What's That Spell? TEAMWORK!

HARMONY. COOPERATION. SYNCHRONIZED EFFORT. It's difficult, but it can be learned. What's the best way to do so? Watch the great teams very closely—and then join one of your own. By Jerry Useem

In 1972 a crack commando unit was sent to prison by a military court for a crime they didn't commit. These four men promptly escaped from a maximum-security stockade to the Los Angeles underground. Today, still wanted by the government, they survive as soldiers of fortune. If you have a problem, if no one else can help, and if you can find them, maybe you can hire the A-Team.

The A-Team went off the air in 1987—still wanted by the government—but television has never produced a better blueprint for team building. The key elements of its effectiveness: a cigar-chomping master of disguise, an ace pilot, a devilishly handsome con man, a mechanic with a mohawk, and an amazingly sweet van. Those particulars might not translate to all business settings. But clear definition of roles is a hallmark of effective collaboration. So is small team size—though four is slightly below the optimal number, 4.6 (see page 120 for an explanation). And the presence of an outside threat—like imminent recapture by gov-

HANGING OUT, TOGETHER Cirque du Soleil performers Tamara Yerofeyeva (top) and Taras Shevchenko (bottom) depend on Oleg Ouchakov (left) and Evgeniya Astashkina (right). For more on what makes Cirque tick, see page 114.

PHOTOGRAPH BY MICHAEL O'NEILL

ernment forces—likewise correlates with high team cohesion. To wit: France and England, which bloodied each other for centuries before they noticed ... Germany. Another universal characteristic of teams is that they're, well, universal. If you work for a living, we're guessing you interact with other humans. (Lighthouse keepers, we'll see you next time.)

MEN AT WORK Workers swarm over a geodesic dome outside Ford's River Rouge Plant, circa 1953.

If you think this is mushy stuff, marginal to the daily battle of business, consider what is happening at Sony (page 70). CEO Howard Stringer and President Ryoji Chubachi are trying to restore the fighting spirit (and higher profits) at a company built on decentralized teams. Their theme: Sony United.

This issue also takes you deep inside a six-man team of Marines operating in Iraq (page 105); the team that built Motorola's RAZR phone (page 124); the cutthroat yet symbiotic pack of cyclists in the Tour de France (page 145); and the world of an open-source software company (page 134). Each of these stories challenges a piece of conventional wisdom. If "hire great people" seems like unassailable advice, for example, then read Geoffrey Colvin's "Why Dream Teams Fail" (page 87).

The fact is, most of what you've read about teamwork is bunk. So here's a place to start: Tear down those treacly motivational posters of rowers rowing and pipers piping. Gather every recorded instance of John Madden calling someone a "team player." Cram it all into a dumpster and light the thing on fire. Then settle in to really think about what it means to be a team.

We're certainly not against the concept of teamwork. But that's the point: All the happy-sounding twaddle obscures the actual practice of it. And teamwork *is* a practice. Great teamwork is an outcome; you can only create the conditions for it to flourish. Like getting rich or falling in love, you cannot simply will it to happen.

We will go further and say: *Teamwork is an individual skill.* That happens to be the title of a book. Christopher Avery writes, "Becoming skilled at doing more with others may be the single most important thing you can do" to increase your value—regardless of your level of authority. As work is increasingly broken down into team-sized increments, Avery's argument goes, blaming a "bad team" for one's difficulties is, by definition, a personal failure, since the very notion of teamwork implies a shared responsibility. You can't control other people's behavior, but you can control your own. Which means that there is an "I" in team after all. (Especially in France, where they spell it *équipe*.)

Yet this is not the selfish "I" that got so much attention during the "me" decade; it's the affiliatory "I" that built America's churches and fought its wars. Neil Armstrong didn't get to the moon through rugged individualism; there is no such thing as a self-made astronaut (see page 116). "Men work together," wrote Robert Frost, "whether they work together or apart."

Here's both the problem and the promise of cooperation. Humans aren't hard-wired to succeed or fail at it. We can go either way. In her study of groupwork in school classrooms, the late Stanford sociologist Elizabeth Cohen found that if kids are simply put into teams and told to solve a problem, the typical result is one kid dominating and others looking totally disengaged. But if teachers take the time to establish norms—roles, goals, etc.—"not only will [the children] behave according to the new norms, but they will enforce rules on other group members." Perhaps to a fault. "Even very young students," Cohen wrote, "can be heard lecturing to other members of the group on how they ought to be behaving."

Economists have long assumed that success boils down to personal incentives. We'll cooperate if it's in our self-interest, and we won't if it's not (sort of like lions, page 121). Then a team of researchers led by Linnda Caporael thought to ask: Would people cooperate without any incentives? The answer was—*gasp!*—yes, under the right conditions. Participants often cited "group welfare" as motivation.

To economists, shocking. To anyone who's been part of a successful team, not shocking at all. Life's richest experiences often happen in concert with others—your garage band, your wedding, tobogganing. The boss who assumes that workers' interests are purely mercenary will end up with a group of mercenaries. No battery of team exercises can fix that situation—especially if they involve spanking your colleagues with yard signs. When a sales office of a home-security company, Alarm One, adopted that practice, a 53-year-old employee later sued for emotional distress. (A jury awarded her $1.2 million in April.)

Again, let the greats show the way. During a public appearance in 2000, an *A-Team* cast member was asked by a fan to name his favorite co-star. "Listen," Mr. T responded. "That's wrong for me to pick a favorite, because I'm a team player and we were a team. Remember, they say"—here it comes again—"there's no 'I' in team." No, but there is a "T." And pity the fool who forgets it. F

These are true stories from the Wall Street Journal.

And we sometimes think we or our clients aren't computer literate !!!

1. Compaq is considering changing the command "Press Any Key" to "Press Return Key" because of the flood of calls asking where the "Any" key is.

2. AST technical support had a caller complaining that her mouse was hard to control with the dust cover on. The cover turned out to be the plastic bag the mouse was packaged in.

3. Another Compaq technician received a call from a man complaining that the system wouldn't read word processing files from his old diskettes. After trouble-shooting for magnets and heat failed to diagnose the problem, it was found that the customer labeled the diskettes then rolled them into the typewriter to type the labels.

4. Another AST customer was asked to send a copy of her defective diskettes. A few days later a letter arrived from the customer along with Xeroxed copies of the floppies.

5. A Dell technician advised his customer to put his troubled floppy back in the drive and close the door. The customer asked the tech to hold on, and was heard putting the phone down, getting up and crossing the room to close the door to his room.

6. Another Dell customer called to say he couldn't get his computer to fax anything. After 40 minutes of trouble-shooting, the technician discovered the man was trying to fax a piece of paper by holding it in front of the monitor screen and hitting the "send" key.

7. Another Dell customer needed help setting up a new program, so a Dell tech suggested he go to the local Egghead. "Yeah, I got me a couple of friends," the customer replied. When told Egghead was a software store, the man said, "Oh, I thought you meant for me to find a couple of geeks."

8. Yet another Dell customer called to complain that his keyboard no longer worked. He had cleaned it by filling up his tub with soap and water and soaking the keyboard for a day, then removing all the keys and washing them individually.

9. A Dell technician received a call from a customer who was enraged because his computer had told him he was "bad and an invalid". The tech explained that the computer's "bad command" and "invalid" responses shouldn't be taken personally.

10. An exasperated caller to Dell Computer Tech Support couldn't get her new Dell Computer to turn on. After ensuring the computer was plugged in, the technician asked her what happened when she pushed the power button. Her response, "I pushed and pushed on this foot pedal and nothing happens."

The "foot pedal" turned out to be the computer's mouse.

11. Another customer called Compaq tech support to say her brand-new computer wouldn't work. She said she unpacked the unit, plugged it in, and sat there for 20 minutes waiting for something to happen. When asked what happened when she pressed the power switch, she asked "What power switch?"

12. True story from a Novell NetWire SysOp:

Caller: "Hello, is this Tech Support?"

Tech: "Yes, it is. How may I help you?"

Caller: "The cup holder on my PC is broken and I am within my warranty period. How do I go about getting that fixed?"

Tech: "I'm sorry, but did you say a cup holder?"

Caller: "Yes, it's attached to the front of my computer."

Tech: "Please excuse me if I seem a bit stumped, It's because I am. Did you receive this as part of a promotional, at a trade show? How did you get this cup holder? Does it have any trademark on it?"

Caller: "It came with my computer,I don't know anything about a promotional. It just has '4X' on it."

At this point the Tech Rep had to mute the caller, because he couldn't stand it. The caller had been using the load drawer of the CD-ROM drive as a cup holder, and snapped it off the drive!

WSJ 8/4/2000 p. W11

Grammar Rules

Maybe the distinction between *flaunt* and *flout* is not the most populist cause these days. Maybe, too, the number of Americans mortified at the mere thought of being caught out with a dangling modifier grows fewer by the day. But with the Republican delegates in Philadelphia having declared themselves on virtually all the minor issues affecting the future of the republic—national security, taxes, the death penalty—the question needs to be asked: Where is the leader willing to stand up for the most basic element of civilization, the simple declarative sentence?

It turns out that he's in New York, and his name is James J. McFadden. A former labor commissioner for the city, Mr. McFadden says that though everyone talks about the problems of education these days, not much of that talk translates into action. His did. With backing from both Verizon and Channel 4 WNBC, Mr. McFadden's Manpower Education Institute yesterday released its own weapon in the fight to improve everyday English: a handbook. The title is as direct and no-nonsense as the man himself: "Promote Yourself With Better Grammar."

In keeping with the market Mr. McFadden is aiming for, "Promote Yourself" is not overly burdened with asides from, say, E.B. White or Edgar Allan Poe ("A man's grammar, like Caesar's wife, must not only be pure but above suspicion of impurity"). To the contrary, the introductory remarks are confined to one page, featuring a quotation from . . . Michael Jordan. "The minute you get away from the fundamentals—whether it's proper technique, work ethic, or mental preparation—the bottom can fall out of your game, your school work, your job, whatever you're doing," says the former Chicago Bulls star. And it is as pithy a statement of purpose as you will find.

Such directness is in keeping with the handbook's mission, which is not so much to persuade people that grammar is important but to help those who already know that their grammar needs help and want to do something about it. That it does with clear, sturdy chapters that begin with sentence structure and the distinctions between subject and predicate, moving on to verb tenses, the diagramming of sentences and the rudiments of a proper business letter.

Already "Promote Yourself" has its own Web site (www.bettergrammar.org) and some enthusiastic takers. Mostly these are schools. The City University of New York, for example, plans to use the handbook on its 20 campuses in tutoring programs. Ditto for the city's St. John's University, which says it will use the book in programs for freshmen whose grammar needs remediation.

But it appears to us that the book might be equally welcome on the desks of many entry-level business employees. As Verizon's Paul Crotty told us over lunch, "the people who get ahead in business are the ones with good presentation skills," of which good grammar is a critical component. And the explosion of e-mail, he suggests, means that writing has again become important.

Good grammar may not make you rich. But it will ensure that, whatever you are, you will never be misunderstood.

[redacted] ELECTRICAL ENGINEERING AND COMPUTER SCIENCES

Department Policy on Academic Dishonesty

Copying all or part of another person's work, or using reference material not specifically allowed, are forms of cheating and will not be tolerated. A student involved in an incident of cheating will be notified by the instructor and the following policy will apply:

1. The instructor may take actions such as:
   a. require repetition of the subject work,
   b. assign an F grade or a 'zero' grade to the subject work,
   c. for serious offenses, assign an F grade for the course.
2. The recommended action for cheating on examinations or term papers is 1(c).
3. The instructor must inform the student and the Department Chair in writing of the incident, the action taken, if any, and the student's right to appeal to the Chair of the Department Grievance Committee or to the Director of the Office of Student Conduct.
4. The instructor must retain copies of any written evidence or observation notes.
5. The Department Chair must inform the Director of the Office of Student Conduct of the incident, the student's name, and the action taken by the instructor.
6. The Office of Student Conduct may choose to conduct a formal hearing on the incident and to assess a penalty for misconduct.
7. The Department will recommend that students involved in a second incident of cheating be dismissed from the University.

EECS Policies and Procedures

Computer Use Policy

In support of the University's mission of teaching, research, and public service, Information Systems and Technology (IST) provides computing, networking, and information resources to the University community of students, faculty, and staff.

Rights and Responsibilities

Computers and networks can provide access to resources on and off campus, as well as the ability to communicate with other users worldwide. Such open access is a privilege, and requires that individual users act responsibly. Users must respect the rights of other users, respect the integrity of the systems and related physical resources, and observe all relevant laws, regulations, and contractual obligations.

Students and employees may have rights of access to information about themselves contained in computer files, as specified in federal and state laws. Files may be subject to search under court order. In addition, system administrators may access user files as required to protect the integrity of computer systems. For example, following organizational guidelines, system administrators may access or examine files or accounts that are suspected of unauthorized use or misuse, or that have been corrupted or damaged.

Existing Legal Context

All existing laws (federal and state) and University regulations and policies apply, including not only those laws and regulations that are specific to computers and networks, but also those that may apply generally to personal conduct.

Misuse of computing, networking, or information resources may result in the loss of computing privileges. Additionally, misuse can be prosecuted under applicable statutes. Users may be held accountable for their conduct under any applicable University or campus policies, procedures, or collective bargaining agreements. Complaints alleging misuse of IST resources will be directed to those responsible for taking appropriate disciplinary action. Illegal reproduction of software protected by U.S. Copyright Law is subject to civil damages and criminal penalties including fines and imprisonment.

Examples of Misuse

Examples of misuse include, but are not limited to, the activities in the following list.

- Using a computer account that you are not authorized to use. Obtaining a password for a computer account without the consent of the account owner.
- Using the campus network to gain unauthorized access to any computer systems.
- Knowingly performing an act which will interfere with the normal operation of computers, terminals, peripherals, or networks.
- Knowingly running or installing on any computer system or network, or giving to another user, a program intended to damage or to place excessive load on a computer system or network. This includes but is not limited to programs known as computer viruses, Trojan horses, and worms.
- Attempting to circumvent data protection schemes or uncover security loop holes.
- Violating terms of applicable software licensing agreements or copyright laws.
- Deliberately wasting computing resources.
- Using electronic mail to harass others.
- Masking the identity of an account or machine.
- Posting on electronic bulletin boards materials that violate existing laws or the University's codes of conduct.
- Attempting to monitor or tamper with another user's electronic communications, or reading, copying, changing, or deleting another user's files or software without the explicit agreement of the owner.

Activities will not be considered misuse when authorized by appropriate University officials for security or performance testing.

Crackers, Hackers, WareZ, and Seku-haru

Cliff Frost

The Berkeley campus computing network has more than 10,000 computers attached, and is growing at a rate of more than 300 per month. Our network is also a part of the Internet, a cooperative collection of networks that has hundreds of thousands of computers attached, appears in many countries around the world, and has millions of users.

This growing and wonderful system brings about new ways for us to communicate with each other, but it also gives some people new ways to cause various sorts of trouble. Of course, the vast majority of people have no trouble in this regard, but a surprising number of new users do.

Thinking that a little education may help, in this article I describe some of the problems that have come up recently, with some discussion about why these are serious problems. My hope is that our readers will be able to discuss these things with new users or, at least, find the topic interesting.

The main ways people have abused the network to cause problems are:

- breaking in to other peoples' computers,
- illegally distributing and receiving proprietary software, and
- harrassing others.

When people do these things, they create very real problems for others (such as wasting system users' and administrators' time) and for themselves (such as being arrested, having their computer equipment confiscated, or being tossed out of the University). It seems that some people behave differently "over the network" than in other situations—somehow they seem to think that normal standards of courtesy and ethics (not to mention the law) do not apply to behavior over a computer network.

An important point is that the campus computers and network are shared resources and that abuse of these systems causes real trouble for many other people.

Breaking into Other People's Computers

When people break into a computer that's not theirs (i.e., they access a computer without the owner's permission), they are doing what is often called "cracking" or "hacking." No matter what it's called, it's a nasty form of harrassment, which is why it's both illegal and against University policy.

Some people think cracking is a harmless game. When crackers break into a computer system, although they may not mean any harm, the system users have no way of knowing the crackers' intentions. And, even if the intentions are benign, crackers end up violating the privacy of system users and causing a lot of work for both the users and the administrators of the system (who may have to spend a lot of time determining if their work has been damaged or compromised by the crackers.)

Illegally Distributing or Receiving Proprietary Software

Many people enjoy sharing software packages with friends and with the world at large, via the Internet. It can be fun, and as long as no laws are broken, it's generally considered a "good thing."

Problems come up when software is shared in violation of the software license or copyright because (of course) this is illegal. We've heard various rationalizations for why some people think it's OK to do this: because they don't agree with the law; because they distribute the software and not the manuals; because they think they aren't doing any harm to anyone. Others have said they do it because they think it's "kool" to share their "WareZ" with others. (WareZ, which is pronounced like "wares" but with an emphasis on the "z" sound, frequently refers to contraband software and sometimes hardware. It also indicates an attitude of considerable nonchalance with respect to the laws being broken.)

Unfortunately, when people are caught doing this, they find that because it's illegal the University and the police have to act. The usual result is a gigantic waste of everyone's time. This time would be far better spent on many other things, such as better services in student computer labs. So the crime is far from being "victimless," even leaving aside the people whose copyrights and licenses are being violated.

Harassing Others

Tools like electronic mail (e-mail), interactive "talk" programs, and electronic bulletin boards make it easy to communicate with lots of different people. A major problem that comes with these tools is that they don't require face-to-face contact (which is, of course, one of their positive features also). There is an article in this newsletter, "Electronic Mail Etiquette," with tips on how to use electronic mail effectively. New and experienced users may both find it useful.

There are a couple of major types of communication traps that many new users fall into:

One is sending electronic chain letters. These letters ask you to send a copy to several other people, who are asked to send a copy to several other people, and so on. If you receive one of these, please throw it away—and let others know to do the same. Most people feel these notes are harassing (system adm inistrators certainly feel this way) and sending them is against University policy.

The other trap is called "seku-haru," which is the Japanese word for sexual harassment (I read it in the New York Times, so it must be as real as "grunge" fashions.)

Sometimes people send e-mail to a user just because the user's account name attracts their attention. Usually this mail is unwanted and is sometimes seen as harassing. The sender may mean no harm, but the person getting the notes may find them annoying and threatening. In fact, it is best not to send e-mail to anyone who has not expressed some willingness to receive it.

University Policy

The University has lots of policies. Not surprisingly, there is a policy that covers use of the student computer labs, UCLink, the network, and other centrally administered systems. We have printed a copy of it on page 7 for your convenience. ■

Electronic Mail Etiquette

Alice Brzovic

Editor's note: This is a revised version of an article originally published in the Spring 1989 issue of the Berkeley Computing Quarterly.

Your office may have a standard format for internal memos, but does it have one for electronic mail? Probably not. Since electronic mail is a relatively recent development in communications, a standard of usage has not been chiseled in stone.

You will find, however, that there are a few courtesies and customs to guide you in your use of electronic mail. Below is a list of some of the do's and don'ts. You might consider them the next time you use electronic mail.

- Be concise. Too much information in one message is a burden on recipients. Screens are harder to read than words on paper.
- Consider the layout of your message. Don't write with uppercase letters only. Consider using short lines and paragraphs.
- Keep your discussion focused. If a new topic is introduced it should be under a separate message with a new subject heading.
- Label your subject clearly and uniquely. Unique subject headings make for easy filing, cataloging, cross referencing, and retrieval.
- Don't send junk mail. Refrain from using electronic mail for unnecessary broadcasting. For example, chain letters are an inappropriate use of electronic mail and can cause excessive loading of mail facilities.
- Pay attention to the distribution list before forwarding received mail to someone else. The recipient might have several copies of that item already.
- Copy others who may be affected by your message, or who may have information or suggestions to add.
- Don't expect instant response to your mail. Not everyone anxiously awaits your message. If you are uncertain of a recipient's electronic mail habits, or are not getting any response to your messages, a phone call or memo may be quicker and more effective. Electronic mail is not a replacement for other communication tools.
- Assume the messages you send and receive are permanent. Don't say anything in electronic mail you might not want to be made public or forwarded to others.
- Cite your information clearly and correctly even if you are paraphrasing. If you are sending information from another source, pay attention to whether the material is copyrighted. Copyright laws apply to electronic mail as well as to printed media.
- Mark text that is not your own. Alterations of a text could confuse the original meaning and embarrass the author.
- Don't forward confidential mail to others without first obtaining permission.
- Be aware that electronic mail might not be as private as you may wish because it works through shared technology. If

confidentiality and privacy are very important, it may be advisable and more appropriate to use other communication methods.

- Don't rush writing your messages. Remember that the recipient will have plenty of time to read your message. Use text editors and spelling checkers to help you organize your thoughts and make sure your text is accurate.
- Use simple English to get your meaning across.
- Remember that you may be sending e-mail to readers with varying levels of expertise. Some of your readers may not understand terminology that is familiar to you.
- Refrain from adding too many attachments to your electronic mail. Large, bulky messages tie up the network and are difficult to read
- Avoid trivial responses unless you really have something important to say. As a receiver it is not necessary to respond to everything.
- Take time to comprehend what has been written before responding, especially if the message makes you angry. The diversity of backgrounds, cultures, opinions, and writing abilities in electronic mail users sometimes makes understanding difficult. If you don't understand a particular item, let the sender know rather than jumping to an incorrect conclusion.
- Remember that you are not interacting with a machine, but with someone who is as sensitive as you are. ■

How to Obtain Computing Accounts

Jacqueline Craig

IST offers a number of shared computing services, which provide a wide range of computing applications to students, faculty, and staff. The following information will assist you in determining which shared services you may choose to meet your needs.

Accounts for Electronic Mail and Access to Information Resources

The new UCLink service provides access to electronic mail and information resources at no cost to students, faculty, and staff. These accounts, which you may use while you are affiliated with UC Berkeley, can be obtained simply by connecting to UCLink and using an account setup program to establish your personal account. Once your account is established, you can use one of the available free software tools for accessing UCLink from Macintoshes or IBM PCs or compatibles, or you can log in directly to UCLink and use the simple e-mail programs provided. See the article "UCLink Pilot Project a Success" in this issue for more information.

Accounts for Computation and Data Processing

Shared UNIX and IBM VM/CMS services are available for those who require a mix of computational or data processing functionality. These shared services offer statistical analysis, numerically intensive computing applications, data storage and retrieval, programming languages, and remote job entry to other machines. If such applications are needed for instruction or for departmentally sponsored research, accounts can be obtained at no cost under the Academic Computing Resources (ACR) program. For extramurally funded research or departmental administration not directly in support of an instructional program you must pay for these accounts with grant or departmental funds. The remainder of this article explains how to obtain accounts and covers general account management issues.

Setting Up Shared Accounts

Accounts for shared UNIX and IBM VM/CMS services may be obtained by completing the *Application for Computing Services* (GEN 4.4.5) form. This form, and the other forms mentioned below, are available from the User & Account Services (UAS) office.

The application form has two parts. The first part sets up the "customer account number" that is used to account for usage of the service. Complete either the green or pink page as applicable for requesting either ACR or grant-funded or departmentally-funded accounts. The second part contains system service pages where you indicate which services you are requesting. It is necessary to obtain the required signatures for ACR accounts as indicated on the form. In particular, students must obtain the signature of a faculty sponsor in order to obtain an ACR account. Grant-funded or departmentally-funded accounts require the

From liza@hera.EECS.Berkeley.EDU Fri Jan 21 14:33:29 1994
Received: from hofmann.CS.Berkeley.EDU (hofmann.CS.Berkeley.EDU [128.32.131.12])
Received: from hera.EECS.Berkeley.EDU (hera.EECS.Berkeley.EDU [128.32.240.97]) b
Received: from mail.EECS.Berkeley.EDU (mail.EECS.Berkeley.EDU [128.32.240.137])
Received: from cory.EECS.Berkeley.EDU (cory-138.EECS.Berkeley.EDU [128.32.138.81
Received: from localhost (lopez@localhost) by cory.EECS.Berkeley.EDU (8.5/8.5) i
Date: Fri, 21 Jan 1994 14:37:40 -0800
From: "Francisco R. Lopez" <lopez@cory.EECS.Berkeley.EDU>
Message-Id: <199401212237.OAA03887@cory.EECS.Berkeley.EDU>
To: csfaculty@eecs.berkeley.edu
Subject: Instructional Support to discourage cheating
Cc: inst@cory.EECS.Berkeley.EDU, kevinm@eecs.berkeley.edu,
        wensel@eecs.berkeley.edu
Status: R

Greetings, Professor.

During the course of the previous semester, there seemed to have been some alarming instances of cheating. The Instructional Support Group would like to remind you we have a software program available for your use to help detect plagiarism of C source code. The program is meant as an aide only; it detects the degree of similarities between C source files. The output is a percentage approximation of this; but the instructor must still make a final determination. It is one of our goals to improve the program's accurracy and to expand the number of programming languages it recognizes.

We can run this software for you, but we will require the assignment due dates for your class and the location of the students' source code. This information should be sent to inst@cory. We then hope to give you a listing of the program's output along with the account names whose source code bears closer scrutiny.

If you wish to run the software yourself, it can be found on our Instructional machines in the /share/b/histok directory which is only accessible by your class master accounts; there you will find a short README file explaining how to use it. If you have any questions/comments or require further information, please send me e-mail (lopez@cs).

Please enjoy your semester,

Francisco Lopez
EECS Instructional Support

Software Finds Student Cheaters

Cal computer science professor's program detects copying of code

By Julia Angwin
Chronicle Staff Writer

Computer science students, take note: You can't cheat technology.

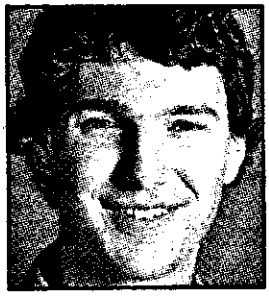
Here's why: A new software program called MOSS roots out plagiarism among computer programming students.

"It will detect cheating even if you only cheated on part of the assignment," said the program's author, Alex Aiken, an associate computer science professor at the University of California at Berkeley.

Aiken wrote MOSS (Measure of Statistical Similarity) to tackle the little-publicized, but fairly prevalent problem of students "borrowing" sections of computer code from one another on their homework assignments.

Teachers don't always find these borrowed sections when they are buried in programs that can be thousands of lines long. In addition, many computer science professors have classes with hundreds of students because of the popularity of the field.

"I have five graders — so even if I had asked them to look for similarities they wouldn't necessarily find them because they're only seeing one-fifth of the class," said Katherine Yelick, a colleague of Aiken's at UC Berkeley who uses MOSS.



UC Berkeley Associate Professor Alex Aiken gives MOSS away on his Web site

She said MOSS is the best anti-cheating program she's used. There are approximately two dozen other programs out there, but MOSS is the only one that displays results in an easy-to-use fashion.

MOSS is a program that displays results in an easy-to-use fashion

Aiken offers access to MOSS for free on his Web site — *www.cs.berkeley.edu/~aiken/moss.html*. He doesn't plan to charge for it unless it starts consuming more than the few hours a week he now spends answering e-mailed questions about it. Professors e-mail their students' homework assignments to MOSS and get an automatically generated Web page of results.

The individually tailored Web pages highlight long sequences of code that are alike and rank programs by similarity. But it's up to the professor to figure out which ones are coincidence and which ones are cheating.

"You still have to try to understand the situation — and that takes a long time," said William Griswold, a UC San Diego computer science professor who uses the program.

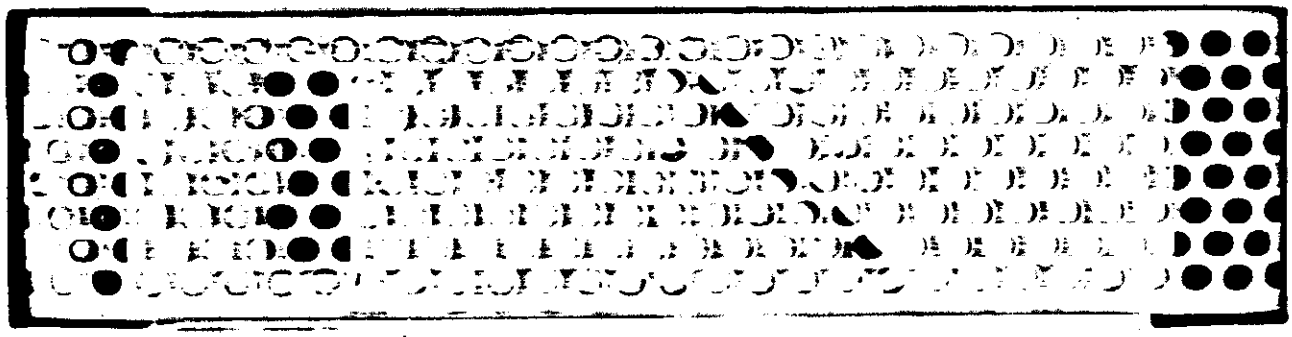
Griswold spent a month this year researching some similarities that MOSS uncovered in his advanced programming class.

"I got old projects and compared them to the new ones to try to find the source of the cheating," he said. "The reason I needed to do this was because when the dean confronted the students he needed to be able to say, 'You got this project from so-and-so.' That basically trumped the student's denial."

He eventually found 12 instances of cheating. The students were flunked, suspended or both. And when he looked at previous year's programs with MOSS, he's also found past instances of possible plagiarism that went unchecked.

Now he's set up stricter guidelines about cheating and plans to use MOSS to enforce them.

"This was an emerging problem which will no longer become a problem," he said.



The Urge to Punish Cheats: It Isn't Merely Vengeance

By NATALIE ANGIER

Over the last couple of weeks, as the Enron fiasco has played itself out like a louche fusion of Shakespeare and the old "Dewey, Cheatum & Howe" routine, Americans have been transfixed by the story, united in a nearly seamless sense of outrage.

Regardless of whether any laws were broken in the spectacular collapse of one of the nation's largest companies, citizens of all political pipings have voiced disgust at accounts of top Enron executives selling off their stock in time to enrich themselves handsomely, while ordinary Enron employees were later forced to sit by in impotent desperation as their retirement savings evaporated.

In the ferocity of the public outcry, and the demand from even those with no personal stake in the Enron collapse that "justice" be done, some scientists see a vivid example of humanity's evolved and deep-seated hatred of the Cheat. The Cheat is the transgressor of fair play, the violator of accepted norms, the sneak who smiles with Chiclet teeth while ladling from the community till.

Human beings are elaborately, ineluctably social creatures, scientists say, and are more willing than any other species to work for the common good — to cooperate with nonkin and to help out strangers, sometimes at great cost to oneself, as the death of hundreds of rescue workers at the World Trade Center only too sadly showed.

Such a readiness to trust others, to behave civilly in a crowd, to share and empathize, to play the occasional Samaritan — all the behaviors that we laud and endorse and vow to cultivate more fully in ourselves — could not have evolved without a corresponding readiness to catch, and to punish, the Cheat.

Only recently have researchers realized that a willingness, even eagerness, to punish transgressors of the social compact is at least as important to the maintenance of social harmony as are regular displays of common human decency. And

In the reaction to Enron, a vivid example of a deep human need.

while the punitive urge may seem like a lowly and unsavory impulse, scientists point out that the effort to penalize cheaters is very often a selfless act.

In an article titled "Altruistic Punishment in Humans," which appears in the Jan. 10 issue of the journal Nature, Dr. Ernst Fehr of the University of Zurich and Dr. Simon Gachter of the University of St. Gallen in Switzerland offer evidence that people will seek to punish a cheat even when the punishment is costly to them and offers no material benefit — the very definition of altruism. The researchers propose that the threat of such pun-

Continued on Page 6

Urge to Punish Cheats Isn't Merely Vengeance

Continued From First Science Page

ishment may have been crucial to the evolution of human civilization and all its concomitant achievements.

"It's a very important force for establishing large-scale cooperation," Dr. Fehr said in a telephone interview. "Every citizen is a little policeman in a sense. There are so many social norms that we follow almost unconsciously, and they are enforced by the moral outrage we expect if we were to violate them."

Dr. David Sloan Wilson, an evolutionary biologist at the State University of New York at Binghamton, said, "People are used to thinking of social control and moralistic aggression as forms of selfishness, and that you must be punishing someone for your own benefit. But if you look at the sort of punishment that promotes altruistic behavior, you see that it is itself a form of altruism."

In their new work, Dr. Fehr and Dr. Gachter put 240 students through a series of "public goods" experiments with real monetary stakes, always a good incentive for cash-strapped young scholars.

Each participant was given an initial lump sum of 20 "monetary units" and allowed to play a series of games with rotating groups of three other participants. By the rules of the game, the members of each group independently decided how much of their sum to contribute to a community project, which in turn determined how much would be divvied up to participants in the end. The more generous each contributor, the better the group did as a whole, but there was always the risk of a participant's trying to freeload off the contributions of others.

From one round to the next, students were kept apprised of the investment decisions by others in their group. In some cases, there was nothing they could do about their teammates' behavior. In other cases, though, participants were allowed to "punish" freeloaders and skinflints after the round was through: one monetary unit from them would cost the shirker three monetary units. Hence, cooperators had to pay out of their own pocket to express their disgust at another's selfish behavior.

The outcome of the study was striking on two fronts. One was the popularity of punishment when it was permitted: 84 percent punished defectors at least once, 34.3 percent took punitive action five times or more and almost 10 percent punished the stingy 10 times or more. And all this, remember, involved the doling out of mad money from people who really needed it.

The second significant result was that when the game was carried out under no-punishment conditions, cooperation among group members quickly broke down, and participants contributed progressively less to the public kitty as the rounds went on. But when the opportunity to punish and be punished was applied, individual contributions to the collective fund jumped sharply, and cooperation among group members grew stronger rather than weaker from round to round.

The researchers also asked participants to describe their feelings toward free-riders on a seven-point scale, from "no big deal" to "very angry," and about 84 percent ranked themselves a five or higher. A sense of emotional outrage is very easily evoked, said Dr. Fehr, and sometimes it feels almost good to indulge and stoke it.

Perhaps part of the reason it feels good to rail against the sinner is that not to do so seems irresponsible, if not cowardly. "Once you think of punishment as a form of altruism, then the kind of person who doesn't punish emerges as a kind of freeloader too," said Dr. Wilson, author with Dr. Elliott Sober of "Unto Others: The Evolution and Psychology of Unselfish Behavior."

The emotional palette behind the effectiveness of social control is a rich one, composed not only of a sharp sense of moral indignation and a fear of being punished, but embarrassment and shame when one violates social norms.

> Moral outrage may have started with the hunter-gatherers, as a survival skill.

Dr. Wilson said that when he and his children, nonbowlers all, recently went bowling, they were mortified when others gently scolded them for failing to observe common bowling etiquette, like taking turns with bowlers in neighboring lanes. "My ears were burning with shame, and we fled as soon as we could," he said.

And sometimes the severity of the emotion far outstrips that of the transgression. Dr. Fehr cited a case during the oil crisis of the 1970's that led to long waits at gas stations, when one motorist shot another to death for attempting to butt into line. Some of the most odious of human behaviors, including torture, public stonings and lynchings, may all be examples of the meting out of altruistic punishment run amok.

The drive to punish selfish transgressors seems to be a basic human predilection. Paradoxically, it stems from something normally associated with rosy-eyed utopianism: according to most anthropological evidence, traditional hunter-gatherer societies have always been highly egalitarian.

In such cultures, there are no kings or commanders, and the bounty of a good hunt or forage is generally shared with the entire community. If one person doesn't like or trust another, the person may walk away, or articulate that distrust with the tip of a spear.

"Hunter-gatherer societies are scrupulously egalitarian, but not harmoniously so," said Dr. Herbert Gintis of the University of Massachusetts, a co-author on a commentary that appears with the current Nature research report. "They're violently egalitarian."

Despite its antiquity, the strength and expression of the urge to scourge is clearly shaped by culture. Anthropological studies by Dr. Fehr, Dr. Gintis and others have shown considerable cross-cultural variation in the ardor with which people seek to punish shifty noncooperators. As a rule, said Dr. Fehr, the more closely a society's economy is based on market rather than kinship ties, the more prevalent the use of altruistic punishment to bring others into line.

In other words, the more likely a person is to be negotiating with nonrelatives, and hence the higher the chances that selfish freeloaders will seek to infiltrate the system, the more important it becomes that everybody play by the rules. Or else.

Other points of view
on the Op-Ed page
seven days a week.
The New York Times

A Question of Rules

Ask not

JOE BOB BRIGGS

I was at a seminar not long ago where a famous teacher of screenwriting threatened to kick a woman out of the room. Her sin: She blurted out a question in the middle of his lecture, interrupting his concentration.

Here are his precise words: "If you speak out like that again, I'll ask you to leave."

(He'd already announced that there would be no questions, except when he asked for them.)

A silence descended over the room. The woman didn't make another sound.

The reason I'm telling this story is not so much because I care about his "no questions" rule — it's his class, so he can run it however he wants to — or about the woman who violated the rule. I'm telling the story because of the *reaction* to the incident:

Half the class was *outraged*. At the next break, several people sent the teacher notes, telling him how "offended" they were, and how they thought it was "inappropriate" to humiliate that woman by threatening to have her removed. When the guy started teaching again, he was obviously rattled by the ferocity of the response — but he refused to apologize. "I'm sorry if you were offended by that," he said, "but I was offended by her interruption."

Second example: Another seminar, another teacher (also semi-famous), at another time, announced that he would take questions, but he would *not* take comments. A lot of teachers do this, to weed out the people who like to hear themselves talk. A student rose, was recognized, asked a question, got his answer. The student then said, "Can I comment on that?"

The teacher said, "No, you may not," and went on to the next question.

Later the guy was recognized again. "Could I just say something?"

The teacher said, "You may ask a question. That's all."

"If I could just give my opinion of that . . . "

"I don't *want* your opinion of that. Do you have a question?"

"I understand where you're coming from, but I . . . "

"Next question."

And he cut him off and refused to recognize him again.

And the class, once again, was *furious*.

Now, here's my point. In both of these examples, all the teacher was doing was announcing rules, and then enforcing them. But there was something about it that made people *extremely* uncomfortable. What they really wanted the guy to do was announce rules but then say, "Oh, OK, I guess we can break the rule this one time." And then, when he didn't, they felt like something was wrong with the universe.

Both teachers were over the age of 60, by the way. And I've noticed that people of that generation find it much easier to simply say what they mean, and then do what they say. Why should this make anybody mad?

I think it makes people mad because they're afraid that, if there really are such things as rules, and the rules really can't be broken, and the guy enforcing the rules can't be manipulated into being "nice," then the whole basis of their lives is threatened.

They hate authority. They hate being the student. I suppose, in their heart of hearts, they think *they* should be the teacher, even though they don't know anything the teacher is talking about.

In Russia and China and certain universities in Europe, they have courses of study where the student doesn't even speak *at all* for the first four years — because they believe it takes that long for the student to learn how to ask an intelligent question. No wonder American education doesn't work. ■

People over the age of 60 find it much easier to simply say what they mean, and then do what they say

TO: EECS FACULTY

FR: EECS Undergraduate Study Committee

GRADING GUIDELINES FOR UNDERGRADUATE COURSES*

New instructors in EECS classes have commented that they do not know what grading policy exists in the department. In some courses the assigned grades show wide fluctuation from section to section, and/or semester, depending on who teaches the section. Furthermore, over the last ten years, the lower and upper division EECS average GPA's have inflated considerably.

The undergraduate study curriculum committee suggests that there should be some uniformity in grading in fairness to our students, and we propose the following guidelines, which were ratified at the faculty meeting of March 11, 1976, and updated in 1989.

1. a. A typical GPA for courses in the lower division is 2.7. This GPA would result, for example, from 17% A's 20% B's, 10% D's, and 3% F's. A class whose GPA falls outside the range 2.5 - 2.9 should be considered atypical. (A Typical GPA for basic prerequisite lower division CS courses (CS 40, CS 41) is 2.5, with GPA's outside the range 2.3 - 2.7 considered atypical.)

   b. A typical GPA for courses in the upper division is 2.9. (This GPA would result, for example, from 23% A's 50% B's , 20% C's, 5% D's and 2% F's.) A class whose GPA falls outside the range 2.7 - 3.1 should be considered atypical. A typical GPA for basic prerequisite upper division courses (EECS 104A, EECS 105, CS 150, CS 153) is 2.7 with GPA's outside the range 2.5 - 2.9 considered atypical.

   These guidelines do not represent a major shift down from current GPA levels, but rather they are intended to prevent inflation.

2. At the end of every semester, please fill out the attached form which will be used to update current departmental grading statistics. The form provides the current GPA typical for each category.

3. Each fall, we will distribute a list of all EECS courses (with no instructor's names) together with their GPA's for the past three quarters.

4. Since some graduate students enroll in upper division undergraduate courses, care should be taken that their performance does not influence the grading of the undergraduate students. The undergraduate students as a **separate** group should first be assigned grades according to the guidelines in 1(b) above. Then the graduate students should be assigned grades using the same boundaries between grades as were used for the undergraduate population. (This technique properly evaluates graduate students against an undergraduate norm in an undergraduate course without skewing the norm because of their presence. The two groups of grades should be recombined for the course report.

5. In addition, there has recently been some concern about inadequate preparation of some students for some courses. The Undergraduate Study Committee was asked to make a preliminary recommendation to faculty regarding this problem. The recommendation is as follows:

   The Undergraduate Study Committee does not believe that grading standards should be lowered even if a larger than usual fraction of students in a course appear to deserve low grades.

---

* Undergoing revision: Fall 1989
231cory@hera/d.admin/undergrad.guidelines

That's Not Computing—*This* Is Computing

Bob Colwell

Pull up that rocking chair and sit a spell. We're going to visit those golden days of yore, when woolly mammoths roamed the earth. No, wait, that's too far back. Let's just go back to the early 1970s, when your relationship with your computer was much more intimate than today, "network" meant your dad's business associates, and "browser" was only useful as a name for your dog.

We've all become blasé about our computing: computers in our pockets, in our washing machines, controlling our car engines, in our greeting cards, in our ski boots and tennis shoes. The giddy sense of amazement, of possibility, has gradually given way to complacency and a vague sense of unfulfilled entitlement. (Well, okay, except for Google. I still think search engines are absolutely amazing.)

To see how we've gotten here, and what we now have, we need to step back from today's reality. You can't get a clear perspective on the reality you're embedded in—you have to find another vantage point. You can't appreciate where you've gotten to unless you know where you came from. Our own computing history is a good place to start.

In the movie *Crocodile Dundee*, there's a famous scene in which a would-be robber confronts Mick Dundee and his girlfriend on a city street. His girlfriend says, "Mick, give him your wallet, he's got a *knife*!" To which Mick replies, "*That's* not a knife—*this* is a knife." Then he whips out a hunting knife with the form factor of a giant machete. The memorable thing about that scene is not just the size of the knife, it's the wide smile and extreme good humor with which Mick delivers the line. Well, in that same vein, today's stuff isn't computing. *This* is computing....

You can't appreciate where you've gotten to unless you know where you came from.

NO CRTS, NO LCDS, NO DISPLAY SCREENS

Initially, computing was paper in, paper out: punched cards in through a card reader—with a probability considerably less than 1.0 of getting all cards read properly—and line printer paper out, as long as the bored student user-abuser didn't misfile your output in some place he knew you'd never look.

People didn't "do computing" with such systems, nor did they "use the computer." In those days, computers were multimillion-dollar monsters, and both those who ran them and the professors who taught programming clearly held them in awe. Users meekly submitted their compute job to The Computer, and at some random time in the future, if the proper obeisances had been made and the wind was at your back, The Computer might deign to run your job and print out page after page of errors on your line printer paper.

Your job was then to slink off to a dark corner and write all over the output with red ink until you were ready to punch new cards and try again. Repeat until computer room closes at 1:00 a.m. Go home exhausted, but thrilled with the possibilities.

THE POSSIBILITIES THEY DIDN'T TELL YOU ABOUT

Very early on, our overactive undergraduate imaginations led to new uses for computer-punched cards. These cards were 7-3/8 inches wide by 3-1/4 inches high by .007 inches thick. After you creased these cards in the middle, you could use them to build houses of cards that reached to the ceiling outside someone's dorm room door or in front of the elevator. Not that *I* ever would have stooped to doing that.

I was also taking electronic music courses as an undergrad, and I had heard that computer line printers generated a lot of electrical noise in the AM radio spectrum. If you held an AM radio near the printer and printed just the right pattern on the paper, you

could get the line printer to, in effect, play music on the radio.

We wrote an ASCII text file and printed it while listening to the radio. With enough iterations of that development loop, we soon had the first half of "Yankee Doodle Dandy" playing. We never did get the second part going, though, because the computer room attendant seemed to lack our peculiar combination of (a) a love of music (the choice of "Yankee Doodle Dandy" notwithstanding), (b) too much idle time on our hands, and (c) complete disregard for the amount of useful paper we were converting to a nonuseful state. It's a wonder that science ever progresses at all with such nonvisionaries at every turn.

Back in the 1970s, a 10-Mbyte (yes, I do mean megabyte) drive was enormous and expensive. When students signed up for college courses that required computing access, they were given a computer account. This account consisted of a userID, a password, and a pitifully small amount of disk space. Now, it's true that we needed considerably less disk space then than today, partly because the field as a whole was less ambitious about how the computer was to be used, and mostly because we were less profligate about how we used it.

The computer systems people had to allocate disk space so that there would be enough for everyone to get their classwork done. But we learned a few things we weren't necessarily supposed to notice, and after that our actions were inevitable.

First, we discovered that the disk usage check was done at logout time—you could use more disk space while logged in than you could keep after you had logged out. Second, we noticed that if we set the file permissions properly, we could reach files from other accounts.

So we pooled our resources: One student would sacrifice his account to be the repository, and promise never to log in. In exchange, he would share another student's userID. And all of us would dump our files to the repository before logging out. I never did understand why the computer systems administrators didn't stop us from doing this, since any cursory check of disk usage would have led them straight to the repository account.

We were supposed to keep large files—say, larger than a few hundred thousand bytes—by putting them on magnetic tape. But the computer room personnel didn't want random students mounting or dismounting mag tapes. Instead, we would post a request on our terminal asking the personnel to mount a tape on a particular drive, then, anywhere from 10 to never minutes later, we could reach the tape at our command line.

In the 1970s, computer workstations had the computing horsepower of today's CD audio players.

TYPHOID HARRY AND BUMBLER BOB

In the 1970s, computer workstations, in the form of the Xerox Alto, first became available. Today's computer users would have felt reasonably at home with these, except that they had the computing horsepower of today's CD audio players, and the hard drives were removable.

On one memorable afternoon, we found that several of these machines had become unusable. After investigating, we noticed that one student was going from machine to machine, inserting his disk, mumbling that "this machine doesn't work either," and moving to the next one. On a hunch, we looked at his disk—it didn't look right. In fact, it didn't even look *planar*. The disk was damaging the drive's head on every machine into which he inserted it, so that the machines not only couldn't read his disk, they could no longer read anyone else's either.

In the late 1970s, a new OS was becoming popular on college campuses: Unix. After having learned my online computing on DEC machinery (Does anyone remember "DECSYSTEM 20 NOT RUNNING"?), I was intrigued with this operating system. Wow, it had hierarchical directory/file structures!

By that time in my college career, I considered myself reasonably well prepared as a computer user, so I sat down to check out this new OS. I logged in okay, but I was at a loss after that. "Dir" didn't work. "Type" didn't work. Even "Help" didn't work.

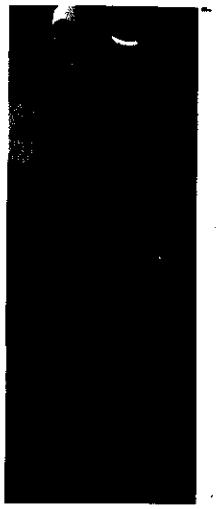
Eventually, I figured out that you didn't "type" in Unix, you "cat"-ed. After spending a few minutes cat-ing files, I proved that a little knowledge was a dangerous thing—I learned that it was possible to cat several files with one cat command. In fact, it was possible to concatenate two files into a third file.

Immediately, a question popped up: If you cat-ed a file onto itself, would you get a new file, twice as long as the first? Would Unix refuse to do that? Surely it wouldn't append the file to itself, notice that it hadn't finished copying the requested file (since it had just become twice as long), and keep going until the disk was full?

Clearly, this question demanded an answer, and there was an easy way to check. However, when I tried it, I didn't get a command prompt back; the system seemed to be frozen. In fact, the users sitting around me were now leaning back in their chairs, with looks of exasperation on their faces. I thought to myself, just my luck that this new OS would crash just as I was about to learn something interesting.

A few minutes later, the system was rebooted. Remember, this was early Unix; when it died with files open, those files were not likely to be in good shape upon rebooting. But I easily recreated my test file and tried it again. Rats, the system again chose just that moment to die.

Then a horrible suspicion dawned in my head: Maybe Unix was *not* going



King Carl Gustaf of Sweden, right, presented the N to Jack S. Kilby in 2000. Mr. Kilby claimed more tl "Crystal Fire: The Invention of the Transistor and the Birth of the Information Age" (W. W. Norton & Company, 1998), noted that Globe Union and Texas Instruments were both pioneers in miniaturization, and that Mr. Kilby "came to T.I. with a drive to make things small."

Mr. Kilby had also been sent by Globe Union to attend an early workshop held by the Bell Laboratories of A.T.& T. to familiarize the technical world with the transistor in the early

Dr. Moore remembered Mr. Kilby as a tall — he was 6-foot-6 — and gentle man with whom he would occa- sionally socialize while attending

Texas Instruments, via Reuters

Jack S. Kilby created this, his first

companies' applications for patents for the integrated circuit. After years of legal battles, Fairchild and Texas Instruments decided to cross-license their technologies, ultimately creating a world information industries market now worth more than $1 trillion annually. Dr. Noyce died in 1990.

worked for a quarter-century.

The integrated circuit that Mr. Kilby designed shortly after arriving at Texas Instruments in 1958 served as the basis for modern microelectronics, transforming a technology that permitted the simultaneous manufacturing of a mere handful of transistors into a chip industry that routinely places billions of Lilliputian switches in the area of a fingernail.

His achievement — the integration

How to Reach *Computer*

Writers

We welcome submissions. For detailed information, write for a Contributors' Guide (computer@computer.org) or visit our Web site: http://computer.org/computer/.

News Ideas

Contact Lee Garber at lgarber@computer.org with ideas for news features or news briefs.

Products and Books

Send product announcements to products@computer.org. Contact computer-ma@computer.org with book announcements.

Letters to the Editor

Please provide an e-mail address or daytime phone number with your letter. Send letters to Letters, *Computer*, 10662 Los Vaqueros Cir., PO Box 3014, Los Alamitos, CA 90720-1314; fax +1 714 821 4010; computer@computer.org.

On the Web

Visit http://computer.org for information about joining and getting involved with the Society and *Computer*.

Magazine Change of Address

Send change-of-address requests for magazine subscriptions to address.change@ieee.org. Make sure to specify *Computer*.

Missing or Damaged Copies

If you are missing an issue or received a damaged copy, contact membership@computer.org.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at whagen@ieee.org. To buy a reprint, send a query to computer@computer.org or a fax to +1 714 821 4010.

Computer

to save me from my own naiveté. Maybe the system really would let something as innocuous as "cat a > a" crash it, something that any user could type at any time. Maybe if these people sitting around me realized that I just cost them all about 20 minutes—or worse, if they had important files open—they might decide that a little vengeance was a good learning experience for me.

I decided discretion was the better part of valor. I had learned enough Unix for one day, and exited the premises with unseemly haste.

PDP-11 STUPID PET TRICKS

For my master's project at CMU, Paul Rodman and I created the world's second digital synthesizer. (We think Hal Alles at Bell Labs beat us by a few months.) To drive the pile of hardware we created, we used a PDP-11/45 minicomputer. In those days, it was rare to actually own one of these machines. Instead, we shared the machine with as many other projects as necessary to justify its purchase.

As we entered the lab one day, the team of physics students that we shared the PDP-11 with informed us that it was down and would be down for several weeks until the service person came. We were hardware guys. To us, "down" meant "figure out what's wrong with this thing and get back to the music project."

When we tried booting it, it hung, and the final program counter address it had reached was now displayed on the front panel lights. From the documentation, Paul deduced that the machine had detected a parity error on a memory location during boot, which trapped to the location we could see on the lights. But there was no handler there, so the boot procedure was stuck.

By toggling the front panel switches and watching the lights, we also deduced that the parity error was, in fact, the parity bit at the memory location in question. The actual memory itself was still working and usable. So we looked up the "return from inter-

rupt" that in reboo we we the PI On again RTI h hardw about Aft such a note i hard c the he We severa in a c bly co batter so tha off th in cas electr above on a c ning t the su rows this n nails And t are n ing a To eral b on t attac In a batte side

I w fo e Dunc

Bob archi and indet at bo

Kilby, an Inventor of the Microchip, Is Dead at 81

Mark Earthy/Scanpix, via Reuters

King Carl Gustaf of Sweden, right, presented the Nobel Prize in Physics to Jack S. Kilby in 2000. Mr. Kilby claimed more than 60 patents.

A Nobel winner who loomed large when it came to making things ever smaller.

:OFF
ctrical engi-
the integrat-
the informa-
explosion of
oducts in the
onal comput-
Monday in

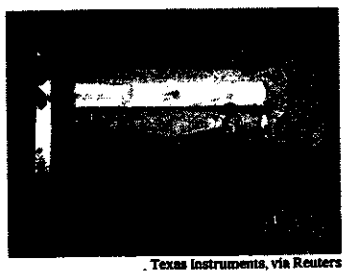
ef battle with
yesterday by
Dallas-based
where he
ntury.
that Mr. Kil-
er arriving at
958 served as
icroelectron-
hnology that
neous manu-
ndful of tran-
stry that rou-
of Lilliputian
fingernail.
he integration
f crystal con-
arate compo-
resistors and
ingle device.
monly called
awarded the
in 2000.
Texas Instru-
e than 60 pat-
ne of the in-
eld calculator
er. But it was
f the integrat-
roadly shaped

lace where the
n't affect your
K. Templeton,
president and
, said in an in-

terview yesterday. "That's how broad its impact is."

It is an impact, Mr. Kilby said, that was largely unexpected. "We expected to reduce the cost of electronics, but I don't think anybody was thinking in terms of factors of a million," he said in an undated interview cited by Texas Instruments.

The remarkable acceleration of the manufacturing process based on the integrated circuit was later described by Gordon E. Moore, cofounder of the Intel Corporation, whose partner, Robert N. Noyce, invented another version of the integrated circuit just months after Mr. Kilby.



. Texas Instruments, via Reuters

Jack S. Kilby created this, his first integrated circuit, in 1958.

In 1965, three years after the first commercial integrated circuits came to market, Dr. Moore observed that the number of transistors on a circuit was doubling at regular intervals and would do so far into the future. The observation, which came to be known as Moore's law, became the defining attribute of the chip-making industry, centered in what is now known as Silicon Valley, where Intel was based, rather than in Dallas.

That was partly because Dr. Noyce's version of the integrated circuit, using silicon and based on a photolithographic printing technology known as the planar process, was easier to manufacture than Mr. Kilby's original invention, which employed germanium and used individual wires.

In 1959 Mr. Kilby and Dr. Noyce, then with Fairchild Semiconductor, were named as inventors in their companies' applications for patents for the integrated circuit. After years of legal battles, Fairchild and Texas Instruments decided to cross-license their technologies, ultimately creating a world information industries market now worth more than $1 trillion annually. Dr. Noyce died in 1990.

Dr. Moore remembered Mr. Kilby as a tall — he was 6-foot-6 — and gentle man with whom he would occasionally socialize while attending technical meetings.

"He was mild mannered," Dr. Moore recalled in a telephone interview yesterday, "but I would never worry when I was walking down the street with him in New York City."

Mr. Kilby's contribution came in an era when manufacturing industries were hunting for new approaches to miniaturization for reasons of both cost and performance. It was a drive that began during World War II and pushed beyond military uses into consumer products in the postwar era.

He began his career in 1947 with the Centralab division of Globe Union Inc. in Milwaukee, developing ceramic-based silk-screen circuits for consumer electronic products.

Michael Riordan, co-author of "Crystal Fire: The Invention of the Transistor and the Birth of the Information Age" (W. W. Norton & Company, 1998), noted that Globe Union and Texas Instruments were both pioneers in miniaturization, and that Mr. Kilby "came to T.I. with a drive to make things small."

Mr. Kilby had also been sent by Globe Union to attend an early workshop held by the Bell Laboratories of A.T.& T. to familiarize the technical world with the transistor in the early 1950's. It was Mr. Kilby who first pulled the idea of miniaturization together with the transistor.

A lifelong optimist who rarely showed signs of anger, according to his daughter, Janet Kilby Cameron, Mr. Kilby took his Nobel Prize in stride. When asked what he did after learning of the award, he said simply, "I made coffee."

Jack St. Clair Kilby was born in Jefferson City, Mo., on Nov. 8, 1923, to Hubert and Vina Kilby. He grew up in Great Bend, Kan., and was exposed early on to the world of engineers: his father ran the local electric utility.

He decided in high school that he would become an electrical engineer and applied to M.I.T., even then the mecca for aspiring engineers. He took a train to Cambridge, Mass., but fell slightly short in his score on the entrance exam in June 1941 and was unable to enroll. A few months later he joined the Army and was assigned to a radio repair shop at an outpost on a tea plantation in northeast India.

After the war he attended college on the G.I. Bill of Rights. After receiving a bachelor's degree in electrical engineering from the University of Illinois and a master's from the University of Wisconsin, he went to work for Globe Union.

He arrived at Texas Instruments in 1958 and during his first summer, working with borrowed equipment, improvised a working integrated circuit. A successful laboratory demonstration of the first simple microchip took place on Sept. 12, 1958. He formally retired from the company in 1983 but continued his association as a consultant.

His other awards included both the National Medal of Science and the National Medal of Technology, the highest technical awards given by the United States government.

His wife, Barbara Annegers Kilby, died in 1982. In addition to Ms. Cameron, of Palisade, Colo., Mr. Kilby is survived by another daughter, Ann Kilby, of Austin, Tex., and five granddaughters.

Remembrances / *William C. Norris, Computing Innovator (1911-2006)*

Super Computer: Tech Guru Tackled Social Ills, Too

Control Data Founder Took A Non-Wall Street Approach

By Don Clark
And Stephen Miller

WILLIAM C. NORRIS built computers that helped address some of the world's toughest scientific problems. He also used his company to take on social problems, an idea that inspired admirers but irritated Wall Street.

Mr. Norris founded Control Data Corp., whose massive machines in the 1960s were more powerful than those of mighty International Business Machines Corp. He also pioneered computer services in areas such as education decades before the Internet made that commonplace. The company employed 60,000 by 1984, and helped turn the Minneapolis/St. Paul area into a technology hub.

A Nebraska farm boy who lived through the Depression, Mr. Norris's firsthand knowledge of hardship helped shape his belief that Control Data should help people the government couldn't. He located plants in inner-city neighborhoods where unemployment was rife. He built training centers for teachers and engineers. Other ventures tried to help farmers, convicts and entrepreneurs.

"I never met a corporate executive with a broader range of insights into the deficiencies of our political economy," consumer activist Ralph Nader says of Mr. Norris, who died Aug. 21 at age 95.

The vision fit the spirit of the 1970s, when Control Data was growing briskly. By the mid-1980s, however, the company was beset by competition from Japan and other problems that led to the sale or closing of several businesses. The Control Data name would eventually die, though some of its computer services live on through Ceridian Corp., specializing in human-resources applications.

When his business faltered, Mr. Norris's social ventures came under stiff criticism from the financial community as a waste of time and money. "The external impact was extremely negative," recalls Robert Price, the longtime No. 2 man at Control Data who became CEO upon Mr. Norris's retirement in 1986.

Never a fan of Wall Street, Mr. Norris was unapologetic. "If people didn't scoff, I'd know immediately I was on the wrong track," he said in a 1980 interview, referring to criticism of a company-supported, small-scale farming venture.

Mr. Norris was educated in a one-room schoolhouse and later the University of Nebraska. He was introduced to calculating machines in World War II, helping the U.S. Navy decode enemy communications. In 1946, Mr. Norris and other veterans set up shop in St. Paul as Engineering Research Associates. In 1951, ERA was purchased by Remington-Rand Corp., whose Univac brand became synonymous with computers of the day. Sperry Corp. later purchased Remington-Rand.

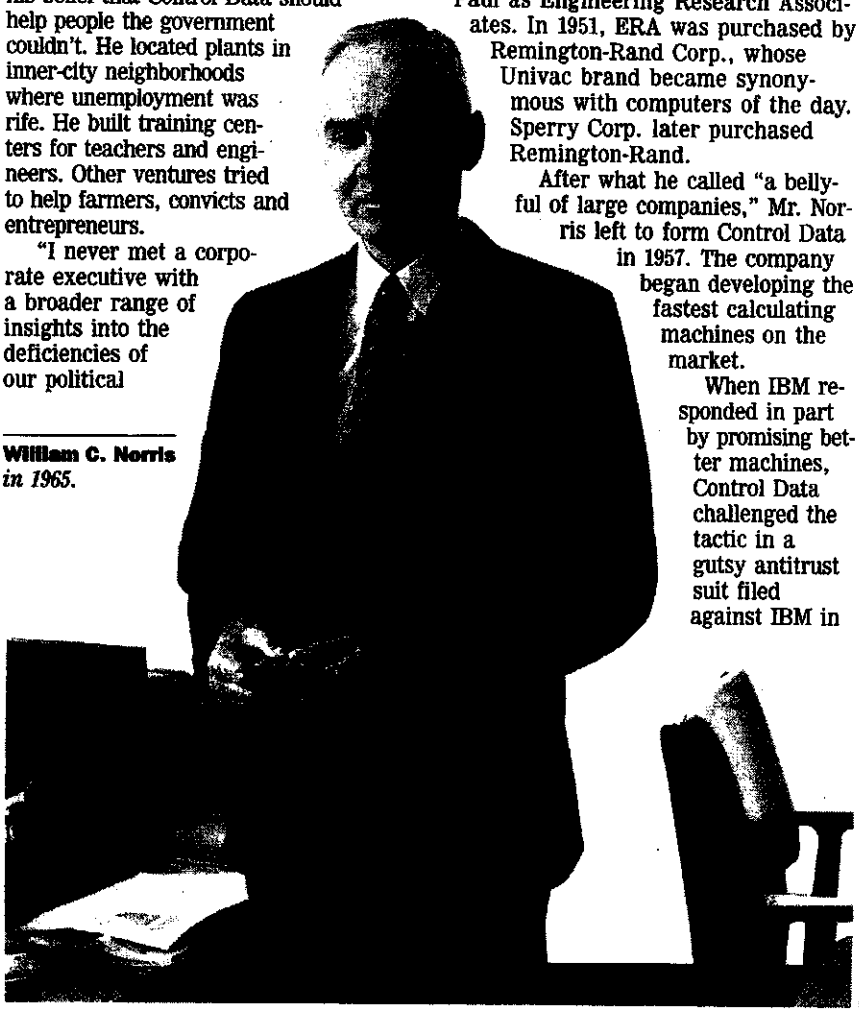
After what he called "a bellyful of large companies," Mr. Norris left to form Control Data in 1957. The company began developing the fastest calculating machines on the market.

When IBM responded in part by promising better machines, Control Data challenged the tactic in a gutsy antitrust suit filed against IBM in 1968. Mr. Norris emerged the victor in a 1973 settlement. But Big Blue would be the winner in large commercial computers, while Control Data in the 1980s was relegated to a high-end niche.

Though Control Data is sometimes described as a "big-iron" dinosaur doomed by the advent of personal computers, Mr. Price notes that Mr. Norris anticipated such pressures and pushed the company into computer services. "He was way ahead of his time in understanding that computer hardware was going to become commoditized," Mr. Price says.

His ideas that generated the most headlines had a social or educational bent. Among them, Control Data Institutes, a network of vocational schools, was set up around the U.S. and other countries. That effort dovetailed with what he called his greatest contribution—Plato, a computerized education system that began in 1963. Though never a money-maker for Mr. Norris, Plato is widely seen as one of the first online communities, where users could exchange messages and play games as well as learn.

After 29 years at Control Data, he founded the William C. Norris Institute, now at the University of St. Thomas in St. Paul, Minn., which operates a seed-capital fund focusing on socially beneficial products and services. Despite his technical innovations, Mr. Norris lived simply, eschewing even computers. He agreed to move from a modest St. Paul home only after insisting on energy-saving features, including a windmill to heat the swimming pool.



William C. Norris *in 1965.*

The William C. Norris Institute

■ Robert Hoffman, a founder of National Lampoon who became a Dallas philanthropist and civic leader. One of three Harvard Lampoon editors who started the irreverent humor magazine in 1969, Mr. Hoffman sold National Lampoon in 1975 and joined his father's Texas bottling firm, which became Coca-Cola Bottling Group (Southwest) Inc. He also led the city's 30-year improvement blueprint adopted in 1994 and last year donated artworks valued at more than $150 million, to the Dallas Museum of Art. Died Aug. 20 at 59.

■ James T. "Red" Hudson, founder of Hudson Foods, a beef and poultry purveyor that was once one of the nation's largest publicly held food processors. In 1996, Hudson, Rogers, Ark., had $1.7 billion in sales, 12,500 employees and 19 plants in 11 states. In 1997, when E. coli contamination was detected in hamburger meat from the firm's Nebraska plant, a major supplier of Burger King, Hudson issued a 25-million pound recall, the nation's largest-ever beef recall. Shortly afterward, Hudson Foods was broken up and sold to rivals Tyson Foods Inc. and IBP Inc. The episode helped increase support for irradiating meat. Died Aug. 20 at age 81.

Email us at remembrances@wsj.com.

William C. Norris, 95, Founder Of an Early Rival to I.B.M.

By JOHN MARKOFF

William C. Norris, an electrical engineer who founded Control Data, the company that created the first supercomputers, died on Monday in Bloomington, Minn. He was 95.

His death followed a long battle with Parkinson's disease, a son, Roger, said.

After World War II, Mr. Norris helped found Engineering Research Associates in St. Paul, which built early digital computers, mostly for the Navy. In 1951, that company merged with another early computer maker, Sperry Rand. Mr. Norris led the Univac division of Sperry Rand until mid-1957, when, after chafing under the restrictions of a large company, he left to found Control Data.

Under his leadership, Control Data grew to become a fierce competitor of I.B.M., then the dominant computer maker. Control Data thrived by making machines that underpriced and outperformed I.B.M. products.

After starting out by making computing peripherals, under the design leadership of Seymour Cray, the company produced an early mainframe computer, the 1604 in 1959, and the next year released the CDC 160A, considered by many to be the first minicomputer, a multiuser machine smaller than a mainframe.

The minicomputer was used by many early computer researchers, including Douglas C. Engelbart, a pioneering computer scientist who invented the mouse as well as many related technologies that underlie the personal computer and the Internet of today.

In 1962 Mr. Cray became frustrated with the emerging bureaucracy at Control Data and demanded that Mr. Norris let him and his team depart for Mr. Cray's hometown, Chippewa Falls, Wis. There, Mr. Cray and a small team designed what would become the CDC 6600, generally recognized as the first supercomputer.

The 6600, which was capable of executing about one million instructions a second in 1964, drew the attention of I.B.M. In August of that year, as Control Data prepared to make its first 6600 delivery, I.B.M. preannounced a competitor, the 360/91 system, which it said would be more powerful and flexible than the 6600.

The larger company made delivery-date pledges and accepted orders, cutting into potential Control Data business. At one point Control Data was unable to book a single order for the 6600 for 18 months, becoming a victim of I.B.M.'s ghost computer strategy.

The strategy forced Control Data into a financial loss in 1966 and led Mr. Norris to complain to the Justice Department's antitrust division. In December 1968 Control Data filed an antitrust lawsuit against I.B.M., and a month later the Justice Department also sued. Although I.B.M. would lose the suit, the damage had been done, and people in the computer industry would come to speak of "I.B.M. and the seven dwarfs": Univac, Burroughs, Scientific Data Systems, Control Data, General Electric, RCA and Honeywell.

Control Data Corporation

William C. Norris, around 1985.

Mr. Norris remained the head of Control Data until retiring in 1986. The company, at one time one of Minnesota's biggest, went into a decline starting in the 1980's, when the personal computer industry came into its own. In 1992, part of Control Data was spun off to become the information services company Ceridian.

A passionate believer in education, Mr. Norris led Control Data in the 1960's in building an innovative computer-based education and training system named Plato.

William Charles Norris was born in Red Cloud, Neb., in 1911 with a twin sister, Willa, and grew up on his parents' cattle, hog and corn farm. Fascinated early by physics, he built a mail-order radio set and became a ham radio operator. In 1932 he graduated from the University of Nebraska and during World War II served in the Navy, getting involved with electronic systems that were forerunners of modern computers.

Besides his son Roger, Mr. Norris is survived by his wife of 61 years, Jane Malley Norris, and seven other children: William, George, Daniel, Brian, David, Constance Van Hoven and Mary Keck.

may be made in his memory to Cornell University or NFTY.

BEESON—Paul Bruce, MD. The Board of Directors of the American Geriatrics Society and its members deeply regret the loss of Paul B. Beeson, MD, one of the fathers of geriatrics. Renowned clinician, teacher, researcher and Editor Emeritus of the Journal of the American Geriatrics Society (JAGS). Dr. Beeson's legacy will have a lasting impact on the well-being of older adults.

BUCK—Ralph D., died in Manhattan on August 21, 2006. He died with grace, peacefully, and in no pain, with his wife Susan beside him. Mr. Buck was tax counsel and associate general counsel for Caltex Petroleum, a division of Chevron Texaco, before retiring in 1986. After graduating from Princeton University in 1943, he served as an officer with the US Army Mule-Drawn Artillery in the Phillipines during WWII. He graduated from Harvard Law School in 1949 and worked with Paul, Weiss, Rifkind, Wharton & Garrison for eight years before joining Caltex Petroleum. RDB, my best friend, my support, my ardent sailing companion, I love you, my husband and dear, dear man, and I will miss you all of my life. May the breeze always be fair. Contributions in Mr. Buck's remembrance to Geriatric Research Fund, c/o The Division of Geriatrics, Dept. of Medicine, New York University Medical Center, 550 First Ave., NY, NY 10016.

DEUTSCHER—Martha. New York Theatre Workshop mourns the passing of Martha Deutscher, former member of the Board of Trustees and beloved aunt of our friend Tony Kushner. Our condolences to Tony and his family.
The Artists, Board and Staff of
New York Theatre Workshop

FOX—Denny. Our beautiful friend, we are heartbroken. We will love you and miss you forever. We extend our heartfelt sympathy to Jay, Rebecca, Andrew, Erin and the entire family. You will always have our love and support.
Jane and Steve Goldstein

FOX—Martin H. , August 19, 2006, in Portland, ME. Born April 29, 1927 in Bronx, NY, he served in the US Navy from 1943-1947. After more than 30 years with Revlon, he retired as Senior Vice President of Purchasing in 1991. He is survived by his wife Jeanne, of Little Fox Island, Freeport, ME and Longboat Key, FL, two sons from a previous marriage and four grandchildren.

FOX—Martin. August 19, 2006. Our dear friend, one of a kind man. We will miss you very much.
Renee & Murray Nadel

FRANKEL—Jerome. The members and staff of J. Streicher & Co. LLC are deeply saddened by the loss of our dear friend and former partner, Jerry Frankel. We will greatly miss "The Doctors" charm, quick wit and warm sense of humor. Our deepest sympathies are extended to his family. Jud, Steve, John, Jimmy and all the J. Streicher & Co. LLC Family

FRIED—Saul. Beloved husband of Norma. Devoted father of Susanne and Eugene. Services today, August 23, 2006, 1pm, at "The Riverside", 76th Street and Amsterdam Avenue.

WSJ
11/22/96
P. B1

THE FRONT LINES

By Thomas Petzinger Jr.

Female Pioneers Fostered Practicality In Computer Industry

Rockville, Md.

GROWING UP near Philadelphia, Betty Holberton was left-handed, cross-eyed and mercilessly teased by her classmates. Her orchestra leader chided her for standing on the wrong side of the double bass. On her first day at the University of Pennsylvania, her math professor said she should be home raising children.

Jean Bartik never quite fit in, either. Growing up in the farm fields of Missouri, she could outpitch her brothers in softball. She studied Latin as a hobby. In college she took physics and trigonometry, usually the only girl in class.

Quirky and tenacious – the perfect attributes for becoming history's first computer programmers. Was their gender also a factor?

Last week, I shared the tale of how the two of them, joined by four other young women, programmed history's first general-purpose digital computer, the Eniac. But Ms. Holberton and Ms. Bartik did not stop there. Their continuing work – spanning nearly 40 years, in Ms. Holberton's case—helped to create the computer industry as we know it today.

They believed computers would flourish only if easy to program and operate. No matter how complicated computers are today, I have no doubt they would be even less accessible if not for the work of these pioneers. "I spent half the day trying to figure out what people needed in computers," Ms. Holberton says, "and the rest of the day trying to convince an engineer it was his idea."

Following the Army's unveiling of the Eniac in 1946, Betty Bartik joined a mathematician named Adele Goldstine in leading a team to revamp the machine as a "stored program" computer. This relieved programmers of the need to configure a new labyrinth of heavy cables for every equation the machine solved. The invention of internal programming is often attributed to the great mathematician John von Neumann, who published a paper on the breakthrough. Ms. Bartik and her team consulted regularly with Dr. von Neumann, but it was she who wrote the code.

A SHORT TIME later, in 1948, she and Ms. Holberton were reunited at a company called Eckert-Mauchly Computer Corp., established by the two principal creators of the Eniac. Eckert-Mauchly was creating a new machine called the Universal Automatic Computer, or Univac—the first general-purpose commercial computer, the device that would compile the 1950 census, predict Eisenhower's victory over Stevenson and ultimately revolutionize business.

While the engineers were absorbed in technical blandishments, Ms. Holberton brooded over "human engineering," her way of saying the machine should be user-friendly. She created an instruction code called C-10, enabling programmers to control the computer with typewritten commands instead of dials and switches. She built the language around mnemonic characters – "a" for add, "b" for "bring," introducing an intuitive aspect to the hypertechnical practice of programming.

The very look of the Univac embodied her thinking. In designing the control panel she put the numeric keypad alongside the keyboard so letters and numbers could be entered with equal ease. She insisted the intimidating black exteriors be abandoned, thus locking in gray as the color of choice in computing.

But to what use would the Univac be put? Ms. Holberton headed the development of a "sort generator," a program organizing payroll, inventory and other data according to the unique parameters of the user. It was probably the first use of a computer to program its own application.

David Smith

Betty Holberton (left) and Jean Bartik

Eckert-Mauchly was sold in 1950 to Remington Rand, which evolved into Sperry Univac and ultimately into the Unisys of today. When M.I.T. Press brought out a history of the history-making machine in 1987, it was titled "A Few Good Men from Univac."

Ms. Holberton remained in the thick of computing – though never in the spotlight – until 1983, first with the Navy and later the National Bureau of Standards. She knew the machines had to talk the same language if computing was to spread, so she led a committee to establish standards for the Common Business Oriented Language, or Cobol; it remains among the most widely used computer languages on Earth. She later played a similar role in the development of Fortran.

TODAY, AT 79, Ms. Holberton is partly paralyzed from a stroke but speeds around her nursing home here in a wheelchair. The one visible memento of her trailblazing years is the steel nameplate, hanging on her wall, from Univac serial number 006. Ms. Bartik, 71, supports herself selling real estate in New Jersey.

Programming started out as an exclusively female field. But as software began catching up to hardware in importance, men moved in and quickly moved ahead for good. Wherein may lie the deepest significance of the story.

The programming challenges of the future are too huge to permit the discouragement of half the population. Yet the proportion of women studying computer science has been declining for a decade. One reason may be the historical dearth of role models in significant positions. "Girls have an image of computer science being for boys," says Denise Gurer, an artificial-intelligence researcher at SRI International.

As for her own pioneering role, Ms. Holberton remains modest. "Somebody would have done it if I hadn't done it," she insists. Maybe so, but would someone else have done it quite the same way?

If operating systems were airlines...

DOS air: Passengers walk out onto the runway, grab hold of the plane, push it until it gets in the air, hop on, then jump off when it hits the ground. They grab the plane again, push it back into the air, hop on, jump off.....

Mac Airways: The cashiers, flight attendants and pilots all look the same, talk the same, and act the same. When you ask them questions about the flight, they reply that you don't need to know, and would you please return to your seat and watch the movie.

Windows Airlines: The terminal is neat and clean, the attendants courteous, the pilots capable. The fleet of Lear jets the carrier operates is immense. Your jet takes off without a hitch, pushes above the clouds and, at 20,000 feet, explodes without warning.

OS/2 Skyways: The terminal is almost empty - only a few prospective passengers mill about. The announcer says that a flight has just departed, although no planes appear to be on the runway. Airline personnel apologize profusely to customers in hushed voices, pointing from time to time to the sleek powerful jets outside. They tell each passenger how great the flight will be on these new jets and how much safer it will be than Windows Airlines, but they will have to wait a little longer for the technicians to finish the flight systems. Maybe until mid - 1995. Maybe longer.

Fly Windows NT: Passengers carry their seats out onto the tarmac and place them in the outline of a plane. They sit down, flap their arms, and make swooshing sounds as if they are flying.

UNIX Express: Passengers bring a piece of the airplane and a box of tools with them to the airport. They gather on the tarmac, arguing about what kind of plane they want to build. The passengers split into several groups and build several different aircraft but give them all the same name. Only some passengers reach their destination but ALL of them believe they arrived.

Early Computer Quotes

The following is from the business section of *The Kansas City Star*, January 17, 1995:

"Computers in the future may weigh no more than 1.5 tons."
– *Popular Mechanics*, forecasting the relentless march of science, 1949.

"I think there is a world market for maybe five computers."
– Thomas Watson, chairman of IBM, 1943.

"I have traveled the length and breadth of this country and talked with the best people, and I can assure you that data processing is a fad that won't last out the year."
– The editor in charge of business books for Prentice Hall, 1957.

"But what . . . is it good for?"
– Engineer at the Advanced Computing Systems Division of IBM, 1968, commenting on the microchip.

"There is no reason anyone would want a computer in their home."
– Ken Olson, president, chairman and founder of Digital Equipment Corp., 1977.

OS Alternatives

Forget Windows and Linux: The lesser-known BSDs may be just what you need. **By Brett Glass**

It's hard to browse the Web or read a computer magazine without finding a reference to Linux, the operating-system wunderkind created by Linus Torvalds and developed by a host of others. But although Linux steals the headlines, ISPs and system administrators often choose one of the BSDs—a group of operating systems based on code polished during more than 20 years of research at one of America's leading academic institutions.

What are the BSDs? And why should you consider using them if you're looking for a non-Windows operating system?

A SCHOLARLY TRADITION

BSD stands for *Berkeley Software Distribution*, a collection of software developed at the University of California at Berkeley. Originally intended as an add-on package for early versions of AT&T's Unix, BSD gradually evolved into a complete, highly sophisticated Unix-like operating system—the first to incorporate built-in networking.

Fussed over by perfectionist academics and then torture-tested by generations of brutal college students, BSD is perhaps the most robust, secure, and reliable of operating systems. It is common for a BSD-based operating system to run for years without maintenance or rebooting. And because it was developed at public expense, BSD was made completely free to all comers—including software developers who wanted to build it into commercial products. Networking code from BSD is at the heart of nearly every modern operating system, including Linux, OS/2, and even every version of Windows since Windows 95.

SECRET WEAPONS

If the BSDs are so good, why aren't they making the headlines, as Linux is? The most important reason is cultural. Many Linux developers see themselves as software revolutionaries. But members of the academic community, where the BSDs have their roots, tended to focus more on results rather than getting the word out. The BSDs also have a dedicated following among system administrators and ISPs, who often prefer to treat their favorite BSD as a secret weapon rather than publicizing the fact that they are using the OS.

As a result, not many people realize that versions of BSD form the foundations of major Internet sites such as Yahoo! and also power highly reliable embedded systems such as the IBM InterJet Internet appliance and Maxtor's network storage servers. One of the BSDs—NetBSD—has the distinction of being the most portable operating system in the world, running on more than 64 different central processing units and hundreds of brands and models of computers.

Today, there are five popular BSD operating systems. Three of them—FreeBSD (*www.freebsd.org*), NetBSD (*www.netbsd.org*), and OpenBSD (*www.openbsd.org*)—are covered by the BSD license, which makes the operating systems and their source code free for use by anyone for any purpose The other two—BSD/OS and Mac OS X—are commercial products that build upon the open-source BSD code and offer unique advantages and distinctive technology. In this article, we'll look at the no-cost BSD versions. For a discussion of BSD/OS and Mac OS X, and the advantage of the BSD license over the Linux General Public License, see the sidebar "More about BSD" on our Web site.

FREEBSD

FreeBSD evolved from 386BSD, a version of BSD ported to the Intel 80386 chip by Berkeley computer scientist Bill Jolitz. Because of its origins, FreeBSD has always worked best on the 32-bit *x86* architecture. Of the no-cost BSDs, this is the most finely tuned for PC-compatible computers and supports the greatest variety of PC-compatible peripherals. Although FreeBSD also runs on the Alpha processor, and ports to other architectures are in the works, you should consider this OS primarily for *x86* systems.

Of the no-cost BSDs, FreeBSD has the most bells and whistles, the largest cadre of developers, the most active collection of mailing lists, and the greatest number of users. It is also probably the easiest to install of all free operating systems—especially if you want to install across the Internet instead of buying a CD.

Linux distributors often make their distributions difficult to install across the Internet. Their businesses, after all, rely on sales of CDs. Not so with FreeBSD. All that's required is a pair of floppy disks (which can easily be created with utilities and image files on the FreeBSD Web site) and a reasonably fast Internet connection. Boot from the first disk, then insert the second. An installation program helps you choose configuration options, downloads the entire system from the Internet, and installs everything in the right place.

Virtual Device Driver Trouble

I'm running Windows 2000. When I start or install certain programs, I get a message saying "16 bit windows subsystem: SYSTEM\CurrentControlSet\Control\VirtualDeviceDrivers Virtual DeviceDriver format in the registry is invalid", and asking me to Close or Ignore. If I choose *Ignore*, everything seems to work properly. What could cause this error?

DIETER ABELE

This problem is caused by invalid Registry data. To fix it:

1. Launch REGEDT32 (you can enter the name in the Start menu's Run dialog). You must use Regedit32, *not* Regedit.
2. Open the `HKEY_LOCAL_MACHINE` subwindow.
3. Navigate to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\VirtualDeviceDrivers`.
4. In the right-hand pane, select the value named VDD.
5. Select *Delete* from the *Edit* menu.
6. Select *Add value* from the *Edit* menu.
7. In the *Add value* dialog, enter `VDD`, and choose `REG_MULTI_SZ` as the data type.
8. When the Multi-String Editor appears, click *OK* without entering anything.
9. Quit the Registry Editor.

For more information see the Microsoft Knowledge Base article Q254914, titled "Virtual Device Driver Error Message in 16-Bit MS-DOS Subsystem," at *http://support.microsoft.com/support/kb/articles/Q254/9/14.ASP.—Neil J. Rubenking*

Add Programs to *Open With* Dialog

Sometimes I want to use the Open With dialog box to open files that are not registered. Not all programs are listed in this box. How can I get a program to appear in the Open With dialog box?

RICH JOHNSON

Windows populates the program list in the Open With box by reading all of the file type information from the Registry and adding every program

USER

that's registered to open at least one type of file. The easiest way to force a program into the Open With list is to associate it with a nonsense extension. Create a file with a name like Getrid.nosuchextension, right-click on it in Windows Explorer, select *Open with...*, and browse for the program you want added. Check the *Always use this program to open this type of file* box and click *OK*. Now that a file type is associated with the program, the program will appear in the Open With dialog.—*NJR*

The Applog Folder

There are over 9MB in the C:\Windows\applog folder. What is this? Can it safely be deleted?

PHILIP S. JOHNSON

In Windows 98 and Windows Me, the Task Monitor tracks which programs are launched and how they load from disk, and it records the info in the Applog folder. The Defrag utility then uses this info to provide special optimization for the program files you use most often.

DESKTOP SOLUTIONS

Those who would rather have a CD, though, can get one from Wind River Systems' FreeBSD Mall (*www.freebsdmall.com*) or from Cheap Bytes (*www.cheapbytes.com*). You can also burn your own CD from an ISO image available on the FreeBSD Web site.

Another of FreeBSD's strengths is its extensive *ports collection*—more than 5,800 free applications you can instantly download and install on FreeBSD systems (*www.freebsd.org/ports/index.html*). Virtually all of the software you need to set up a desktop workstation or an industrial-strength server is either installed with FreeBSD or available via the ports collection.

In addition, FreeBSD, like OpenBSD and NetBSD, can run virtually any program that's sold for Linux, for SCO Unix, or for the Intel version of Solaris. Like Linux, FreeBSD uses the X Window system and all the desktops and graphical user interfaces developed for that protocol, including KDE, GNOME, and the programs written for both.

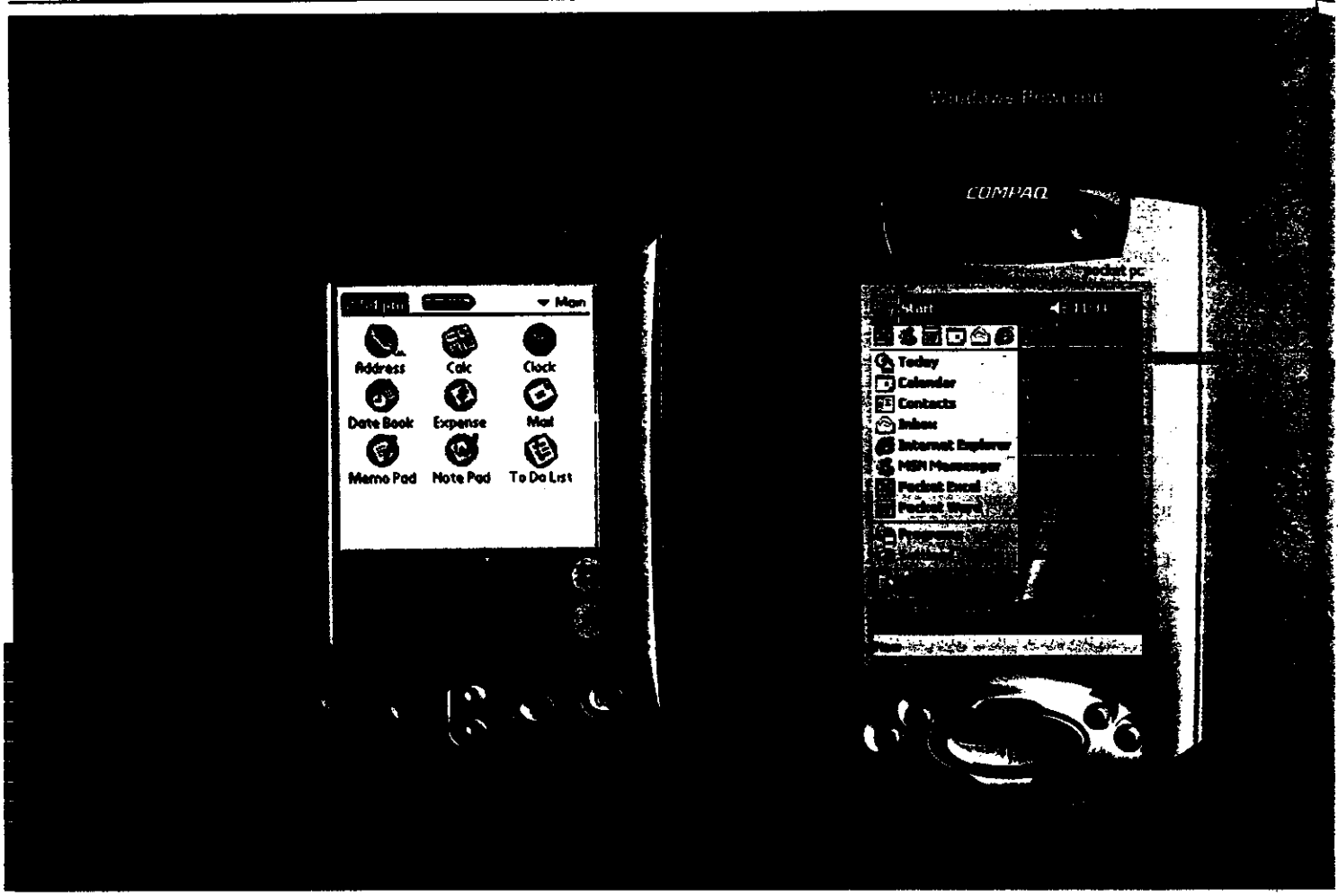
Finally, FreeBSD has the most literature for new users. The FreeBSD Handbook (*www.freebsd.org/handbook*) provides excellent instructions for newcomers. And books for beginners and more advanced users are arriving from many publishers.

In short, FreeBSD is a strong challenger to Linux—and may offer superior stability, ease of installation, and convenience.

OPENBSD

OpenBSD, another no-cost BSD derivative, is reputed to be the most secure operating system in the world. OpenBSD's home page (*www.openbsd.org*) reports that OpenBSD has gone "four years without a remote hole in the default install." What this means is that—as far as anyone knows—no version of OpenBSD released in the past four years can be taken over from across the Internet. (You can, of course, make any operating system—including OpenBSD—vulnerable to a takeover if you configure the server badly or run insecure software that lets intruders in.) OpenBSD also has fully integrated cryptographic software to keep data safe from prying eyes.

OpenBSD isn't the most feature-packed Unix implementation, nor is it the fastest. But it's no slouch in these areas. And the OS is so small and efficient, it can run well on an old 486 with 16MB of RAM—hardware that couldn't even boot Windows 2000. And no other operating system shares OpenBSD's sterling security track record. Several remote root vulnerabilities are typically discovered every month in Microsoft operating systems such as Windows 2000. And



many recent Linux distributions have been vulnerable to exploits such as the Ramen worm.

Like FreeBSD and NetBSD, OpenBSD has a ports collection that contains customized versions of many free software packages (*www.openbsd.org/ports.html*). The collection is not as large as FreeBSD's but contains most of the utilities you'd want for a Unix server or workstation. The *x*86 version of OpenBSD can also run programs created for FreeBSD, Linux, and Solaris. OpenBSD supports ten computer architectures—more than FreeBSD but not as many as NetBSD.

OpenBSD CD-ROMs are available from OpenBSD itself or from Cheap Bytes. An ISO image isn't available, however, because the group prefers to sell CDs as a way of supporting its efforts. You can install the OS over the Internet, though the installer's user interface is not as polished as that of FreeBSD.

Of the three no-cost BSDs, OpenBSD has the smallest group of developers and the most difficult learning curve for Unix newbies. But if you need to construct a bulletproof Internet firewall or server, OpenBSD is the way to go.

NETBSD

NetBSD, also available at no cost, is the most portable of the BSD-based operating systems. Currently running on 46 different hardware architectures (and more are in the works), NetBSD is at home running on everything from an ancient 68K-based Macintosh or Amiga to AMD's as-yet-unreleased *x*86-64 Hammer architecture.

This portability makes NetBSD one of the best choices for embedded systems—computers that churn away, unseen, inside other devices. Because porting code from one platform to another tends to expose bugs that might otherwise go unnoticed, code from NetBSD is unusually solid; both OpenBSD (originally a spin-off of the NetBSD Project) and FreeBSD have borrowed from it over time. NetBSD is also the operating system of choice for many types of orphaned hardware, including old Sun workstations.

Like FreeBSD and OpenBSD, NetBSD has a large collection of free ports and packages (fewer than FreeBSD and more than OpenBSD) and can run commercial programs compiled for Linux and other versions of Unix.

NetBSD CD-ROMs are available from Wasabi Systems (*www.wasabisystems.com*), Cheap Bytes, and several other sources listed on the NetBSD Web site. An ISO CD-ROM image is available for download, and you can also install via FTP.

Rather than rearranging the clusters of these files in sequential order, Defrag rearranges them in the order they're loaded when the program launches.

This specialized processing reduces the time required to launch certain programs, but you may prefer to trade that speed for disk space. You're perfectly free to delete the entire contents of this folder, but Task Monitor will start refilling the folder right away. To prevent this, you need to tweak a Registry setting. Launch Regedit from the Start menu's Run dialog and navigate to `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Applets\Defrag\AppStartParams`. In the right-hand pane, find or create a `DWORD` value named `UseProfile`, and set its value to `0`. For more information, see the Microsoft TechNet article at *www.microsoft.com/technet/Win98/Reskit/Part2/wrkc10.asp.*—*NJR*