MICRO-COMPUTERS-A POWERFUL TOOL FOR BEEF PRODUCERS

Ted R. Nelson

Department of Agricultural Economics
Oklahoma State University

Recent improvements in desktop-sized computers and programs to run on these machines has made this equipment practical as a management tool. They are being used for a number of applications which enhance the manager's knowledge about his operation and the cattle production and marketing environment. Some of these applications are:

Least-Cost formulation of rations, by specifying either the daily requirments for the stock to be fed or the amounts of each nutrient desired per 100 pounds of feed (the conventional "per centage" approach).

Accounting programs which track feed and cattle through the operation, keeping track of inventories of feed in storage and stock in each pen, with billing of feed and care to to owner on a regular interval.

General business accounting for tax and credit and overall management purposes. Accounts payable, net worth, cash flow, and payroll processing often are included.

Budgets and projections to predict the performance of possible combinations of cattle and rations to search for the most profitable combination under a unique resource situation.

When making an initial entry into an unfamiliar area, it is useful to review items which should be carefully considered to reduce the potential for errors, frustrations and lost time and money.

1. Match the hardware with the software and the job(s) to be done. It is as easy to become over-equipped or under-equipped with computing 'lines' as it is with any equipment.
2. Look at several brands and models and the programs available for them. It is much less frustrating, expensive and much more satisfying to acquire a system which you merely need to learn to operate rather than to construct while you are learning.
3. Plan from the start for the system you eventually intend to acquire and be aware of the advantages of getting as much of it as possible from the same responsible source. With scattered sources you are exposed to the possibility that when something doesn't function the way it should each vendor is tempted to alibi that the problems are with the components you got from the other guy. This does not necessarily mean that you must buy a turn-key system with all of the software from one place any more than that all of your farm machinery must

be from the same dealer, but there are good parallels to be drawn.

4. First and last, remember to have "backup" (duplicate) copies of all programs and data records on disk or tape. If its worth having one copy it is worth protecting against human, mechanical and electronic errors.

Disks-- are very important if you are to be involved with storage and retrieval of data--and that is what accounting and production records and word processing is all about so don't plan on getting by with less than 300K of rapid access storage. Convenience is served by having two or more disks, but it may not be worth the extra cost if you have more than 300K on one disk and have a good utility program for duplication of back-up copies of data files and programs.

Disk drives are available for micros in a large number of capacities, determined by four characteristics, which are:

1. Size
   - A. 5 1/4 inch diameter floppy disk.
   - B. 8 inch diameter floppy disk.
   - C. Sealed "hard" disks.
2. Density of data storage on the disk.
   - A. Single
   - B. Double
3. Number of tracks formatted per side. (35,40,76,77,80,etc.)
4. Number of sides of the disk used.
   - A. One side only.
   - B. Both (dual sided).

At the extremes of these variables a five inch disk may contain as little as 80,000 bytes (80K) or up to over 400,000 bytes (400K) at prices which vary from $270 to $1000 per drive. "Hard" disk drives are available to hold up to 23,000,000 bytes (10,000K or 10 "MEGA bytes") at about $6,500.

Printer-- is nearly a necessity, but there is a lot of choice on the quality of print, printing speed, and fancy features which determine the cost. There are several very readable printers which will do the job in the $400 to $700 price range. If you insist on lightning speed or quality typewriter print it is hard to stay below $1500 on cost.

Telephone modem (or Accoustical coupler) is required to enable a computer to communicate by phone with another machine, and an RS-232 Interface-- must be present in the system to support the phone modem. Many producers will want to subscribe to commodity market analysis services which provide

timely information in this form and/or to central program and data bases. These two items cost about $150 each.

Tape Recorders and cassette tapes are the least expensive medium for storage of programs and data. They have the reputation of being slow, unreliable, and inflexible. Most low cost systems start, however, with tape. Any kind of data can be stored and retrieved using tape, and some of the disadvantages may be alleviated with various modifications at reasonable cost.

Table II covers some of the characteristics of magnetic tape storage equipment.

TABLE II - Cassette Tape System Characteristics

Recorder	Char/Sec.	10 Min.	60 Min.	Cost.
Mdl I CTR-80	50	10K	60K	60
add TC-8	250	50K	300K	120
Mdl III CTR80A	150	20K	120K	60

Software Compatability

The hardware selection decision must not be made without regard to the programs which are to be used on that hardware. While it is possible to translate programs from any brand or model to any other brand or model (assuming that a common language compiler or interpreter has been acquired for both machines) one should be fully aware that the time and labor required to do so may be at least a nuisance and more likely a problem, and that time and expertise are required to make the translation. The only safe recommendation is to demand that you see the programs you want run on the machine, or at least that you see representative samples of printouts and screens produced by the programs you want on the system in question. Even a change in printers may make a difference, since printers have different commands for some functions such as page control or printing of graphic characters.

The cost of some software is also different on some brands and models than on others. For example Visi-Calc and Scripsit (word processing) are substantially more expensive for the RS Model II than for the Model I and Model III. When these costs are higher, it usually means that the program contains added features or run more efficiently in the more expensive version, but that is not always necessarily the case -- so ask around and check it out. You may not need the faster or "better" version that badly!!

General purpose software may do a fine job of doing many of the programs you need, so check them out also. The most common examples of this type of program are:
1. Data Base Management or File Management for storage and searching and limited calculations such as field records.
2. Visi-Calc provides an "electronic worksheet" for calculations and storage of budgets, feasibility plans, etc.
3. Word Processors for construction, modification and printing text material such as letters, reports and secretarial jobs.

TYPICAL SYSTEMS

It is extremely difficult to describe systems which are "adequate" or "practical" because of the many, many options which are available and the limits of experience possible with any one author. The following is supplied with notice served that the author has very limited experience with any micro equipment except the TRS80 Model I and Model II. But, because it is an important matter for individuals who have not made an investment, I make the attempt to describe five "systems" which I believe are considered by producers to be adequate for their needs.

1. The "Get-By" System. This is the type of system that will do the basic jobs for the user who places a high priority on low cost and is willing to go without any frills. He is willing to allow some extra time for processing those jobs which many people believe should be done on a disk system.

a. 16K Memory, 64 x 16 screen & Tape Recorder	$ 850
b. 80,40,132 Character printer (Lprtr VII or MX70)	450
c. Hi-Speed Tape Controller (JPC-8), etc.	120
Total Cost (No disk storage--all tape)	1420

2. The "Standard Mdl III" System. This system provides compatibility with the large number of Model I programs available and provides 40 track dual disks and a flexible printer.

a. 48K memory, 64 x 16 screen, 2 disks drives	$2500
b. 80,40,132 char printer with graphics (MX-80)	650
Total Cost (300K disk storage on 2 diskettes)	3150

3. The "Standard AppleII" System. This system provides compatability with software developed for Apple systems during the past 2 years.

a. 48K with 2 disks, 12 inch 40 x 25 Green,Blk Screen	$3000
b. 80,40,132 Character printer with graphics (MX-80)	650
c. Apple interface for printer	170
Total cost(With 220K disk storage on 2 diskettes)	3720

4. The "First Class" System. This system provides 80 column screen and 500 thousand characters of disk storage and print quality comparable to that from a IBM Selectric typewriter.

a. 64K, 13" 80 x 24 screen and one 8 inch disk drive	$3900
b. 132 column Daisy Wheel printer	1800
Total Cost (with 500K disk storage on 1 disk)	5700

5. The "Large First Class". 128K & Four 8" Disks.	$5000
132 column Daisy Wheel printer	$1800
Total Cost (with 2000K disk storage on 4 disks)	$6800

Software Availability

Microcomputers for cattlemen now a partially realized fact. Interest in their use has grown as programs have become available and the number of beef producers who now have these machines is in the hundreds compared to a few dozen just two years ago. The level of satisfaction felt by these owners is highly variable, however, ranging all the way from nearly complete frustration (or even disgust) to an extremely high level of pride and feeling of accomplishment. This variation is seldom caused by lack of performance of the equipment. While there are a few scattered cases of "hardware" problems, most of the disappointments come from the problems associated with learning the use of the "software" or programs which enable these versatile machines to do their clerical and mathematical tasks.

The obvious answer to the question,"Is software available?" is YES. There has been software available to some extent since desktop computers costing less than $5,000 came on the market. The supply of that software has been growing and improving steadily,and will grow (at an increasing rate) for some period of years and then continue to increase as long as this technology is in general use. The software availabile is much more attractive today than it was a year ago, and the software industry for agricultural software is still in it's infancy. Most of the programs available now will be obsolete when substantial investments have been made in the development of sophisticated programs which perform faster and more conveniently. Development of really good, convenient, and fast programs is an expensive and time consuming task. It will be developed only when a large market is perceived to exist by organizations which have a sufficient financial base. At this point in time there is no large organization which has a staff of programmers and scientists which anticipates a large market for special software for agriculture or for any other specific industry.

The large software houses are presently working in the area of operating systems for micro-computers and programming tools such as interpreters and compilers, and on general business software which has utility in nearly any type of business and any industry. So it is in the area of general purpose software that the YES answer is most resounding.

Types of Software Available

SYSTEMS SOFTWARE is developed to a higher degree than other types and it is the foundation for all that follows. Most microcomputer vendors havereasonably efficient operating systems, (instructions coded to handle the general purpose tasks necessary to do the system functions such as storage of programs and data on disks and routing of information to printers, etc.).

But even these basic functions are under continual study and improvement. One step closer to the practical user is the development of interpreters, editors, and compilers which make the programing process more efficient and easier to learn. Time of this development is going in two directions at the same time; i. e. the programing becomes more difficult to learn on compilers built for increased machine efficiency and eaiser to learn on interperters aimed at the novice programer. This is of little interest to the farmer who has no intention of writing programs but he should realize that the capabilities of his machine are inceased as this type of progress is made which make better, bigger, faster programs a possibility for his small machine.

GENERAL BUSINESS SOFTWARE is written with the intent that it is of value to any type of business in any location. There are many of these on the market and there is a large investment being made in this type of sofware because of the wide market known to exist for it. Examples are:
WORD PROCESSORS such as the one used to prepare this paper. The text is typed into the machine memory and stored on disk with options to move words, sentences, or paragraphs around or to change the width of the page or to search for certain words and replace them with others and a number of other functions to conveniently work with text.
SPREADSHEETS such as VISICALC, TARGET, SPECALCULATOR, ect., which provide a screen table of rows and columns and for user-defined calculations to be performed on the cells within the table. These are very useful for entry and updating of budgets, cashflows, balance sheets, and other general purpose reports.

GENERAL LEDGER accounting programs which provide for entry of income and expense data and generate income statements and balance sheets.
ACCOUNTS PAYABLE programs which maintain records of vendors and the pending transactions and totals due to each and print checks for amounts due and summaries of totals outstanding.
ACCOUNTS RECEIVABLE programs which maintain records of customers and the pending transactions and amounts due from each and summaries of total amounts due.
DATA BASE MANAGEMENT programs which enable the user to set up electronic file cards which may be sorted, selected, updated corrected or otherwise modified with input formats and report formats designed by the the user. One use of these is for mailing lists, and many of them may be coordinated with the Word Processor to combine addresses from the DBM file with form letters and to print corresponding mailing labels.
MAILING LIST programs perform a specific form of data base management which is intended only for mailing addresses.
PAYROLL programs maintain records for employees such as time worked, computation of overtime, deductions for state and federal taxes, fringe benefits, etc. and print payroll checks and related tax reports such as W-2 forms.
INVENTORY CONTROL programs track the receipt and distribution of retail items and maintain continuous on-hand reports for the items in stock. Some of them also provide for automatic re-order of items to maintain minimum stock levels.

AGRICULTURAL PROGRAMS

The availability of industry-specific software for agriculture is not well defined at this time. There are dozens of programs in agriculture as well as in many other industries which have been developed for use in a particular firm, or in this case, for a particular farm or ranch. So, while the existence is known, the availability is not generelly programs are often prepared for use by the author and operating instructions are virturally non-existant. The methods of data entry may be unique, the formulas used in computation are vague or difficult to discern, and the existence of the programs is not well advertised, so there is no way for us to know where they are, how they work, or exactly what they do.

There is a growing number of firms which have been founded in the past two years which offer software to the public. The quality of this software is generally unknown except to those who have acquired and used it. A definition of quality of software is even difficult, however, as the content and structure of such software is often very individualized and personal. Formats of how the data is input and the form and content of results may be ideal for some

users and considered inappropriate by others. The setting of standards for such creative material is a difficult process, and there is no agency in our society which has a responsibility to put a seal of approval on software.

There are newsletters for agricutural computing which may attempt to evaluate software, but their performance in this role remains to be revealed. And it will take some time before the market will be able to appraise these evaluations for validity and effectiveness. So beef producers who wish to purchase software in the near future will need to rely on the experience of others with whom they have direct contact for recommendations. They will need to carefully make evaluations of this software to decide whether or not it meets their minimum needs.

Categories of Agricultural Software

One way to classify software is according to the general form in which it is offered to the buyer. There are two common forms.

(1). Source statements in the BASIC language which may be "run" using the interpreter provided with the machine on which the software was written.

The advantages to this form are that these programs are relatively inexpensive, and they may be easily altered by the user. They utilize the operating system provided by the vendor with the machine, so everyone who has the same model can use them without purchasing a special operating system.

The disadvantages to this form are that they must be translated to run on brands other than the one for which they were written, they execute slowly on tasks which entail large amounts of mathematical and disk operations, and they pose problems for the originator because they can be easily pirated and it is therefore more difficult to protect authorship and to recover large development costs.

(2) Compiled load-save modules require that the originator purchase a compiler which translates the original statements into code which cannot be read by the user, but which runs more efficiently in terms of both execution time and machine memory required. Much of this type of software commercially available is also compiled under a standardized operating system such as CP/M or the PASCAL UCSD operating system, which require that the user also uses the special operating system to execute the program. The advantages and disadvantages of this form are essentially the inverse of the BASIC source statement form in (1) above--Protection of original program code makes proprietory investment in the software more attractive, execution time is much faster,

memory required is reduced, but the user cannot change the program to his liking.

HOW TO WRITE, BUY, ADAPT, OR HIRE SOMEONE TO ADAPT SOFTWARE

Software acquisition is often said to be at least as important and difficult as the computer hardware. It would be presumptuous to guess at how many producers could, or would want to, become proficient in some programming language to be able to write useful programs. So here we discuss the primary points of program acquisition.

Let's start on some rather critical definitions.

SOFTWARE is the instructions which make the computer do what we want it to. It includes everything listed below, plus anything else in the general category.
MACHINE LANGUAGE is the form which instructions take in the machine; all numbers which make sense to the electrical engineers who design computers. It takes a long long time to code it to do very little-BUT it is a necessary foundation for all that follows.
PROGRAMMING LANGUAGE is a set of rules which permits communication between humans and a computer. There are many languages each different than others. BASIC (Beginners All-purpose Symbolic Instruction Code) is the microcomputer "Standard".
SOURCE LISTING is a series of statements stated in a programming language. Line numbers are used to aid location of specific instructions.
STATEMENT is the smallest instruction in a line such as read, print, add, multiiply, etc.
INTERPRETER is a machine language program which translates each statement in a program to machine language, and executes the instruction as it goes. It is relatively easy to use and relatively slow in execution.
ASSEMBLER LANGUAGE is a hard-to-learn programming language which closely resembles machine language so it executes rapidly.
HIGH LEVEL LANGUAGES are "near-English" programming languages which require translation to machine language before they can be executed ("run").
COMPILER is a computer program which produces a machine language module from a high level language (like BASIC, FORTRAN, COBOL, PL/I, PASCAL) which produces an efficient executable machine language module which can be stored on disk and later loaded and run quickly. (It cannot be changed except by going back and changing the SOURCE code and COMPILING a replacement).

OPERATING SYSTEM is the set of control programs which directs the use of the computer and usually comes with the machine.(Other versions may be purchased for use instead, such as CP/M below).
CP/M is a specialized operating system - a pioneering effort to create data and program storage in forms transferable between brands of machines.
P-CODE is an intermediate storage form created by a special complier to be transferable among brands for which a RUN-TIME System has been installed.(Example PASCAL UCSD (Univ.Cal.San Diego).
FORTRAN is the "old standby" language (Granddaddy of the high-level languages named from FORmula TRANslator.)
COBOL is a high level "self-documenting" language popular in corporate business application because of ease of reading by other programmers. (Common Business Oriented Language)

A. How to write software.
1. Get access to a micro-computer.
2. Get a BASIC language manual which may come with your machine.
3. Spend a few evenings going through the exercises outlined in the manual very closely following the examples carefully watching what happens as you follow those instructions. Do the exercises.
4. Run some BASIC programs which deal with games or business problems you are familiar with, study the listings and see how the other guy does it.
5. Write a few very small programs addressing problems which you completely understand. Change those programs to work in various ways and print in various ways on the screen and with the printer.
6. VERY CAREFULLY describe the problem you wish to program on paper, specifying as completely as you can what you want the screen and the printout to look like at each stage of the game.
7. Get after it with as much continuous effort as you can muster, but plan to spend at least three times as much time doing it as you really thought it was going to take when you start. (Save backup copies every half hour as you go on storage medium and run frequent check runs on your program.)
8. Enjoy the satisfaction of having been creative and of having learned a lot and of having a program you thoroughly understand and can perfect to your liking any time you want to take the time to do it.

B. How to Buy Software. This approach is much more attractive to most people since it is more convenient less time consuming, and less expensive than alternative A. And it

can be all those things-- it WILL BE all those things for many people today, and for more and more people as time flys by. There are a number of myths involved in some literature today about buying software which need to be challenged.

M1."Producers generally should plan to spend as much (or even more) for software as they do for hardware". It is always prudent to plan for the worst contingency, but caution and worry can surely go too far! We can be enthusiastic about the versatility of the microcomputer (and there will be problems addressed which we admit to being completely unaware of at the moment). Good software can and should be shared over many users. We can think of few better examples of the economic efficiency of mass production to reduce the very high cost of the original model to a modest price to the majority of users. It seems that this concept is much of what microcomputing is all about--economic efficiency from which everybody gains. So software should not be prohibitively "expensive" when it is handled efficiently!

M2."You don't really need to worry about software, just buy the hardware and your problems will be solved." This one is at the other end of the pessimism-optimism scale--at least for now. There will be continuous development of better and more comprehensive programs for decades, but we are still in the early stages of this gigantic task. There is a lot of useful software in existence BUT it is not well cataloged, not well evaluated and often not well documented. Getting the software needed at a reasonable cost is a still a big problem now, which will gradually and slowly be solved-- But it is a problem now.(Period.) OK- It's a big problem, but how do we attack this problem?

1. Attend meetings, subscribe to microcomputing magazines, look for articles in farm magazines, talk to consultants, contact your University Extension Service, visit computing stores in farming areas - anything you can reasonably do to become informed will help but this problem will cost you in time and money one way or another. It is axiomatic that you COULD spend more time and money than the result of the search is worth, but it is important.
2. Define needs as completely and carefully as you can, both for the present and for the next few years. Assign priorities as to just how important it is to you to have the software on that want-list.
3. Confirm that the software you propose to buy will come close to your present needs on the equipment you have or plan to add to your system. The acid test is to sit down and run the programs-hands on. That is not always possible, so come as close to that ideal as you can. (Printed samples of screens

and printouts seem to be minimal to me - I like to see what I get before I pay, don't you?)
4. Buy the program- Read the instructions; Test run the program; Apply it to your situation- IN THAT ORDER.

C. How to Adapt Software. This one turns out to be a combination of A and B above: You have to acquire a program and know something about programming to do this one, so the steps itemized above apply more or less, with the following caveats and conditions.
1. Check with the source to see if they know of anyone who might have already added a similar enhancement.
2. Acquire programs capable of being changed. This means they must be provided in a dialect of a language available on your machine. If you have not added special compilers (at a cost) this spells BASIC for most microcomputers. It is the easiest language to learn and is therefore the one provided "On-Board" most micros from the factory.
3. Get a listing, preferably from the source, but certainly of the original program in the form that you received it. A list of variable names is also an extremely useful tool which most good programmers will have in the file for their program. A flowchart or block diagram provides an additional useful tool, but is less critical for micros which have the capability of running with a statement number trace on the screen as the program executes.
4. Insert your needed modifications into the program. Some hints on this are: a. Screen and printing cosmetics are usually easy to do and "safe". b. Use "GOSUB" subroutines which RETURN to the sending point to add features without affecting the logic or results.
5. Save backup copies of the original program and of various stages of your enhanced program and checkrun with constant test-data often. (Introduction of undreamed-of bugs is a constant hazard in this game.)

D. How to Hire Someone to Adapt Software. This one may prompt you to ask "How about another topic called "Hiring someone to write original software? If it does not prompt such a question, I want to answer it anyway because it seems to be a logical and common question. Original non-trivial software is expensive, much like the first prototype of a machine, but nearly all of that cost is expensive labor. It's just cheaper to use something on which the cost has been shared among a few hundred cohort users. Even if the problem you have is entirely unique, there is probably a program or subroutine which is available which can be modified and/or extended. Software houses which are knowledgeable about your industry should either have them or know about them.

Custom programming is a very unique, personal kind of area which must be approached gingerly and with large doses of skepticism. Programmers come in all degrees of expertise, experience, knowledge of your particular system and of your farming situation. Those without farm experience may have to depend on you for every little detail of what appears to producers as "common sense" (I do not intend to be critical, we are all products of our environment and can get easily lost when we are "out of our element"). But I do want to serve notice that it can be a real problem.

It is difficult to make a quick evaluation of a running program to decide whether it will fit your bill. Supervising custom programming is a far more difficult task and paying the bill can be worse.

Some general rules for custom program adaptation.
1. Draw and write out specific and complete "pictures" of what is to be done.
2. Find a programmer familiar with your machine and the dialect to be used.
3. Be sure the programmer has a good understanding of all that is to be done. Talk it over at length--You are getting into some sort of "partnership".
4. Write a contract which will set some limits on cost for your sake. (And the programmer should want some limits on what is to be accomplished- these things have a way of "growing" beyond original expectations.)
5. Include an agreement on who will have the ownership and distribution rights on the finished product. (This can be a bargaining tool in the costing process).

A careully planned programing project with one who understands you and the problems you want solved can be a rewarding project. A poorly planned project with inherent misunderstandings is almost sure to end in frustratrion and hard feelings for both parties concerned.

Purchase Decisions

Once you have made it through the agonizing process of deciding that you can no longer get along without having that modern wonder of technology and you have worked out the money problem in your mind it is only human to get right into it and bring home the new "baby". Suppress that urge- for a little while and recheck your goals. "Reset" and decide:
A. What job(s) are really important to you.

B. What software is most important to you and where you will get it and what it will cost.
C. That you will devote several hours a week for several weeks getting familiar with the operation of the machine and the programs.

When these conditions are resolved, go ahead and get the best deal you can considering the software, service and prices for any of the established brands.
Among the established brands it is my opinion that they all do a good job of running the software designed to run on them: but there are important considerations:
SERVICE- Is the repair procedure satisfactory to you? (Sooner or later it will be needed).
SOFTWARE- Are acceptable programs available at the right price?

Now That You Have a Machine!

1. GET ACQUAINTED- With the manual. It is time-consuming, YES, BUT it will save a lot of time later if you really understand the Operating System (Whether you use "canned" programs or do your own.
2. GET SOME PROGRAMS- Be prudent and try to be sure you will use them before you lay out bigbucks, but you can learn a lot by seeing how some other programmers do things--and they give you something concrete to work with.
3. DIG INTO THOSE PROGRAMS- If they are in BASIC, you can hit BREAK at any time and see what line they are working on; then LIST the surrounding lines and tell what is going on and how it is done. A critical point in all programs is at the "Main Menu". Write down that line# so when the program breaks with data in it, you will not endure the frustration of losing all that information by starting the program "from the top". Just type GOTO Ln# and get back to the place where you can try it again.
4. LEARN to EDIT so you can personalize your programs to do it YOUR WAY if you feel the need.
5. RECOGNIZE that some "BlackBox" programs or routines are needed.

Programs- BASIC vs. OTHER-

Acquiring programs is complicated by the wide variety of forms in which programs are available. Several definitions are useful:
INTERPRETIVE Languages are those for which the instructions (Source Code) are changed into machine instructions as the program is executed, BASIC on most micros is an example. The

best thing about them is that they are easy to program and to modify using the EDIT features. The machine instructions are never stored, either in memory or on disk, but are used momentarily as they are created when the program is executed.
Advantages= 1. Lower programing cost. 2. Easy to change. 3. Relatively easy to translate between brands. 4. They come with the machine at no extra cost.
Disadvantages= 1. Slow execution and loading. 2. Functions limited to those provided by the dialect of the language used. 3. Difficult to maintain proprietary control.(From the author and vendor's point of view.)
COMPILERS convert source statements to Load Modules for more efficient storage and execution. Examples are FORTRAN(FORmula TRANslator) & COBOL(COmmon Business Oriented Language).
Advantages=1. Five to ten times faster loading and execution. 2. Distribution of load modules rather than source code protects proprietorship.
Disadvantages= 1. Must buy the compiler ($100-Up.) 2. More difficult for user to change.
ASSEMBLERS create very efficient load modules which require minimal space to store and execute very fast with maximum availability of functions possible on the machine.

Advantages= 1. Very fast execution. 2. Protected code.
Disadvantages= 1. Tedious, slow (expensive) to program. 2. Very difficult to modify.

WHICH DOS?

DOS is computereeze for DISK OPERATING SYSTEM. It does for the micro with disk the control functions and additional interpreting and utility functions that the ROM(Read Only Memory) chip usually does for the system with no disk drives. Machine vendors usually supply a DOS disk with the first disk drive purchased, and it is different for each model. Updated (improved) versions are released from time to time and packed with equipment shipped after the release date.
The version supplied by the vendor is "supported" by the vendor; i.e. It is designed to work with the peripheral equipment (printers,etc.) sold by that vendor and if serious errors are discovered a free replacement will often be supplied.

Other DOS's such as NEWDOS, VTOS, CP/M, etc. are for sale by various software houses which contain additional or special features. These often have to do with management of disc files, improved source-code editing properties or more convenient communications capabilities.

Some of these are compatable with the original DOS. Others are not because they are created to provide a common

operating system for the purpose of transfer of programs between brands and models of computers.(Such as CPM).
Advantages= 1. Compatability of programs between brands and models, allowing multi-brand distribution of programs.
Disadvantages 1. Extra cost ($100 up) for the DOS.
2.Inability of users who do not have it to use programs created for it.

So-- WHAT DO I BUY?

The choices are many, but so are the objectives and sophisication of users, so there is no "pat" answer to what fits where without examining the specific application. That's part of management!