

# The Amateur Computerist

Webpage: <http://www.ais.org/~jrh/acn/>

June 29 1988

Volume 1 No. 2

## THE BIG MACHINE

by Floyd Hoke-Miller

The white-hot sparks a flyin'  
An' Bulls a' standin' 'round  
A thinkin' 'bout their bonus  
If speed-up plans are sound.

They're puttin' in a big machine  
With cats an' elbow grease  
An' when the jigs are set  
One dozen jobs will cease.

A dozen men to take the air  
An' line up for the dole—  
The big machine is master  
To pawn the worker's soul.

Each man of toil a glancin'  
In pangs of sheer regret  
'Cause most of modern methods  
Are made the workers' debt.

---

# Table of Contents

The Big Machine .....	Page 1
Pass the Profits, Please.....	Page 2
Technology Editorial. ....	Page 3
Sample BASIC Program. ....	Page 6
Try This (IBM).....	Page 6
Telecommunication Correction. ....	Page 7
German Vocabulary Helper Program.....	Page 8
Programming in BASIC or C?.....	Page 10
Configuring Your System.....	Page 15
Letter to the Editor.....	Page 20

---

## **PASS THE PROFITS, PLEASE** **A Treatise on the Shorter Hour Contention**

by Floyd Hoke-Miller

**SHORTER HOURS** – Indeed it is the only answer less hours means more workers.

Shorter hours is not just a Labor necessity. It is an economic and social must to prevent a complete collapse of any industrial society. The reason being patent, it is a “for hire” framework completely.

You are still working an 8 hour day gained by a hard struggle of the WWI. But the ratio of productivity and services per man-hour wages have lost any semblance of equation – my assumption would be about a thousand to one in favor of the controlling few, as against the working element, mental or muscular as long as they are under “for-hire” pattern.

We, as the working class, are the “low men on the totem pole” or like Atlas “bearing the world on our shoulders.”

To accomplish these things we need more thinking and less “sweeping-under-the-rug” sort of human behavior. There is definitely no way to change the face of the world equitably unless you are able to change the hearts of men. All levels of society need this kind of restructuring.

In the phraseology of contractual language, we need the application of COLA (Cost of Living) and less “payola.” For instance, the kind of payola I am referring to is that which Bill Casey and some of the ilk like

Colonel North and Admiral Poindexter were found to participate in as revealed by the Senate Select Committee. Their hearings recently did help to begin to find a solution for the constitutional crisis or country is in.

Since this treatise is meant to deal with the amateur and the professional, let me differentiate. The amateur is dedicated to the love of computers for the sake of the common cause. The professional is dedicated to money and self aggrandizement. The computer can be and often is a bane instead of a boon. It might be well and wise to consider the changes that have come about for mankind beginning with the end of the Dark Ages and continuing to the Age of Enlightenment and the Machine Age. We have come a long way, but the gains have been in a state of imbalance the minority is still skimming the milk, so to speak.

---

## **TECHNOLOGY: TO DEVELOP OR STAGNATE?**

This second issue of the *Amateur Computerist* newsletter was to be published on May 29, 1988. (We are one month late.) This date was chosen to commemorate the 40<sup>th</sup> Anniversary of the introduction of the Cost of Living Adjustment (COLA) and the Annual Improvement Factor (AIF) into labor contracts in the U.S.

These contract gains were first agreed to in the GM-UAW contract signed May 29, 1948. They were won as a result of a hard fight waged by UAW rank and file workers. But the story of how these gains were won is shrouded in darkness.

After WWII, the major U.S. companies planned to scale back production. They wanted to limit industrial capacity so as to keep profits up and wages down. UAW rank and file members, however, challenged this scenario. Writing in local union newspapers like "The Searchlight" (the newspaper of UAW Local 659 in Flint, Michigan), UAW members like Howard Foster, dispelled the myth that higher wages would just lead to higher prices. He explained that the price of an article depended on "the number of hours of labor workers had to put in on it" using the

most advanced technology. He wrote:

“You may as well know it. There is no ‘law’ of supply and demand. Under competitive conditions it is value which will determine price over a period of time. Value is determined by the amount of necessary human labor required to produce the commodity under the latest technological development. Prices will fluctuate around value. When supply is below demand, prices will temporarily rise above value. If supply becomes greater than demand, prices will fall below value until production is reduced. That value is the basic determining factor in determining price can be seen when we ask ourselves what the price would be when supply and demand are equal. It is obvious, price would then equal value.”

(The Searchlight, Aug. 14, 1947, p. 7)

If higher wages didn't cause higher prices, then why did employers so hardly oppose them? This UAW member explained:

“Wages don't determine prices, but they do affect profits. Both profits and wages come from one source – the value of the goods produced by labor. This means that if pay is raised it must come out of profits. Likewise, if wages are cut profits go up. That is why employers never willingly give increases. They know it comes from their pockets. They know it is wages and profits that are tied together, not wages and prices.... The fact that they raise such a fuss every time we ask for more money shows that they don't believe their own argument.”

(The Searchlight, June 5, 1947, p 3)

Companies using the most advanced technology can afford to pay high wages because with advanced technology, less of a worker's time is needed to produce a product. And the price of the product has historically fallen when a company adopts advanced technology. This UAW columnist describes why:

“In fact, when mass production methods were introduced in the automobile industry, the price of cars went down. This was because the labor time on each car was greatly reduced. Yet we automobile workers got higher wages through our union.”

(The Searchlight, June 5, 1947)

The discussion of the relationship between wages and profits and prices and technology that appeared over 40 years ago in the UAW press, led to the introduction in the 1948 contract of two gains that helped establish the basis for the technological development of U.S. industry. The COLA established a wage level in the auto industry that meant it was profitable for the auto companies to invest in new technology. Low wages would have acted as a disincentive to investment as it would then have been cheaper to hire another low paid worker, rather than replace a relatively better paid worker with a machine. The AIF language in the UAW contract by 1950 explains this commitment to updating technology, rather than lower wages. The contract says that the wage gain “depends on technological progress, better tools, methods, processes and equipment and a cooperative attitude on the part of all parties in such progress. *It further recognizes the principle that to produce more with the same amount of human effort is a sound economic and social objective.*” (Our emphasis)

Yet as of 1982, there has been an effort to reverse this commitment to updating technology. Management now tries to claim that the AIF in the UAW contract means that they should get more and more labor out of their workers at the same or lower wages, so they can make more profit. They claim that cooperation means helping them to make greater profit with more and more stagnant technological conditions, rather than the commitment to high wages and the updating of technology. But as the progress from 1948 on demonstrates, it is high wages and a commitment to new technology, not high profits, that leads to social progress.

COLA and the AIF were contractual provisions whereby the UAW was able to force the American auto industry to develop advanced technology. With the attack on these gains by U.S. management, the press, government etc., the technological competence and advancement of U.S. society is being stymied. Thus it is important that COLA and the AIF be defended and their achievement 40 years ago, be celebrated.

---

# Sample Basic Graphic Program

by Morris Darga

(This program should work on any computer - Editor )

```
1 PRINT
2 PRINT
3 PRINT
4 PRINT
5 PRINT
6 PRINT
7 PRINT
81 PRINT TAB(12);";"\ "
82 PRINT TAB(12);"*****~"
83 PRINT TAB(11);"*****~"
84 PRINT TAB(10);"*0*****~"
85 PRINT TAB(10);"*****~"
86 PRINT TAB(9);"*****~"
87 PRINT TAB(8);"****  *****"
88 PRINT TAB(7);" @*/  *****"
89 PRINT TAB(13);"*****\" TAB(37);" (( "
91 PRINT TAB(9);"*****/*****\" TAB(36);" (/"
100 PRINT TAB(8);"*****/*****\          /\)"
110 PRINT TAB(7);" ***      ****/*****\      /\)"
120 PRINT TAB(6);" ***      ***      *****/)"
130 PRINT TAB(5);"/_" TAB(11);" ***      *****/"
140 PRINT TAB(10);"***" TAB(22);"*****"
150 PRINT TAB(9);"***" TAB(24);"*****"
160 PRINT TAB(8);"/_" TAB(26);"*****"
170 PRINT TAB(27);"*****"
180 PRINT TAB(26);"****  ****"
190 PRINT " HELLO *****" TAB(26);"***  ***"
210 PRINT TAB(24);" <_  <_"
999 END
```

---

## TRY THIS (IBM)

by Michael Hauben

(From DOS, type in BASICA)

```
1 cls:key off
```

```
10 gosub 5000
25 cls: screen 2
30 for L= 1 to 25
40 line (x1 + L, y1 + L) - (x2 - L, y2 - L),,b
45 line (x1 + L, y1 + L) - (x2 - L, y2 - L), 0, b
50 next
60 end
5000 print "Enter your coordinates for the box:"
5015 locate 7, 12: input "Top left corner (x1, y1):", x1,
y1
5020 locate 9, 12: input "Bottom right corner (x2,y2):",
x2, y2
5030 return
```

---

## **The World of Telecommunications Corrections and Additions**

The following bbs's listed in the last issue are no longer functioning:

PC Playhouse 313-381-8633  
Chessie's BBS 313-291-2160

Following are additional BBS's  
The BallRoom BBS 313-885-2766  
Discussion and IBM files  
Sietch Tabr 313-255-2767  
Discussion and IBM and Amiga Files  
Fallout Shelter 313-464-4417  
Discussion  
Treasury Chest BBS 313-226-2567  
IBM Files  
Nightmare 313-563-1390  
Discussion  
Promegia 313-386-5301  
For Amiga

Apple Gram 313-292-0389  
For Apple  
Atlatsis BBS 313-936-6339  
Echo-Mail and Conferences and Discussion  
Arthur 313-879-2318  
Discussion, Files for IBM and CoCo

---

## GERMAN VOCABULARY HELPER

(This program will give you a form to study German nouns – their plurals, gender, and meaning – and German verbs. This program can be typed on an IBM compatible by installing BASICA from DOS and then typing in the following lines.)

```
10 REM THIS PROGRAM IS TO HELP STUDY GERMAN VOCABULARY
20 REM DATA IS FROM LINE 320 ON.
25 REM ADD NEW VOCABULARY WORDS FROM LINE 450 ON, USING
FOLLOWING PATTERN
27 REM DATA DIE, ABFAHRT, ABFAHRTEN, DEPARTURE
29 REM DATA ARTICLE, SINGULAR, PLURAL, MEANING
30 KEY 1,"a":KEY 2,"u":KEY 3,"e": REM DEFINE KEYS FOR
UMLATES USING ASCII # 132, 129, AND 137. HOLD DOWN
ALTERNATE KEY, TYPE ASCII NUMBER ON NUMBER PAD, THEN
RELEASE ALTERNATE KEY.
40 FOR V=4 TO 10:KEY V,"":NEXT
45 KEY ON
50 N = 10: REM NUMBER OF WORDS. CHANGE THIS IF YOU ADD
WORDS
60 CLS
70 PRINT"use F1 for a, F2 for u and F3 for e ":PRINT:
PRINT
80 FOR T = 1 TO N
90 READ ARTICLE$, NAM$, PLURAL$, MEANING$
95 KEY 10,NAM$
100 PRINT TAB(15)NAM$
110 PRINT:PRINT:PRINT:PRINT
115 IF PLURAL$="none" THEN PRINT "This word isn't a Noun
so it can't have a plural.": GOTO 165
```



```

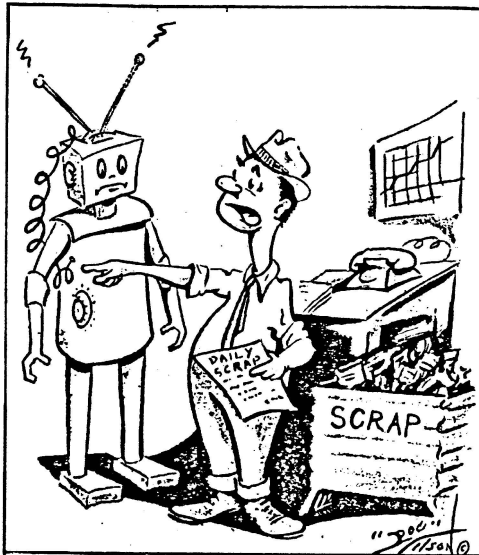
120 PRINT TAB(15); "What is the plural";
130 INPUT P$
135 PRINT
140 IF P$ = PLURAL$ THEN PRINT TAB(15)"Du hast recht!":
GOTO 165
150 PRINT TAB(15); "falsch": PRINT: PRINT: PRINT "The
correct plural for ";NAM$; " is ";PLURAL$
165 IF ARTICLE$ = "none" THEN PRINT "This word is NOT a
NOUN so it doesn't have an article.": PRINT: GOTO 220
180 PRINT: PRINT TAB(15); "What is the gender?(der/die/or
das)";
190 INPUT A$
195 PRINT
200 IF A$ = ARTICLE$ THEN PRINT: PRINT TAB(15)"Du hast
recht!": GOTO 230
210 PRINT TAB(15); "falsch": PRINT: PRINT "The correct
gender is "; ARTICLE$; " "; NAM$
230 PRINT: PRINT TAB(15); "What is the meaning";
240 INPUT M$
245 PRINT
250 IF M$ = MEANING$ THEN PRINT TAB(15)"Du hast recht!":
GOTO 280
260 PRINT TAB(15); "falsch": PRINT: PRINT "The correct
meaning for ";NAM$;" is ";
280 PRINT: PRINT: PRINT: INPUT "Do you want to try another
(Y/N) :"; ANSWER$
290 IF ANSWER$ = "N" OR ANSWER$ = "n" THEN END
300 CLS
310 NEXT T
320 REM DATA FOR ENTRIES
330 DATA die, Abfahrt, Abfahrten, departure
340 DATA die, Abfahrtszeit, Abfahrtszeiten, departure time
350 DATA der, Auslander, Auslander, foreigner
360 DATA der, Flug, Fluge, flight
370 DATA die, Frau, Frauen, wife
380 DATA der, Himmel, Himmel, sky
390 DATA die, Richtung, Richtungen, direction
400 DATA none, umsteigen, none, to transfer
410 DATA der, Prospekt, Prospekte, brochure
420 DATA die, Mahlzeit, Mahlzeiten, meal
430 REM Add more data from here if you wish.
440 REM For each new line of data you add, add 1 to the N
in line 80.

```

---

# PROGRAMMING IN BASIC OR C?

The personal computer is not a dedicated word processor or



Turn in your Robot ears, Bud;  
You're All Done!

Cartoon by "Doc" Wilson

~~spreadsheet machine or game machine or desktop publisher.~~ These are all important uses for the computer. But the wonder of the personal computer is it is an all-purpose machine. Its capabilities are only beginning to be understood. Though there has been an effort to create such a machine since the 1600's and though mathematicians and scientists and science fiction enthusiasts dreamed of its achievement for hundreds of years, it is only 14 years old. With the appearance of the

Mark-8 created by Jonathan Titus on the front cover of the July 1974 issue of *Radio Electronics*, the personal computer became a reality. Not only is it rapidly developing technology, it is also in people's homes. They can experiment with it and try to get it to serve uses they have in mind.

But to be able to take advantage of the all-purpose capability of the computer, to be able to customize the personal computer to one's own needs, it's important to know a programming language.

There are many programming languages. Originally computers used a form of input in 1's and 0's known as binary notation. But coding lots of 1111100000's etc. was a very difficult process for the programmer, and thus other languages were created. By the 1950's, a language called Fortran (Formula Translation) was created. It greatly simplified the job of the programming. By the 1960's two Dartmouth University professors, Kemeny and Kurtz, created the language BASIC, (Beginner's All Purpose Symbolic Instructional Code). Their object was to make it possible for beginners to write programs, without having to learn the intricacies of previous programming languages.

When the personal computer – the Altair 8800 – became available in 1975, two computer hackers, Bill Gates, and Paul Allen realized there was a need for a language interpreter. They set to work and created a BASIC interpreter. People could now use their computers for their own purposes.

The history of the development of home computers shows that people who want to be able to deal with their computers will benefit from learning a programming language.

The question of what programming language is it worth learning is often asked. Any programming language that will work on one's own computer can be useful, but there are 3 that seem particularly available for home computers.

For beginners, BASIC would seem the most helpful choice. Most personal computers come equipped with a version of the BASIC language. Also, BASIC is the only interpreted language, and thus you get the quickest feedback if you are typing in a program. If you type in a line in BASIC such as:

```
10 PRINT "HELLO"
```

and you run the program, the machine converts that line into machine code, tests it for errors in syntax, and gives you feedback if the line is not usable by the computer immediately. For example, if you typed in a line such as

```
10 PRIN "HELLO"  
RUN
```

The computer will give you back

```
SYNTAX ERROR
```

The word **PRIN** is not a word that the computer is set to recognize when it is functioning in the BASIC language and the computer cannot figure out that you meant to type **PRINT** rather than **PRIN**.

Thus it's easy to learn when you are making errors when you are working in BASIC. It is easy to learn to correct the errors. Also, BASIC is easy to learn, as it was created for the Beginner. And it lets the user access the computer's functions and capabilities with a minimum of experience and knowledge.

If you were to begin to learn BASIC, you would have to get into the mode in which your computer can use the language. For IBM PC variety machines, you must have the DOS disk in the disk drive and you type in **BASICA** and hit the return key. (With MS DOS, you type in **GW BASIC**). In an **APPLE 2-E** or **Commodore 64** you are in BASIC when you turn on the machine, as long as there is no software application disk in the machine.

Once in the BASIC language, if you were to type in

```
HELLO WORLD
```

and hit the return key, the computer would return

```
SYNTAX ERROR
```

Why?

**Hello World** is not something the computer can understand. The computer, contrary to all its wondrous capabilities, is only a machine that needs to have switches turned on or off for it to be able to perform various functions. In a computer, the switches are set via typing in various combinations of keystrokes from the computer's keyboard, rather than playing with wires or circuits directly.

If you want to get the computer to be able to deal with putting the message "Hello World" on the screen, you need to tell it to do so by

typing in:

**PRINT "Hello World"**

Once you hit the return key, the computer should return:

**Hello World**

But had you typed in:

**PRINT Hello World**

The computer will not return the words. People are often surprised that forgetting a “ at the beginning of the words you want put on the screen will change what the computer will do. But for the computer the “ is translated into a 0 or 1 switch and forgetting to set the proper switch on any machine does have significant effects. Thus the importance of putting in an “ shows that the computer is merely a machine that one must set the switches on.

If you want to get the computer to put any other message on the screen, just type it in place of the words Hello World. As long as the message you want is enclosed in quotes, the computer will put it on the screen. (As long as it is not more than 256 characters or whatever number of keystrokes your computer will allow you to put between one set of quotes.)

If you want to try to put this same message on the screen in another computer language, you will have to learn the codes the computer recognizes in that language.

For example, Atari, IBM, APPLE, Commodore, etc. all can also work in a language known as Assembler. That language is a fast language as it requires the programmer to put in code that is close to what the machine can handle. The machine has a minimum of translation of the code left for it to do. But IBM and APPLE, taking two examples, are based on different computer chips, and thus their assembler language's are different. There is another language, however, that combines some of the versatility of Assembler, but the code is relatively transportable, from one machine to another. That language is “C.” Thus whether the computer is an IBM or a UNIX machine, an APPLE machine, etc. the kind of “C” code one learns will be similar. Thus “C” has gained popularity among programmers in recent years.

“C” programming language was created by Dennis Ritchie for the DEC PDP-11 computer in the 1970's. It was designed to be used with

the UNIX operating system. But it is now being used on IBM machines under DOS, on Amiga's etc. It is helpful, however, when starting to learn "C" to have some previous background in programming in some other language so that one is already familiar with concepts like "assignment statements," "loops," "functions (i.e. subroutines)," etc.

To be able to get the computer to print "Hello World" on the screen using the "C" programming language, one would have to purchase a "C" program language like Turbo "C" from Borland or Quick "C" from Microsoft. Using Turbo "C" (which is possible to use without a hard drive), while Quick "C" I've been told requires a hard drive, one would have to read through the manual and set up a set of disks so you have one disk for use as a work disk and one for libraries. Then working in the drive where the work disk is located, you would need to type in:

### **tc hello.c**

The tc will load the program tc (which is short for turbo c). "hello.c" is just the name I chose for my first program. You can choose other names, as long as you use ".c" after the name.

Once you hit the return key, you will find yourself in the turbo "C" editor. Now it is necessary to set up the program. This will require what is called a "main function." (In "C" the word function represents what is known as a subroutine in BASIC.) Each program in "C" must have a main function, and then it can have numerous other functions. Keywords in "C" are typed in lowercase, as the UNIX operating system has been based on using lower case commands and "C" was originally written for a UNIX environment.\*

Now you can type in:

```
main()
{
printf("hello world\n");
}
```

To save this program in turbo c, you hit f2 on your function keys, or RUN on the choice given on the Editor. Then the program will be compiled and executed and you should see on the screen

**hello world**

In "C," you use the word "printf" to tell the computer you want to print something. The set of characters or "string" as it is known in

computer language is then enclosed in (“.....”);If you want to skip to a fresh line after printing your string “hello world,” you put in \n inside the quotes.

Each function in “C” has to be enclosed in curly brackets { goes at the start of the set of instructions. } ends the instructions. And the keystrokes main() indicate that you are typing in the function called main. () indicates what is being dealt with by the computer is an integral whole known as a function.

Try experimenting with either BASIC or “C.” Try to print out some message on the screen. Experiment by putting each word on separate lines. The main lesson of learning to program, whatever the language, however, is that the computer can only do what some human being tells it to do. Thus for the computer to be able to achieve the wonders it is capable of, human beings need to learn the computer’s capabilities and its limitations. The computer is an important tool, but it is not to make people proof computers, but rather for people to learn as much as they can how they can make the computer serve their needs.

(\* Note: A useful book for learning “C” is *The C Programming Language* by Brian W. Kernighan and Dennis M. Richie. (1978) For learning BASIC, *A Guide to Programming the IBM Personal Computer* by Bruce Presley (1982) is useful and it also comes in versions for Apple, Commodore, etc.)

---

## **CONFIGURING YOUR SYSTEM**

By Gene St George

### A. Start-Up

1. When you turn on your computer, the CPU kicks into action and goes to ROM for instructions for diagnosing the system. The POST (power on self test) is made and memory is checked. The CPU is then told by ROM to check the disk drives.

If a DOS diskette is in the default drive, the CPU reads the “bootstrap” and a chain of events is started that loads the DOS system files.

If a non-DOS disk is in the default drive the system asks that a system diskette be placed therein. The system will then proceed as described above.

2. If you have a hard drive and there is no diskette in the default diskette drive, the hard drive becomes the default drive and a process similar to the above occurs.

3. The DOS “hidden” system files are loaded and the computer searches for a CONFIG.SYS file. If one is found, its commands are executed. The COMMAND.COM is loaded into memory, and DOS is now controlling the system. After all of the above is completed, DOS executes the AUTOEXEC.BAT file, if it exists, and the user proceeds to use the system. If it doesn’t exist. COMMAND.COM exits with the date and time and the system is now usable.

## B. CONFIG.SYS FILE

1. Again, every time you turn on, or reset, your computer DOS searches for this file. The file tells DOS how to configure your system. When the files in this command are executed nothing appears on the screen. But, the system becomes designed as you want it. Should you have no needs, defaults are provided for some of the commands described in the file.

2. CONFIG.SYS should be located in the root of the directory you will boot from. The file is optional. Because of defaults, it is not essential to system use. However, if you use a “mouse,” a “virtual” drive, or other devices, you need this file.

3. Some of the files (or commands) described hereinafter consist of filenames of software that come with DOS (e.g. \*.SYS) or other sources (e.g. MOUSE.SYS). These are integrated into your system with an appropriate CONFIG.SYS file. The order of the commands in a CONFIG.SYS generally are irrelevant to system performance.

4. The CONFIG.SYS file must be written in standard ASCII. Today, it may include any or all of the DOS commands described below.

## C. BREAK

1. Toggles CTRL-BREAK (or CTRL-C) “ON” AND “OFF.” When ON it enables DOS to check for CTRL-BREAK (or CTRL-C) during disk reads and writes. It should be ON when using most compilers and



programs that have few or no operations with standard devices. This slows the system slightly but it is worth using.

2. The line - BREAK ON - in a CONFIG.SYS file will cause the system to check for CTRL-BREAK (or CTRL-C) more often than the normal default mode which is BREAK OFF. There is no need for BREAK OFF in the CONFIG.SYS file.

3. BREAK ON or OFF may be entered any time during system operation in DOS or it may be started in an AUTOEXEC.BAT file.

#### D. BUFFERS

1. This file reserves memory for holding information moving from, or to the disks. Its use may increase speed of access to data in the drives.

2. The form of this command is - BUFFER = n - where n is the number of buffers you want. The default is 2. But, it may be set from 1 to 99. Buffers take up memory. They expedite accesses by the disk drive. It is usually not practical to set up more than 20. Diskette systems may use 10. XT's may use 20. AT's may use 50. If a lot of subdirectories are in use, increase the value from 20 to 30 or even 65.

3. Once having set a number of buffers, it is necessary to re-boot to change that number.

#### E. COUNTRY

1. This file sets: date, time, currency, and case conversion formats as used in various countries. This command must be included in the CONFIG.SYS file if you use the SELECT command of DOS.

2. The format for this command is - COUNTRY = nnn - where n is a code unique to each of several countries - (e.g. 039 is Italy). The default is for the U.S. of A. where nnn = 001.

#### F. DEVICE

1. This command makes device drives part of DOS.

2. The format of this command is - DEVICE = filespec. Where the filespec is an assembly language program that tells DOS how to use the particular device. These programs are not needed for use of standard devices such as: keyboard, floppies, screen, modems, and hard drives.

3. Example of device drivers are: ANSISYS, VDISK.SYS,

RAMDISK.SYS, and DRIVER.SYS - all of which are on your DOS disks. Device drivers such as MOUSE.SYS etc. are obtained elsewhere. The format described above must be in your CONFIG.SYS file if you are using device drivers.

## G. ANSI.SYS

1. This file permits extended input and output control. If you use it, the file must be included in the root directory and a line - DEVICE = ANSI.SYS - must be included in the CONFIG.SYS file.
2. The ANSI.SYS device driver permits increased control over the cursor position, erasing the screen, changing colors and the screen mode, and redefining the meaning of keys.

## H. VDISK.SYS or RAMDRIVE.SYS

1. Provides a “virtual” disk in RAM. The disk will be the size of your choosing.
2. If you use the command, the file VDISK.SYS must be included in the root directory and a line - DEVICE = VDISK...(parameters)... - must be included in the CONFIG.SYS file.
3. To change the characteristics of the created “virtual” drive, the system must be re-booted and a new description of the device must be included in the CONFIG.SYS file.

## I. DRIVER.SYS

1. Provides support for an external diskette drive, or assigns a second drive letter to a diskette drive.
2. Format is - DEVICE = Driver.SYS ...(parameters)...

## J. FCBS

1. Specifies the number of control blocks that DOS uses at one time. It is not recommended to use this command unless you need it -- e.g. if you are using a LAN (local area network).
2. Format for this command in the CONFIG.SYS file is - FCBS = m,n -where m = the number of files opened by FCBS and n = number of files opened by FCBS that DOS must close if a program requests more than 8 files opened at once. Default for m and n are 4 and 0. Allowable values

are 1 to 255.

3. If you have a program that works with DOS 2.x but does not work with 3.x using 12 for m and n may allow it to work.

#### K. FILES

1. This command permits the opening of more than 8 files at a time. Format is - FILES = nnn - where nnn is between 8 and 99. Default is 8 and a value > 30 serves no function.

#### L. LASTDRIVE

1. Format is - LASTDRIVE = x - where x is a letter equal in value to the number of drives you have on your system (e.g.: C=3,D=4 ... Z=26). Default is E. You should not specify a larger “number” than needed. Use especially in networking.

#### M. SHELL

1. Load an alternate command interpreter. It provides a way of expanding the “environment” where DOS stores and passes information to programs. The development of alternate command processors is not a minor task. It is used by system programmers who write “COMMAND.COM’s” for their programs.

2. The format for this command in the CONFIG.SYS file is - SHELL = filespec... - where the filespec is the name of the alternate command processor which should also be in the root directory.

#### N. EXAMPLE OF LINES IN CONFIG.SYS FILE:

**break on**

**buffers = 20**

**country = 039**

**device = d:ansi.sys**

**device = d:vdisk.sys...paramaters...**

**device =d:driver.sys...parameters... fcbs = 12,12**

**files = 20**

**lastdrive = d**

**shell = filespec...parameters**

**\*\*\*SOURCES OF INFORMATION\*\*\***

1. *THE POWER OF RUNNING PC-DOS* by Siechert & Woods (1986).
  2. *PC-DOS and MS-DOS a Guide for Beginning and Advanced Users* by Sheldon (1985).
  3. *MICROSOFT(tm) MS-DOS* by Microsoft Corp. (1986).
  4. *INSIDE THE IBM PC* by Norton (1986).
  5. *YOUR CONFIG.SYS and AUTOEXEC.BAT* by Simon (1986).
  6. Various magazines (various dates)
- 

## **Letter to the Editor**

May 31, 1988

Dear Editor,

I just reread your recent publication, for the third time and I find your efforts interesting. Your aim of disseminating computer knowledge is commendable but I don't agree with you that an organized conspiracy exists in withholding computer knowledge from the General Public. In order to use computers Industry needs a trained cadre of employees who can program and run these machines. Therefore, you will find that Industry provides much support to the Universities in computer training. They do provide such support for selfish reasons, and your feeling that they discourage employees using menu driven programs from learning how to manipulate these programs is correct. However, in many instances there may be valid reasons for doing this. I had much experience with this problem when I worked in Europe. People with limited knowledge would manipulate and mess up their programs and create chaos. I do believe that computers are not a mass media. It takes motivation, intelligence and education to master the computer.

There is a great need for standardizing hardware, software,

operating systems etc. Doing this would give the computer a big push forward at least initially, however, this may be very detrimental to the long range growth and development. Standardization of equipment and software would freeze computer development to the current state of the art and it could limit the future development.

I do agree with you that Companies and Individuals involved in computer development are generally acting in their self interest. SO WHAT'S NEW? I find it hard to believe that they are so organized that a conspiracy exists to control future computer development.

There is a great need for education at the grass roots level to teach people to use computers. Also there is great need for publications which are written so that they are understandable to the novice and also the average computer buff. I am sure that the *Amateur Computerist* could make a good contribution in this area.

You keep saying that the teaching of programming would aid workers in learning to use computers involved in automated production. I find this hard to accept, with the exception of a small group of highly motivated individuals. My experience with computers has been that learning computer skills by the average individual is only effective when he is highly motivated and when the individual has the facilities to practice his or her skills while learning. Except for a relatively small group of people, courses have to be specialized and based upon a need. A worker assigned to a automated production line if motivated could be upgraded and should be provided the opportunities. The mass teaching of computer programming could be prohibitively expensive for the small gain achieved. I do agree that most computer courses today are a turn-off for the average student. I know many individuals who have taken such courses and few if any have maintained an active interest in computers.

You indicated that you thought computers could be of major benefit to man and it could be. What I see to date scares me. I see big brother in the making. Every day more and more data on each individual is collected in data banks. So far these banks are fragmented. However computers are getting faster and more sophisticated and it is only a matter of time that all data will be centralized. I shudder to think of the result.

Last year a friend of mine who works for the City of Philadelphia

told me about a laser storage device which he was testing. It is a cabinet about twice the size of a two drawer file cabinet and it is capable of permanently holding all records for Philadelphia for many years. Access time to an individual record is seconds, and all records are permanent and cannot be erased.

With regard to your request for a contribution of an article or a suggestion for bettering your paper. I would suggest that you use my comments on big brother. Solicit comments from your readers on whether or not they agree. You might also encourage readers to write about their views on whether computers would improve or make the lot of the average worker worse. It could generate some interesting discussions and comments. I believe that this should be an area of genuine concern.

Sincerely,  
Dave Pollack

**[Note: We invite responses to this letter from readers, particularly on the question “Whether Computers Will Improve or Make the Life of the Average Worker Worse.” We will respond to some of the issues raised in this letter in the next issue - editors]**

---

## STAFF

**Ronda Hauben  
William Rohler  
Norman O. Thompson**

**The *Amateur Computerist* invites contributions of articles, programs, etc. Send submissions to:**

**R. Hauben  
P.O. Box 4344  
Dearborn, Mi. 48126**

---

The opinions expressed in articles are those of their authors and not necessarily the opinions of the *Amateur Computerist* newsletter. We welcome submissions from a spectrum of viewpoints.

## **ELECTRONIC EDITION**

ACN Webpage:

<http://www.ais.org/~jrh/acn/>

All issues of the *Amateur Computerist* are on-line.

Back issues of the *Amateur Computerist* are available at:

[http://www.ais.org/~jrh/acn/Back\\_Issues/](http://www.ais.org/~jrh/acn/Back_Issues/)

All issues can be accessed from the Index at:

<http://www.ais.org/~jrh/acn/NewIndex.pdf>

## **EDITORIAL STAFF**

Ronda Hauben

William Rohler

Norman O. Thompson

Michael Hauben (1973-2001)

Jay Hauben

The *Amateur Computerist* invites submissions.

Articles can be submitted via e-mail: [jrh@ais.org](mailto:jrh@ais.org)

Permission is given to reprint articles from this issue in a non profit publication provided credit is given, with name of author and source of article cited.