

"A remarkable collection of characters...courageously exploring mindspace,
an innerworld where nobody had ever been before." —The New York Times

hackers

heroes of the computer revolution

Free Sampler

steven levy

OREILLY

O'Reilly Ebooks—Your bookshelf on your devices!



When you buy an ebook through oreilly.com, you get lifetime access to the book, and whenever possible we provide it to you in four, DRM-free file formats—PDF, .epub, Kindle-compatible .mobi, and Android .apk ebook—that you can use on the devices of your choice. Our ebook files are fully searchable and you can cut-and-paste and print them. We also alert you when we've updated the files with corrections and additions.

Learn more at <http://oreilly.com/ebooks/>

You can also purchase O'Reilly ebooks through [iTunes](#), the [Android Marketplace](#), and [Amazon.com](#).

Hackers

by Steven Levy

Copyright © 2010 Steven Levy. All rights reserved.
Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North,
Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (*my.safaribooksonline.com*). For more information, contact our corporate/institutional sales department: (800) 998-9938 or *corporate@oreilly.com*.

Editor: Mike Hendrickson

Production Editor:

Rachel Monaghan

Production Services:

Octal Publishing, Inc.

Copyeditor: Amy Thomson

Indexer: John Bickelhaupt

Cover Designers:

Monica Kamsvaag and

Edie Freedman

Interior Designer: Ron Bilodeau

Printing History:

May 2010: First Edition.

The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. *Hackers* and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-449-38839-3

[SB]

Contents

CAMBRIDGE: The Fifties and Sixties

Preface	ix
Who's Who: The Wizards and Their Machines	xi

Part One. TRUE HACKERS

CAMBRIDGE: The Fifties and Sixties

Chapter 1	
The Tech Model Railroad Club	3
Chapter 2	
The Hacker Ethic	27
Chapter 3	
Spacewar	39
Chapter 4	
Greenblatt and Gosper	61
Chapter 5	
The Midnight Computer Wiring Society	83
Chapter 6	
Winners and Losers	101
Chapter 7	
Life	123

Part Two. HARDWARE HACKERS

NORTHERN CALIFORNIA: The Seventies

Chapter 8	
Revolt in 2100	151
Chapter 9	
Every Man a God	179
Chapter 10	
The Homebrew Computer Club	201
Chapter 11	
Tiny BASIC	227
Chapter 12	
Woz	249
Chapter 13	
Secrets	275

Part Three. GAME HACKERS

THE SIERRAS: The Eighties

Chapter 14	
The Wizard and the Princess	289
Chapter 15	
The Brotherhood	313
Chapter 16	
The Third Generation	325
Chapter 17	
Summer Camp	345
Chapter 18	
Frogger	365

Chapter 19	
Applefest	389
Chapter 20	
Wizard vs. Wizards	413

Part Four. THE LAST OF THE TRUE HACKERS

CAMBRIDGE: 1983

The Last of the True Hackers	437
Afterword: Ten Years After	455
Afterword: 2010	463
Notes	479
Acknowledgments	485
Index	489

Who's Who: The Wizards and Their Machines

Bob Albrecht Founder of People's Computer Company who took visceral pleasure in exposing youngsters to computers.

Altair 8800 The pioneering microcomputer that galvanized hardware hackers. Building this kit made you learn hacking. Then you tried to figure out what to *do* with it.

Apple II Steve Wozniak's friendly, flaky, good-looking computer, wildly successful and the spark and soul of a thriving industry.

Atari 800 This home computer gave great graphics to game hackers like John Harris, though the company that made it was loath to tell you how it worked.

Bob and Carolyn Box World-record-holding gold prospectors turned software stars, working for Sierra On-Line.

Doug Carlston Corporate lawyer who chucked it all to form the Brøderbund software company.

Bob Davis Left a job in a liquor store to become the bestselling author of the Sierra On-Line computer game *Ulysses and the Golden Fleece*. Success was his downfall.

Peter Deutsch Bad in sports, brilliant at math, Peter was still in short pants when he stumbled on the TX-0 at MIT—and hacked it along with the masters.

Steve Dompier Homebrew member who first made Altair sing, and later wrote the *Target* game on the Sol, which entranced Tom Snyder.

John Draper The notorious “Captain Crunch” who fearlessly explored phone systems, was jailed, and later hacked microcomputers. Cigarettes made him violent.

Mark Duchaineau The young Dungeonmaster who copy-protected On-Line’s disks at his whim.

Chris Espinosa Fourteen-year-old follower of Steve Wozniak and early Apple employee.

Lee Felsenstein Former “military editor” of the *Berkeley Barb* and hero of an imaginary science-fiction novel, he designed computers with a “junkyard” approach and was a central figure in Bay Area hardware hacking in the seventies.

Ed Fredkin Gentle founder of Information International, he thought himself the world’s greatest programmer until he met Stew Nelson. Father figure to hackers.

Gordon French Silver-haired hardware hacker whose garage held not cars but his homebrewed Chicken Hawk computer, then held the first Homebrew Computer Club meeting.

Richard Garriott Astronaut’s son who, as Lord British, created the *Ultima* world on computer disks.

Bill Gates Cocky wizard and Harvard dropout who wrote Altair BASIC, and complained when hackers copied it.

Bill Gosper Horowitz of computer keyboards, master math and *LIFE* hacker at MIT AI lab, guru of the Hacker Ethic, and student of Chinese restaurant menus.

Richard Greenblatt Single-minded, unkempt, prolific, and canonical MIT hacker who went into night phase so often that he zorched his academic career. The hacker’s hacker.

John Harris The young Atari 800 game hacker who became Sierra On-Line’s star programmer, but yearned for female companionship.

IBM PC IBM’s entry into the personal computer market, which amazingly included a bit of the Hacker Ethic and took over.

IBM 704 IBM was The Enemy and this was its machine, the Hulking Giant computer in MIT's Building 26. Later modified into the IBM 709, then the IBM 7090. Batch-processed and intolerable.

Jerry Jewell Vietnam vet turned programmer who founded Sirius Software.

Steven Jobs Visionary, bearded, nonhacking youngster who took Wozniak's Apple II, made lots of deals, and formed a company that would make a billion dollars.

Tom Knight At sixteen, an MIT hacker who would name the Incompatible Time-sharing System. Later, a Greenblatt nemesis over the LISP machine schism.

Alan Kotok The chubby MIT student from Jersey who worked under the rail layout at TMRC, learned the phone system at Western Electric, and became a legendary TX-0 and PDP-1 hacker.

Efrem Lipkin Hacker-activist from New York who loved machines but hated their uses. Cofounded Community Memory; friend of Felsenstein.

LISP Machine The ultimate hacker computer, invented mostly by Greenblatt and subject of a bitter dispute at MIT.

"Uncle" John McCarthy Absentminded but brilliant MIT (later Stanford) professor who helped pioneer computer chess, artificial intelligence, LISP.

Bob Marsh Berkeley-ite and Homebrewer who shared garage with Felsenstein and founded Processor Technology, which made the Sol computer.

Roger Melen Homebrewer who cofounded Cromemco company to make circuit boards for Altair. His "Dazzler" played *LIFE* program on his kitchen table.

Louis Merton Pseudonym for the AI chess hacker whose tendency to go catatonic brought the hacker community together.

Jude Milhon Met Lee Felsenstein through a classified ad in the *Berkeley Barb* and became more than a friend—a member of the Community Memory collective.

Marvin Minsky Playful and brilliant MIT professor who headed AI lab and allowed the hackers to run free.

Fred Moore Vagabond pacifist who hated money, loved technology, and cofounded Homebrew Club.

Stewart Nelson Buck-toothed, diminutive, but fiery AI lab hacker who connected the PDP-1 computer to hack the phone system. Later cofounded Systems Concepts company.

Ted Nelson Self-described “innovator” and noted curmudgeon who self-published the influential *Computer Lib* book.

Russell Noftsker Harried administrator of MIT AI lab in late sixties; later president of Symbolics company.

Adam Osborne Bangkok-born publisher-turned-computer-manufacturer who considered himself a philosopher. Founded Osborne Computer Company to make “adequate” machines.

PDP-1 Digital Equipment’s first minicomputer and in 1961 an interactive godsend to the MIT hackers and a slap in the face to IBM fascism.

PDP-6 Designed in part by Kotok, this mainframe computer was the cornerstone of the AI lab, with its gorgeous instruction set and sixteen sexy registers.

Tom Pittman The religious Homebrew hacker who lost his wife but kept the faith with his Tiny BASIC.

Ed Roberts Enigmatic founder of MITS company who shook the world with his Altair computer. He wanted to help people build mental pyramids.

Steve (Slug) Russell McCarthy’s “coolie” who hacked the *Spacewar* program, first videogame, on the PDP-1. Never made a dime from it.

Peter Samson MIT hacker (one of the first), who loved systems, trains, TX-0, music, parliamentary procedure, pranks, and hacking.

Bob Saunders Jolly, balding TMRC hacker who married early, hacked til late at night eating “lemon gunkies,” and mastered the “CBS strategy” on *Spacewar*.

Warren Schwader Big blond hacker from rural Wisconsin who went from the assembly line to software stardom, but couldn’t reconcile the shift with his devotion to Jehovah’s Witnesses.

David Silver Left school at fourteen to be mascot of AI lab; maker of illicit keys and builder of a tiny robot that did the impossible.

Dan Sokol Long-haired prankster who reveled in revealing technological secrets at Homebrew Club. Helped “liberate” Altair BASIC program on paper tape.

Sol Computer Lee Felsenstein’s terminal-and-computer, built in two frantic months, almost the computer that turned things around. Almost wasn’t enough.

Les Solomon Editor of *Popular Electronics*, the puller of strings who set the computer revolution into motion.

Marty Spergel The Junk Man, the Homebrew member who supplied circuits and cables and could make you a deal for anything.

Richard Stallman The Last of the Hackers, he vowed to defend the principles of hackerism to the bitter end. Remained at MIT until there was no one to eat Chinese food with.

Jeff Stephenson Thirty-year-old martial arts veteran and hacker who was astounded that joining Sierra On-Line meant enrolling in Summer Camp.

Jay Sullivan Maddeningly calm wizard-level programmer at Informatics who impressed Ken Williams by knowing the meaning of the word “any.”

Dick Sunderland Chalk-complexioned MBA who believed that firm managerial bureaucracy was a worthy goal, but as president of Sierra On-Line found that hackers didn’t think that way.

Gerry Sussman Young MIT hacker branded “loser” because he smoked a pipe and “munged” his programs; later became “winner” by algorithmic magic.

Margot Tommervik With her husband Al, long-haired Margot parlayed her gameshow winnings into a magazine that deified the Apple Computer.

Tom Swift Terminal Lee Felsenstein’s legendary, never-to-be-built computer terminal, which would give the user ultimate leave to get his hands on the world.

TX-0 Filled a small room, but in the late fifties, this \$3 million machine was world’s first personal computer—for the community of MIT hackers that formed around it.

Jim Warren Portly purveyor of “techno-gossip” at Homebrew, he was first editor of hippie-styled *Dr. Dobbs Journal*, later started the lucrative Computer Faire.

Randy Wigginton Fifteen-year-old member of Steve Wozniak’s kiddie corps, he helped Woz trundle the Apple II to Homebrew. Still in high school when he became Apple’s first software employee.

Ken Williams Arrogant and brilliant young programmer who saw the writing on the CRT and started Sierra On-Line to make a killing and improve society by selling games for the Apple computer.

Roberta Williams Ken Williams’ timid wife who rediscovered her own creativity by writing *Mystery House*, the first of her many bestselling computer games.

Stephen “Woz” Wozniak Openhearted, technologically daring hardware hacker from San Jose suburbs, Woz built the Apple Computer for the pleasure of himself and friends.

Greenblatt and Gosper

Ricky Greenblatt was a hacker waiting to happen. Years later, when he was known throughout the nation's computer centers as the archetypal hacker, when the tales of his single-minded concentration were almost as prolific as the millions of lines of assembly-language code he'd hacked, someone would ask him how it all started. He'd twist back in his chair, looking not as rumpled as he did back as an undergraduate, when he was cherub-faced and dark-haired and painfully awkward of speech; the question, he figured, came down to whether hackers were born or made, and out came one of the notorious non sequiturs which came to be known as Blatt-isms: "If hackers are born, then they're going to get made, and if they're made into it, they were born."

But Greenblatt would admit that he was a born hacker.

Not that his first encounter with the PDP-1 had changed his life. He was interested, all right. It had been freshman rush week at MIT, and Ricky Greenblatt had some time on his hands before tackling his courses, ready for academic glory. He visited the places that interested him most: the campus radio station WTBS (MIT's was perhaps the only college radio station in the country with a surfeit of student audio engineers and a shortage of disc jockeys), the Tech Model Railroad Club, and the Kluge Room in Building 26, which held the PDP-1.

Some hackers were playing *Spacewar*.

It was the general rule to play the game with all the room lights turned off, so the people crowded around the console would have their faces eerily illuminated by this display of spaceships and heavy stars. Rapt faces lit by the glow of the computer. Ricky Greenblatt was impressed. He watched the cosmic clashes for a while, then went next door to look over the TX-0, with its racks of tubes and transistors, its fancy power supplies, its lights and switches. His high school math club back in Columbia, Missouri, had visited the state university's batch-processed computer, and he'd seen a giant card-sorting machine at a local insurance company. But nothing like this. Still, despite being impressed with the radio station, the Model Railroad Club, and especially the computers, he set about making dean's list.

This scholastic virtue could not last. Greenblatt, even more than your normal MIT student, was a willing conscript of the Hands-On Imperative. His life had been changed irrevocably the day in 1954 that his father, visiting the son he hadn't lived with since an early divorce, took him to the Memorial Student Union at the University of Missouri, not far from Ricky's house in Columbia. Ricky Greenblatt took to the place immediately. It wasn't merely because of the comfortable lounge, the television set, the soft-drink bar . . . It was because of the students, who were more of an intellectual match for nine-year-old Ricky Greenblatt than were his classmates. He would go there to play chess, and he usually had no problem beating the college students. He was a very good chess player.

One of his chess victims was a UM engineering student on the GI bill. His name was Lester, and Lester's gift to this nine-year-old prodigy was a hands-on introduction to the world of electronics. A world where there were no ambiguities. Logic prevailed. You had a degree of control over things. You could build things according to your own plan. To a nine-year-old whose intelligence might have made him uncomfortable with his chronological peers, a child affected by a marital split which was typical of a world of human relations beyond his control, electronics was the perfect escape.

Lester and Ricky worked on ham radio projects. They tore apart old television sets. Before finishing college, Lester introduced Ricky to a Mr. Houghton, who ran a local radio shop, and that

became a second home to the youngster through high school. With a high school friend, Greenblatt built a gamut of hairy projects. Amplifiers, modulators, all sorts of evil looking vacuum tube contraptions. An oscilloscope. Ham radios. A television camera. A television camera! It seemed like a good idea, so they built it. And of course when it came time to choose a college, Richard Greenblatt picked MIT. He entered in the fall of 1962.

The course work was rigid during his first term, but Greenblatt was handling it without much problem. He had developed a relationship with a few campus computers. He had gotten lucky, landing the elective course called EE 641—Introduction to Computer Programming—and he would often go down to the punch-card machines at EAM to make programs for the Hulking Giant 7090. Also, his roommate, Mike Beeler, had been taking a course in something called Nomography. The students taking the class had hands-on access to an IBM 1620—set in yet another enclave of those misguided priests whose minds had been clouded with the ignorant fog that came from the IBM sales force. Greenblatt would often accompany Beeler to the 1620, where you would punch up your card deck, and stand in line. When your turn came, you'd dump your cards in the reader and get an instant printout from a plotter-printer. "It was sort of a fun, evening thing to do," Beeler would later recall. "We'd do it the way others might watch a sports game, or go out and have a beer." It was limited but gratifying. It made Greenblatt want more.

Around Christmas time, he finally felt comfortable enough to hang out at the Model Railroad Club. There, around such people as Peter Samson, it was natural to fall into hacker mode. (Computers had various states called "modes," and hackers often used that phrase to describe conditions in real life.) Samson had been working on a big timetable program for the TMRC operating sessions on the giant layout; because of the number crunching required, Samson had done it in FORTRAN on the 7090. Greenblatt decided to write the first FORTRAN for the PDP-1. Just why he decided to do this is something he could never explain, and chances are no one asked. It was common, if you wanted to do a task on a machine and the machine didn't have the software to do it, to write the proper software so you *could* do it. This was an impulse that Greenblatt would later elevate to an art form.

He did it, too. Wrote a program that would enable you to write in FORTRAN, taking what you wrote and compiling the code into machine language, as well as transforming the computer's machine language responses back into FORTRAN. Greenblatt did his FORTRAN compiler largely in his room, since he had trouble getting enough access to the PDP-1 to work online. Besides that, he got involved in working on a new system of relays underneath the layout at TMRC. It seems that the plaster in the room (which was always pretty grungy anyway, because custodial people were officially barred entry) kept falling, and some of it would get on the contacts of the system that Jack Dennis had masterminded in the mid-fifties. Also, there was something new called a wire-spring relay which looked better than the old kind. So Greenblatt spent a good deal of time that spring doing that. Along with PDP-1 hacking.

It is funny how things happen. You begin working conscientiously as a student, you make the dean's list, and then you discover something that puts classes into their proper perspective: they are totally irrelevant to the matter at hand. The matter at hand was hacking, and it seemed obvious—at least, so obvious that no one around TMRC or the PDP-1 seemed to think it even a useful topic of discourse—that hacking was a pursuit so satisfying that you could make a life of it. While a computer is very complex, it is not nearly as complex as the various comings and goings and interrelationships of the human zoo; but, unlike formal or informal study of the social sciences, hacking gave you not only an understanding of the system, but an addictive control as well, along with the illusion that total control was just a few features away. Naturally, you go about building those aspects of the system that seem most necessary to work within the system in the proper way. Just as naturally, working in this improved system lets you know of more things that need to be done. Then someone like Marvin Minsky might happen along and say, "Here is a robot arm. I am leaving this robot arm by the machine." Immediately, nothing in the world is as essential as making the proper interface between the machine and the robot arm, and putting the robot arm under your control, and figuring a way to create a system where the robot arm knows what the hell it is doing. Then you can see your offspring come to life. How can something as contrived as an engineering class compare to that? Chances are that

your engineering professor has never done anything half as interesting as the problems you are solving every day on the PDP-1. Who's right?

By Greenblatt's sophomore year, the computer scene around the PDP-1 was changing considerably. Though a few more of the original TX-0 hackers had departed, there was new talent arriving, and the new, ambitious setup, funded by the benevolent Department of Defense, nicely accommodated their hacking. A second PDP-1 had arrived; its home was the new, nine-story rectangular building on Main Street—a building of mind-numbing dullness, with no protuberances, and sill-less windows that looked painted onto its off-white surface. The building was called Tech Square, and among the MIT and corporate clients moving in was Project MAC. The ninth floor of this building, where the computers were, would be home to a generation of hackers, and none would spend as much time there as Greenblatt.

Greenblatt was getting paid (sub-minimum wages) for hacking as a student employee, as were several hackers who worked on the system or were starting to develop some of the large programs that would do artificial intelligence. They started to notice that this awkwardly polite sophomore was a potential PDP-1 superstar.

He was turning out an incredible amount of code, hacking as much as he could, or sitting with a stack of printouts, marking them up. He'd shuttle between the PDP-1 and TMRC, with his head fantastically wired with the structures of the program he was working on, or the system of relays he'd hacked under the TMRC layout. To hold that concentration for a long period of time, he lived, as did several of his peers, the thirty-hour day. It was conducive to intense hacking, since you had an extended block of waking hours to get going on a program, and, once you were really rolling, little annoyances like sleep need not bother you. The idea was to burn away for thirty hours, reach total exhaustion, then go home and collapse for twelve hours. An alternative would be to collapse right there in the lab. A minor drawback of this sort of schedule was that it put you at odds with the routines which everyone else in the world used to do things like keep appointments, eat, and go to classes. Hackers could accommodate this—one would commonly ask questions like, "What phase is Greenblatt

in?” and someone who had seen him recently would say, “I think he’s in a night phase now, and should be in around nine or so.” Professors did not adjust to those phases so easily, and Greenblatt “zorched” his classes.

He was placed on academic probation, and his mother came to Massachusetts to confer with the dean. There was some explaining to do. “His mom was concerned,” his roommate Beeler would later say. “Her idea was that he was here to get a degree. But the things he was doing on the computer were completely state-of-the-art—no one was doing them yet. He saw additional things to be done. It was very difficult to get excited about classes.” To Greenblatt, it wasn’t *really* important that he was in danger of flunking out of college. Hacking was paramount: it was what he did best and what made him happiest.

His worst moment came when he was so “out of phase” that he slept past a final exam. It only hastened his exit from the student body of MIT. Flunking out probably wouldn’t have made any difference at all in his life had it not been for a rule that you couldn’t be a student employee when you were an exiled student. So Greenblatt went looking for work, fully intending to get a day-time programming job that would allow him to spend his nights at the place he wanted to spend his time—the ninth floor at Tech Square. Hacking. And that is exactly what he did.



There was an equally impressive hacker who had mastered the PDP-1 in a different manner. More verbal than Greenblatt, he was better able to articulate his vision of how the computer had changed his life, and how it might change all our lives. This student was named Bill Gosper. He had begun MIT a year before Greenblatt, but had been somewhat slower at becoming a habitué of the PDP-1. Gosper was thin, with bird-like features covered by thick spectacles and an unruly head of kinky brown hair. But even a brief meeting with Gosper was enough to convince you that here was someone whose brilliance put things like physical appearance into their properly trivial perspective. He was a math genius. It was actually the idea of hacking the world of mathematics, rather than hacking systems, that attracted Gosper to the computer, and

he was to serve as a long-time foil to Greenblatt and the other systems-oriented people in the society of brilliant foot soldiers now forming around brand-new Project MAC.

Gosper was from Pennsauken, New Jersey, across the river from Philadelphia, and his pre-MIT experience with computers, like Greenblatt's, was limited to watching Hulking Giants operate from behind a pane of glass. He could vividly recall seeing the Univac at Philadelphia's Franklin Institute churn out pictures of Benjamin Franklin on its line printer. Gosper had no idea what was going on, but it looked like great fun.

He tasted that fun himself for the first time in his second MIT semester. He'd taken a course from Uncle John McCarthy—open only to freshmen who'd gotten disgustingly high grade point averages the previous term. The course began with FORTRAN, went on to IBM machine language, and wound up on the PDP-1. The problems were nontrivial, things like tracing rays through optical systems with the 709, or working routines with a new floating-point interpreter for the PDP-1.

The challenge of programming appealed to Gosper. Especially on the PDP-1, which, after the torture of IBM batch processing could work on you like an intoxicating elixir. Or having sex for the first time. Years later, Gosper still spoke with excitement of “the rush of having this live keyboard under you and having this machine respond in milliseconds to what you were doing . . .”

Still, Gosper was timid about continuing on the PDP-1 after the course was over. He was involved with the math department, where people kept telling him that he would be wise to stay away from computers—they would turn him into a clerk. The unofficial slogan of the math department, Gosper found, was “There's no such thing as Computer Science—it's witchcraft!” Well then, Gosper would be a witch! He signed up for Minsky's course in artificial intelligence. The work was again on the PDP-1, and this time Gosper got drawn into hacking itself. Somewhere in that term, he wrote a program to plot functions on the screen, his first real project, and one of the subroutines contained a program bum so elegant that he dared show it to Alan Kotok. Kotok by then had attained, thought Gosper, “godlike status,” not only from his exploits on the PDP-1 and TMRC, but from the well-known fact that his work at DEC included a prime role in the design of a new

computer, a much-enhanced version of the PDP-1. Gosper was rapturous when Kotok not only looked over his hack, but thought it clever enough to show to someone else. *Kotok actually thought I'd done something neat!* Gosper hunkered down for more hacking.

His big project in that course was an attempt to “solve” the game *Peg Solitaire* (or *HI-Q*), where you have a board in the shape of a plus sign with thirty-three holes in it. Every hole but one is filled by a peg: you jump pegs over each other, removing the ones you jump over. The idea is to finish with one peg in the center. When Gosper and two classmates proposed to Minsky that they solve the problem on the PDP-1, Minsky doubted they could do it, but welcomed the try. Gosper and his friends not only solved it—“We demolished it,” he'd later say. They hacked a program that would enable the PDP-1 to solve the game in an hour and a half.

Gosper admired the way the computer solved HI-Q because its approach was “counterintuitive.” He had a profound respect for programs which used techniques that on the surface seemed improbable, but in fact took advantage of the situation's deep mathematical truth. The counterintuitive solution sprang from understanding the magical connections between things in the vast mandala of numerical relationships on which hacking ultimately was based. Discovering those relationships—making new mathematics on the computer—was to be Gosper's quest; and as he began hanging out more around the PDP-1 and TMRC, he made himself indispensable as the chief “math hacker”—not so much interested in systems programs, but able to come up with astoundingly clear (nonintuitive!) algorithms which might help a systems hacker knock a few instructions off a subroutine, or crack a mental logjam on getting a program running.

• • • • •

Gosper and Greenblatt represented two kinds of hacking around TMRC and the PDP-1: Greenblatt focused on pragmatic systems building, and Gosper on mathematical exploration. Each respected the other's forte, and both would participate in projects, often collaborative ones, that exploited their best abilities. More than that, both were major contributors to the still nascent culture that was

beginning to flower in its fullest form on the ninth floor of Tech Square. For various reasons, it would be in this technological hot-house that the culture would grow most lushly, taking the Hacker Ethic to its extreme.

The action would shift among several scenes. The Kluge Room, with the PDP-1 now operating with the time-sharing system, which Jack Dennis had worked for a year to write, was still an option for some late-night hacking, and especially *Spacewarring*. But more and more, the true hackers would prefer the Project MAC computer. It stood among other machines on the harshly lit, sterilely furnished ninth floor of Tech Square, where one could escape from the hum of the air conditioners running the various computers only by ducking into one of several tiny offices. Finally, there was TMRC, with its never-empty Coke machine and Saunders' change box and the Tool Room next door, where people would sit at all hours of the night and argue what to an outsider would be bafflingly arcane points.

These arguments were the lifeblood of the hacker community. Sometimes people would literally scream at each other, insisting on a certain kind of coding scheme for an assembler, or a specific type of interface, or a particular feature in a computer language. These differences would have hackers banging on the blackboard or throwing chalk across the room. It wasn't so much a battle of egos as it was an attempt to figure out what "The Right Thing" was. The term had special meaning to the hackers. The Right Thing implied that to any problem, whether a programming dilemma, a hardware interface mismatch, or a question of software architecture, a solution existed that was just . . . it. The perfect algorithm. You'd have hacked right into the sweet spot, and anyone with half a brain would see that the straight line between two points had been drawn, and there was no sense trying to top it. "The Right Thing," Gosper would later explain, "very specifically meant the unique, correct, elegant solution . . . the thing that satisfied all the constraints at the same time, which everyone seemed to believe existed for most problems."

Gosper and Greenblatt both had strong opinions, but usually Greenblatt would tire of corrosive human interfacing, and wander away to actually *implement* something. Elegant or not. In his thinking, *things had to be done*. And if no one else would be

hacking them, he would. He would sit down with paper and pencil, or maybe at the console of the PDP-1, and scream out his code. Greenblatt's programs were robust, meaning that their foundation was firm, with built-in error checks to prevent the whole thing from bombing as a result of a single mistake. By the time Greenblatt was through with a program, it was thoroughly debugged. Gosper thought that Greenblatt loved finding and fixing bugs more than anybody he'd ever met, and suspected he sometimes wrote buggy code just so he could fix it.

Gosper had a more public style of hacking. He liked to work with an audience, and often novice hackers would pull up a chair behind him at the console to watch him write his clever hacks, which were often loaded with terse little mathematical points of interest. He was at his best at display hacks, where an unusual algorithm would evoke a steadily unpredictable series of CRT pyrotechnics. Gosper would act as tour guide as he progressed, sometimes emphasizing that even typing mistakes could present an interesting numerical phenomenon. He maintained a continual fascination with the way a computer could spit back something unexpected, and he would treat the utterances of the machine with infinite respect. Sometimes the most seemingly random event could lure him off into a fascinating tangent on the implications of this quadratic surd or that transcendental function. Certain sub-routine wizardry in a Gosper program would occasionally evolve into a scholarly memo, like the one that begins:

On the theory that continued fractions are underused, probably because of their unfamiliarity, I offer the following propaganda session on the relative merits of continued fractions versus other numerical representations.

The arguments in the Tool Room were no mere college bull sessions. Kotok would often be there, and it was at those sessions that significant decisions were made concerning the computer he was designing for DEC, the PDP-6. Even in its design stage, this PDP-6 was considered the absolute Right Thing around TMRC. Kotok would sometimes drive Gosper back to South Jersey for holiday breaks, talking as he drove about how this new computer would have *sixteen independent registers*. (A register, or accumulator, is a place within a computer where actual computation occurs. Sixteen of them would give a machine a heretofore

unheard-of versatility.) Gosper would gasp. *That'll be, he thought, the greatest computer in the history of the world!*

When DEC actually built the PDP-6 and gave the first prototype to Project MAC, everyone could see that while the computer had all the necessary sops for commercial users, it was at heart a hacker's machine. Both Kotok and his boss, Gordon Bell, recalling their TX-0 days, used the PDP-6 to demolish the limitations that had bothered them on that machine. Also, Kotok had listened closely to the suggestions of TMRC people, notably Peter Samson, who took credit for the sixteen registers. The instruction set had everything you needed, and the overall architecture was symmetrically sound. The sixteen registers could be accessed three different ways each, and you could do it in combinations, to get a lot done by using a single instruction. The PDP-6 also used a "stack," which allowed you to mix and match your subroutines, programs, and activities with ease. To hackers, the introduction of the PDP-6 and its achingly beautiful instruction set meant they had a powerful new vocabulary with which to express sentiments that previously could be conveyed only in the most awkward terms.

Minsky set the hackers to work writing new systems software for the PDP-6, a beautiful sea-blue machine with three large cabinets, a more streamlined control panel than the One, rows of shiny cantilevered switches, and a winking matrix of lights. Soon they were into the psychology of this new machine as deeply as they had been on the PDP-1. But you could go further on the Six. One day in the Tool Room at TMRC the hackers were playing around with different ways to do decimal print routines, little programs to get the computer to print out in Arabic numbers. Someone got the idea of trying some of the flashy new instructions on the PDP-6, the ones that utilized the stack. Hardly anyone had integrated these new instructions into his code; but as the program got put on the blackboard using one instruction called Push-J, to everyone's amazement the entire decimal print routine, which normally would be a page worth of code, came out only six instructions long. After that, everyone around TMRC agreed that Push-J had certainly been The Right Thing to put into the PDP-6.

The Tool Room discussions and arguments would often be carried over to dinner, and the cuisine of choice was almost always Chinese food. It was cheap, plentiful, and—best of all—available

late at night. (A poor second choice was the nearby greasy spoon on Cambridge's Main Street, a maroon-paneled former railroad car named the F&T Diner, but called by hackers "The Red Death.") On most Saturday evenings, or spontaneously on weeknights after 10 P.M., a group of hackers would head out, sometimes in Greenblatt's blue 1954 Chevy convertible, to Boston's Chinatown.

Chinese food was a system, too, and the hacker curiosity was applied to that system as assiduously as to a new LISP compiler. Samson had been an aficionado from his first experience on a TMRC outing to Joy Fong's on Central Square, and by the early sixties he had actually learned enough Chinese characters to read menus and order obscure dishes. Gosper took to the cuisine with even greater vigor; he would prowl Chinatown looking for restaurants open after midnight, and one night he found a tiny little cellar place run by a small family. It was fairly dull food, but he noticed some Chinese people eating fantastic-looking dishes. So he figured he'd take Samson back there.

They went back loaded with Chinese dictionaries, and demanded a Chinese menu. The chef, a Mr. Wong, reluctantly complied, and Gosper, Samson, and the others pored over the menu as if it were an instruction set for a new machine. Samson supplied the translations, which were positively revelatory. What was called "Beef with Tomato" on the English menu had a literal meaning of Barbarian Eggplant Cowpork. "Wonton" had a Chinese equivalent of Cloud Gulp. There were unbelievable things to discover in this system! So after deciding the most interesting things to order ("Hibiscus Wing? Better order that, find out what that's about"), they called over Mr. Wong, and he jabbered frantically in Chinese disapproval of their selections. It turned out he was reluctant to serve them the food Chinese-style, thinking that Americans couldn't take it. Mr. Wong had mistaken them for typically timid Americans—but these were explorers! They had been inside the machine, and lived to tell the tale (they would tell it in assembly language). Mr. Wong gave in. Out came the best Chinese meal that any of the hackers had eaten to date.

So expert were the TMRC people at hacking Chinese food that they could eventually go the restaurateurs one better. On a hacker excursion one April Fools' Day, Gosper had a craving for a little-known dish called Bitter Melon. It was a wart-dotted form of

green pepper, with an intense quinine taste that evoked nausea in all but those who'd painfully acquired the taste. For reasons best known to himself, Gosper decided to have it with sweet-and-sour sauce, and he wrote down the order in Chinese. The owner's daughter came out giggling. "I'm afraid you made a mistake—my father says that this says 'Sweet-and-Sour Bitter Melon.'" Gosper took this as a challenge. Besides, he was offended that the daughter couldn't even read Chinese—that went against the logic of an efficient Chinese Restaurant System, a logic Gosper had come to respect. So, even though he knew his order was a preposterous request, he acted indignant, telling the daughter, "Of course it says Sweet-and-Sour Bitter Melon—we Americans *always* order Sweet-and-Sour Bitter Melon the first of April." Finally, the owner himself came out. "You can't eat!" he shouted. "No taste! No taste!" The hackers stuck to the request, and the owner slunk back to the kitchen.

Sweet-and-Sour Bitter Melon turned out to be every bit as hideous as the owner promised. The sauce at that place was wickedly potent, so much so that if you inhaled while you put some in your mouth you'd choke. Combined with the ordinarily vile bitter melon, it created a chemical that seemed to squeak on your teeth, and no amount of tea or Coca-Cola could dilute that taste. To almost any other group of people, the experience would have been a nightmare. But to the hackers it was all part of the system. It made no human sense, but had its logic. It was The Right Thing; therefore every year on April Fools' Day they returned to the restaurant and insisted that their appetizer be Sweet-and-Sour Bitter Melon.

It was during those meals that the hackers were most social. Chinese restaurants offered hackers a fascinating culinary system and a physically predictable environment. To make it even more comfortable, Gosper, one of several hackers who despised smoke in the air and disdained those who smoked, brought along a tiny, battery-powered fan. The fan was something kluged up by a teenage hacker who hung around the AI lab—it looked like a mean little bomb, and had been built using a cooling fan from a junked computer. Gosper would put it on the table to gently blow smoke back into offenders' faces. On one occasion at the Lucky Garden in Cambridge, a brutish jock at a nearby table became outraged when the little fan redirected the smoke from his date's cigarette

back to their table. He looked at these grungy MIT types with their little fan and demanded the hackers turn the thing off. “OK, if she stops smoking,” they said, and at that point the jock charged the table, knocking dishes around, spilling tea all over, and even sticking his chopsticks into the blades of the fan. The hackers, who considered physical combat one of the more idiotic human interfaces, watched in astonishment. The incident ended as soon as the jock noticed a policeman sitting across the restaurant.

That was an exception to what were usually convivial gatherings. The talk revolved around various hacking issues. Often, people would have their printouts with them and during lulls in conversation would bury their noses in the reams of assembly code. On occasion, the hackers would even discuss some events in the “real world,” but the Hacker Ethic would be identifiable in the terms of the discussion. It would come down to some flaw in a system. Or an interesting event would be considered in light of a hacker’s natural curiosity about the way things work.

A common subject was the hideous reign of IBM, the disgustingly naked emperor of the computer kingdom. Greenblatt might go on a “flame”—an extended and agitated riff—about the zillions of dollars being wasted on IBM computers. Greenblatt would go home on vacation and see that the science department at the University of Missouri, which allegedly didn’t have any money, was spending four million dollars a year on the care and feeding of an IBM Hulking Giant that wasn’t nearly as nifty as the PDP-6. And speaking of grossly overrated stuff, what about that IBM time-sharing system at MIT, with that IBM 7094 right there on the ninth floor? Talk about waste!

This could go on for a whole meal. It is telling, though, to note the things that the hackers did not talk about. They did not spend much time discussing the social and political implications of computers in society (except maybe to mention how utterly wrong and naive the popular conception of computers was). They did not talk sports. They generally kept their own emotional and personal lives—as far as they had any—to themselves. And for a group of healthy college-age males, there was remarkably little discussion of a topic, which commonly obsesses groups of that composition: females.

Though some hackers led somewhat active social lives, the key figures in TMRC-PDP hacking had locked themselves into what would be called “bachelor mode.” It was easy to fall into—for one thing, many of the hackers were loners to begin with, socially uncomfortable. It was the predictability and controllability of a computer system—as opposed to the hopelessly random problems in a human relationship—which made hacking particularly attractive. But an even weightier factor was the hackers’ impression that computing was much more *important* than getting involved in a romantic relationship. It was a question of priorities.

Hacking had replaced sex in their lives.

“The people were just so interested in computers and that kind of stuff that they just really didn’t have time [for women],” Kotok would later reflect. “And as they got older, everyone sort of had the view that one day some woman would come along and sort of plunk you over the head and say, *you!*” That was more or less what happened to Kotok, though not until his late thirties. Meanwhile, hackers acted as if sex didn’t exist. They wouldn’t notice some gorgeous woman at the table next to them in the Chinese restaurant, because “the concept of gorgeous woman wasn’t in the vocabulary,” hacker David Silver later explained. When a woman did come into the life of a serious hacker, there might be some discussion—“What’s happened to so-and-so . . . the guy’s just completely falling apart . . .” But generally that kind of thing was not so much disdained as it was shrugged off. You couldn’t dwell on those who might have fallen by the wayside, because you were involved in the most important thing in the world—hacking. Not only an obsession and a lusty pleasure, hacking was a mission. You would hack, and you would live by the Hacker Ethic, and you knew that horribly inefficient and wasteful things like women burned too many cycles, occupied too much memory space. “Women, even today, are considered grossly unpredictable,” one PDP-6 hacker noted, almost two decades later. “How can a hacker tolerate such an imperfect being?”

Maybe it would have been different if there had been more women around TMRC and the ninth floor—the few that did hang around paired off with hackers. (“*They found us,*” one hacker would later note.) There were not too many of these women, since outsiders, male or female, were often put off by the group: the

hackers talked strangely, they had bizarre hours, they ate weird food, and they spent all their time thinking about computers.

And they formed an exclusively male culture. The sad fact was that there never was a star-quality female hacker. No one knows why. There were women *programmers* and some of them were good, but none seemed to take hacking as a holy calling the way Greenblatt, Gosper, and the others did. Even the substantial cultural bias against women getting into serious computing does not explain the utter lack of female hackers. “Cultural things are strong, but not *that* strong,” Gosper would later conclude, attributing the phenomenon to genetic, or “hardware,” differences.

In any case, only rarely were women in attendance at the Chinese restaurant excursions or the sessions at the Tool Room next door to TMRC. So naturally, one did not have to look one’s best. Greenblatt, perhaps, took this to an extreme. He worked on several mammoth projects in the mid-sixties, and would often get so wrapped up in them that his personal habits became a matter of some concern to his fellow hackers.

After he dropped out of school, Greenblatt had taken a job at a firm called Charles Adams Associates, which was in the process of buying and setting up a PDP-1. Greenblatt would work at their offices near Boston’s “Technology Highway” outside the city during the day and drive thirty miles back to MIT after work for some all-night hacking. Originally he moved from the dorms to the Cambridge YMCA, but they booted him out because he wouldn’t keep his room clean. After his stint at Adams, he got rehired at the AI Lab, and though he had a stable living situation—as a boarder in a Belmont house owned by a retired dentist and his wife—he would often sleep on a cot on the ninth floor. Cleanliness was apparently a low priority, since tales abounded of his noticeable grunginess. (Later Greenblatt would insist that he was no worse than some of the others.) Some hackers recall that one of the things Greenblatt’s hacking precluded was regular bathing, and the result was a powerful odor. The joke around the AI lab was that there was a new scientific olfactory measure called a milliblatt. One or two milliblatts was extremely powerful, and one full blatt was just about inconceivable. To decrease the milliblatts, the story goes, hackers maneuvered Greenblatt to a place in

the hallway of Building 20 where there was an emergency shower for cases of accidental exposure to chemicals, and let it rip.

Gosper would sometimes tweak Greenblatt for his personal habits, and was particularly bothered at Greenblatt's habit of rubbing his hands together, which resulted in little pieces of dirt falling out. Gosper called these blattlies. When Greenblatt worked on Gosper's desk and left blattlies behind, Gosper would make a point of washing the area with ammonia. Gosper would also sometimes kid Greenblatt about his awkward speech patterns, his frequent coughing, his poor spelling, his mumbling—even though many of Greenblatt's expressions became integrated into the specific vernacular which all the hackers used to some degree. For instance, it was probably Greenblatt who popularized the practice of doubling words for emphasis—like the times he'd get revved up explaining something to Gosper, Kotok, and Samson, and the words would get tangled up, and he'd sigh, saying, “Oh, *lose-y lose-y*” and begin over. Gosper and the others would laugh—but, like the way a family will take on a baby's speech patterns and cute malapropisms, the community adopted many Greenblattisms.

Despite these odd personal traits, the hackers held Greenblatt in awe. He was the way he was because of conscious priorities: he was a hacker, not a socialite, and there was nothing more useful than hacking. It so consumed him that he sometimes would go six months without finding time to pick up his MIT paycheck. “If he randomly sat around and tried to articulate what he was thinking and doing all the time, he wouldn't have gotten anything *done*,” Gosper would later say. “If he worried how to spell things, he wouldn't have gotten anything written. He did what he was good at. He was a complete pragmatist. What people thought, be damned. If anyone thought he was stupid or nerdy, that was their problem. Some people did, and they were wrong.”

Gosper could appreciate Greenblatt's single-mindedness because his own insistence on graduating (which he did in 1965) had led him to trouble. It was not that his final year at MIT was an academic disaster, because he managed to fulfill the graduation requirements by a slim margin. The problem was a pact he had made with the United States Navy. Before he entered MIT, he'd taken a civil service exam and placed high enough to be included in an exclusive student engineering development program. He worked

summers for the Navy, which paid half his tuition and required him to work there for three years after graduation. When Gosper signed up, there had been an escape clause that allowed you to postpone your commitment if you went to graduate school; and if you could get a corporation to pay off the Navy's three-thousand dollar investment after that, you'd no longer be obligated. But during Gosper's senior year the graduate school loophole closed. Only a buyout would save him, and he didn't have the money.

The prospect of going into the Navy was hideous. During his summer employment stints he had been exposed to a pathetic system that was antithetical to the Hacker Ethic. Programmers were kept in a room totally separated from the machine; sometimes, as a reward for years of service, they would let a particularly obedient worker venture into the computer room and actually see his program run. (One woman, the story goes, was allowed this privilege, and the sight of the lights flashing and disks whirring caused her to faint.) In addition, Gosper's Navy boss was a man who could not understand why the logarithm of the sums in a given equation was not the sum of the logarithms. There was no way in hell Bill Gosper was going to work under a man who did not know why the logarithm of the sum was not the sum of the logarithms.

Then there was Gosper's perception that the Navy was in bed with Univac. He considered the Univac machine a grotesque parody of a computer, a Hulking Giant. The Navy had to know it was a basically phony computer, he figured, but used it anyway—it was a classic example of the inevitably warped outcome of Outside World bureaucracy. Living with that machine would be immersion in hell. Gosper used computers to seek things that no one had ever found before, and it was essential that the computer he used be optimal in every way. The PDP-6 was the best thing he had found so far, and he was determined not to leave it, especially for a dog like the Univac. “If I see a machine has some incredibly stupid thing wrong with it, some error in its design or whatever, it just irritates the hell out of me,” Gosper would later explain. “Whereas the PDP-6 always seemed like an infinitely perfectible machine. If there was something wrong, you would change it. In some sense, we lived inside the damn machine. It was part of our environment. There was almost a society in there . . . I couldn't imagine being without a PDP-6.”

Gosper was determined to find the money to pay back the Navy, and to earn it while working for a company with a PDP-6. He fulfilled these rigid criteria by landing a job with the firm that Greenblatt had worked for that past year, Charles Adams. The fact that the Adams company never quite got their PDP-6 working right (Greenblatt insists that he did his part of the preparation adequately) did not seem to upset Gosper: what freaked him was the fact that Charles Adams scrapped the project and bought a carbon copy of the same Hulking Giant Univac that the Navy had.

But by that time more funding for Project MAC had come through, and Bill Gosper found his way onto the payroll. He hardly had to change his habits, since during his whole stint at Adams he had been working on the PDP-6 on the ninth floor every night.

• • • • •

By then, Greenblatt was in full hacking swing. One of the first projects he worked with on the PDP-6 was a LISP compiler, to allow the machine to run the latest and most nifty version of John McCarthy's artificial intelligence language. Young Peter Deutsch had written a LISP for the PDP-1, but it was not too effective, since the One had less memory; and LISP, which works with symbols and not numbers easily translated to binary, consumes an incredible amount of memory.

Some people, notably Gosper, thought that LISP would be a waste of time on the PDP-6 as well. Gosper was always concerned with what he considered the atrocious lack of computer power in those days, and later would marvel at how ignorant they all were in the AI lab, trying impossible tasks and blaming their failures not on the piddling machines they had, but on themselves. In his senior year, Gosper had been put to work by Minsky on a display that would test whether a certain visual phenomenon was binocular or monocular. Gosper did manage to come close with a clever, clover-leaf shape, which at least *displayed* the phenomenon, but generally was banging his head against the wall trying to make the machine do more than it could do. One of the tasks that Gosper considered impossible was a useful LISP on a PDP-6—it might be nice as a symbol evaluator, but not to *do* anything. He considered

it one of Minsky's follies that Greenblatt and the others had been tricked into implementing.

But Greenblatt saw more. Though he realized that LISP on the PDP-6 would be to some extent a hack, not fully pragmatic, he did see the need to move toward it. It was a powerful language that would help the field of artificial intelligence move forward: it was the language by which computers would do extremely difficult tasks, by which they could actually learn. Greenblatt was just starting then to have a certain vision of the future, an inkling of a technical implementation of the hacker dream. So he and some others—even Kotok came down from DEC—began implementing LISP on the PDP-6. They filled the blackboards of TMRC with layers and layers of code, and finally got it going on the machine.

The crucial sections were written by Greenblatt and another hacker. Two or three people on a project were considered The Right Thing—far fewer than IBM's so-called “human wave” style of throwing dozens of programmers at a problem and winding up with junk. And it was better to rely on two or three people than on a single crusader—so that when one person was at the end of his thirty-hour phase, someone else could come in and keep hacking. Kind of a tag team project.

With PDP-6 MacLISP (named for Project MAC), the hackers began integrating that computer language into their programs, and even into their conversation. The LISP convention of using the letter “p” as a predicate, for instance, was the inspiration for a common hacker style of asking a question. When someone said “Food-P?” any hacker knew he was being asked if he wanted to get something to eat. The LISP terms “T” and “nil” came to stand, respectively, for “yes” and “no.” LISP's acceptance did not diminish the hacker love for assembly language, particularly the elegant PDP-6 instruction set. But as Greenblatt and even Gosper later realized, LISP was a powerful system builder that fit neatly into the hands-on Hacker Ethic.

DEC had shown an interest in MacLISP, and Kotok arranged for Greenblatt and the others to go to Maynard late at night to work on the program, type in their code, and debug it. It was all part of the easy arrangement between MIT and DEC, and no one questioned it. The Right Thing to do was to make sure that any good

program got the fullest exposure possible, because *information was free* and the world would only be improved by its accelerated flow.

After working on MacLISP, Greenblatt was perhaps the most authoritative of the systems hackers on the PDP-6. The new administrator of the AI lab, a young man from the Southwest named Russell Noftsker, had hired Greenblatt mainly to maintain and improve the organic creation that is a computer operating system. But Greenblatt's vision did not stop at systems; he was intensely drawn by the concepts of artificial intelligence. He decided to use the system to actually *do* something in that realm, and, since he had been a chess player all his life, it was only logical that he work on a chess program that would go far beyond Kotok's effort and beyond the other AI chess projects that had been attempted at various labs around the country.

Like any good hacker, no sooner did he decide to do something than he began work on it. No one asked him for a proposal. He didn't bother to notify his superiors. Minsky did not have to ponder the relative virtues of the project. There were no channels to go through because in the mid-sixties, in those early days of the AI lab, the hackers themselves were the channels. It was the Hacker Ethic put to work, and Greenblatt made the most of it.

He'd seen a game played by the Kotok program and thought it was crap. Basically, those guys did not know how to play chess: swayed by the romance of a computer making moves, they had somehow forgotten the idea that the name of the game was to take the other guy's pieces. Greenblatt's program used sophisticated artificial intelligence techniques to try and figure out moves in accordance with certain criteria that he considered good chess. Working with a couple of other hackers, Greenblatt went on a coding blitz. He'd manage to get four hours of PDP-6 time a day, and he'd keep writing offline when he wasn't on the machine. He got the program actually playing chess in one week. The program was debugged, given features, and generally juiced up over the next few months. (Greenblatt was eventually offered an MIT degree if he would write a thesis about his chess program; he never got around to it.)

Circulating around MIT around 1965 was a notorious Rand Corporation memo called "Alchemy and Artificial Intelligence."

Its author, an academic named Herbert Dreyfus, lambasted the field and its practitioners. To hackers, his criticism was particularly noxious, since the computer was their implicit model of behavior, at least in their theories of information, fairness, and action. Dreyfus focused on the computer's ridiculously limited structure (compared to the structure of the human brain). His coup de grace was the blunt assertion that no computer program would be able to play a good enough game of chess to beat a ten-year-old.

After Greenblatt finished his chess program, called *MacHack*, MIT invited Dreyfus to play the PDP-6. The hackers gathered round to watch the computer surrogate of Richard Greenblatt play this cocky, thin, red-headed, bespectacled anticomputer opponent. Artificial intelligence pioneer Herbert Simon, who watched the match, later was quoted as saying that it was

... a real cliffhanger. It's two woodpushers ... fighting each other ... Dreyfus was being beaten fairly badly and then he found a move which could've captured the opponent's queen. And the only way the opponent could get out of this was to keep Dreyfus in check with his own queen until he could fork the queen and king and exchange them. And the program proceeded to do exactly that. As soon as it had done that, Dreyfus' game fell to pieces, and then it checkmated him right in the middle of the board.

Peter Samson later recalled the scene immediately following Dreyfus' loss: the defeated critic looked around at the assembled MIT professors and hackers, including a victorious Greenblatt, with a look of puzzlement. Why weren't they cheering, applauding, rubbing it in? Because *they knew*. Dreyfus was part of that "real world" that couldn't possibly comprehend the amazing nature of computers, or what it was like working with computers so closely that a PDP-6 could actually become your environment. This was something which Dreyfus would never know. Even Minsky, who never really immersed himself in the thirty-hour-day, seven-day-week assembly-language baptistery, had not experienced what the hackers had. The hackers, the Greenblatts and the Gospers, were secure in having been there, knowing what it was like, and going back there—producing, finding things out, making their world different and better. As for convincing skeptics, bringing the outside world into the secret, proselytizing for the Hacker Ethic—all that was not nearly as interesting as living it.

Want to read more?

You can find this [book](#) at oreilly.com
in print or ebook format.

It's also available at your favorite book retailer,
including [iTunes](#), [the Android Market](#), [Amazon](#),
and [Barnes & Noble](#).



O'REILLY[®]

Spreading the knowledge of innovators

oreilly.com