From data entry supervisor to data entry specialist

by CAROLYN M. DUNNING
Pertea
Santa Ana, California

ABSTRACT
For many years data entry lagged behind all other aspects of data processing. Originally, there were card punches where data was entered a character at a time. Then came key tape making buffering a common concept to data entry. Buffered key punches provided the same advantages of key tape, but had more flexibility. Then key-disk entered the scene and data entry became computerized. Data could be entered faster and more accurately than ever before.

What does this mean to the data entry industry? Operators do not have to be as well trained as before. But the greatest impact is felt at the supervisory level. ‘Mini-analyst’ training should be provided for those individuals interested in supervising key-disk installations. These classes should cover terminology, hardware components and some form of programming as well as the use of peripherals and options common to key-disk systems. Typical problems that can occur, both from an operator and the system should be discussed with methods to assist in analyzing them. Control procedures should be outlined.

Well-trained supervisors can help provide a more efficient data entry department and can be an asset in designing the data entry systems of the future.

For many years, the lowliest task associated with data processing was data entry. The manipulation and output of information progressed by leaps and bounds. Faster accounting machines, calculators and the advent of computers. Look how far computers have advanced in 20 years! The growth has been phenomenal. But what happened with data entry? Computers were processing more information at faster speeds every day. It was beginning to become a problem to find enough input to feed these hungry monsters. Why? Because data was still being keyed into cards one character at a time on equipment that was only as fast as the mechanics of the machines would permit. Keyboards had to be interlocked to prevent the operators from getting ahead of the card punches.

The introduction in the early 1960’s of the IBM 029-059 card punches and verifiers produced two features that were indicative of things to come in the world of data entry. One was the automatic left zero option that gave birth to the concept of buffering data as it was keyed. Data could be keyed and corrected before it was actually punched into the card. Not only that, the operator didn’t have to count and key leading zeroes. Granted, only integers could be buffered and the maximum was eight digits, but it was a beginning. The second feature that was new, was the photosensing on the 059 verifier. Cards could be read more quietly and much faster without waiting for any mechanical action to occur.

The next milestone for data entry was the introduction of the key-tape. Now an entire record could be buffered and massaged and corrected before being output. Just as important, the operator did not need to be inhibited by the mechanical limitations of the machine. Faster electronic keyboards were made which had a feature called n-key roll over. This ‘remembered’ keystrokes therefore assuring none were omitted, nor the operator locked out. Record sizes could be increased from 80 characters providing the ability to enter more data at one time.

Although key-tape machines were much faster, they had other serious limitations that made their use somewhat impractical for many applications. It was impossible for the operator to see an entire record all at once. Data would have to be read character-by-character. Another limitation was the inability to insert records within a batch. Missing records had to be appended to the end of the data batches which was not always feasible. Also, although both data entry and verification could be performed on the same machine, it was not convenient to do so. Since it is common practice for an operator not to enter and verify the same batch of data, this meant either switching tapes or machines. Therefore, key-tape was found to be very inflexible.

However, cards hadn’t given up the battle yet. Then came the advent of the buffered card punches. Now an entire card image could be keyed into a buffer without physically punching a card until column 80 had been passed. While still being limited by the 80 columns in a card, the number of program levels was increased to provide the ability to key more data from a source
two worlds; the speed and accuracy of key-tape with the flexibility of cards. Gone were the clumsy drums and their nemesis, starwheels. Programs could be stored in memory. Card punches could have the faster, electronic keyboards. As one card was being punched, the operator could begin keying the next one. Skips and dupes were performed at electronic speeds. Operator statistics and the generation of batch totals was possible. Automatic emitting of data was available on some machines, as well as the automatic sequencing of program levels. Both punch and verifier were housed in the same machine. What a tremendous help this was in scheduling! No more worrying about the ratio of punches to verifiers. Throughput was increased tremendously.

Then all of a sudden, data entry grew up. Key-disk entered the scene and “Look Ma, we’re computerized!” Now data entry had entered the same league as the big guys. Information is entered onto a disk with the assistance of a mini-computer. This data can be massaged and manipulated as much as desired. It is then usually written to tape for the mainframe computer to process. On most systems, it can optionally be transmitted directly to the mainframe, either over a telephone line or a direct interface. The advantages to this type of system are numerous. Data can be validated as it is keyed. Calculations can be performed and the results can be automatically entered into the records. Likewise, constant data. Now it isn’t a question of keystrokes, but truly a matter of throughput. The data entry department has finally become a respected member of the data processing community. And this is just a beginning.

What does all of this mean? First of all, operators do not need to be as well trained as with the earlier key punches. Although key punch keyboards are standard, optional typewriter keyboards make it easy for typists or clerical personnel to become data entry operators. Information can be easily entered at the source, eliminating or greatly reducing the possibility of errors, which typically increases turn-around. Programming is no longer a factor since most systems are ‘preprogrammed’ with all operators using the same programs. Extensive procedures need not be remembered as most key-disk systems can prompt the operators as well as alert them to error conditions.

Well and good for the operators. But what happens at the supervisory level? Here is where the greatest impact is felt. Now to be really effective and utilize a key-disk system to its greatest potential, the supervisor should be a ‘mini-analyst’ with a fundamental knowledge of basic computer systems and data management. Gone are the cards with the visual evidence of work that has been completed. Data is now in a ‘never-never land’ called a disk. But how do we really know it is there? What does it look like and what happens to it now? These and many more questions plague a supervisor who is frequently answered with a “Don’t worry about it. Just do exactly what you have been told to do and everything will be alright.”

Key-disk has been over-simplified by many vendors. Who wants to frighten the supervisor with a lot of confusing terminology? After all, a very competent mainframe programmer wrote all of these programs and all that needs to be done is for the operators to enter and verify the data, the supervisor to write it to tape for the computer and then delete it from the disk. Simple, right? But what is wrong with this logic?

Let’s start with programming. There’s a world of difference between a program for a mainframe where an operator selects it and it executes with very little interference, and a program for a key-disk system where many operators sit and use it for hours at a time. A poorly written mainframe program can be slow and inefficient memory-wise, but this has little, if any, effect on the operator. On the other hand, a badly written key-disk program may not only be inefficient as far as the processing goes, but it can be very tedious for an operator to use. Who is best qualified to write a data entry program? A mainframe programmer or systems analyst whose primary key punch experience comes from punching their own programs (usually freeform) while in school or a key punch operator or supervisor who has been making program cards and punching and verifying data for years. These people know what is necessary for entering data easily and accurately and should be best qualified for writing programs that are going to be used by all operators on a system.

Another area of concern that is created by a key-disk system is the terminology that is used. A supervisor must be familiar, not only with standard data entry terms, but also those associated with computer operations. While most key-disk manufacturers have attempted to keep the terms data entry oriented and as simple as possible, this is not always feasible. For example, what about tape writes with blocking factors, translations, labels, etc.? And how about bits, bytes and words? We didn’t have all of that with cards. The worst problem we had may have been trying to translate the holes in the cards into characters if the print mechanism failed. Otherwise, we just handed a deck of punched and verified cards to the computer operator and everything else was taken care of.

Some key-disk systems must compile programs before they can be used. That really is scary. With card punches all we had to do was create a program card, either put it on the drum and lower the starwheels or read it into memory and we were ready to go. If the program card wasn’t quite right it was a simple matter to change it. Simplicity was the key word.

What is the point of all of this? Well, many schools and colleges provide very extensive computer training, both for programming and operations. And some schools also provide key punch training. But what
about a person who is interested in learning about a key-disk system? Oh yes, manufacturers provide training, but only on their equipment and usually only for customers. Normally, these are rather abbreviated and do not cover a lot of things a key-disk supervisor might want to know. A mainframe programming or systems class taught at the college level is more than most data entry supervisors would need, or want. They aren't interested in becoming mainframe programmers. So why not provide educational classes tailored to suit the needs of a key-disk supervisor? Data entry is advancing at a rapid pace and experienced, qualified individuals are necessary to create efficient key-disk departments. Again, who is more qualified to handle this task than an operator or supervisor who has been entering data for a long time and understands the needs and problems associated with data entry? This is a 'people-oriented' environment more so than any other area of data processing. It can be a very emotional situation.

What kind of course would this be? It does not need to be concerned with all of the problems of a large mainframe computer, but should be more like a 'mini-analysts' course, tailored for individuals who want to work with some type of computerized data entry system. It must be generalized to cover the common features and functions of most key-disk systems, which is a large order since none are totally alike. However, there are many characteristics that are common to all, or most, systems.

An explanation of computer concepts and terminology is vital. Bits, bytes, words, binary, ASCII, EBCDIC, software, hardware. These are terms that are thrown around by the customer engineers or systems analysts working with a key-disk system. But what do all of these words mean to the supervisor? I think an understanding of some of these terms can help the supervisor communicate better with the people who have to answer questions and service the system. It will also help allay some fears of the 'unknown'.

The difference between hardware and software should be explained. As any one who has been in the real world of data processing knows, it is often very difficult to isolate problems between software and hardware. A supervisor who has some understanding of them can possibly better define a problem that may be occurring with the system. The hardware components that are part of the system should be explained, the most common factor being the disk. This should be covered in sufficient detail so the supervisor has a clearer understanding of its function. Tracks, sectors and AU's should be explained, as well as some explanation of a volume table of contents (VTOC). Some systems require the disk be initialized regularly. The purpose of this and its effect should be explained. Most systems also use some type of WARM and COLD START procedures. The difference between the two functions should be defined as well as what will happen if the wrong option is accidentally selected.

Tape drives are very important to most key-disk installations. Their care and use should be thoroughly covered. Proper cleaning and usage of both tape drives and tapes is very vital to producing good, clean output as the improper handling of tapes can create many errors from the data entry department. The cleaning of tape heads and a recommendation of what should be used to clean them is important. The benefit of having tapes certified regularly should be discussed. Since most drives are loaded and unloaded in much the same way, this could be demonstrated with students actually mounting and dismounting their own tapes. Terminology connected with tape drives should be discussed. For example, the difference between 7-track and 9-track tape drives should be explained, bits per inch and the difference between 800 bpi and 1600 bpi as well as parity and its purpose should be covered. An explanation of the appearance and purpose of BOT (beginning of tape) and EOT (end of tape) markers and how they sensed is vital. Cutting the lead from a tape and putting on a new BOT marker should be demonstrated. EOF (end of file) should be defined. These are things that will confront a supervisor who may have no idea as to what is meant.

A most important aspect of any key-disk training is some form of programming. Although all systems use their own type of programming, there are many common factors that can be discussed in a generalized way. Most systems use at least some type of 'checkbox' programming. This is usually merely a fill in the blanks procedure to tell the system what is wanted. It usually contains field descriptors, the most common being field size and shift (alphabetic, numeric and in some systems, lower-case alphabetic). Other controls can also be defined in the checkboxes. One of these might be data type which specifies exactly what types of characters can be entered into a field and provides a character-by-character validation as the field is keyed. For example, a field may be programmed as alphabetic, but numerics and blanks are also allowed. Data type would have to specify this. Or perhaps a field can only be numeric, with no alphas or blanks permitted. Another common control is how the field can be exited. Perhaps it doesn't need to be exited at all and as soon as it is completed, control automatically goes to the next field. Or maybe a field release must be depressed before continuing to prevent overflowing into the next field. The extent to which a field must be specified may also be specified. Perhaps a field must have every position keyed (none can be blank), or maybe it must have some data keyed, but not necessarily be filled. Maybe it must not be keyed at all and the system will insert some type of constant or calculated data. Justification and fill can be programmed so the operator does not need to remember whether a field must be right-justiﬁed and zero-ﬁlled or left-justiﬁed and blank-ﬁlled. The system will handle it automatically. Range checking
and batch balancing frequently can be programmed in a checkbox program.

Since most key-disk systems allow rearranging the data for output (commonly called reformatting), fields can be organized so that it is easier for the data entry operator. Various types of verification can be programmed, such as key, sight, to-balance or not verified. In many instances, verification can be greatly reduced, or even eliminated.

In addition to checkbox programming, many systems use some type of higher level programming language, the most common being similar to COBOL. It might be well in a class of this type to cover the general concepts of computer programming, particularly the logic of COBOL such as IF, MOVE and the arithmetic statements ADD, SUBTRACT, MULTIPLY and DIVIDE.

Most key-disk systems have some method of prompting the operator. The purpose and general use of prompts should be discussed, such as the most effective way to program them so that they are meaningful to the operator. Peripheral equipment such as teletypes and line printers should be discussed. Since each system probably utilizes these options differently, their use can only be discussed in a general way. However, feed controls, carriage tapes, top-of-form, and other common characteristics might be explained. Since most data is usually entered in an unedited form, (no commas, periods, dollar signs, etc.) and printed reports will have these characters inserted, editing might be mentioned. Examples of the most common editing functions should be provided.

With the increasing use of data communications, a discussion of this feature might be appropriate. Since this is a very complex area it must be generalized and greatly simplified. A brief description of the most common protocols might be good as well as factors necessary for initiating a transmission or receive.

Since most systems use some type of full CRT display, the generalized areas of display should be discussed. For example, there are usually one or more status lines, a message line and several lines for displaying data. An explanation of what is displayed in these three areas is appropriate.

A discussion of error messages and the most common types of operator errors is necessary. This might include common errors such as keying an invalid character, attempting to key too many characters in a field or keying data that is not within a specific range. Most systems also provide the capability of creating some types of individual ‘tailor-made’ messages. A recommendation of what these messages could be and where they should be used would be helpful.

Relatively common system problems should be covered. This might include such things as not being able to bring up the system, a terminal or entire system hanging or not being able to locate a batch of data or a program. Things to look for when any of these conditions occur should be outlined. It can help a customer engineer isolate a problem if a supervisor has previously tried to analyze what is happening. Common operator problems should also be explained, such as keying an incorrect batch name or number, or opening a batch under the wrong mode. Most systems also have some type of search procedure. The parameters available should be discussed.

The supervisory functions are basically the same for all systems. The purpose of the most common ones should be explained. This might include things such as tape writes, deleting data from the disk, system saves, obtaining a disk status, assigning a new name or batch number to an existing batch of data, printing to a line printer or requesting batch status. Generalized operator statistics should be discussed.

Control procedures are a very important area of a key-disk system. Batches must be carefully and accurately logged and monitored as the work progresses. The log must be periodically checked against what is on disk and the status of the batch. (Has the batch been completely entered? If so, has it been verified? Does it contain any invalid data? Is it in- or out-of-balance?) These are things that are very important when the data is output. When a write is performed, batches must be checked against the log to be certain that all desired batches were written, and no extraneous batches were included. Disk status must be checked regularly to assure that the disk does not get full since in most instances, this can create severe problems. Deletes must be very carefully monitored and system saves regularly scheduled. System crashes where data is lost is not a common occurrence, but just one crash can be a disaster in that hours, or even days of work can be lost if the data is not properly backed up.

Well-trained key-disk supervisors can be a great asset to the industry in optimizing the quality of data going to mainframe computers. They can also be very instrumental in providing good, valuable input for creating better data entry systems in the future. Data entry is still in its childhood. There will be many changes in the years to come and the term ‘data entry specialist’ will be just as common to data processing as systems analyst or programmer. Now is the time to start training these specialists for tomorrow.