Notes on English Word Processing

Tamami Kimihira

It is only these two or three years since the phrase "word processor" has become known to most ordinary (non-technical) people in Japan. Sometimes one speaks even of 'the word processor revolution', which refers to the tremendous change of preparing Japanese documents. Compared with English, the Japanese language has an unbelievably complex system of writing or printing. The chief obstacle to typing or printing is undoubtedly the use of Kanji (Chinese characters). The average Japanese is estimated to use about two thousand Kanji daily, so the typewriter coping with this need must have that many keys, a situation which has prohibited most Japanese other than professional typists from using typewriters. The very remarkable developments in electronic technology, as represented by LSI chips, has made it possible to construct computerized typewriters which enable ordinary English or Kana typewriters with standard keyboards to print Japanese documents using in them as many Kanji characters as are needed. The immediate input from keyboard is either in alphanumeric letters or Kana characters (sometimes, so-called Kanji codes as defined by JIS standard), which are then converted to Kanji characters when desired. This is the beginning of the Japanese word processor, a discussion of which I have no intension of entering into further here. My concern will be with the English word processing, which has a much longer history of development. Finally I will touch upon the significance of the word processor as a useful tool in preparing English documents.

1. the concept of word processing

It is a commonplace to say that the invention of printing machines heralded the beginning of the modern world, the Renaissance. Printing is in a word the technology of producing documents displayed on paper (so-called 'hard copy'). Though its value is of course inestimable in making available one person's ideas to manay (that is, playing the precious role of the mass media), it must depend upon manuscripts written by authors; as often as not it is likely to be restricted to the documents worthy of being conveyed to many readers. In the case of one-to-one communication written documents (for example, letters) are far cheaper and easier and more valued than printed materials. The appearance of typewriters, around the late nineteenth century, enabled an ordinary individual to own a personal printing machine, a blessing to poor hands (like myself).

The traditional typewriter, however, was lacking in one important condition that qualified it as a word processor: it had no memory function. The word processor must possess at least the following four functions:

(1) input function (like the keyboard of a typewriter)

(2) editing function (which can correct and edit texts)

(3) output function (which ordinary typewriter or printers have)

(4) memory function (which can store input data)

Of these an ordinary typewriter has functions (1) and (3), and furthermore an up-to-date one has a correction ability with so-called correct ribbons; it has a minimum editing function. The last function, memory, is the most sophisticated, which at last gives the machine the appearance of a word processor. As mentioned in the above passage, the development of electronic technology provided the strongest weapon for this function. Here I will first take up a typewriter-like processor and then focus my attention on devices which benefit amply from microprocessors.

2. A word processor without a video screen (Memory Card Typewriter)

IBM Mag Card II Typewriter (IBM MC82) is the usual typewriter with a ball-shaped element plus memory unit and the magnetic cards onto which the memorized texts can be written. In the sense defined above this may be qualified as a kind of word processor. The greatest difference from the now most popular processor is that this typewriter has no display device such as a CRT video screen. It has thus a considerable disadvantage, but otherwise a tremendous progress has been made in that it can now store and memorize input data which can be corrected or modified (ie. edited) later on.

Equipped with any kind of memory device, the most important functions (at least to the users) of the word processor will be those of input and editing, the output function (such as the quality of printing) depending upon the accompanying hardware devices (eg. a printer). Input function is chiefly concerned with how easily and conveniently we can make a document, whereas the editing function greatly influences the speed and manageability of correcting, revising and editing a first draft of a document.

As for the input function of the IBM typewriter, it is supposed to aim at producing a 'firs-time final' copy of the document; it achieves this, it is asserted, by providing the typewriter with various means of immediate correction and the control signal which help to fix a printing format beforehand (ie. at the input time). As this is a typewriter without the video screen which serves to echo back the input data, the input data (ie. a string of characters) must be checked by 'hard copy' on the paper (which it has printed out as ordinary typewriters do).

Deletion of a wrong (eg. misspelled) word or character can be easily made by backspacing, which delete the character both from memory and the paper of the hard copy. The initial ('default') line length is set at six inches (about 15.2cm) from the left margin;

which allows 60 characters pica or 72 characters elite per one line. The bell rings eight characters from the line end. The important feature of the typewriter is that of line end zone. The zone consists of the last six columns on the line. The zone has to do with the problem of syllabification at the line. This is perhaps among the biggest obstacles in English word processing. It poses, as we see below, various difficulties to the word processing programs. Anyway the zone here tries to leave room for a decision on the part of the typist to make adjustments in syllabification in the automatic playing back of the text from memory.

Here several important control commads are mentioned in anticipation of common word processing commands to be mentioned below. Commands can be entered by depressing both the CODE button and one other key at the same time. CODE corresponds obviously to CTRL (control) key introduced below. IR (Index Return) command is typed on the first page of a document to assure that playback will always begin at the left margin. Required Hyphen is always necessary when a hyphen is needed, because the machine automatically drops a hyphen during playback unless it is positioned at the end of a line. It will be seen easily that the distinction is being made here between the 'hard' and 'soft' hyphens which is introduced in an up-dated per se word processor. Required Space command prevents the carrier (ie. carriage) from being automatically returned at the end of a line after a space, which is troublesome in the case of names, dates, and abbreviations. Similarly Required Carrier Return command is necessary to prevent a single carrier return in the middle of a line during playback from being changed to a space; this is required in the case of a name ane address. Other now familiar commands, such as Word Underscore, Centering, Changing Line Length (or Tabs), Stop Instruction or Required Tab are conveniently provided for their respective functions.

Before the consideration of the editing function of this typewriter, a brief mention will be made of its memory mode. Precise technical specifications are not given in the manual, but the memory unit seems to have a memory capacity (buffer) of about 8 Kbyte; that is, it can memorize 8000 characters (including command codes and special symbols). A mag (netic) card is said to have sufficient capacity to record two pages of a standard manuscript (more than three thousand characters). Page End code typed at the end of a page commands the machine to finish recording that page. The card unit has a pack feed slot to contain a stack of cards (up to 50). Onto each card one page of the document is written, the next page being automatically recorded on another card. At playing back (ie. 'loading' in computer terminology), the machine can print out the text at a speed of 930 characters per minute.

Finally the most important feature of this word processor, editing, will be considered. Remember this machine has no video screen, so the method of editing is classified as 'line editing'. This means that the unit of editing is a line. The more precise notion of a line will be defined below. Let us here tentatively assume that a line is one that is shown on a paper of a 'hard' copy (a actual copy on a paper). Furthermore, the notion of a paragraph is very important in the editing mode. The paragraph can be defined as the line(s) plus these conditions; (1) two or more carrier returns; (2) a carrier return followed by a 'tab' or space(s) or a required backspace(s); (3) a required carrier return; (4) an Index Return. Lines and paragraphs can be 'accessed' by using the control keys in order to move to the part of text in memory which is to be edited; PAR ADV (paragrph advance) key moves the pointer of memory to the beginning of the next paragraph, while LINE ADV moves the pointer to the beginning of the next line and LINE RET (urn) moves it back one line at a time.

Line editing in the computer usually depends on the line number which is attached to each line (though in memory numbers are commonly ignored), but in this typewriter no line number is typed, which may delay a little (or considerably) the 'access' time to the line to be edited. For this deficiency, however, the above-mentioned control keys make up very much. Most of the editing work consists of insertion, deletion or both. Insertion in this machine is of most facile type. Insertion is accomplished simply by typing the additional text to be inserted. As typing proceeds, the text following the inserted part is moved in memory. Deletion is made by depressing both CODE (ie. CTRL) button and the deletion key at the same time. One time deletion is possible for a whole line (including CR(=carrier return)), a word (with a space, CR, or tab), or one word.

To do justice to the full competence of the machine, we must add that it has 'scanning' capability of creating final memory cards without playing back to rearrange lines and pages, where the line ending and page length decisions, together with hyphenation decisions if necessary, are made in memory. This may be compared to the 'macro' commands in the microprocessors to be considered below.

3. The line editor of the microcomputer as word processor

CP/M (Control Program for Microprocessors: a registered trade mark of Digital Research) has long (according to the standard of the computer world) been a standard disk operating system for microcomputers in the U. S. A. In Japan it is rapidly becoming one of the most influential tools for memory disk operation. CP/M has a powerful text program called ED (editor) that can be used to create text files. It is undoubtedly meant for writing and manipulating programming texts mostly in an assembly (mnemonic) language which can be directly assembled for creating an object array of machine codes. But once the use of ED commands and control orders has been mastered, it can be utilized as a fairly convenient and interesting tool for word processing in general. This may be especially true where CP/M is one's main operating system. So I will here examine ED as a word processor (of ordinary documents) which may be regarded as a typical editing tool using 'line editing'.

ED in CP/M can be regarded as a word processor without any recourse to the video (screen) mode currently common in many processors (eg. WordStar or WORD-MASTER). The advantange and convenience of a video editing, which will be examined later in a little more detail, lies mainly in the ease of inspecting and updating texts by using a CRT terminal (ie. video screen). There we can move at will the cursor on the screen and position it by line, word, or character, and immediately modify the text and ascertain the change visually. On the contrary, in a line editor editing is confined within one specified line and it is not presupposed that the cursor can move beyond the line (upward or downward). The function of locating the cursor at an arbitray position on the screen, which is ofter referred to as 'cursor addressing', depends heavily upon the hardware construction of the input mechanism (eg, keyboard) and any program employing cursor addressing should be adjusted ('installed') to the particular machine to be used. The elimination of this function therefore allows the scope of a program (in our case a word processing program) to be greatly widened, and this is counted as one advantage of a line or character-oriented editor such as ED in CP/M. Notice that line editing is in accord with traditional microprocessor programming; for example even in the most popular BASIC (MicrDoesn't matterrsoft BASIC-80) program texts are edited line by line and cannot be modified directly by moving the cursor freely as we can in almost all Japanese machines.

The fundamental logical notion of line editing is that of 'character pointer' (abbreviated as CP). The CP is not actually displayed, but is to be regarded as a logical indicator pointing to a particular character on a line; more precisely a CP is usually located at the position just after the particular character. In ED of CP/M a line is to be defined as a string of characters ending with the two code charactors represented by CR (carriage return) and LF (line feed). Here no restriction is placed on the length of the string in a line, the only necessary condition being that the concatenated CR+LF ends the string. Each line has an imaginary line number attached to itself which is not actually part of the text. The number exists only in the edit buffer (ie. a memory block inside the computer reserved for ED's text processing). The line number serves much to specify a position in the editing text. LF which follows CR is automatically attached in memory on entering CR at the end of a line. In the IBM typewriter considered above, we saw that two CR's are necessary to indicate the end of a paragraph; the concatenated CR+LF might be regrarded as a refinement of showing the end of a line. Thus defined, the way of saying that CP is at the beginning of the nth line can be easily interpreted as meaning that CP is just after the CR+LF of the (n−1)th line. In the newer version of CP/M (mine is ver. 2.2), the line numbers are automatically displayed, unless the command '−V' is entered to turn off the display. In the IBM typewriter in section 2, the notion of lines is of course made much use of (hence the command LINE ADV plays an important role), though line numbers are never printed out.

The notion of CP is very important in line editing because it is in reference to CP that position to be edited in the edit buffer is specified (had 'access' to) as quickly as possible. This is why in employing ED we should always bear in mind where CP is now. Insertion, deletion, and modification are all based on moving CP to the point to be edited. This is tantamount to saying that CP in line editing corresponds to the cursor in video editing to be discussed below. Digital Research's manual explains that CP is between two characters, while Zaks (1980) speaks of its pointing to the rightmost of those two characters. The difference may be interpreted as a matter of explanation, and is not of theoretical importance. In particular, CP is either 'before' or 'at' the first character when it is moved to the end of a line. The visual cursor points to the character which it is at, whereas the imaginary CP may be freely interpreted as pointing either at or just after it.

Access to the character(s) to be modified can anyway be made by moving CP first to the line where the character is and then moving it to the point where it is located. For this pupose several commands are provided; +/−nL, which moves CP n lines forward (from the beginning) or backward, or +/−nC which moves CP n characters forward or backward, or +/−B, which moves CP to the beginning (end) of the text, etc. Notice that at the end of a line there are supposed to be two characters, ie. CR and LF, which must be taken into account in counting the CP position.

On arriving at the point to be edited, a variety of commands makes it possible to execute such indispensable editing processes as insertion or deletion; the typical insertion command is I (or i, the capital changing every character entered into a capital) which leads us into the insert mode; D command deals with character deletion, while K command delete lines. Beyond the mere editing functions, ED has a poewerful 'search' function which reminds us of the competence of microprocessors. F command searches a specified string anywhere or as many times as we order, and N is provided with the additional capacity of searching the string not only in the memory buffer but also in the unappended file on the disk. Further refinements can be found in the S (subtitute) command which substitutes one string for another string, while J (juxtaposition) command can execute the more complicate operation of finding one string and inserting another string after it, and then deleting a string till CP reaches a specified string; J finds a specified string of characters, adds an arbitrary string after it, and delete any string after the inserted string. These search and substitute functions add greatly to the capacity of ED as word processor, and may be regarded as a beginning of information retrieval.

The function of ED, though apparently crude in comparison with later developments and rather inconvenient mainly due to its lack of screen editing function, may safely be regarded as possessing almost all the necessary requirements. Though we cannot afford to refer to all all the commands of ED here, at least it must be necessary to add that many commands can be given as a command line (eg. *3f Jones ^Z−3Cia+CR: fined the third Jones, insert 'a' after 'o', which rewrite Jones as Joanes). This culminates in so-called 'macro' commands, which allow a command line to be executed repeatedly. Finally the R command enables another file (ie. so-called 'library file') on the disk to be inserted.

The most remarkable feature of ED's memory function is that it is (perhaps) too meticulous in preserving the original text and in utilizing memory capacity, a fact that has undoubtedly had a great influence over later editors and processors. Whenever ED is executed for CP/M's command line, a temporary file (having the extension of $$$) is created. At the end of editing, when the buffer is copied into the temporary file, ED renames the original file as the file with the extension of BAK (ie. backup) and the newly edited text is the most updated one. The provision for memory capacity is made by writing onto the disk at any time using W command, thus emptying the edit buffer. These considerations for backup files and memory capacity are worth mentioning in discussing the ED of the CP/M.

4. Screen editors as word processors (WORD-MASTER and others)

What are called here 'screen editors' differ from the line editor in the previous sections, primarily in that they have a 'video eiditing' mode. Instead of having recourse to imaginary (or logical) CP, a portion of the document being edited is always displayed on the CRT screen and access to any point can be made by moving the visible cursor. One well-known example of this type of editors is WORD-MASTER (a trade mark of Micro Pro International Corp.)

The important point is, it may be asserted, that though the introduction of the video screen is most convenient, the traditional line editing method has greatly influenced these new developments. Here we can see again a gradual accumulation of knowledge and technology in the history of science in general, of various tools in particular.

In using WORD-MASTER (abbreviated as WM), we are in one of three modes: video mode, command mode or insert mode. At the start-up video mode is always chosen. As mentioned just now, WM's commands in the command mode, which roughly correspond to the line editing above, are easily recognized as being a superset of ED's commands, which thus are very easily handled by those familiar with ED. Hardware restrictions are limited to the video mode which requires a video display (CRT terminal) capable of at least random cursor addressing

Employing the concept of CP here again, we may say that in video mode CP may be regarded as being located visibly after the character on which the cursor is (usually blinks). In the command mode, however, it is invisible as it is in ED's line editing. Corresponding to ED's temporary buffer, the 'Scratch-pad Memory' serves to store the editing text for the present. Parallelism between the command mode and ED's line editing mode considered above are thus fairly close, though remarkable refinements are made in WM's commands. Our attention will therefore be focussed on the video mode editing.

The manual says that Video Edit mode might better be described as 'video display and edit mode', because a screenful of text from the file is on display and immediately updated to show the effect of each key struck. In this mode we are concerned with moving the cursor (not invisible CP) to the desired position by entering various control characters (by depressing usually CTRL key plus some specified key). For example, typing CTRL and O at the same time turns on insertion (ie. entering the insertion mode), and any non-control characters typed are inserted at the cursor position, the rest of the line moving to the right and being redisplayed.

For the purposes of cursor motion and text edition in the video mode, WM defines a 'word' as a string of characters delimited either by blank characters (where 'blank' is broadly interpreted as space, tab, CR or LF) or by punctuation characters if no blank is present. This definition allows a string to be a word where it is separated by special characters without surrounding spaces. Some of the control characters used to move the cursor are: ^F (^ stands for CTRL) for cursor forward one word; ^A (cursor back word), while to move the cursor by a single character, ^S (move left one character) and ^D (right one character) are to be entered. ^Q is the command needed to move the cursor right tab stop, and in order to move it by line ^E (up one line) and ^X (down one line) are to be keyed in. ^^ (ie. CTRL+^) moves the cursor to the top (or bottom) of the screen, while ^B moves it to the left (or right) of the line; these are 'toggle' commands (on and off every time they are typed). For the larger-scale screen movement, ^W (roll down one line to display one additional line) and ^Z (roll up one line to show at the bottom, a line disappearing off top of screen) are provided, while ^R and ^C are the commands to move the cursor to the very beginning (end) of the text.

The control commands for deleting characters (or the whole text) can also be entered in the video mode such as ^G (delete one character), ^T (or ¥ for word deletion), and ^K (^U, ^Y for line deletion).

As mentioned above, the various command mode can safely be regarded as being a superset of those available in ED of CP/M, and thus are fairly easy to execute for those familiar with ED commands. In the command mode familiar CP movement can be entered, moving n characters (lines) forward or backward, deleting charaters (lines) and so on. The method employed in line editing can in most cases be applied here. Commands can be given similarly in the form of a properly formed string of commands (ie. as a command line) which execute a series of command from left to right. From the command mode we can enter the insert mode by I (i) command as in ED, which enables the text typed in to be entered into the file at the CP position until ^Z is input, terminating the insert mode. Of course insertion (or deletion) is possible or more efficient when we are in the video mode.

The 'macro' commands mentioned in ED are further enhanced as 'loop commands' and serve to execute commands repeatedly. 'Search' function too is strengthened: a distinction between a 'short search' and a 'long search' is made, the former being suitable for finding a nearby text because it stops at once when the text is not found nearby, the latter continuing to search to the limit of the file. Obviously these correspond roughly to F and N commands in ED. Likewise besides substitution commands similar to ED's an extension is made by providing additional commands lest the processing of the command line should not terminate; the notion 'branch on failure' is introduced to continue processing even if some commands of a loop cannot be extecuted. The substitution commands may serve to find and delete a part of the text if a null argument is used as a string to be substituted (eg. S×^Z substitutes the × for a null string, that is, deletetes the string).

Finally mention should be made of the 'Scratchpad' capability of WM, a refinement of ED's temporary buffer function. A general form of nQP put n lines (forward from CP) of the text into the Scratchpad, the temporary text being deleted from the whole file. The Scratchpad (so-called Q buffer) has additional functions of a loop, repeating commands for arbitray times (corresponding to ED's R command), of copying Q buffer in the file, and so on. The only improvement, which I would like to have made in actually using WM, is that the line number of the present CP should somehow be displayed, because, even though in the video mode lines could be be counted visually, it is much more convenient to know the line number of CP by seeing it displayed on the screen. The Screen Editor of DUAD-88D (a trademark of ASCII CONSUMER PRODUCTS INC.) is by far superior in this respect.

This editor of Japanese make is installed for the Japanese microprocessor (NEC PC-8800 series) and can be acclaimed as providing a new level of ease and convenience in editing text files and in our case as a word procesor. Of course the editor is restricted to one specified machine which is operated by the particular operating system (OS) different from CP/M, but for the user/owner of the machine it is certainly of powerful editing capability because it utilizes to the utmost the machine's excellent function of screen editing. Though not so variegated as in WM, sufficient commands and control codes for screen addressing are provided, and I am certain that the users of the machine will get accustomed to manipulating the editor in no time. I cannot afford to discuss these kinds of editors which are provided for various machines by various manufacturers, but let me add that it is regretable that some excellent editors are useless because they are restricted to some specified machines, and that we hope some common operating system such as CP/M enables them to be coupled to a variety of machines.

5. An example of per se English word processors (WordStar)

In the previous two sections I have been concerned with some microcomputer softwares which can be utilized to process ordinary or scientific documents. As stated above, these are primarily designed for editors which help us to create program files to be run on the machines. At least for us Japanese, the 'word processor' means a tool for preparing ordinary documents. As things stand, we have hardly any Japanese 'word processors' for preparing programming texts, chiefly because there are scarecely any programming languages based on Japanese, from an assembly (machine) language to a higher language which is apparently at least akin to an ordinary natural language. Of course matters are changing rapidly; we have seen the invention of Kana BASIC or Kanji CP/M, for example. In view of the Japanese complex writing system mentioned at the beginning of this essay, it might be inconceivable or fairly difficult to write programs in Japanese (esp. using Kanji) and process particular strings of characters efficiently in the near future (so it seems to ts amateurs of microcomputers). In English, on the other hand, higher languages tend to become more and more like ordinary English, so that editors originally designed for preparation of programming texts can be diverted to edit ordinary documents. Indeed, I have used above these text editors as the 'word processor' of this essay.

The utility programs for preparing documents, that is, 'word processing program per se,' are being devised and appear on the market, some of which are of Japanese make. Among the most popular examples of this per se processors, those which can be regarded as sophisticated versions of the traditional text editors, is WordStar (which is a trade mark of MicroPro Internation Corp.)

WordStar is often said to be an excellent word processor except for the fact that there are so many functions it is apparently impossible for ordinary users to be a complete master of them; indeed there are about hundred commands to control preparing, editing, storing and printing documents. This comment is, however, obviously misleading. Once one gets used to the main operations of the program, the chief and important commands are easily found to be in accord with those of WORD-MASTER and therefore of CP/M's ED. (Notice that WordStar and WORD-MASTER are the products of the same software manufacturer.) My overview of WordStar is that it is thus comparatively easy to use in view of the analyses made in the previous sections.

WordStar is a fully screen-oriented word processing system, compatible with CP/M, and is intended for the preparation of either either programming texts (which are called 'non-documents') or ordinary English (or other alphabet-using) documents by non-technical persons. The processor is, as mentioned immediately above, rather more congenial to CP/M users than to those of any other disk operating system. The function of video screen editing is enhanced here to the utmost, the most updated text being always shown on the screen. The modifications which I wished to have made to WORD-MASTER are almost completely realized; the helpfulness of line numbers to which I referred in considering line-editors is full taken into consideration. Here not only line numbers, but also the page and the column of the line at which the cursor is located in the memory buffer are always displayed and made much use of. The 'status line' which is the uppermost (ie. first) line is used for this purpose, where other information (eg. command in current operation, or file name) is displayed besides the one about the cursor position.

The rather complicated commands, though at first sight formidable, can gradually be mastered by consulting Help menus which can always be displayed on the screen and at four levels, according to the operator's knowledge, eliminating the need to refer to the manual, and at the last top level the menu completely disappearing, which allows full screen utilization for the text. The commands to move (address) the cursor to any position are mostly in accord with those of WM's; eg. cursor left (right) one character (word), cursor up (down) one line, roll up (down), or file up (down) one screen (page), etc. The additional function of rapid ('Q'uick) cursor movement is performed by prefixing Q to control codes(eg. ^QE (^QX) to move the cursor to the top (bottom) line of the screen; especially convenient is ^QS (^QD) which moves the cursor to the leftmost (rightmost) of the line). Deletion commands too are almost the same; eg. ^G(^T) to delete one character (word). The insertion mode is turned on and off by ^V, where any numbers or characters (including of course spaces) can be inserted from the cursor position on. The function of screen editing culminates in that of 'on-screen' printer-image formatting; the effect of margin type and margin settings is displayed on the screen just as the text will be printed on paper. The most desired capability of right-margin justification (or sometimes ragged-right margin) in ordinary (non-programming) English word precessors is fully supported and faithfully displayed on the screen. One of the most knotty problems in English documents, syllabification, is settled dexterously by introducing the functions of 'word-wrap' and 'hyphen-help'. The function of word-wrap eliminates the need to enter CR at the end of each line, a word too long to fit at the line end being automatically moved to the next line; CR is requested to be used only to indicate the end of a paragraph. When the margin settings are changed, the useful function of 'reform' control realigns margins and reforms the whole paragraph; it may be necessary to change right-margin hyphens which have been set according to syllabification rules; then the hyphen-help asks whether and where a hyphen should be put. Here a distinction is made between 'soft' and 'hard' hyphens, the former being necessary only when it is at the end of a line while the latter is always necessary. (In the IBM typewriter examined above, this distinction is accounted for by Required Hyphen control.) Anyway the hyphen-help function, together with the notion of soft and hard hyphens, enables us to manage a rather delicate problem of syllabification in English. It would be rather tedious to mention other convenient features of screen editing which can be realized visibly on the screen; such now familiar functions of a WP as automatic-centering, or tabulation commands, for example can be freely employed.

Search and substitution commands are similarly provided as in WM and far easier to handle for non-specialists; a dialog form questions are submitted, by answering which the commands can be executed. Many optional commands are provided to execute such functions as global search, automatic replacement, backward search or whole word search. Memory saving is quite in parallel with that of WM (therefore of ED). As a CP/M compatible program, WS allows CP/M's command to be executed; this is very convenient, for instance, in finding out the memory capacity of the disk by using CP/M's STAT command.

As a per se word processor WordStar's capability of printing edited documents is the most powerful imaginable; there are more than thirty commands for this purpose. They are mostly intended for controling printing mode; eg. boldface or double prints, subscript/superscript, underscore, pitch setting (pica/elite), printing pause (which enables the printer to stop at a specified point, the changing of a printing ball or wheel being then possible), or overprint character which is necessary in French accent marks or German umlaut sings and so forth. Some of these printing commands depend upon hardware capability and may need user's patches of the program itself. Of course the right-margin justification, as mentioned above, is set on and off (in this case a ragged-right margin is printed) already in the video editing mode (on-screen formatting). Further refinement can be seen in so-called 'dot commands' (because they are in the format of dot plus two consecutive characters) which can specify the configuration of the text on a page, the layout of the text, the printing of a heading (footnote, page number) or the command of page change (T. O. F) and so on.

Thus surveyed, WordStar as a typical example of the most updated word processor softwares may well be evaluated highly both in terms of its editing capability and of its rich printing-mode multi-functions. Other word processors (including their softwares) may presumably be following suit in their processing functions. Lastly what I wish to emphasize boils down to the power of know-how of traditional editing algorithms from the time of ordinary typewriters.

6. The significance of word processing for preparing English documents

I cannot afford here to enter into a detailed discussion of word processing in general as a tool for making documents, so I will conclude this overview of English word processing with a brief consideration of its possible impact upon the prepartion of English documents, especially scientific papers.

On many occasions the documents or papers we Japanese draw up are of academic nature in its broader sense, including, for example, publication of a newly developed device for commercial purposes. Increasingly more learned circles in Japan tend to publish their research results in English, not to mention those pioneering works which are to be disseminated worldwide as quickly as possible. Now even in academic journals written mostly in Japanese the abstracts and illustrating figures and diagrams containing explantory comments are supposed to be written in English. Thus papers which are to be valued first of all in the academic world are inevitably to be written in English. In this sense the preparation of an academic document requires a thorough command of written English. For us teachers of English teaching how to write or compose a lucid expository paper in English should be an important and 'practical' task at least in an upper-grade class. There are many problems involved here, grammatical, collocational or stylistic. And after these obstacles have temporarily been overcome, drawing up a neat draft is necessary. It is perhaps at this stage that our word processor is required to make its appearance.

Word processing as considered in this essay, it might be asserted, appears to be at best of secondary importance, the primary significance lying in composing documents that are linguistically and stylistically tolerable. Admitting this criticism, I should like to stress the immense value of word processors. They may well be criticized as the electronic substitute of a traditional printer. This remark is quite to the point. I would conclude, however, that this is the very point in which the word processor is valuable. Notice that I am not speaking of its value as a powerful weapon of the fashionable Office Automation. Throughout the essay I confine myself to its use as a valuable tool in my study or in your university office. Here the processor enables an ordinary individual to possess his/her own printing shop. Let me go further and say that the printer is not here only to print out a neat document; we should not forget the processor's indispensable role of clearing up our thought itself. As a concrete example, I will take up Word Star's function of 'block' movement of some passage in a document. After drawing up a first draft, it often occurs on rereading that some passage had better be placed in another place and then by using the block movoment command we can move any passage to any other desired place. Can't we say that, by surveying the newly arranged passages visually displayed on the screen, we are able to clarify our own thinking? At least my experience in handling the processors led me to believe that the use of a word processor undoubtedly enhances the efficiency of preparing documents, improving at the same time the thought process itself.

The software programs which I happen to consider in this paper are merely ones available to me, but they are some of the most fundamental and traditionally signigicant ones, and there is no doubt that further improvement both in hard-and soft-ware aspects contribute enormously to drawing up English documents.

(30th April 1983)

*Devices used in writing the paper are:

IBM Mag Card Ⅱ typewriter (IBM Corp.)
PC-8800 series system (Nippon Electric Co.)
MZ-80B series system (Sharp Corp.)

*Software programs utilized are:

DUAD-88D (ASCII Consumer Products Inc.)
FDOS (Sharp Corp.)
MZ-80B CP/M version 2.2A (Digital Research/Microsoft Associates)
NEC PC-8000 series CP/M version 2.2 (Digital Research/NEC Co.)
WORD-MASTER release 1.07 (MicroPro International Corp.)
WordStar release 3.0 (MicroPro International Corp.)

*References:

DUAD-88D user's manual (1982): ASCII Consumer Products Inc.

IBM MAG CARD Ⅱ Twypewriter operating instructions and applications supplement (1977): IBM Corp.

MZ-80B CP/M 2.2A user's manual (1982): Microsoft Associates/ Digital Research (1978)

NEC CP/M 2.2 user's manual (1982): Nippon Electric Co./Digital Research (1978)

WORD-MASTER release 1.07 reference guide and operator's guide (1980): MicroPro International Corp.

WordStar release 3.0 reference manual and training guide (1982): MicroPro International Corp.

Zaks, R (1980) the CP/M handbook with MP/M: SYBEX Inc.

(Received 4 May, 1983)

P.S.

At the time of the proof-reading (Sept. '83), my greatest benefit from WordStar is its capability as a data-processing tool. Without sophisticated data-base utilities, it can supply me with a sufficient weapon for processing text data. Details aside, my procedure is roughly as follows: a key word (or string) can be identified and replaced throughout the whole text by an appropriate mark, using ^QA (Find & Replace) command and the passage containing the specified mark then is saved onto a disk as a new file by ^KW (Write Block) command, thus a set of files containing the key string being easily obtained, which, if necessary, can be merged onto a new file by CP/M's PIP command.