

# PHRAINE

Computer Technology Enlightenment *Without* the Noise  
<http://pearlyfreepress.madoshi.com/phraine/>



Richard Stallman (rms)

## Editor's Corner

It hardly seems possible that another issue of PHARINE is finished and ready for release. It seems as if it was yesterday I was searching for the best way to write and distribute the first issue of PHRAINE, but it is not yesterday and another issue of PHRAINE is ready.

In this issue of PHRAINE you will find an interview with Jim Hall (FreeDOS project creator). JH took time out of busy schedule to speak with us and we cannot thank him enough.

I hope that you, the reader, recognizes this issue of PHRAINE as an improvement over the previous and that each subsequent one is better than the last.

Enjoy.

Editor-in-Chief  
Scott C. MacCallum  
[scm@madoshi.com](mailto:scm@madoshi.com)

Vol. 1 Num. 2(4)  
Summer/Fall 2004  
\$2.50

---

## Policy

### Free Flow of Information

PHRAINE believes in the free (as in freedom) flow of information, even when this information has been deemed controversial.

### Reader Advisory

Due to recent and questionable laws/lawsuits in the United States and abroad over the right to freely experiment with technology, we advise our readers to research state and federal laws before attempting any projects related to the information written in this publication. Although no information in this publication was written with the intent that its readers would engage in any unlawful conduct, incorrect use of the information found in PHRAINE might result in this.

### Disclaimer

While every precaution has been taken in the preparation of this electronic quarterly, the editor assumes no responsibility for errors, omissions, misuse, or damages resulting from the use of the information contained herein.

Opinions expressed in articles and letter submissions are *not* necessarily those held by The Pearly Free Press (<http://pearlyfreepress.madoshi.com/>), PHRAINE (<http://pearlyfreepress.madoshi.com/phraine/>), or Madoshi (<http://www.madoshi.com/>).

## Submission

### Article Submission

Do you have a computer, telephone, programming, or electrical engineering article, commentary, or book review that you are the owner of and that you think deserves publication in PHRAINE? Please send it via e-mail in ASCII character format to [scm@madoshi.com](mailto:scm@madoshi.com). Please be sure to include "article" in the subject line of your e-mail.

If we print your article, you get the satisfaction of "mission accomplished".

### Letter Submission

Do you have a comment or question for PHRAINE? Did you find an error that you feel should be corrected? Please send it via e-mail in ASCII character format to [scm@madoshi.com](mailto:scm@madoshi.com). Please be sure to include "letter" in the subject line of your e-mail. If we print your letter, you get the satisfaction of "immortalization".

### Pen Names

PHRAINE believes in openness. We encourage those who submit articles or letters for consideration in future publications to use their own name as opposed to a computer nick-name, but this is not required. If you would like your name withheld, please make that known upon submission.

If you do not see either your article or letter appear in two sub-

sequent releases of PHRAINE, it is a good bet that it will not be featured at all. Submitters of an article, letter, and/or program source code(s) maintain their copyright; but submissions cannot be returned.

By submitting the aforementioned material to PHRAINE, you proclaim that you are indeed the copyright owner of the said material and that any program source code(s) submitted are compliant with the GNU General Public License as published by the Free Software Foundation; either version 2 of the license or (at your option) any later version (see the Copyright section of this issue for more details).

### Copyright

PHRAINE Copyright © Scott C. MacCallum. All Rights Reserved.

PHRAINE may not be reproduced in whole or part without written permission.

PHRAINE is published quarterly by the Pearly Free Press (<http://pearlyfreepress.madoshi.com/>).

### Programs

Programs' source codes found in this publication are free software; you can redistribute them and/or modify them under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the license or (at your option) any later version.

These programs are distributed in the hope that they will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

An ASCII copy of the GNU General Public License is located at (<http://www.gnu.org/licenses/gpl.txt>); if not write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA, 02111-1307, USA.

### Trade/Service Marks

All terms used in this publication that are known to be trademarks or service marks have been appropriately capitalized. The Pearly Free Press cannot attest to the accuracy of this information. Use of a term in this publication should not be regarded as affecting the validity of any trademark or service mark.

### Cited Material

#### Jargon

The definition of "hacker" adapted by Scott C. MacCallum for PHRAINE, Taken from "The Hacker Jargon File" version 4.4.7 (<http://www.catb.org/~esr/jargon/html/H/hacker.html>).

The definition of "TM" adapted by Scott C. MacCallum for PHRAINE. Taken from "The Hacker Jargon File" version 4.4.7 (<http://www.catb.org/~esr/jargon/html/O/TM.html>).

#### Richard Stallman's Action Items

"Urgent action items and Long-term action items" adapted by Scott C. MacCallum for PHRAINE. Taken from "Richard Stallman's Personal Page" (<http://www.stallman.org/>). Please send comments on these web-pages to rms at stallman period org. Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004 Richard Stallman. Verbatim copying and redistribution of this entire page is permitted provided this notice is preserved.

## Advertising

Only subscribers can submit ads for the "bazaar" section of PHRAINE. Each subscriber is allowed one ad per issue. Ads for reprint must be resubmitted by the deadline located in the "bazaar" section of each issue.

PHRAINE reserves the right not to print any ad that it deems unsuitable for the magazine and is not responsible for the possible lack of goodwill of the advertiser. Contact these individuals at your own peril.

## Jargon

**Hacker:** n.

1. A person who enjoys exploring the details of programmable systems and how to stretch their capabilities, as opposed to most users, who prefer to learn only the minimum necessary.
2. One who programs enthusiastically (even obsessively) or who enjoys programming rather than just theorizing about programming.
3. A person who is good at programming quickly.
4. An expert at a particular program, or one who frequently does work using it or on it; as in a "Unix Hacker".
5. An expert or enthusiast of any kind. One might be an electronics hacker, for example.
6. One who enjoys the intellectual challenge of creatively overcoming or circumventing limitations.

### Note

It is better to be described as a hacker by others than to describe oneself that way. Hackers consider themselves something of an elite grouping based on ability to which new members with similar traits are gladly welcomed.

### Origin

The term "Hacker" seems to have been first adopted as a badge in the 1960's by the hacker culture surrounding the MIT's TMRC (Tech Model Railroad Club) and AI (Artificial Intelligence) Lab.

**(TM):** //[[Usenet]

1. ASCII rendition of the <sup>TM</sup> appended to phrases that the author feels should be recorded for posterity.
2. Used ironically as a form of protest against the recent spate of software and algorithm patents and look and feel lawsuits.

## Hacker Biography: Richard Stallman

Our featured hacker for this issue of PHRAINE is Richard Stallman (rms). I think you would be hard pressed to find someone that has contributed to the hacker culture in more positive ways than he.

The following are two biographies, one from 1983 that was first published in the Hacker Jargon File, and the other a more recent one from his home-page.

### 1983 Biography

I was built at a laboratory in Manhattan around 1953, and moved to the MIT Artificial Intelligence Lab in 1971. My hobbies include affection, international folk dance, flying, cooking, physics, recorder, puns, science fiction fandom, and programming; I magically get paid for doing the last one. About a year ago I split up with the PDP-10 computer to which I was married for ten years. We still love each other, but the world is taking us in different directions. For the moment I still live in Cambridge, Massachusetts, among our old memories. "Richard Stallman" is just my mundane name; you can call me "rms".

## Recent Biography

Richard Stallman is the founder of the GNU Project, launched in 1984 to develop the free software operating system GNU. The name ``GNU" is a recursive acronym for ``GNU's Not Unix".

GNU is free software: everyone is free to copy it and redistribute it, as well as to make changes either large or small. Non-free software keeps users divided and helpless, forbidden to share it and unable to change it. A free operating system is essential for people to be able to use computers in freedom.

Today, Linux-based variants of the GNU system, based on the kernel Linux developed by Linus Torvalds, are in widespread use. There are estimated to be some 20 million users of GNU/Linux systems today.

Richard Stallman is the principal author of the GNU Compiler Collection, a portable optimizing compiler which was designed to support diverse architectures and multiple languages. The compiler now supports over 30 different architectures and 7 programming languages.

Stallman also wrote the GNU symbolic debugger (gdb), GNU Emacs, and various other programs for the GNU operating system.

Stallman graduated from Harvard in 1974 with a BA in physics. During his college years, he also worked as a staff hacker at the MIT Artificial Intelligence Lab, learning operating system development by doing it.

He wrote the first extensible Emacs text editor there in 1975. He also developed the AI technique of dependency-directed backtracking, also known as truth maintenance. In January, 1984 he resigned from MIT to start the GNU project.

Stallman received the Grace Hopper award for 1991 from the Association for Computing Machinery, for his development of the first Emacs editor. In 1990, he was awarded a MacArthur Foundation Fellowship, and in 1996 an honorary doctorate from the Royal Institute of Technology in Sweden. In 1998, he received the Electronic Frontier Foundation's Pioneer Award along with Linus Torvalds. In 1999, he received the Yuri Rubinski award. In 2001, he received a second honorary doctorate, from the University of Glasgow, and shared the Takeda Award for social/economic betterment with Torvalds and Ken Sakamura. In 2002, he was elected to the US National Academy of Engineering, and in 2003 to the American Academy of Arts and Sciences. In 2003, he was named an honorary professor of the Universidad Nacional de Ingenieria in Peru, and received an honorary doctorate from the Free University of Brussels.

## C Programming: Hello, user x!

It is time for another lesson in the C language. Last time you learned how to print "Hello, world!" to your computer's monitor. This time you will learn how to write a C program that says hello to you, the user, by giving the computer input.

Here is the C source code to do accomplish this:

```
(1) #include <stdio.h>
(2)
(3) int main(void)
(4) {
(5)     char name[9];
(6)
(7)     printf("What is your name: ");
(8)
(9)     scanf("%s", name);
(10)
(11)    printf("Hello, %s!\n", name);
(12)
(13)    return(0);
(14) }
```

### Source Code Breakdown

- (1) Include the stdio header file in your program (required for input/output).
- (2) This is white space, we use it to make your code easier to read.
- (3) The main function (required for every C program) is returning an integer type (null 0), and nothing (void) is being sent to another function.
- (4) This is an open brace, start your statements after it.
- (5) We use a character variable type (char) with the designation "name" to hold the name of the user upon input. Notice that the variable can only hold nine characters ([9]), eight maximum for the users name and one for the null zero (/0).
- (6) More white space for clarity.
- (7) Use this, the print function, to print to the user's monitor. This is where we ask for the name of the user for input.
- (8) More white space for clarity.
- (9) This is the scan function, it is used to scan what the user inputs as a string (%s) of characters and puts them into our name character variable.
- (10) More white space for clarity.
- (11) Use this, the print function to print to the user's monitor. This is where we print the users inputted name found in the name variable.
- (12) More white space for clarity.
- (13) The return function, a null zero (0) , needs to be returned at the end of our program to show the program is finished.
- (14) This is a close brace, use it when you have no more statements to add.

## Interview with FreeDOS Creator Jim Hall

**Q:** Where are you from/reside?

**A:** I was born in Virginia, but I currently live in Minnesota, in St Paul (part of the Twin Cities), I'm an American. :-)

**Q:** Do you have significant other in your life?

**A:** I am married to my high-school sweetheart. We've been

dating/married longer than half our lives.

**Q:** Do you own any animals?

**A:** I have three cats Linus, Murphy, and Vita.

**Q:** What kind of educational background do you have?

**A:** I have a BS in Physics, from the University of Wisconsin- River Falls. I try to remain involved with the UW-RF Physics program, and I attend the (mostly) annual alumni dinners that they have.

**Q:** What do you do for work?

**A:** I work at the University of Minnesota, where I'm the Manager of Central Computing Operations' Systems Administration groups, which includes: Solaris, AIX/PIA, Linux/Web, and Windows/Novell.

At the University, I founded the Web Development Advanced Labs. We officially retired the Labs in March 2003 when I moved out of Web Development into my current role in Central Computing Operations but work still goes on for Advanced Labs (under other names).

Our last completed project may interest you, I created an experiment to run Linux on my desktop half-time (dual-boot). It was a good experiment since I was in my WebDev office for half the day and my CCO office for the other half, I would run Windows when in WebDev and Linux when in CCO. After a short while, I stopped booting into Windows altogether, except to do a few things that I can't do in Linux, like run Visio and change my Novell password. I'm 99+% Linux at work now.

**Q:** When was your first computing experience, and what kind of hardware and operating system were used?

**A:** Wow, that takes me back. My dad recognized very early that computers were going to be very big. He borrowed some kind of desktop computer from his work (this thing was huge!) and my brother and I would take turns reading a programming manual, and practice writing simple programs in BASIC. Really simple math games mostly. Then we did some games that did primitive text graphics (mazes and such).

My mom worked at Plato, a spin-off from Control Data Corporation. She brought home a Plato terminal for a weekend. That was amazing! The Plato terminal had a touch screen as well as a keyboard, and there were some simple games that used the touch screen and communicated with a mainframe over a modem. I remember that I liked the horse racing game the best. As I recall, you could even play against other people who were also connected to the mainframe.

But I suppose the first time I *really* got into computers was when my dad bought us a Franklin Ace 1000 (<http://oldcomputers.net/ace1000.html>) an early clone of the Apple ][e. It had the same hardware and ran the same software as an Apple, but was a bit cheaper. I wrote lots of programs for it in BASIC, including a nuclear war simulator ala the "War Games" movie. At one time, my brother and I tried to write a BBS, but that turned out to be too much for us.

Later, my parents bought an IBM PC so they could do some work at home, and I moved onto that. This was a huge leap, and I was thrilled about the features of the IBM DOS. Commands could be connected together using pipes, so now a simple batch file could become (almost) like a program if you could connect the right stuff together. There was a BASIC interpreter, so I was able to convert most of my Apple BASIC programs to GW-BASIC and run them on the IBM. But I didn't need to convert all of my BASIC programs, since there were all these programs for DOS that I

could use, including some text processors and (gasp!) a spreadsheet.

**Q:** Why did you begin the FreeDOS project?

**A:** I was still a student at UW-RF, and I did all my work using DOS. I liked the simplicity that DOS offered. Anything that you want to do on the PC is possible. Nothing is really stopping you, other than hardware limitations. I found that this degree of freedom was nice to have.

But that year (1994) there was an announcement from Microsoft that they would stop support for DOS, and a new version of Windows was going to be released that completely removed DOS from the picture. (Of course, this was Windows 95, and it still did have DOS.) Everyone was pretty shocked. We didn't want to be forced to use Windows, which at the time really sucked. In DOS, everything is done on the command line, and a true command line "guru" can do amazing things there. In Windows, you are stuck with the mouse, and if the menus don't let you do something, it pretty much can't be done. So things were looking pretty bleak. We were all very upset about Microsoft's decision to ditch the DOS platform for this sucky version of Windows.

Then, I saw a discussion thread on the DOS newsgroups asking "hey, why doesn't someone write their own free version of DOS?" Remember, this was about three years after Linus Torvalds announced his work on the Linux kernel, and by 1993 Linux had shown that free software can achieve incredible results. So in 1994, the suggestion that we could write our own free version of DOS, and give it away with the source code so others could work with it and improve it, really didn't sound all that far-fetched.

Unfortunately, no one did anything about it. The idea sort of sat there, waiting. I didn't have tons of experience in writing C or Assembly programs (most of my analytical work in physics was limited to FORTRAN) but I had written some C programs. So I sat down one weekend and hacked out code for a bunch of DOS file utilities. I posted what I had done to the DOS newsgroups, and announced that I intended to form a group on the Internet to write our own free version of DOS. That's what became the FreeDOS Project.

**Q:** How do you envision FreeDOS being used by others?

**A:** I don't really have a vision for how people should use FreeDOS. I'm just thrilled that people still find it useful to them, and they are able to do so many things with it.

Besides the obvious use of running old DOS programs on your computer, FreeDOS is used in DOS emulators, and in embedded systems. FreeDOS is also an invaluable resource for people who would like to develop their own operating system. I still get e-mail from students who are working on an operating systems project, and they have found FreeDOS to be very helpful in understanding the basics. DOS is not a complicated operating system, and is fairly easy to figure out.

**Q:** Why did you choose to go with the Borland C compiler and the MASM assembler for FreeDOS's development?

**A:** These are decisions that were made fairly early in the project, and by now the standards have changed, but I haven't updated all the docs yet to reflect that. When we began, there wasn't a free C compiler available for people to use. Sure, there was djgpp (the GNU C Compiler, ported to DOS by DJ Delorie) but that required at least an 80386 to run. We wanted to make Free-DOS available on low-end hardware including the 8088, so we needed to choose a different compiler.

Looking around at the contributors to FreeDOS at the time, most people had a copy of Borland C (or a variant, such as Borland

Turbo C) and Microsoft's MASM. So, to make code sharing easier, it was a fairly easy decision to settle on Borland C as the default C compiler, and MASM as the default assembler. Arrow ASM provided a superset of MASM instructions, and was freely available, so that was an option for people who didn't have Microsoft MASM. And Borland provides free copies of their Turbo C compiler.

Of course, this didn't mean that you had to have either tool to contribute to FreeDOS. Rather, this meant that any code contributed must be compilable with Borland C 3.1 and MASM.

Today, the standard C compiler is OpenWatcom, and the standard assembler is NASM. Both are Free Software.

**Q:** Have you ever met any of the core development team in-person?

**A:** I've only met two FreeDOS contributors in person: Joe Cosentino and Russ Nelson. Joe Cosentino wrote several of the base utilities that are distributed with FreeDOS, including the current FreeDOS EDIT and Russ Nelson wrote Freemacs, a clone of GNU emacs for DOS.

When we started the FreeDOS Project, Freemacs was our replacement for MS-DOS EDIT, although later Joe wrote a FreeDOS EDIT that could be used by non-programmers. Russ also wrote the packet drivers that are used with FreeDOS.

Russ stopped by my session at the O'Reilly Open Source Convention in 2001, and it was nice to chat with him for a while before he had to run off to a meeting.

Joe's work brought him to the Twin Cities, and he e-mailed me one day to say he'd be in town and did I want to meet. I thought this was great, so I took him out to dinner, and we've been good friends ever since. He's one of the few people in the FreeDOS Project that I consider a close friend.

That said, there are lots of others in the FreeDOS Project that I'd like to meet in person. Maybe one day I'll get to meet some of them.

**Q:** What are some of the biggest challenges you have faced in the development of FreeDOS?

**A:** Overall, I think the biggest challenge was being taken seriously as a free software project. When you first start a project like FreeDOS, the only people who take an interest in you are other DOS users. It's hard to get others who might not consider themselves DOS users to find your project interesting.

**Q:** Where do you stand on intellectual property i.e. do you favor more of an individual whatever license works for a given piece of software, or a free software for all position?

**A:** No, I don't believe that one license fits all software. You have to evaluate needs and demands when you choose a software license. I'd prefer that all software remain free and open, but in the end I do recognize that sometimes that's not possible. In my personal life, I tend to use free and open software instead of proprietary "closed source" software.

**Q:** What games would you love to see work in FreeDOS?

**A:** A lot of DOS games already work in FreeDOS. The ones that I have always enjoyed are 'Epic Pinball' (works great in DOSEmu), 'DOOM' (works great!), 'Wolfenstein 3D' (works fine), 'Dark Forces' (movies played very slow, last time I tried it), and the 'Commander Keen' series (works great!).

If I had an old-style joystick port on my IBM Thinkpad, I'd love to try 'TIE Fighter' again, but alas I think that's a pipe dream!

**Q:** What is your position on abandonware, do you see such softwares source code particularly useful to the FreeDOS project?

**A:** Depends on what you mean by "abandonware", I suppose. There are two different meanings to that term: either as "obsolete" software that is now distributed freely (although illegally), or as old software that the copyright owner has decided to distribute for free (but without support.) 'Lotus 1-2-3' would be an example of the first type, and 'Turbo C' would be an example of the second type.

I prefer that any software that is distributed be done so with the copyright owner's permission. That is, I would like to see more of the second type of abandonware. Even better would be if the copyright owner decided to open up the source code to their software, so that others could improve it.

**Q:** What do you tell people when they question the legitimacy of working on a DOS operating system project since the most well known DOS, MS-DOS, has now been abandon by Microsoft.

**A:** In the first few years of the FreeDOS Project, I got a lot of emails asking me "why are you doing this?" or "why DOS?" The answer is "because I find it interesting" and "because I like DOS", but those kinds of answers don't move non-DOS users to support you. But it's an honest answer, and there wasn't much I could do to sway opinion.

Today, things are much different. Do a search on "FreeDOS" and you'll find thousands of hits. FreeDOS gets mentioned in documentation and on various web sites ala "you'll need a copy of FreeDOS to run that." So we're a very common thing now, which is quite a step up from the days when no one really took us seriously.

**Q:** Do you think now that MS-DOS has been abandon by Microsoft that its source code should be released, and if so do think that source code would be of any significant use to the FreeDOS project?

**A:** Yes, I think it would be great if Microsoft released the source code for MS-DOS. However, I don't know that the source code would be of too much value to the FreeDOS Project, since we've somewhat diverged from the designs of MS-DOS, and our source trees will look completely different making merging impossible. But the MS-DOS source code could be useful to add better compatibility to FreeDOS.

**Q:** Where does the FreeDOS project currently need the most help?

**A:** We have needs in a few specific areas:- We're making the big push towards the "1.0" release, finally. Aitor has created a list of things that need to be finished before we can consider ourselves at the "1.0" release point. So we need some help from developers in finishing up the few things that should be done.

FreeDOS really needs a rock-solid GUI. There are several GUIs available, including Desktop2, SEAL, and Octane. Unfortunately, much effort seems to go into writing a "new" GUI for FreeDOS, and not as much into improving an existing GUI and adding new features.

Also, our documentation effort (<http://fd-doc.sf.net/>) needs an overhaul. I tried to take this mini-project forward, but I wasn't able to balance FD-DOC, the FreeDOS.org site, and all the little things that go with the role of "coordinator". I'd like to see someone (preferably someone who does professional technical writing) come to the aid of FD-DOC and help make some major

updates to our documentation.

**Q:** Is there anything that you would like to clarify that you think others have gotten wrong about the FreeDOS project?

**A:** I guess I'd like to reiterate the point that DOS is not dead. I get that comment a lot from people. With Linux and Windows being the popular operating systems these days, it's hard for people to recognize the value in a text-based, single-tasking operating system like FreeDOS.

DOS will be around for quite some time yet. DOS remains a great environment to work in if you are building an embedded system, for example. The operating system is light, so it will run well in a device that doesn't have a lot of memory. You can burn it into ROM, boot from a floppy, or a small micro-drive. There aren't many operating systems that you can find these days that will boot from a floppy, yet still leave you enough room on the disk for your embedded program and maybe some room for data files.

The embedded systems market is one of the markets that will keep DOS alive, at least for several years to come. Look at some of the alternatives that you have today if you want to build a small embedded device, and you'll see that as footprint becomes an issue for you, DOS is really the only way to go. Linux and Windows both require too much overhead to build an efficient embedded system.

Another thing that will keep DOS around, even behind the scenes, is the PC market. A lot of companies that make computer hardware (disk arrays, video cards, controller cards, etc.) use some kind of basic BIOS. At some point, you'll need to provide a patch, and the user will need to flash the BIOS on their device. The hardware manufacturer may not know or care exactly what operating system you use. If you are the hardware manufacturer, how do you distribute a flash update to all your users? Will you provide a different update program for every operating system platform that might be using your hardware? For a lot of companies, the answer is no. It's a lot simpler if they can give their users a bootable DOS floppy disk that they put into their computer, and the boot floppy automatically flashes the BIOS. You see that in several places today, although you may not realize that you are booting FreeDOS.

## Encryption from Addition

The idea to write a C program that enciphered a plain text message by adding the values of each character in the plain text message to those of secret key came to me one day. But how to go about it in took some pondering.

The following is the C code that I came up with.

```
#define max_array_size 1000

#include <stdio.h>

int input_array(void);
int build_array_c(char array[], int array_size);
int build_array_k(char array[], int array_size);
int encipher(char message[], char key[]);
{
    int main(void)
    {
        input_array();

        return(0);
    }

    int input_array(void)
```

```
{
    char message[max_array_size];
    char key[max_array_size];

    build_array_c(message, max_array_size);

    build_array_k(key, max_array_size);

    encipher(message, key);

    return(0);
}

int index_c = 0;

int build_array_c(char array[], int array_size)
{
    int input;

    while(index_c < (array_size - 1) && ((input =
    getchar()) != '\n'))
    {
        array[index_c] = input;
        index_c++;
    }

    array[index_c] = '\0';

    return(0);
}

int index_k = 0;

int build_array_k(char array[], int array_size)
{
    int input;

    while(index_k < (array_size - 1) && ((input =
    getchar()) != '\n'))
    {
        array[index_k] = input;

        index_k++;
    }

    array[index_k] = '\0';

    return(0);
}

int encipher(char message[], char key[])
{
    int loop;
    int matrix_c = -1;
    int matrix_k = -1;
    int message_element_value;
    int key_element_value;
    int enciphered_element_value;

    char message_element;
    char key_element;
    char enciphered_message;

    for(loop = 0; loop < index_c; loop++)
    {
        matrix_c++;

        if(matrix_k < index_k)
        {
            matrix_k--;
        }
    }
}
```

```

matrix_k++;

message_element = message[matrix_c];

message_element_value = message_element;

key_element = key[matrix_k];

key_element_value = key_element;

enciphered_element_value = key_element_value +
message_element_value;

enciphered_message = enciphered_element_value;

printf("%c", enciphered_message);
}

return(0);
}

```

It worked very well when compiled with "gcc" in FreeBSD 4.8, but for some reason when I compiled and ran it on a computer running Slackware 10 GNU/Linux, it did not work. The ciphered text results were not correct. Until I have sorted through the reason for this, please feel free to give my code a try and write PHRAINE with any suggestions.

For the next issue I will present the method to reverse the process and decipher a ciphered message that uses this C code.

## Enciphering Your E-mail With GNU Privacy Guard

With a recent court ruling in the First Circuit Court of Appeals in Massachusetts that held that it was not a violation of federal, criminal wiretap laws for the provider of an e-mail service to monitor the content of users' incoming messages without their consent, I decided that it was time to put an end to sending my e-mails for someone other than the intended recipient to read.

I think that this court ruling should send the message home to the reader, too. If you don't want your e-mails read by who knows who at your Internet Service Provider or anyone else who has intercepted them, then I think the choice is not why you should encipher your e-mail - but how to encipher it.

A program called Gnu Privacy Guard is freely available. It can be downloaded at <http://www.gnupg.org/>. If you have gotten this far with this article, I assume you would like to get started with setting up in Gnu Privacy Guard to encipher your e-mail messages, so read on.

Once you have installed "gpg" type the following: **gpg --gen-key**. You should have something on your screen that looks like the following:

```

you@yourbox:~$ gpg --gen-key
gpg (GnuPG) 1.2.4; Copyright (C) 2003 Free Software Founda-
tion, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

```

Please select what kind of key you want:  
(1) DSA and ElGamal (default)  
(2) DSA (sign only)  
(4) RSA (sign only)  
Your selection?

The next step is to choose what kind of encipherment method you would like to use. I will be choosing the default for this purpose of this article. To do so, just press your **Enter** key.

You should now have something on your screen that looks like the following:

```

DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
  minimum keysize is 768 bits
  default keysize is 1024 bits
  highest suggested keysize is 2048 bits
What keysize do you want? (1024)

```

The choice here is really yours. The higher the key size the stronger the key (theoretically) is from cracking. For our purposes we will be going with the default by pressing the **Enter** key.

You should now have something on your screen that looks like the following:

```

Please specify how long the key should be valid.
  0 = key does not expire
  <n> = key expires in n days
  <n>w = key expires in n weeks
  <n>m = key expires in n months
  <n>y = key expires in n years
Key is valid for? (0)

```

The next choice you will have to make is whether the key you create expires. An expiring key theoretically adds security, but it means you have to go through the process of creating another key once it does expire. Again, this is your call here. We will be going with the default by pressing the **Enter** key.

You should now have something on your screen that looks like the following:

```

Key is valid for? (0)
Key does not expire at all
Is this correct (y/n)?

```

You should now press the **y** key followed by the **Enter** key.

You should now have something on your screen that looks like the following:

```

You need a User-ID to identify your key; the software con-
structs the user id
from Real Name, Comment and Email Address in this form:
  "Heinrich Heine (Der Dichter)
<heinrichh@duesseldorf.de>"

```

Real name:

Now it is time to type your real name, a comment, and your e-mail address. We typed each one in turn:

```

Real name: john doe
Email address: johndoe@private.org
Comment: key

```

You should now have something on your screen that looks like the following:

```

You selected this USER-ID:
  "john doe (key) <johndoe@private.org>"

```

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit?

continued next issue...

## Boolean Algebra Basics

Boolean algebra is the logical basis for computer operation. It was originally developed as a way to analyze philosophical debates. I have decided to use one of the biggest debates there is to teach the three precise definitions boolean algebra utilizes: "NOT", "AND", and "OR". The philosophical debate we will be utilizing is whether God exists.

In Boolean algebra, two conditions can be met: existence or nonexistence. Each of these conditions is represented by a number: existence is a "1" and nonexistence is a "0". Now that we know this, we need a variable to represent God. It seems logical to me to use the letter "g". Knowing these things has prepared us for the debate!

Let's say you believe in the existence of God, so to express this side of the philosophical debate of whether God exists using Boolean algebra you would write  $g = 1$  which precisely states that God "g" exists "1". Now it is my turn to debate.

I believe in the nonexistence of God, so I would express this in Boolean algebra as  $g = 0$  which in Boolean algebra precisely states God "g" does not exist "0".

At this point, we are tired of arguing this debate and want another way to state the position each of us has in the debate.

Let's say our positions in the debate are not the position of another person. To express this in Boolean algebra, we would use the NOT definition expressed with a line above the variable, which in our case would look like  $\bar{g}$  with line above it because this mean NOT. We will sum up the belief in the existence of God with Boolean Algebra as follows  $g = 0$ . In English, this says God is not nonexistence but on the other side.

continued next issue...

## Richard Stallman's Action Items

### Urgent

In the US with a computer, a phone and some time? [Help the MPP's Montana medical marijuana initiative.](#)

You can [help the Starbucks Coffee workers unionize](#) and overcome the dirty tricks of their employer. Sign the petition against execution of [four Indian Dalits who resisted a campaign of massacres by upper-caste landlord armies.](#)

[A protest is planned in New York City on Sep 2 against Bush policies in Iraq, Palestine, and Haiti.](#)

US citizens: [contact your senators](#) to oppose a new law that would prohibit photocopiers, tape recorders, VCRs, CD writers, peer-to-peer software, etc.

Call on the Olympic Games to insist on [fair wages for the people who make clothing for it.](#)

US citizens: add your name to the letter to support and add pressure to Senator Dayton's objections to some of the flaws in the official 9/11 investigation.

[Sign Ralph Nader's petition](#) calling for impeachment of Bush. US citizens: [add your name to the pressure for voter-verified paper ballots](#), to protect against theft of the coming election

through black box voting machines.

### Long-term

[Boycott Liberty Apparel](#)- for running sweatshops.

US voters, [support Kucinich's campaign to create a Department of Peace](#) which would encourage nonviolent solutions to international problems.

UK citizens: read about the mandatory ID card plan and then [sign Liberty's petition against it.](#)

Support <http://www.no2id.net>.

Also see <http://www.defy-id.org.uk/> for another action. In France, people are vowing to [boycott the record companies](#) due to the laws that they are lobbying for.

[Evidence of wide-ranging and persistent FBI misconduct](#), including sheltering murderers, shows that other instances of alleged FBI misconduct, such as the fabricated evidence against Leonard Peltier, deserve fresh investigations. [Sign this petition.](#)

US citizens: you can sign your name to the [campaign to repeal part of the U.S.A. P.A.T. R.I.O.T. Act](#). This would make the police once again have to get search warrants before they can get information from bookstores or libraries about what you have read or borrowed.

This is a step in the right direction, but it does not go far enough. The PAT RIOT act was extended in December 2003 to give the police equally easy access to many kinds of transaction records about you. The PAT RIOT act attacks your freedom in other ways, too, and not all of them will expire in 2005. See

<http://www.aclu.org/safeandfree/>. MoveOn suggests [contributing to the campaign of Congressman Doggett.](#)

UK citizens: renew your passport before 2005, so you can [avoid contributing your biometric to the national ID card plan](#). If you don't have a passport, get one now!

Israelis who refuse military service in the occupation of Palestine have trouble supporting their families. Yesh Gvul is asking for [donations of funds to help these families](#). (The idea is that this will help more people decide to refuse.)

European citizens: tell your government [you do not want digital fingerprints in your passports](#). Push back against the US! Why should you be treated like a criminal just to get permission to leave your country?

Support the call for a [Nobel Prize in Sustainable Development](#). A proposal by Dave Winer to ask the leading Democratic candidates to pledge to [keep the Internet free from media company control](#).

Citizens of Europe: sign the [resolution that demands a voter-verifiable audit trail](#) for electronic voting machines.

Please sign the [petition against unsafe computerization of elections and careless purging of voter lists](#). [Boycott Coca Cola Company.](#)

US voters: [please ask your senators and representatives to support the Truth in Trials Act](#), which would protect state-authorized medical marijuana growers from federal prosecution. Massachusetts citizens: [help convince the legislature to pass a bill](#) to make emergency contraception available from pharmacies without seeing a doctor first.

Support federal medical marijuana legislation.

Join the boycott of Chinese products to support human rights for Tibetans and Chinese.

Residents of Massachusetts: the ACLU is collecting a petition (on paper) asking the Massachusetts congressional delegation to repeal the USA patriot act. Please help collect signatures. You can get the form through the web from [aclu-mass.org/petition.html](http://aclu-mass.org/petition.html).

US voters: the US government is pushing a draconian law that could put you in prison for 20 years just because someone smokes pot in your back yard. It would also be used to shut down political activity.

Precisely how the Bush team stole the Florida election is documented. Florida Secretary of State Katherine Harris, who was co-chair of the Bush campaign in Florida and a close associate of Bush's brother, deliberately blocked careful checking of the list of felons, with the result that thousands of Black people who were not felons and were legally entitled to vote were blocked from voting.

This gun is smoking enough for me. Dubya's forces stole the election; Dubya should resign!

*To follow any of the underlined text featured above, visit (<http://www.stallman.org/>)!*

## The Bazaar

### The Free Software Foundation Needs Your Help

Help protect your rights to run, copy, distribute, study, change and improve software by becoming an associate member of the FFS by visiting the following link: (<http://members.fsf.org/>).

### The board at Madoshi dot com

Join in on the discussions and games at (<http://board.madoshi.com/>).

### Subscribers add you computer related message here!

Send it to [scm@madoshi.com](mailto:scm@madoshi.com); subject: "The Bazaar" by 11-30-04.

