.edu](mailto:gnu@prep.ai.mit.edu)

---



## Contents

GNU'S Who	10
What Is the Free Software Foundation?	11
What Is Copyleft?	12
GNUs Flashes	13
“Programming Freedom Activism” by Richard Stallman	14
GNU Wish List	16
“Porting BSD Unix Through GCC” by John Gilmore	17
GNU Project Status Report	19
GNU Documentation	24
GNU Software Available Now	25
How To Get GNU Software	29
Thank GNUs	30
GNU Order Form	31

## GNU's Who

Some new people have joined our full-time staff. **Joseph Arceneaux** is working on Emacs version 19. Soon to arrive are **Karl Berry**, **Kathy Hargreaves**, and **Jim Kingdon**. **Randy Smith**, who has been working on GDB, will be leaving us for grad school in the fall.

**Mike Haertel** is back with us, working on a new, more efficient malloc and on finishing the C interpreter started by Nobuyuki Hikichi. **Roland McGrath** has been hired for the summer to complete the ANSI C library which he started.

**Brian Fox** is still working for us at UC Santa Barbara. He is beta-releasing BASH, the 'Bourne Again SHell' which is the GNU version of **sh** that incorporates extensions found in the Korn and C shells. **Jay Fenlason** is writing the GNU spreadsheet program **Oleo**, and maintaining the GNU assembler, **tar** and **sed**. Jay also takes care of our backups and creating distribution tapes. **Diane Barlow Close**, our first full-time technical writer, is writing the documentation for all of the small Unix utilities that have been completed for us, while living in San Diego, CA.

**Opus Goldstein** is our jack-of-all-trades office staff. If you call our office, she is the one who answers. She fills the orders, and handles the day-to-day operations of the Foundation. **Robert Chassell** is our Treasurer. Besides dealing with foundation issues not related to programming, he is working on an elementary introduction to programming in Emacs Lisp.

**Richard Stallman** continues as a volunteer to do countless tasks, including refining the C compiler, GNU Emacs, etc. and their documentation. Finally, volunteer **Len Tower** continues as our electronic JOAT (jack-of-all-trades), handling mailing lists, information requests, system mothering, etc.

---

## GNU's Bulletin

Copyright © 1989 Free Software Foundation, Inc.

Written by: Joseph Arceneaux, Robert Chassell, John Gilmore,  
Leonard H. Tower Jr., and Richard Stallman

Illustrations: Etienne Suvasa

Permission is granted to anyone to make or distribute verbatim copies of this document as received, in any medium, provided that the copyright notice and permission notice are preserved, and that the distributor grants the recipient permission for further redistribution as permitted by this notice.



## What Is the Free Software Foundation?

The Free Software Foundation is dedicated to eliminating restrictions on copying, redistribution, understanding and modification of computer programs. We do this by promoting the development and use of free software in all areas of computer use. Specifically, we are putting together a complete integrated software system called "GNU" (GNU's Not Unix) that will be upward compatible with Unix. Some large parts of this system are already working and we are distributing them now.

The word "free" in our name refers to two specific freedoms: first, the freedom to copy a program and give it away to your friends and co-workers; second, the freedom to change a program as you wish, by having full access to source code. Furthermore, you can study the source and learn how such programs are written. You may then be able to port it, improve it, and share your changes with others.

Other organizations distribute whatever free software happens to be available. By contrast, FSF concentrates on development of new free software, building toward a GNU system complete enough to eliminate the need to purchase a proprietary system.

Besides developing GNU, the Foundation has secondary functions: producing tapes and printed manuals of GNU software, carrying out distribution, and accepting gifts to support GNU development. We are tax exempt; you can deduct donations to us on your tax returns. Our development effort is funded partly from donations and partly from distribution fees. Note that the distribution fees purchase just the service of distribution: you never have to pay anyone license fees to use GNU software, and you always have the freedom to make your copy from a friend's computer at no charge (provided your friend is willing).

The Foundation also maintains a Service Directory: a list of people who offer service for pay to users of GNU programs and systems. Service can mean answering questions for new users, customizing programs, porting to new systems, or anything else. Contact us if you want to be listed.

After we create our programs, we continually update and improve them. We release between 2 and 20 updates a year for each program. Doing this while developing new programs takes a lot of work, so any donations of pertinent source code and documentation, machines, labor or money are always appreciated.

The board of the Foundation is: Richard Stallman, President; Robert Chassell, Treasurer; Gerald J. Sussman, Harold Abelson and Leonard H. Tower Jr., Directors.

## What Is Copyleft?

In the section entitled “What Is the Free Software Foundation,” we state that “you never have to pay anyone license fees to use GNU software, and you always have the freedom to make your copy from a friend’s computer at no charge.” What exactly do we mean by this, and how do we make sure that it stays true?

The simplest way to make a program free is to put it in the public domain. Then people who get it from sharers can share it with others. But bad citizens can also do what they like to do: sell binary-only versions under typical don’t-share-with-your-neighbor licenses. They would thus enjoy the benefits of the freeness of the original program while withholding these benefits from the users. It could easily come about that most users get the program this way, and our goal of making the program free for *all* users would have been undermined.

To prevent this from happening, we don’t normally place GNU programs in the public domain. Instead, we protect them by what we call *copyleft*. A copyleft is a legal instrument that makes everybody free to copy a program as long as the person getting the copy gets with it the freedom to distribute further copies, and the freedom to modify their copy (which means that they must get access to the source code). Typical software companies use copyrights to take away these freedoms; now we software sharers use copylefts to preserve these freedoms.

The copyleft used by the GNU project is made from a combination of a copyright notice and the *GNU General Public License*. The copyright notice is the usual kind. The General Public License is a copying license which basically says that you have the freedoms we want you to have and that you can’t take these freedoms away from anyone else. (The actual document consists of several pages of rather complicated legalbol that our lawyer said we needed.) The complete license is included in all GNU source code distributions and many manuals, and we will send you a printed copy on request.

Recently the Foundation made a dramatic change in the General Public License. The terms for copying remain unchanged, but the structure of the license is different, and it is now easier to copyleft programs. The License is now essentially a subroutine, and programs need only state that the General Public License applies to them. Specifics on using the License now accompany it, so refer there for details.

*“As we enjoy great advantages from the inventions of others, we should be glad of an opportunity to serve others by any invention of ours”*

—Benjamin Franklin

## GNUs Flashes

- **Anonymous Donation**

We have received a surprise donation of \$100,000 from an English person who wishes to remain anonymous. This, plus the \$100,000 we have just received from Hewlett-Packard, and the \$25,000 we have received from the Open Software Foundation, has enabled us to hire several new employees.

- **New Version of RCS Distributed with Compiler**

We are now distributing the latest version (V4 from Purdue) of RCS, the Revision Control System.

- **BASH Now Available**

Brian Fox has just released the 'Bourne Again SHell' for Beta testing.

- **GNU Emacs Lisp Manual**

The GNU Emacs Lisp Reference manual should be published this summer.

- **New qsort, gdbm**

We now have an improved qsort which is faster than Berkeley's and is also reentrant. It will be released in the future with the GNU C Library.

Our gdbm library is about to be tested. A gdbm data base consists of one file with no large holes. gdbm supports fancy automatic crash recovery and interlocking. It handles keys and data of unlimited size.

- **File Manipulation Utilities**

A collection of utilities for file manipulation, including ls, mv, cp, cat, rm, du, head, tail and cmp will be released soon.

- **GPC: GNU Pascal Compiler**

Some volunteers from Helsinki University of Technology are now working on a Pascal front-end for GCC. Currently they support a subset of the language.

- **Some Parts of BSD are Becoming Free**

Berkeley has now released those parts of their Unix system not containing AT&T code, including the TCP/IP code. The files freed in the 4.3 BSD-Tahoe distribution are now on our beta test tape.

Also, the next release of Berkeley Unix may contain Make, AWK, and sh from the GNU Project. Berkeley will also be distributing GCC in place of their PCC-based compiler.

- **NeXT, Inc. Using GCC as Production Compiler**

NeXT uses GCC to compile their entire system, including their kernel.

- **Distribution of 80386 Floppies Happening Soon**

This will probably take two forms, via 1.2 megabyte diskettes and through an electronic BBS. It's possible that the newly reactivated Programming Language SIG of the Boston Computer Society will be one of the major routes of distribution.



## Programming Freedom Activism

by Richard Stallman

*To benefit by the discoveries of his fellowmen is thus not only a natural right, it is also the natural duty which every man owes to himself and to society; and the mutual, universal progress thence resulting is the fulfillment of the earthy destiny of the human race.*

Robinson, "Treatise on the Law of Patents."

## Interface Copyright Battle Gaining Speed

The battle against user interface copyrighting is gathering momentum. GNU founder Richard Stallman joined MIT professors Gerald J. Sussman and Marvin Minsky in placing an ad in *The Tech*, MIT's student newspaper, warning of the harm that such monopolies could do to computer users and the computer industry. Here is the text of the ad:

---

### Computer Scientists, Watch Out!

Apple and Lotus are trying to create a new form of legal monopoly: a copyright on a class of user interfaces. These monopolies would cause serious problems for users and developers of computer software and systems.

Until two years ago, the law seemed clear: no one could restrict others from using a user interface; programmers were free to implement any interface they chose. Imitating interfaces, sometimes with changes, was standard practice in the computer field. The interfaces we know evolved gradually in this way; for example, the Macintosh user interface was developed over fifteen years at Stanford, SRI, Xerox and other places. Hundreds of students and researchers contributed to this effort, and no one has a right to own it all now.

Most computer companies, and nearly all computer users, are happy with this state of affairs. Lotus and Apple say it does not offer "enough incentive" to develop their products, but they must have considered it "enough" when they made their decision to do so. It seems they are not satisfied with the opportunity to continue to compete in the marketplace—not even with a head start.

If Lotus and Apple are permitted to make law through the courts, the precedent will hobble the software industry:

- Gratuitous incompatibilities will burden users. Imagine if each car manufacturer had to arrange the pedals in a different order.
- Software will become and remain more expensive. Users will be "locked in" to proprietary interfaces, for which there is no real competition.
- Large companies have an unfair advantage wherever lawsuits become commonplace. Since they can easily afford to sue, they can intimidate small companies with threats even when they don't really have a case.
- User interface improvements will come slower, since incremental evolution through creative imitation will no longer be permitted.

- Even Apple and Lotus will find it harder to make improvements if they can no longer adapt the good ideas that others introduce, for fear of weakening their own legal positions. Some users suggest that this stagnation may already have started.

Express your opinion! Reconsider your plans! You can make a difference.

This is a paid political advertisement, sponsored by  
Marvin Minsky, Richard Stallman, Gerald J. Sussman

## Keep Their Lawyers Off Our Computers!

---

Reporters from the Boston Globe and Computer Reseller News saw the ad and then published articles in their papers. Followups to the Globe article then appeared in Computerworld and InfoWorld. The InfoWorld article was seen by a public relations agent who is donating time to help fight the user interface copyright. Additionally, several lawyers have asked for our help to find useful expert witnesses.

On Wednesday, May 24th the League for Programming Freedom picketed Lotus Development Corporation in Cambridge. Despite a heavy threat of rain, some 200 people showed up for the demonstration carrying signs and handing out flyers. Attendees included Bryan S. Kocher, President of the ACM, and Patrick Winston, head of MIT's AI Lab. The protest was led by FSF founder Richard Stallman.

If you would like to help fight for the freedom to write programs, you can do so by joining and working for the League for Programming Freedom. Since the League is still being organized, it does not have an address of its own. For the moment, you may phone or write to the League in care of the Free Software Foundation. Note, however, that the League will be an entirely separate organization from the Foundation with different agendas and personnel. The League will be dedicated solely to the fight against restrictions on development of monopolistic software.

## Dangerous Legislation Proposed

Legislation has been proposed by Senator Orrin Hatch to prohibit the lending and rental of software by *anyone*, including private individuals. Similar legislation has been considered for musical recordings. Such legislation would be an aggressive attack on the traditional freedom to borrow and lend. Libraries could eventually be forced into oblivion if they begin limiting their contents to media which become obsolete.

Hatch and other supporters of this bill reason that software publishers are losing much money because people only borrow the software to make copies. Libraries that lend software say this is untrue, but even if it were, this would be insufficient reason for such an attack on our liberties. There is essentially no resistance on Capitol Hill to this legislation; the software publishers have been very outspoken in their position, but there has been little or no effort made to explain the interests of users.

The only resistance to this proposal has been from the American Library Association, who have obtained language in the bill which would exempt libraries from this legislation for a three year test period. It seems likely, however, that software publishers will only continue complaining about "lost" profits and at the end of the test period even libraries will be prohibited from lending software.

Regaining these rights will be much more difficult than making an effort to preserve them now. Please help alarm people about this problem. Write to Senator Hatch and Representative Dennis DeConcini as well as your own legislators and ask them to vote against bill S.198, the Computer Software Rental Amendments Act. A sufficient address is:

Senator So and So  
United States Senate  
Washington, DC 20510

Representative Such and Such  
U.S. House of Representatives  
Washington, DC 20515

You can also phone senators and representatives at (202) 225-3121, or look in your phone directory for their local offices.

## Common Knowledge's Universal Index

There is also an international group called Common Knowledge working to compile public domain, copyright free and machine-readable information. The group, consisting of journalists, scientists, librarians and others, is amassing a database of non-copyrighted information which they call the "Universal Index". They are doing this to provide an alternative to the information merchants, who are increasingly successful at reducing the amount of material available to traditional libraries. Their address is:

Common Knowledge Jefferson, MD 21755, USA Phone: (301) 695-3100

## GNU Wish List

Wishes for this issue are for:

- Someone skilled in compiler maintenance who could take over GCC maintenance for RMS. This would probably be a full-time job.
- Volunteers to help write utilities and documentation. Among others, we need manuals for X-Windows and the C Library.
- Professors who might be interested in sponsoring or hosting research assistants to do GNU development, with full or partial FSF support.
- Fonts. We are looking for Ghostscript format fonts.
- Speech generation and character recognition software (if the devices aren't too weird), hopefully with the device drivers. This would help at least one partially disabled programmer we know to be productive.
- Grammar checking software for English or other natural languages.
- Copies of newspaper and journal articles mentioning the GNU Project. Send these to the addresses on the front cover, or send a citation to [gnu@prep.ai.mit.edu](mailto:gnu@prep.ai.mit.edu).



- Money, as always. Please remember, donations are tax-deductible. With the latest donations, we have been able to expand our staff.

One way to give us a small amount of money is to order a distribution tape or two. This may not count as a donation for tax purposes, but it can qualify as a business expense.

## Porting BSD Unix Through the GCC

by John Gilmore

I have ported the University of California at Berkeley's latest Unix sources through the GNU C Compiler. In the process, I made Berkeley Unix more compatible with the draft ANSI C standard, made many programs less machine-dependent and less compiler-dependent, and tested GCC.

Berkeley Unix has set the standard for high powered Unix systems for many years, and continues to offer an improved alternative to AT&T Unix releases. However, Berkeley's C compiler is based on an old version of PCC, the Portable C Compiler from AT&T. By merging GCC into the Berkeley release, we provided ANSI C compatibility, better optimization, and improved compiler maintenance. The GNU project gained an important test case for GCC, and a strong collaborator in the free software movement.

The project was conceived by John Gilmore, and aided by Keith Bostic and Mike Karels of Berkeley, and Richard Stallman of FSF. I did most of the actual porting, while Keith and Mike provided machine resources, collaborated on major decisions, and arbitrated the style and content of the changes to Unix. Richard provided quick turnaround on compiler bug fixes and problem solving.

We are producing a Unix source tree which can be compiled by both the old and the new compilers. Rather than introducing new `#ifdef`'s, we are rewriting the code so that it does not depend on the features of either compiler. Whenever we have to make a change, we are moving in the direction of ANSI C, POSIX compatibility, and machine independence.

We have used GCC releases 1.15 through 1.35. I did four complete "passes" over the Unix source tree; each involved running "make clean; make" on the entire source tree, and examining 500K to 800K of resulting output. I'd fix as many errors as I could, testing small parts of the source tree in the process, then merge my changes back into the master sources and rebuild the whole thing again.

The errors fell into two general categories: language changes in ANSI C, and non-portable code. In some cases it was hard to tell the difference.

The major ANSI C problem was the generation of character constants in the preprocessor. Excessive use of this now-obsolete feature in system header files caused us to change about 10 include files and about 45 source modules. Another preprocessor problem was that ANSI C uses a different syntax for token concatenation; we rewrote pieces of five modules to avoid having to concatenate tokens. ANSI C clarified the rules for the scope of names declared `extern`. We moved extern declarations around, or added

global function declarations, in more than 38 files to handle this. Nine programs used new ANSI keywords, such as `signed` or `const`, as identifiers; we picked new identifiers. Eleven modules used typedefs as formal parameters names, or used `unsigned` with a typedef.

The worst non-portable construct we found in the Unix sources was the use of pointers with member names that aren't right for the pointer type. Fixing this problem caused a lot of work, because we had to figure out what each untyped or mistyped pointer was really being used for, then fix its type, and the references to it. We changed 5 modules due to this, and abandoned one program, `efl`, which would have required too much work to fix.

Another problem was caused by using CPP as a macro processor for assembler source. We circumvented this problem by making the assembler source acceptable to both old-CPP and ANSI CPP.

A major problem was `asm` constructs in C source. Some programs were written in C with intermixed assembler code, producing a mess when compiled with anything but the original compiler. Other routines, such as `compress`, drop in an `asm` here or there, as an optimization. Still more modules, including the kernel, run a `sed` script over the assembler code generated by the C compiler, before assembling and linking it. We eliminated as many uses of `asm` as we could, and turned others into assembler language subroutines in `.s` files. Both the Pascal and Lisp interpreters used heavy hacking with `sed` scripts; each of these took several days to fix.

We fixed three programs that used multi-character constants; two were clearly errors. Fifteen programs tried to declare functions or variables, while omitting both the type and storage class; we added `int` to the declarations. In two modules this diagnosed errors caused by use of `';` where `'.'` was intended. Changes to the rules for parsing declarations made us fix five modules, and declaration bugs in six more were caught by GCC's improved error checking. Fifteen programs had miscellaneous pointer usage bugs fixed. GCC caught bugs in five modules caused by misunderstood sign extension. Five or ten other miscellaneous bugs were caught and fixed.

We are pleased with the results so far. Most of the Unix code compiled without problems, and the parts which we have executed are free from code generation bugs. The worst of the ANSI C changes only required roughly fifty modules to be changed, and there were only two problems of this magnitude. A total of twenty bugs in GCC have been found so far, and most of them are now fixed. We expected several times this many bugs; the compiler is in better shape than any of us expected.

Many minor problems and nit incompatibilities with ANSI C have been removed from the Unix sources. Far fewer user programs should require attention when doing a BSD Unix port now. However, we did not attempt to make Berkeley Unix fully ANSI C compliant. In particular, we kept preprocessor comments (`#endif F00`) as well as machine-specific `#define`'s (`#ifdef vax`). GCC supports these features even though ANSI C does not.

Unfinished work remains. The BSD kernel has not yet been ported to GCC, though it has been syntax-checked. Optimization of the kernel will

cause problems until `volatile` declarations are used in all the right places. Pieces of the Portable C Compiler are still used inside `lint`, `f77`, and `pc`. Various sources still need their `setjmp` calls fixed so that only volatile variables depend on keeping their values after a `longjmp`.

Our changes will be available to recipients of Berkeley's next software distribution, whenever that is. We will also make diffs available to others involved in porting Unix to ANSI C.

Future projects include building a complete set of ANSI C and POSIX compatible include files and libraries (including function prototypes), and converting the existing sources to use them. An eventual goal is to produce a fully standard-conforming Unix system—not only in the interface provided to users, but with sources which will compile and run on any standard-conforming compiler and libraries.

The success of this collaboration between GNU and Berkeley has encouraged further cooperation. The GNU project is working to provide reimplementations of System V features that Berkeley Unix lacks, such as improved shells and make commands. In return, Berkeley has released much of its software to the public, eliminating the AT&T license requirement for programs that AT&T did not supply. A large set of "freed" BSD software is available by `uucp` or `ftp` from `uunet.uu.net` in the subdirectory 'bsd-sources', as well as on the GNU Compiler tape and the UUNET tapes.

## GNU Project Status Report

- **GNU Emacs**

GNU Emacs version 18 is in wide distribution. Version 18.55 will soon begin distribution.

Berkeley is distributing GNU Emacs with the 4.3 distribution, DEC is distributing it with Unix systems on Vaxes, and NeXT is shipping it.

Emacs 18 development has now ceased (although bugs are still fixed, of course) and version 19 is being worked on. New features so far include: multiple X windows (a `screen` object in Emacs Lisp); scroll bars; per-buffer redefinition of mouse commands; support for European character sets; source-level debugging for Emacs Lisp; and floating point numbers.

We are also considering these new features: associating property lists with portions of the text in a buffer, and using them to control visibility of the text; specifying different visibility conditions for multiple windows showing one buffer; incremental syntax analysis for various programming languages; a visually enhanced Info mode; an object-oriented graphics-drawing extension; hooks to be run if point moves outside a certain range.

It will take a while to do this (plus any other ideas we get), so please don't ask when Emacs 19 will be available for beta test. We will announce it.

- **Shells**

Brian Fox has now completed GNU's version of `sh`, called BASH, the 'Bourne Again SHell'. In addition to Korn shell features, it now has job



control and both Emacs-style and `csh`-style command history manipulation. Look for it soon on our distribution.

There is a good chance that the C Shell from BSD will be declared free software by Berkeley, so we won't need to write that.

- **Kernel**

We hope to use the MACH message-passing kernel being developed at CMU. The current distributed version of MACH is not free because it contains code from BSD of AT&T origin. However, the MACH developers have been working to separate this code from the kernel and they now say that they have a first version of this running in alpha test. Once this is stable, the MACH kernel is supposed to become free.

Should MACH not become available, then we will start the kernel with either MIT's TRIX kernel or Berkeley's Sprite system.

Another thing we are considering is using the high-performance, distributed Sprite file system with MACH.

- **GNU Debugger**

The GNU source-level C debugger, GDB, is now being distributed along with Emacs version 18 as GDB version 2.8. GDB version 3.2 is distributed on the beta-test (compiler) tape, and as soon as it is stable it will replace version 2.8 on the Emacs tape.

GDB 3.\* reads symbol tables incrementally; this results in much faster startup and much less memory use. GDB also provides command line editing with history substitution and completion of command, filename, and symbols. Recently added FORTRAN support has not been tested. New commands include: `until`, which executes until a certain line; and `bt -n`, to show only the outermost *n* stack frames.

The current GDB can code-grind (pretty-print) structures and can conditionally avoid printing unions. C++ support has been improved.

There is a version of GDB which can run stand-alone so that we can use it to debug the kernel, and we also have a serial line interface for running GDB remotely. A French volunteer is now working on a UDP based over-the-ethernet debugging interface. Work is also being done on debugging multiple process, parallel programs.

Future versions of GDB may support watchpoints. We also hope soon to merge the diffs for the Altos and Sun 386i machines.

- **C Compiler**

The GNU C Compiler GCC is now fairly reliable. It supports the May 1988 draft of ANSI C. People are still reporting bugs, but they also say they think there are fewer bugs than in commercial compilers. Read John Gilmore's article to see how some of these bugs were uncovered. Next builds their entire system, including their port of the MACH kernel and NFS, with GCC. Someone has also told us that GCC successfully compiled a System V.3 kernel.

The compiler performs automatic register allocation, common subexpression elimination, invariant code motion from loops, induction vari-

able optimizations, constant propagation and copy propagation, delaying popping of function call arguments, tail recursion elimination, plus many local optimizations that are automatically deduced from the machine description. We are also currently implementing delayed-branch fill and pipeline scheduling (experimentally).

GCC has recently been ported to the Convex, Tahoe and the MIPS processors. A Pyramid version is expected soon, and ports are underway to the IBM 370, IBM PC/RT, 3B2, HP Spectrum, Motorola 88000, some sort of Gould machine, and (perhaps) the AMD 29000.

GCC makes shorter and faster 68020 code than the new Sun compiler with -O. The new Sun compiler can't beat GCC despite taking 3 times as long. As of version 1.31, GCC also wins on the Sun 4. GCC makes shorter Vax code than the Tartan C compiler with -O4, but we haven't been able to compare the running speed of that code. A National 32000 port done by Michael Tiemann on a Sequent 32000 system is said to be more reliable than Sequent's compiler and yields a 40% speedup for several programs including a Prolog interpreter.

We are also working on merging the C Compiler with the C++ compiler so that there would be only one distribution for both languages.

In the future, if we have time, we would like to implement more language extensions (we will probably add nested scoping), plus facilities for precompilation of header files to save time when they are large and the source files are small. We might also do other language front ends, but there seem to be enough remote GNUers willing to do this job.

Enough internal documentation is included for people interested in re-targeting the compiler to other CPUs to do so.

- **GNU C for VMS**

GCC can run under VMS. However, the ordinary VMS C compiler has bugs and cannot compile GNU C, so you need an executable of GNU C to get started. This comes on the VMS tape along with the source.

Other GNU programs for 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.

- **COFF Support**

It is now possible to run the entire suite of GNU software tools on System V, replacing COFF entirely. First you install the GNU compiler, assembler, linker and other utilities. Then you use the program *robotussin*—COFF medicine for your computer—to convert the system libraries from COFF format to GNU (i.e. BSD) format.

When you compile programs, you will get ordinary GNU/BSD object files. Linking these with the GNU linker will produce GNU/BSD executables with a COFF header that the kernel will accept. The other GNU utilities such as *size*, *nm* and *strip* know how to operate on these encapsulated files.

As true COFF support is peripheral to the GNU project, please don't ask us to expend effort on it.

- **Compiler-related Programs**

- **C++**

Michael Tiemann of Stanford University has written a C++ compiler as an extension of GNU C which is distributed with GCC. This was the first compiler to compile C++ directly instead of preprocessing it into C with great benefits for debugging and efficiency.

Recently Michael added multiple inheritance and other new features promised by AT&T at the first USENIX C++ workshop (AT&T has yet to release their version).

GDB version 3.\* includes support for debugging C++ code, which merges in the functionality of the old program GDB+. GDB+ was a source code debugger for C++, but is now being withdrawn.

- **Assembler and Object File Utilities**

GAS is a fairly portable, one pass assembler that is almost twice as fast as Unix as. It now works for Vaxes, 680x0, 32x32 and 80386. A port for Sparc (Sun 4) will be available soon.

The GNU replacements for `ld`, `nm`, `size`, `gprof`, `strip`, `ranlib`, etc., have been released with the GCC beta-test distribution.

The GNU linker `ld` runs significantly faster than the BSD version. Our `ld` is the only one that will give you source-line numbered error messages for multiply-defined symbols and undefined references.

- **C++ Library**

Doug Lea is writing `libg++`, a library with utility classes for C++.

- **GNU Make Extended**

We have been distributing GNU Make for several months. An extended version including more text-processing capabilities and static rules is now available. It also supports parallelism.

- **C Library**

Roland McGrath, who contributed a great deal to GNU Make, has a nearly complete set of ANSI C library functions. He will work full time this summer to complete them. These join the GNU `malloc`, `regex` and `termcap` libraries. A better `malloc` may be written soon, and we will shortly add our `qsort` library routine. `gdbm` is about to enter beta test.

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

- **Preliminary Ghostscript**

We now distribute Ghostscript, the free GNU software that provides nearly all the facilities of a Postscript interpreter. It supports X version 11. Peter Deutsch is still doing some work on it.

Right now, Ghostscript will accept commands in Postscript and execute them by drawing on an X window. Karl Berry and Kathy Hargreaves will be working on adding more fonts. They could use the help of volunteers. Beside additional fonts, Ghostscript needs these enhancements:



to serve as a previewer for multi-page files; to serve other X clients by drawing on their windows; to improve both its performance and visual quality. Other suggestions for enhancements are welcome.

Ghostscript also includes a C-callable graphics library (for client programs that don't want to deal with the Postscript language), and also supports IBM PCs and compatibles with EGA graphics (but please don't ask the FSF staff any questions about this; we don't use PCs and don't have time to learn anything about them).

- **Finger and Send**

We soon will have a daemon-based Finger program. It polls a selection of hosts and is thus able to tell you where each person is logged in.

We are also testing a Send program for sending immediate messages to other users across the net.

- **Oleo**

Jay Fenlason is writing a spreadsheet named Oleo (better for you than the more expensive spreadsheet).

- **GNU Mailer**

Landon Noll and Ronald Karr of Amdahl are writing a mail queuing and delivery system, called `smail`. This project is a supported part of the Amdahl UTS system—as free software!

We may use `smail` for GNU, or `zmailer`, which Rayan Zachariasen is writing, or perhaps both.

- **Window System**

We are using the MIT X window system, which is free software.

- **Other Utilities**

Our free replacement for Yacc is called Bison. We also have `cron`. We now have the world's fastest `grep/egrep` and the world's fastest `diff`. A new fast `sort` has just been finished. A "fast lex" called FLEX recently became available; we are now distributing it.

- **Long Term**

Volunteers are working on a Smalltalk system and an APL system. Also, software for editing and playing music.

- **Possible Target Machines**

The GNU operating system will require a CPU that uses 32-bit addresses and integers and addresses to the 8-bit byte. 1 megabyte of core should be enough, though 2 meg would probably make a noticeable improvement in performance. Running some of the system in 1/2 meg may be possible, but certainly not GNU Emacs, which requires more than 1 meg of addressable memory. Virtual memory will be required.

A hard disk will be essential; at least 20 meg will be needed to hold a minimal system. Plus more space for the user's files, of course. Perhaps 80 meg altogether for a personal GNU system.

Not that it will be impossible to adapt some or all of GNU for other architectures; but we don't currently consider it part of our job.

- **Porting**

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

## GNU Documentation

GNU documentation is distributed as Texinfo source files. Texinfo source yields both a typeset hardcopy and an on-line format, accessed by a menu-driven system.

To make the printed manual, the Texinfo source file is processed through the `TeX` typesetting program. To make the on-line documentation using GNU Emacs, the Texinfo source file is processed with the `M-x texinfo-format-buffer` command; the resulting Info file is installed in the 'info' directory which you reach by typing `C-h i`.

The following manuals, provided with our software, are also available in hardcopy; see the order form on the inside back cover.

The **Emacs Manual** describes how to use GNU Emacs. It also explains advanced features, such as outline mode and regular expression search. The manual tells how to use the special modes for programming in languages such as C and Lisp, how to use the tags utility, and how to compile and correct code. It also describes how to make your own keybindings and make other elementary customizations.

This manual does *not* cover programming in Emacs Lisp. This topic will be handled in an introductory Emacs Lisp programming manual and an Emacs Lisp reference manual. The reference manual should be out this summer. Watch for the announcement. The **Texinfo Manual** describes how to write documents in Texinfo source code. It describes the markup language used to create both an Info file and a printed document from the same source file: how to make tables, lists, chapters, nodes, indices and cross references. It also describes how to use Texinfo Mode in GNU Emacs and how to catch mistakes.

The **Termcap Manual** is often described as "Twice as much as you ever wanted to know about Termcap". It describes the format of the Termcap data base, the definitions of terminal capabilities and how to interrogate a terminal description. This manual is primarily for programmers.

The **Bison Manual** describes how to write a grammar description that Bison can convert into a C program that can parse that grammar. This manual assumes no prior knowledge of parser generators. It describes the concepts and then provides a series of increasingly complex examples before describing what goes on in considerable detail.

The **GAWK Manual** describes how to use the GNU implementation of AWK. It is written for someone who has never used AWK before, and describes all the features of this powerful string manipulating language.

The **Make Manual** describes the GNU Make utility, a program used to rebuild parts of other programs when and as needed. The manual tells

how to write a makefile, which describes how to recompile the parts of your program and how they depend on each other.

The **GDB Manual** explains how to use the GNU DeBugger. It describes how to run your program under control of the debugger, how to examine and alter data within the program, and how to modify the flow of control within the program. It also explains how to use GDB through GNU Emacs, with auto-display of source lines.

## GNU Software Available Now

We now offer three Unix software source distribution tapes, plus VMS tapes for GNU Emacs and GNU C that include sources and VMS executables. The first Unix tape (called the "Release" or "Emacs" tape) contains GNU Emacs as well as various other well-tested programs that we consider reliable. The second Unix tape (called the "Beta test" or "Compiler" tape) contains the GNU C compiler and related utilities, and other new programs that are less thoroughly tested. The third Unix tape (called the "X11" tape) contains the X11 distribution from the MIT X consortium. See the order form, on the inside back cover, for details about media, etc.

## Contents of Release Tape

The software on this tape is considered to be fairly stable, but as always, we welcome your bug reports.

- **GNU Emacs**

In 1975, Richard Stallman developed the first Emacs: the extensible, customizable real-time display editor. GNU Emacs is his second implementation of Emacs. It's the first Emacs available on Unix systems that offers true Lisp, smoothly integrated into the editor, for writing extensions. It also provides a special interface to MIT's free X window system, versions 10 and 11, which makes redisplay very fast. The current version is 18.55.

GNU Emacs has been in widespread use since 1985 and often displaces proprietary implementations of Emacs because of its greater reliability as well as its good features and easier extensibility. Isaac Salzman of *Unix Review* magazine planned to compare the various publicly available Emacs', but only one company wanted their product to be compared with GNU Emacs. The review should appear in the June, 1989 issue.

GNU Emacs (as of version 18.55) runs on many kinds of Unix systems: those made by Alliant, Altos 3068, Amdahl (UTS), Apollo, AT&T (3B machines and 7300 pc), CCI 5/32 and 6/32, Celerity, Convex, Digital (Vax, not PDP-11; BSD or SysV), Dual, Elxsi 6400, Encore (DPC and APC), Gould, HP (9000 series 200, 300 or 800 (Spectrum) but not series 500), HLH Orion 1/05, IBM (RT/PC;4.2 and AIX), Integrated Solutions (Optimum V with 68020 and VMEbus), Intel 80386 (BSD,



SysV, and Xenix), Iris (2500, 2500 Turbo and 4D), LMI (Nu), Mass-comp, Megatest, MIPS, NCR (Tower 32), Nixdorf Targon 31, Plexus, Pyramid, Sequent Balance, SONY News, Stride (system release 2), Sun (all kinds), Tahoe, Tektronix (NS16000 system & 4300), Texas Instruments (Nu), and Whitechapel (MG1).

GNU Emacs is described by the GNU Emacs Manual, which comes with the software in Texinfo form. See "GNU Documentation" above.

- **GDB**

GDB 2.8 (GNU's Debugger) is the source-level C debugger written in 1986. It offers many features not usually found in debuggers on Unix, such as Emacs-style command history and substitution, a history that records all values examined within the debugger for concise later reference, multi-line user-defined commands, and good self-documentation.

GDB 2.8 currently runs on Vaxes under 4.2 and 4.3bsd, on Sun 3 under systems version 2 and 3 and 4, on the SPARC (Sun 4) under systems version 3.2 and 4.0, HP9K320, ISI, Merlin, SONY News, Gould NPL and PN, i386, and on some 32000 systems. GDB 3.\* is currently being distributed on the beta tape, and supports several more systems.

On-line help and a users' manual for GDB comes with the software; the printed version of the manual is also available from the Foundation.

- **Bison**

Bison is an upward-compatible replacement parser generator for Yacc, with additional features. It has been in use for several years. Bison is used for compiling GNU C, so it is included on the GNU beta tape as well. A users' manual for Bison comes with the software; the printed version is also available from the Foundation.

- **X Window System, V10R4**

Version 10 of X Windows is distributed on the GNU Emacs tape; version 11 (which is totally incompatible) is distributed on the X11 tape. GNU Emacs version 18.55 supports both versions 10 and 11.

- **MIT Scheme**

Scheme is a simplified, lexically scoped dialect of Lisp, designed at MIT and other universities to teach students programming and research new parallel programming constructs and compilation techniques. Scheme is written in C and runs on many Unix systems.

It now conforms to the *Revised<sup>3</sup> Report On The Algorithmic Language Scheme* (MIT AI Lab Memo 848a), for which T<sub>E</sub>X source is included in the distribution. Another good source of documentation for Scheme is *Structure and Interpretation of Computer Programs*, by Harold Abelson and Gerald J. Sussman with Julie Sussman, the MIT Press & McGraw-Hill Book Company, 1985.

- **Yale T**

A variant of Scheme developed at Yale University, T is intended for production use in program development. T contains a native-code optimizing compiler that produces code that runs at speeds comparable to

the running speeds of programs written in conventional languages. It runs on BSD Vaxes and a few types of 68020 systems. T is written in itself and cannot be bootstrapped without a binary (included), but it is great if you can use it. Some documentation is included.

- **texi2roff**

**texi2roff** translates GNU Texinfo files into a format that can be printed by the Unix [n]roff programs utilizing the mm, ms or me macro packages. It is included on both tapes so that people who don't have a copy of TeX can print out GNU documentation.

- **GNU Chess and NetHack**

GNU Chess is a chess program written in C by John Stanback and Stuart Cracraft. It includes an extensive opening book and was recently rated by USCF Senior Master IM Larry Kaufman at around USCF 1950 (close to expert level) when run on a Sun 3 workstation. On a Sun 4, it should play at nearly master level.

Hack is a display oriented adventure game similar to Rogue.

## Contents of Beta Test Tape

The programs on this tape are all recent releases and can be considered to be at various stages of user testing. As always, we solicit your comments and bug reports. This tape is also known as the Compiler tape.

- **GNU CC**

The GNU C compiler is a fairly portable optimizing compiler. It generates good code for the 32000, 680x0, 80386, Alliant, SPARC, SPUR, Tahoe, and Vax CPUs. Machines using these CPUs include the Encore NS32000, Genix NS32000, Sequent NS32000, AT&T 3B1, HP-UX 68000/68020, ISI 68000/68020, SONY News, All Sun's, Intel 386, Sequent Intel 386, Alliant FX/8. The MIPS processor is also supported. It supports full ANSI C as of the latest draft standard. Included with the compiler are the GNU assembler GAS, Make, Bison (also on the Emacs release tape), and the perfect hash-table generating utility (Gperf), plus the object file utilities ld, nm, size, strip, ar, ranlib and gprof and the Texinfo source of *The GCC Manual* (for those interested in extending or retargeting GCC).

- **GDB**

Version 3 of GDB runs under BSD 4.2 and 4.3 on Vaxes and Suns (2, 3, and 4), Convex, HP 9000/300's under BSD, HP 9000/320's under HP-UX, Intel 386 workstation (with either GNU or native object file format), ISI Optimum V, Merlin under Utek 2.1, SONY News, Gould NPL and PN machines, Sequent Symmetry (a 386 based machine), and Encores under Umax 4.2.

- **GAWK, FLEX and tar**

GAWK is GNU's version of the Unix AWK utility; it comes with a Texinfo manual. FLEX is a mostly-compatible replacement for the Unix

**lex** scanner generator written by Vern Paxson of the Lawrence Berkeley Laboratory. **FLEX** generates far more efficient scanners than **lex** does. **GNU tar** includes multivolume support, automatic compression and decompression of archives, remote archives, and special features to allow **tar** to be used for incremental and full backups of file systems.

- **Freed Files from the U.C. Berkeley 4.3-tahoe Release**

These files have been declared by Berkeley to be free of AT&T code, and may be freely redistributed. They include complete sources for some utility programs, games, library routines and partial sources for many others.

- **RCS and BASH**

The latest version of the Revision Control System for version control and management of large software projects.

The GNU Shell **BASH** (for Bourne Again SHell) provides compatibility with the Unix **sh** and extensions from both **csk** and **ksh**.

- **diff and grep**

These programs are GNU's versions of the Unix programs of the same name. They are much faster than their Unix counterparts.

- **Ghostscript and gnuplot**

Ghostscript is GNU's graphics language. It is almost fully compatible with the postscript language. It supports X version 11.

**gnuplot** is an interactive program for plotting mathematical expressions and data. Oddly enough, the program was neither done for nor named for the GNU Project—the name is a coincidence. However, we are distributing it anyway. If you can put us in contact with the author of this program, please do!

- **g++, libg++, OOPS, and InterViews**

**G++** is a set of changes for **GCC**, that compiles **C++**, the well-known object-oriented language. Since **G++** depends on **GCC**, it must be used with the correspondingly numbered version of **GCC**.

**libg++** (the GNU **C++** library) is a collection of **C++** classes and support tools for use with **G++**.

**OOPS** (Object-Oriented Program Support) class library is a portable collection of classes similar to those in **Smalltalk-80** that has been developed by Keith Gorlen of NIH, using the **C++** programming language.

**InterViews** is an object-oriented, **C++** library to support the design and implementation of user interfaces.

- **GnuGo**

**GnuGo** allows the user to play the machine in a game of Go (Wei-Chi). It is an updated version of the program called **Hugo**.

## Contents of X11 Tape.

The X11 tape contains Version 11, Release 3 of the MIT X window system. X11 is more powerful than, but incompatible with, the no-longer-



supported version 10. MIT no longer labels Version 11 'beta test' but is still releasing frequent patches and updates.

## VMS Emacs and Compiler Tapes

We offer a VMS tape of the GNU Emacs editor, and a separate VMS tape containing the beta-test GNU C compiler. The VMS compiler tape also contains Bison (needed to compile GCC), GAS (needed to assemble GCC's output) and some library and include files. Both VMS tapes include executables that you can bootstrap from.

## How to Get GNU Software

All the software and publications from the Free Software Foundation are distributed with permission to copy and redistribute. The easiest way to get GNU software is to copy it from someone else who has it.

If you have access to the Internet, you can get the latest software from the host 'prep.ai.mit.edu'. For more information, read the file '/u/emacs/GETTING.GNU.SOFTWARE' on that host. Please note that the internet address of 'prep' has changed to 18.71.0.38.

If you cannot get the software from a friend or over the net, or if you would like to contribute some funds to our efforts and receive the latest versions, the Free Software Foundation distributes tapes for a copying and distribution fee. See the order form on the inside back cover.

There are also third party groups that distribute our software: people and organizations that do not work with us, but have our software in other forms. For your convenience, some of them are listed below. Please note that the Free Software Foundation is *not* affiliated with them in any way, and is not responsible for either the currency of their versions or the swiftness of their responses.

These Internet sites providing for anonymous FTP:

scam.berkeley.edu, spam.istc.sri.com, bu-it.bu.edu,  
wsmr-simtel20.army.mil (under 'PD:<UNIX.GNU>'),  
cc.utah.edu (VMS GNU Emacs), and uunet.uu.net.

Those on the SPAN network can ask rdss::corbet.

Information on how to obtain some GNU programs using UUCP is available via electronic mail from:

hao!scicom!getzal!upba!ugn!nepa!denny, arnold@skeeeve.UUCP,  
uunet!hutch!barber, hqda-ai!merlin, acornrc!bob,  
ames!killer!wisner, mit-eddie!bloom-beacon!ht!spt!gz,  
sun!nosun!illian!darylm, or info@uunet.uu.net.

Ohio State also makes GNU programs available via UUCP. They post their instructions monthly to newsgroup comp.sources.d on USENET. Current details from Karl Kleinpaste karl@tut.cis.ohio-state.edu or ...!osu-cis!karl; or Bob Sutterfield (substitute bob for karl in the above addresses).

Information on getting floppy disks of GNU Emacs for the AT&T Unix PC (aka 3B1 or PC7300) is available from: [brant@manta.pha.pa.us](mailto:brant@manta.pha.pa.us).

## Thank GNUs

Thanks to our **Anonymous Contributor**, and thanks to Hewlett-Packard for their donations of a \$100,000 each. Also, thanks to the Open Software Foundation for their donation of \$25,000.

Many thanks to the following people for copying Sun cartridges: **Barry Kleinman** and **Andre Mesarovic** of Index Technology; **George Brown**; **Devon McCullough** and **Nick Papadakis**; **Mark Nahabedian** and **Shaun Keller** of Phoenix Technology; and finally thanks in advance to **Mark Hannon** of ICAD.

Thanks to all those mentioned in GNU Flashes and the GNU Project Status Report.

Thanks to the MIT Laboratory for Computer Science, and its director, **Professor Michael Dertouzos**. LCS has provided FSF with the loan of a Microvax for program development.

Thanks to the MIT Artificial Intelligence Laboratory for invaluable assistance of many kinds.

Thanks to **Arnold Robbins** and **Dave Trueman** for their work on GAWK and the GAWK manual.

Thanks to **John Klensin** of the INFOODS Project at MIT for use of the project's machine for making our VMS master tapes.

Thanks go out to all those who have lent us machines, including **Brewster Kahle** of Thinking Machines, Inc. for the Sun 4/110, **K. Richard Magill** for the AT&T Unix PC, CMU's Mach Project for the Sun 3/60, Intel Corp. for their 386 workstation, and SONY Corp. and Software Research Associates, Inc., both of Tokyo, for the SONY News workstations.

Thanks to all those who have contributed ports and extensions, as well as those who have contributed other source code, documentation, and good bug reports.

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

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

675 Massachusetts Ave.  
Cambridge, MA 02139

## Free Software Foundation

### ORDER FORM

EFFECTIVE from JUNE 1989 to JANUARY 1990

All software and publications are distributed with permission to copy and to redistribute. TeX source for each manual is on the appropriate tape. All software from the Free Software Foundation is provided on an "as is" basis, with no warranty of any kind. Prices and contents of tapes are subject to change without notice.

Qty	Price	Amt	◇
___ @ \$150 = \$			GNU Emacs source code and other software, for Unix systems, on 1600 bpi reel-to-reel 9-track tape in tar format. The tape includes Scheme, T, Bison, Hack, GNU Chess, GDB, and the X Window System Version 10R4.
___ @ \$150 = \$			GNU Beta Test software, for Unix systems, on tape as in previous item. The tape includes the C compiler, G++, Bash, Bison, Flex, Diff, Grep, Tar, Make, Gawk, Ghostscript, the GNU Assembler, Gnu Plot, Compress, object file utilities, and the freed files from the 4.3 BSD-Tahoe distribution.
___ @ \$150 = \$			MIT X Window System V11R3, for Unix systems, on tape as in previous item.
___ @ \$175 = \$			GNU Emacs and other software, for Suns, as in the first item, on DC300XLP 1/4" cartridge tape, in QIC-24 encoding in tar format.
___ @ \$175 = \$			GNU Beta Test software, for Suns, as in the second item. on tape as in previous item.
___ @ \$175 = \$			MIT X Window System V11R3 for Suns, as in the third item, on tape as in previous item.
___ @ \$150 = \$			GNU Emacs source code and binaries, for VMS on 1600 bpi reel-to-reel 9-track tape in VMS backup format.
___ @ \$150 = \$			GNU C Compiler source code and binaries, for VMS on tape as in previous item. Includes Bison and GAS.
___ @ \$ 15 = \$			GNU Emacs manual, ≈ 300 pages. The manual is phototypeset, offset printed, and spiral bound; it includes a reference card.
___ @ \$ 60 = \$			Box of six GNU Emacs manuals, each with reference card.
___ @ \$ 1 = \$			GNU Emacs reference card, without the manual.
___ @ \$ 5 = \$			Packet of ten GNU Emacs reference cards.
___ @ \$ 10 = \$			GDB manual, ≈ 70 pages, side stapled.
___ @ \$ 10 = \$			Texinfo manual, ≈ 100 pages, side stapled.
___ @ \$ 10 = \$			Termcap manual, ≈ 60 pages, side stapled.
___ @ \$ 10 = \$			Bison manual, ≈ 80 pages, side stapled.
___ @ \$ 10 = \$			Gawk manual, ≈ 100 pages, side stapled.
___ @ \$ 10 = \$			Make manual, ≈ 100 pages, side stapled.
Subtotal \$			
+ \$			If ordering from Massachusetts: add 5% sales tax.
+ \$			If outside of North America and Hawaii: for shipping costs, please add \$60 for each box of Emacs manuals; for tapes or unboxed manuals, add \$15, and then add \$15 more for each tape or manual in the order.
+ \$			Optional tax-deductible donation.
TOTAL \$			

◇ Orders are filled only upon receipt of check or money order. Please make checks payable to the Free Software Foundation, and mail orders to the above address.

◇ Name: \_\_\_\_\_

Organization: \_\_\_\_\_

Address: \_\_\_\_\_

City/State/Zip: \_\_\_\_\_



