WORKING SMART

HOW TO USE MICROCOMPUTERS TO DO USEFUL WORK

Colin K. Mick

Working Smart

How to Use Microcomputers to Do Useful Work

Colin K. Mick
with the assistance of
Stefan T. Possony

MACMILLAN PUBLISHING COMPANY
A Division of Macmillan, Inc.
NEW YORK

Collier Macmillan Publishers
LONDON

Macmillan Publishing Company
866 Third Avenue, New York, N.Y. 10022

Collier Macmillan Canada, Inc.

Printed in the United States of America

printing number
1 2 3 4 5 6 7 8 9 10

Library of Congress Cataloging in Publication Data

Mick, Colin K.
Working smart.

Includes index.
1. Microcomputers. 2. Microcomputers—Programming.
3. Computer literacy. I. Possony, Stefan T. II. Title.
QA76.5.M5213 1984 001.64 83-43009
ISBN 0-02-949580-6

Working Smart

Acknowledgments

This book would not have been possible without the assistance and support of Ulla Mick and Betyar Mick, who hung in there through the learning pains of personal computers and the birthing pains of this book. Gerald Papke and Eileen DeWald of Macmillan provided considerable emotional and technical support throughout the extensive process of getting the manuscript published.

Introduction

I saw my first computer in the early 1960s when I was an undergraduate at the University of Alaska. Actually I didn't *see* it; I peeked at it through an open door to an electrical engineering laboratory. I vaguely knew what it was, and that one had to be an engineer to play with it. I experienced a brief pang of regret that I had abandoned engineering and physics for anthropology and journalism. But it was too late for me to change my major once again.

I encountered my next computer several years later while a graduate student in communication at Stanford. Things weren't going so hot. The faculty at the Institute for Communication Research and I were having disagreements over whether I was going to remain a graduate student. And then, during the third quarter I took a required class on data analysis. It involved learning to run statistical programs on Stanford's new IBM 360/67. Having stronger mathematics skills than the rest of the class, and being less intimidated by machines, I ended up as the star of the data-analysis class. The professor's final report read: "You spent more money than the rest of the class combined. We winced, but we paid it." Little did "we" know.

The Institute had an unofficial policy of designating one student as the computer expert—to run analyses for the faculty and help the other students cope with IBM 360/67. The current "consultant" was getting ready to graduate, and I was soon drafted to help input and edit papers for advanced graduate students and faculty members. My future at the Institute began to seem more secure. I got the job because of my text-processing skills—a cinch for someone who had supported himself as a journalist.

During the following summer I received novitiate instruction. The IBM 360/67 sat in a sealed room. You could look at it through the windows, but you couldn't touch. Most instruction was by trial and error. I learned computers can't spell, and are intolerant of poor typing ability. I had to learn FORTRAN to conduct a computer analysis of educational television programming. I also had to learn about arrays, control language, and loops. And, I learned to write a special purpose program to generate all the tables we wanted in one pass. I rewrote the program three times. In final form there was a respectable stack of boxes full of computer cards, 1500 cards for the program alone and another 10,000 cards for the data, which I hauled to the machine on a hand truck. The program printout measured three feet high—we used the hand truck to haul it back to the office. My future as a graduate student was secure!

Power corrupts. As the only computer expert in a group professionally committed to data analysis, I was easily corrupted. In two years I was the department's consultant, controlling one of the largest academic computer budgets on campus.

In 1970 I designed a course called "Computers for Duffers," to teach people how to make computers work for them, particularly in their writing and data-analysis work. At this time academic computing was divided into two camps: the computer scientists, who viewed the computer as a big toy and invented creative, but generally unproductive things for it to do; and the number crunchers, who saw the computer as an enormous calculator. Both approaches emphasized using the computer for specialized tasks. I was showing people how to use the computer as a general-purpose tool to make their work and life easier and a little less tedious. Students took the course and went through the motions, but few seemed to get the message. One student summed it all up by saying, "I don't give a damn how the computer works, just so it gives me a significant F." Who could argue with a conceptual framework like that?

After graduate school I began research in the field of information behavior—how people create and use information and how new technologies influence those processes. I had been writing with and studying about computers since 1967, but their benefits were still unavailable to the masses. Then, the December 1975 issue of *Popular Electronics* heralded the new age. Featured on the magazine cover was the Altair 8008—the first commercially available microcomputer.

I immediately began following the development of microcomputers. I already had a word processor—a full-screen VYDEC 1146. (This may not impress you, but to me it was like a Rolls convertible parked in the driveway.) I awaited the time when the micros would bring the advantages of computers to those who needed them most—information workers.

You have to understand that to a user of big, macho IBM mainframes, a micro looks the way a rubber duck does to the captain of a cruise ship. It's an interesting toy, appealing to the uninitiated, perhaps, but hardly suitable for my needs. I mean, I once ran a statistical analysis that required a 750 by 750 matrix. It took every bit of memory we could squeeze out of the Stanford 360/67, and tied up the damn machine for forty minutes. There's no way I can do my work on a little bitty 64K machine, I thought.

The reality was, however, that I had run that giant program only once. I did large scale statistical analysis once a year. Most of what I did was writing, processing simple calculations, preparing budgets, federal research grant proposals, and keeping track of my growing collection of articles, books, journals, loose information, and appointments. I was writing on my word processor (a big, pretty, but very dumb micro). I was budgeting on an electronic calculator (a small, very dumb micro). I was filing, indexing, and scheduling by making piles on my desk and occasionally throwing them into file cabinets and boxes by myself (a slightly overweight, dumb contract researcher and consultant).

Finally, in the summer of 1980, I became convinced that the microcomputer was mature and I should buy one. Feeling insecure, I talked to a few colleagues, and we all started investigating the field. We bought magazines—*Byte, Interface Age,* and *Kilobaud*—to read the microcomputer reviews, not just the ads. (This is difficult because it seems that 75 percent of the column inches in these magazines are filled with ads promising everything.) It was clear an incredible treasure existed for the taking. The only problem was deciding what to select from this tremendous pile of

goodies. The magazines were no help at all. The writers were all from the indiscriminate, high-fidelity school of reviewing.

We decided to go out into the world and see for ourselves. We went to computer shows and visited computer stores. We talked to people. We learned a number of great truths and a few GREAT TRUTHS. We learned that a lot of salesmen don't know very much about computers. Remember now, we are pros. I can talk BASIC, FORTRAN, ALGOL W, PL/1, and smatterings of SNOBOL and LISP. My hunting companion Jesse Caton could talk assembly language, knew the difference between a compiler and an interpreter, had even written his own language, and was director of computing at a small research company. Between us we destroyed the egos of many sales representatives. We learned by not talking with them. Instead, we played with the machines.

After months of indecision, we finally moved. We agreed to bite the bullet and build three S-100-based systems. We decided to buy the components separately and integrate them ourselves, trusting our own abilities. Jesse could handle the software, and I could solder and puzzle through the instructions and assembly. The third member of our group, Bob Mason, was an honest-to-God Ph.D. engineer. There was no way we could fail.

Our logic went like this: By building the system ourselves we would save money and learn more about microcomputers. We had minimum dollars to spend and we wanted to get the most computing power we could. We were buying three systems at once, so we started negotiating with the mail-order house that advertised the lowest prices. We kept at it until we finally got a full S-100 system—CPU board, memory board, disk controller, and mainframe—two disk drives, and cabinets for just over $2,600. We made the buy.

Then we learned another great truth: A mail-order house advertisement does not guarantee that a component is in stock. Over the next two months equipment arrived sporadically, interspersed with frequent calls to discuss FTC regulations, nonperformance, and return of equipment with mail-order house representatives.

When all the components finally arrived, Jesse and I assembled them and got them working. To our surprise, the machine actually came up quickly. But as we became more sophisticated (we pros "sophisticate" rapidly) we realized our systems were performing at less than optimal levels. While we puzzled over

performance, fate stepped in and dealt me an ace named John Mason—my sister's fiancée. While fumbling through the social amenities of our first meeting, I interjected my experience with the new computer. John assumed a somewhat amused expression.

As it happened, he is a computer designer—not just any computer designer—but the cofounder of the company that had manufactured our computer equipment. And he had designed all of it.

John put my equipment through one of the most drastic transformations in the history of computers, and I embarked on a learning curve that would shame 99 percent of the world's students. I got exactly what I needed. I already knew computing from the user end. What I learned from John was the business end of the computer industry: how they are designed, made, and priced; what distinguishes good and bad products. He gave me an inside view of the industry—one that few people ever get.

During this learning process, I was using my new micro increasingly in my work. I transferred my writing from the word processor to the micro the first week it was up. I initiated the filing operations a few weeks later. But the most mindboggling advance was in budgeting, which is, after all, just an exercise in basic mathematics: adding, subtracting, multiplying, and dividing. For a research proposal you usually prepare both a summary budget on a standardized form and a breakdown that details costs by category and task. The goal is to arrive at matching totals on the rows and columns of the detail budget, and have them equal the final total on the summary budget. In ten years of proposal writing my totals agreed only once, and that was when I had made fortuitous mistakes in both budgets, discovered by someone who checked my figures.

For my first budgeting exercise with the computer I used a program called Forecaster, which allowed me to create basic budget worksheets and then vary the data that went into them. It was amazing. I could change an estimate and push a button, and out popped a new set of budgets. The row and column totals were equal. The final total of the detail budget matched that of the summary budget. Two hours of work and fifteen feet of calculator tape were replaced by one minute and a few strokes of the key. Far out! I was hooked. I started thinking about information behavior and the new micros. I started working my theories into my

lectures. I bought a second micro for my home, so I didn't have to go into the office whenever the creative urge struck.

All this background brings me finally to this book. In the midst of all this enthusiasm I was talking with Jerry Pournelle, who writes a personal computer column for *Byte* and is a fellow enthusiast on the subject. He asked if I would be willing to assemble a machine like mine for his friend Steve Possony, and help him learn to use it. Nothing to it, I thought. I'll whip this one off in nothing flat, save Jerry some bucks, make a few myself, and also do him a favor. Little did I know.

Jerry, Steve, his wife Regina, and I got together a few weeks later for a test drive on the computer. Steve got interested, and the great undertaking lurched into motion. Assembling the system took longer than I planned. (It always does.) Even with John's help it took time to acquire the components.

Thoroughly intimidated by his test drive, Steve wisely decided to pick up some pointers before he got his machine. He agreed to come by my office occasionally for an hour or so of instruction. After the first few sessions, Steve got hooked. That's not unusual—it happens to everyone who isn't a latent Luddite. Steve quickly discovered that he had found a tool for cutting his workload by a high order of magnitude. He had been writing by dictating to Regina, irritating her by returning totally obscured edited texts and demanding retyped copies. The thought of generating documents by keyboard instead of by dictation didn't thrill him—he was used to giving dictation. What got him hooked to the computer was that it made editing so much easier. He could revise the text, move blocks of it around, and go through permutations in a matter of minutes.

The problem was that Steve didn't behave the way I expected. I had been living with computer technology for so long that its jargon was part of my vocabulary. I was used to working with people who, as information professionals, interact with computers on a regular basis. To Steve I was talking in a foreign language whose vocabulary he had never heard before.

Steve represents a generation that did not grow up with electronic technology. The computer to him was an abstract concept, as it had been to me when I saw the machine in the engineering lab twenty years ago. His self-assurance, however, allowed him to tell me abruptly when he didn't understand what I was saying. His Germanic stubbornness and attention to detail forced him to persist in asking questions.

Steve initially tried the traditional approach to learning. I was teaching him WordStar, a word processing program. He read the entire manual. From cover to cover. He must be the only person in the world to accomplish that thankless task without being paid for it. It didn't help. Bad as our communication problems were, they were nothing compared to penetrating the obscurities of the WordStar manual (much less any other microcomputer manual).

In frustration Steve starting writing down his perceptions of what I was telling him and of what the manual said. He also listed what he didn't understand. (Naturally, he wrote all this as part of his WordStar practice exercises.) His text confirmed that I had fallen victim to jargonitis—the dread disease of the technical communicator. I also began to understand Steve's frustration as that of any novice encountering the computer world for the first time.

I began writing responses to Steve's questions, formulating instructions and tutorials on what I thought he should know to use his system effectively. Jerry Pournelle (Remember him? Steve's friend, the microcomputer writer) monitored all this through telephone conversations with Steve and me. With characteristic entrepreneurial spirit he suggested we expand the lessons into a book.

Most of the material in this book is based on dialogues between Steve and me. Some has roots in the computer-literacy course I taught at Stanford. Some comes from the seminars and workshops I have taught on micros, information behavior, and the use of computers to support information work. All of it has been thrown at Steve, who listened, interpreted, inquired persistently, and synthesized. I shouldn't forget Regina. Steve is a classic European intellectual—left-brain dominant—who likes his concepts broken down so he can understand them. In . . . small . . . digestible . . . pieces. Regina is right-brain dominant. She works with gestalts and feelings. She learned to use the system along with Steve, but she saw it very differently. I have tried to preserve some of her insights here as well.

This book is organized in seven sections:

Section I: A discourse on the usefulness of personal computers.

Section II: Personal-computer applications at work, in education, and in the home.

Section III: Personal-computer hardware. A description of the physical components of a personal computer system and an explanation of how they work and interact.

Section IV: Personal-computer software—the instructions that tell the computer what to do. Description of types of software—operating systems, utilities, languages, and applications programs—and what they do. Emphasis on practical applications.

Section V: How to get started in personal computing. Suggestions on buying hardware and software.

Section VI: How to put a system together and use it. Guide to developing good computer work habits and avoiding problems. Preventive maintenance and trouble-shooting.

Section VII: Working smart: a philosophical discourse on the impact of personal computers on society.

Because the seven sections and their chapters are somewhat independent of each other, it is unnecessary to read them in sequential order. When writing about a technical topic it is difficult to entirely avoid using jargon. I've tried to purge the text of as many undefined terms as possible, but sometimes there just aren't any appropriate synonyms. Since one purpose of this book is to advance your computer literacy, I've included an extensive glossary, which should accelerate your fluency in "computerese."

Contents

SECTION I

INTRODUCTION

This section provides a philosophical introduction to personal computing and a discussion of how best to use personal computers for business and professional support. It includes guidelines to help you decide if you should invest time and money in acquiring a personal computer and, if you do, to learn how to use it productively.