International Atomic Energy Agency

INDC(CCP)-250/L

INDC INTERNATIONAL NUCLEAR DATA COMMITTEE

# DATA INPUT TECHNOLOGY FOR PRODUCING A SECONDARY INFORMATION DOCUMENT IN "GENERALIZED EXFOR" FORMAT ON A MINICOMPUTER

G.M. Zhuravleva, A.E. Ingnatochkin
and F.E. Chukreev

(Translated from Nuclear Constants 2(56) 1984, pp. 60,
Russian Original distributed as INDC(CCP)-232/G)

Translated by the IAEA

October 1985

IAEA NUCLEAR DATA SECTION, WAGRAMERSTRASSE 5, A-1400 VIENNA

INDC(CCP)-250/L

# DATA INPUT TECHNOLOGY FOR PRODUCING A SECONDARY INFORMATION DOCUMENT IN "GENERALIZED EXFOR" FORMAT ON A MINICOMPUTER

G.M. Zhuravleva, A.E. Ingnatochkin
and F.E. Chukreev

(Translated from Nuclear Constants 2(56) 1984, pp. 60,
Russian Original distributed as INDC(CCP)-232/G)

Translated by the IAEA

October 1985

Reproduced by the IAEA in Austria
October 1985

85-04962

Translated from Russian

INDC(CCP)-250/L

UDC 681.327.12:539.1

# DATA INPUT TECHNOLOGY FOR PRODUCING A SECONDARY INFORMATION DOCUMENT IN "GENERALIZED EXFOR" FORMAT ON A MINICOMPUTER

G.M. Zhuravleva, A.E. Ignatochkin and F.E. Chukreev

Abstract

Technology for computerized input of scientific and technical abstracts in the international "Generalized EXFOR" format using a minicomputer is described. The possible applications of this technology are considered and it is compared with existing data input systems.

One of the main operating modes of any computerized information system is the data input mode. There are various types of data input technology, depending on the specific features of the data processing system. The most important of these features are:

- The extent to which primary information is ready for input to the system;

- The composition and characteristics of the system's hardware and software;

- The amount of data involved;

- The extent of human participation in data preparation;

- The use of different data preparation devices (in particular, manual input devices);

- Rules for the preparation of data used in the operation of processing systems.

Data preparation by recording information directly from the keyboard of a data preparation device onto a magnetic carrier has by now become common [1-3]. Reference [1] describes four basic data input schemes for converting non-formalized primary information into data which can be accepted by the computer for further processing.

The process of converting non-formalized information and putting it into a form which is convenient for further processing is a most laborious one. Non-formalized primary information includes, in particular, books, periodicals, handbooks and similar material. It is characteristic of this type of information, first, that it is voluminous, and second, that many information publications duplicate each other. This is why the selection work necessary for input into the computer of data derived from the primary information is so laborious. For this reason the primary analytical processing and indexing of information, on the basis of which a secondary information document (SID) is drawn up, is performed by highly qualified experts. In addition, operators are needed to key all the material prepared into the computer. It is also important to make proper use of experts on different subjects during the process of converting primary information into data which are subsequently stored in data bases and banks for further processing and utilization, since this factor determines the productivity and quality of the entire data processing system.

The first stage in the schemes presented in Ref. [1] is that of primary analytical processing. Basically, this involves identifying the semantic information (meaning) contained in the primary information document (PID) and composing an SID consisting of formatted data stored on some carrier which can then be entered into the computer by means of input devices.

In the first scheme the SID is composed on paper and then transferred to punched cards or tape. The main feature of this scheme is dual input with character-by-character comparison of the information as a check. For the input of information using this scheme, staff are needed in four process sectors: primary analytical processing, punching 1, punching and checking 2, data input to the computer, as well as accountancy, checking and transfer of SIDs and punched cards between these process sectors.

In the second scheme the SID is also prepared on paper, but transmitted to a visual display unit (VDU) operator who keys in an image of the SID on the screen in accordance with set rules, checks the correctness of the text and presses the "Enter" button to put the SID into the computer. In the third scheme, the staff responsible for the primary analytical processing key in all their "records" directly on to the screen. In the fourth scheme, the SID is composed on paper, and strict standardization is ensured by special forms or printers. Usually, a data input scheme of this type is used when primary analytical processing and SID composition are carried out at different locations some distance from where data input takes place.

The analysis in Ref. [1] brings out the advantages of schemes using data input from VDU screens rather than punched cards or tape. It is pointed out that productivity in the third scheme is two or three times higher than in the second and three or four times higher than in the first. The main advantages of the third scheme are as follows:

- The SID composed on the VDU screen after primary analytical processing is verified by the staff responsible for such processing, and there is no further human interference with it;

- Composition of the SID on paper ceases to be necessary, which eliminates operations such as accountancy, transmission, storage, etc.

Reference [3] describes technology of this type for the input of abstracts into an integral information system and discusses the extent to which it is possible and desirable to combine the functions of indexer, abstracter and operator. Even if it is considered advisable to combine the functions of indexer and abstracter, combining those of abstracter and operator is useful only where the entire flow of information into the system is processed by staff abstracters who are in charge of maintaining the system. Where outside abstracters are used in this scheme, the information produced by them is transmitted to an electronic input section for operators to record on machine-readable data carriers. In addition, the use by abstracters themselves of direct input technology is subject to the following conditions:

- The volumes of information to be recorded must be small (the standard time for composing a document description and recording it on magnetic tape is approximately 12 minutes);

- Sufficient machine time must be made available (not less than two hours in accordance with a strictly determined schedule);

- The electronic input terminals must be part of the work station or in the room of the abstracter;

- There must be fast channels of communication with the computer room;

- The computer operating modes must be reliable;

- There must be an appropriate system of monetary incentives.

The evaluation of direct input technology given in Ref. [3] does not cover the stages of correcting and editing the documents recorded. In the proposed scheme, these operations would presumably be performed by the abstracter, who would thus have to learn not only the data input process but also how to carry out formal logic verification procedures with subsequent editing of the documents on the computer. Although experience shows that abstracters can successfully perform the various functions outlined using VDUs as the main technical basis for processing and recording the documents they have composed on an appropriate medium (magnetic tape), the conditions mentioned above make it inadvisable to entrust the abstracters with these functions, especially when the volumes of data to be entered are large.

Large volumes of data and the need for the most reliable and useful data to be selected and for these data to be correctly placed in the SID all provide an impetus to devise systems where intellectual processes are replaced by automatic ones to the maximum extent possible so as to reduce the amount of manual work; this would include the automation of SID input [4, 5].

For the preparation of quantities of data with closely related subject matter by centres using systems application resources, a scheme is proposed in which the functions of abstracter and operator are separated. The abstracter, who is engaged in analysis, synthesis and indexing of information, is responsible for the quality and reliability of the data selected from the primary source and transferred to the SID being "assembled" (coded). The operator is in charge of processing these SIDs on the computer (input, editing, formal logic verification analysis and so on). Direct co-operation between the abstracter and the operator throughout the processing cycle for SID files ensures high quality of data assembly for final inclusion in the data base.

At the same time, unlike technical sources of interference, the human operator is a specifically "active" source of error, since he can not only commit errors but also identify and correct them. Although it would appear impossible to give a full account of the influence of operator errors on the reliability of data processing in computerized information systems (in any case, the data in the literature are partially contradictory [6-8]), it is believed that information is distorted by errors committed by the operators of manual input devices to a much greater extent than by other sources of interference and error.

There are, however, ways of improving the effectiveness of operators' data input work. One of these is to reduce the volumes of data recorded on punched tape or cards. At a computer centre engaged in the processing of scientific and technical information which is content-linked and structurally dependent, the operator does not key in the whole of the SID being assembled (the structure of which must in any case be consistent with the requirements of further processing), but only its semantic part, which is prepared by the abstracters on paper, often in free handwritten form. By this means the productivity of operators' work and the reliability of the data can be improved. A second way, supplementary to the first, is to write input programs containing functions for assembling formatted SIDs in accordance with the program's composition rules.

The data input procedures can be programmed in various different ways. In the first place, they can be programmed anew for any particular case in any procedure-oriented language. Secondly, they can be written as universal programs of the interpreting type. Thirdly, they can take the form of program generators.

Unlike the first of these techniques, the second and third do not require a specified data processing technology, only the input and output parameters need to be described. However, adjustment to a particular medium with universal programs is so complex that it is usually necessary to develop different programs for input from different media [9]. This problem does not arise with program generators, since a block for information input from a particular medium is produced in the generating process [10]. However, program generators are usually designed to process and assemble amounts of data from different types of structure with large numbers of parameters. The data to be assembled may have the same parameters but different structures for the output files. The program generator, using input and output data on the links established between parameters - data which the user specifies by means of order forms - assembles the actual program which corresponds to the structure of the output file required. But if the file being assembled has a particular structure, there is no need to design logically complex programs capable of producing all necessary output file structures while making allowance for the special conditions of working together with users.

In order to assemble quantities of data with a constant formalized output file structure, it is thus more efficient to write an input program oriented towards the specific type of processing; this will lead to a program which is more effective in terms both of speed and of the volume of the working memory. Minicomputers represent a satisfactory solution to this question, and the use of specialized programs is more rational (optimal) for solving the particular problem. The fact is that when data are processed for a single user, operations hardly need to be carried out with extreme speed, and an enormous working memory is not required when peripherals of sufficiently high performance are available. Moreover, use of the central processing unit of a minicomputer results in additional convenience in that the opportunities for dialogue between the operator and the computer are extended, so that the process of data input is simplified and the language of communication between the computer and the operator can most closely resemble natural language. At the same time, the application of single-user minicomputers makes it easy to overcome one of the main psychological problems, that of reducing to a minimum the time elapsing between asking the "question" and receiving the "answer" in a pre-planned dialogue.

In such a pre-planned dialogue the initiative is usually taken by the program. However, after receiving the "invitation" from the program to key in control information, the operator will select one or other of its branches. Unfortunately, analysis of symbolic information in high-level languages is relatively laborious and even an operation such as recognition of the standard answers "Yes" or "No" takes a fair amount of time. The program's response time to operator questions can be reduced to a minimum by using languages of the assembler type (ASSEMBLER or MAKROYAZYK) [11, 12]. Also, if the output information is formatted and content-linked, it becomes possible to have certain obligatory information assembled automatically, and thereby to exclude the possibility of its being distorted or lost during input. With a formatted structure, each record consists of a number of sectors (fields and elements) and the "address" of any field in the file can be determined definitely and unambiguously. This opens up opportunities for filling the files with all sorts of data, such as data entered automatically from different carriers or obtained by conversion, for example directly from printed publications.

A specialized, semi-automatic data input program written in the international "Generalized EXFOR" (GE) format meets the above requirements [13]. This program is written in a language using the ASSEMBLER macro-extension for the 1010B minicomputer [14].

The international GE format is intended for the "factographic" description of physics experiments. Use of a fixed format means that the input of data and their partial coding can be automated. The input process consists in a pre-planned dialogue between operator and computer in which transaction processing is used to specify the sequence of computer operations. With this approach it is possible to start entering an SID practically without any detailed instructions or study of a special language. Manual input is combined with automatic data assembly and recording in a form dictated by the formatting rules. The operator's answers to the program's proposals constitute instructions for further computer operations. In this way errors are avoided, since the operator need not remember the sequence and format of the information. During the dialogue each line of SID is displayed on the VDU screen as it is keyed in. This enables the operator to recognize and correct mistakes committed in that line.

Checking the reliability of input data when they are first entered into the system is one of the most important tasks. Reference [7] shows that the stages most likely to distort the results are those of data preparation and output, so the most thorough check on the reliability of data is essential at these stages.

Program checks on data reliability can be divided into a number of categories: checks covering individual characters, the field, the line, the file section and the whole file. Since these checks represent a large body of program effort it is advisable to separate the input program from the checking and editing program. The input program is thus left only to check essential data and those which do not increase the response time in the pre-planned dialogue.

The program for SID input and assembly performs the following functions:

- It monitors the correctness of data and structures the data in accordance with the rules of the international GE format;

- It checks the characters keyed in;

- It indicates any errors committed on the screen;

- It conducts a pre-planned dialogue with the operator about the correctness of data input;

- It records the coded and formatted SID (the ENTRY) on an interim magnetic tape for further processing;

- It prints out the coded ENTRY on an alphanumeric printer;

- It informs the operator when peripherals are ready for operation.

The input process begins, after the program name has been entered, with selection of the input mode. The SID can be entered in two modes: input of a new SID and addition of information from an SID the beginnning of which has already been stored as a formatted ENTRY in the working library. The library will have been recorded on magnetic tape with a file storage structure. Access to any ENTRY in the library takes place by entering two names: the name of the library and the name of the ENTRY. The elements of the last line of an ENTRY being supplemented are necessary parameters for ensuring that the structure of all the information subsequently entered from the SID is correct. The program ensures that the ENTRY with the added information has the correct structure for subsequent storage in the working library. When a new SID is entered, the input data are specified as initial conditions of the program.

The hierarchical structure of the format is such that the input program can be divided up into procedurally linked modules. This makes it easy to develop programs both for improving input modes and for extending the service capabilities and processing procedures.

Four basic program levels can be identified in accordance with the hierarchical structure of the format:

- "Assembly" (generation) of the overall structure, which is determined by a fixed sequence of system identifiers (SI);
- Assembly of the bibliographical data section (BIB section);
- Assembly of the numerical data tables (COMMON and DATA sections);
- Assembly of line identifiers.

The SID of the ENTRY is placed in a particular zone of the disk after input. The address of the beginning of the ENTRY is fixed. After the input process has ended and the operator has informed the input program accordingly, the ENTRY is copied from the disk onto magnetic tape. The reasons for recording the ENTRY first on a disk and then on magnetic tape are: the requirements of automatic data assembly, the need to be able to handle emergencies in the event of system failure and the requirement of a degree of mobility for programs with a view to further processing of the information entered.

In an emergency the operator enters the name of a program which copies the data entered from the disk onto magnetic tape so that the information entered before the failure is not lost.

Data are entered line by line. The program codes the data in punched card format as required by the rules of the international GE format. Each line is divided into three zones. The first eleven spaces are assigned to system and information identifiers. This zone is filled semi-automatically.

Spaces 12-66 are reserved for data, which are all entered by the operator. The program merely assigns positions to the information entered in accordance with the structure. The third zone (spaces 67-80) is filled with line identifiers automatically by the program.

Assembly of the overall structure

The overall structure of the coded SID is assembled by specifying the appropriate sequence of system identifiers: ENTRY, SUBENT, NOSUBENT, BIB, NOBIB, COMMON, NOCOMMON, DATA, NODATA, ENDDATA, ENDCOMMON, ENDBIB, ENDSUBENT, ENDENTRY. The system identifiers (SIs) are called up on the screen by means of functions included in the program. A special area is allocated to the list of SIs (called the V1 file), as follows:

| | | |
|---|---|---|
| ENTRY | BIB | DATA |
| SUBENT | COMMON | ENDSUBENT |

At any moment in time the V1 file index determines the SI which corresponds to the ASSEMBLY SEQUENCE ENTRY in accordance with the GE formatting rules.

The SIs SUBENT, BIB, COMMON and DATA determine, in accordance with the formatting rules, the beginning of the corresponding data unit (DU) of the sub-entry or section. This means that, if a given DU is filled with data, the operator must confirm that it continues to be present in the ENTRY by keying in a confirmation symbol (any symbol except N). If there are no data for filling this DU, the operator states this by entering N on the proposed line with the SI; the line containing an SI indicating a DU is then replaced on the screen by a line with a correspnding SI indicating its absence, i.e. by the lines NOSUBENT, NOBIB, NOCOMMON, NODATA.

For the definitive assembly of the BIB, COMMON and DATA lines, information is required which can be obtained only after these sections have been completely filled. Thus, after receiving from the operator an instruction about how to continue filling the section with data, the program notes the address on the disk of the line containing this SI. The instruction to end filling in of the DU is given by the operator in the form of a termination signal which causes lines with SIs indicating the end of the DU (ENDBIB, ENDCOMMON or ENDDATA) to be called up on the screen and lines with SIs indicating its beginning to be copied from the address noted.

The assembly of the next sub-section (the SUBENT DU) is terminated fully automatically after the COMMON or DATA DU assembly has finished by means of the line ENDSUBENT. A line with the SI SUBENT appears automatically on the screen after this line. Thus, depending on the operator's response, the program proceeds either to fill in the next sub-entry or to terminate the input process.

The process of input and ENTRY assembly ends with operator responses to the program's messages and recommendations. Depending on the answers, either a complete ENTRY or an ENTRY to which information from the SID is added later can be assembled.

Assembly of the bibliographical data section

The BIB section can be represented as an object-characteristic table. The names of the objects in the table are determined by a finite set of information identifiers (IIs). The properties of each of the objects are determined either by all of a finite number of characteristics whose codes are found in dictionaries or by free text or by both. In the program an area called file V2 is allotted to the list of IIs (Table 1).

The II presence flag. Each II can be present in the BIB section only once. Thus, as soon as the operator calls up the II and fills it with information, the program responds by changing the value of the presence flag. Before the functional module for filling the BIB section is called up, the presence flag for each II is left blank.

The II availability flag. In accordance with the formatting rules, certain IIs can occur in any sub-entry of an ENTRY, while others can be present only in the first; a third group, if present in the first sub-entry, must not appear in any of the others or, if absent from the first, must be found in all the others. The availability flag thus determines the possibility of a given II being present in any of the sub-entries of an ENTRY. It takes the form of a fixed parameter which does not vary during the operation of the program.

The sequence of IIs in the ENTRY is random, and the next II is called up when the operator enters its first letter. Since there may be a number of IIs with the same first letter, the operator must enter the confirmation symbol for the II shown on the screen. Only after this can the operator begin to fill in the information field.

While filling the information field in the line being assembled the operator performs a visual check of the text being entered. If he wishes, he can correct errors and even delete the whole line and begin to re-assemble it as of the II call (if necessary, another II can be called up). When the symbol for termination of line assembly is keyed in, the line is finished automatically and transferred to the buffer store on disk. The symbol for terminating line filling can be keyed in by the operator (RT); otherwise, termination can take place automatically if the operator is keying in the text without looking at the screen and hence does not notice that the whole information field has been filled. Each successive line of information received by the buffer store on disk becomes inaccessible for correction.

It is posible for the BIB section information field to be filled in automatically. In this case the operator must specify a transaction and, after a dialogue, the name of the file which is to be transferred to the information field. Before being called up, the file to be transferred must be present in the working library. Using a special symbol in the file being transferred, it is possible to create the desired structure for the output file. The BIB section is closed by calling up on the screen the SI ENDBIB specified by the operator in accordance with the II call-up scheme.

Assembly of the numerical data tables

In assembling the COMMON and DATA sections, a number of particular features imposed by the GE formatting rules must be taken into account. The first is that these sections are tables with columns of variables. Each column has eleven spaces. Thus, not more than six columns of data table can be entered in each line. Experimental data tables may, however, contain a larger number of columns. The formatting rules provide for the structure of these tables to be organized in a special way. The second feature is that numerical values can be recorded in the data table in two formats: in decimal format and in "E-format". In the former the numerical value can begin in any space in the column. However, for reasons of table presentation the program places values so that the decimal point is always in the middle of the column. In "E-format" the figure must be placed with its last digit in the last space of the line (i.e. so that it is justified on the right-hand side). The third feature is that the names of the columns and data units must start in the first spaces of a column.

These features are taken into account automatically by the functional module for assembling numerical data sections. After entering the number of table columns as requested by the module, the operator enters the numerical values of the variables. If a column is empty the operator enters the symbol for moving on to the next column (RT). When this key is pressed a check is made of the numerical field and then the value entered is automatically copied into the appropriate column spaces. This process is monitored visually, but the operation is so fast that it does not interfere with the operator's work even if he is entering data "without looking". If necessary, the operator can correct errors occurring within the limits of the line being assembled or delete a line entirely.

The numerical value entered is checked for the following:

- Presence of digits and special symbols (., E, +, -);

- Compliance with the specified format;

- Length of the value entered, which must not exceed 11 characters including special symbols.

The type of error committed determines whether the program corrects the numerical value in the column or whether it queries it again. The COMMON and DATA sections are closed when the table assembly termination symbol ⋕ is keyed into the first space in the first column, in which case the module checks that the number of columns specified is equal to the number entered. If they are not equal a message is shown on the screen to the effect that table assembly is incomplete.

Assembly of line identifiers

In accordance with the formatting rules and the values worked out by the program on the basis of the actual data entered, the value of the identifier for each line of ENTRY is calculated automatically. The input program also checks that the hardware is ready to receive data. If a hardware component is not ready to receive data when the program addresses it, the operator is informed about the need to take specific action in order to proceed with normal operation. Since the input program is part of the overall data processing cycle for the EXFOR system [15], it supplies the following inter-program parameters needed for operation of the system:

- The name of the ENTRY being assembled;

- The date of assembly;

- The name of the old ENTRY if an addition has been made to the ENTRY from the SID;

- The symbol denoting a complete or incompletely recorded ENTRY;

- The number of lines of the ENTRY assembled.

To judge how well the program works, let us compare it (with some idealization) with the input systems used at other international centres, namely at the Nuclear Data Section (NDS) of the IAEA, Vienna, Austria, at the Brookhaven National Laboratory (BNL), Brookhaven, USA, at the Nuclear Data Centre (CJD), Obninsk, USSR, and at the Centre for Nuclear Structure and Reaction Data (CAJaD), Moscow, USSR. The figures shown relate to the period 1982-83.

At the IAEA, data input takes place from a VDU terminal which enables the operator to perform self-checking of the data entered; he is, however, himself responsible for the positional structure of the data. This means that the input speed is affected by the time it takes to place a marker in the apropriate space in a line. The number of fields in a line varies between one (where only an SI such as NOCOMMON is entered) and seven (in assembly data tables). The operator enters the whole line, which is 80 characters long. It should be noted that input is usually off-line onto floppy disks.

At Brookhaven, punched card readers are used to enter large data files. The input speed is thus reduced because errors are corrected by replacing the punched card. For the input of small volumes of data a system is used for preparing data in CSISRS format which is similar in structure to the GE format. Data are then converted from CSISRS to GE format by a special translation program.

At CJD (Obninsk), data input takes place as follows: the textual information up to the next information symbol is entered by the operator, who is himself responsible for the positional structure of the line. When a data table is assembled, a marker is placed in the first space of a column automatically, but the operator is responsible for the positional arrangement of values within a column.

The calculated values given in Ref. [16] were used for the comparison (see Table 2). By way of example, let us examine the structure of ENTRY (A0118). The ENTRY consists of 341 lines, as follows:

| | | |
|---|---|---|
| - | Lines with system identifiers | 63 |
| - | having: | |
| | fields N1 | 27 |
| | fields N1 and N2 | 27 |
| | SI only | 9 |
| - | Lines with information identifiers | 203 |
| - | Lines with columns of variables | 75 |

Since the input program used at CAJaD assembles part of the information automatically, the ENTRY will be made up as follows:

| | | |
|---|---|---|
| - | Lines with system identifiers | 12 |
| - | Lines with information identifiers | 196 |
| - | Lines with columns of variables | 75 |
| - | Lines assembled automatically | 58 |

The following limitations were adopted for the calculations:

| | |
|---|---|
| Number of columns in data tables | 6 |
| Maximum number of digits in numerical values for data tables | 7 |

- Average number of characters in assembly:

  the BIB section information field 50

  lines with SIs 4

The calculation is performed using the formulae $T^L_{key} = nT_{ch}$; $T^L_{pm} = mT_{pm}$; $T_{key} = T^L_{key} + T^L_{pm}$; $T^E_{key} = \sum_{i=1}^{t} T_{key}(i)$, where n is the length of the message being keyed in; m is the number of fields in the line; k is the number of lines in the ENTRY being assembled; $T^L_{key}$ is the time taken to key in the message in a line; $T^L_{pm}$ is the time taken to place a marker in a field of a line; $T_{key}$ is the time taken to assemble the line; and $T^E_{key}$ is the time taken to assemble the ENTRY. The results are shown in Table 3.

In Ref. [1] calculations are given for the labour effort expended on data input into a computerized data bank with an SID input flow of 617 280 punched cards per year. This means that the input speed is 28.2 punched cards per hour per operator. For the EXFOR system with the input program in operation at CAJaD, an input speed of 155 punched cards per hour can be derived from the data in Table 3. Thus, by using a special program for entering content-linked and structurally dependent data, productivity is inceased by a factor of more than five.

In addition to speed, the operator's work can be gauged in terms of reliability, i.e. the absence of errors. In Ref. [16] reliability is defined as the probability $\beta$ that a message will be keyed in correctly, as determined by the formula $\beta = (\beta_{ch})^n$, where n is the length of the message being keyed in and $\beta_{ch}$ is the accuracy with which a single character is transferred from the keyboard to the screen (for VDUs with a MARKER LEFT key in self-checking mode $\beta_{ch} = 0.9994$, while for VDUs without a MARKER LEFT key not in self-checking mode $\beta_{ch} = 0.9950$).

The calculation results presented in Table 4 clearly demonstrate the advantages of computerized input.

In the common information system for processing data in the international GE format, the input program is regarded as a procedural module carrying out a specific data conversion procedure: the input of data from a VDU keyboard into a computer with automatic assembly of these data in a set format.

On the basis of the experience accumulated with operation of the program over a number of years, the following conclusions can be drawn:

(1) The program fulfils one of the most important conditions for operation of computerized information systems, namely once-only data input;

(2) The program does all that it is designed to do with great speed and reliability of data input;

(3) The fact that data input can be performed piecemeal means that it can be spread over short intervals of available computer time;

(4) The time spent by untrained operators on learning data input by means of this program is less than one working week;

(5) The use of structuring methods for the composition and organization of the program renders it easy to read and accessible for improvements shown to be desirable by practical experience.

REFERENCES

[1] OVCHAROV, L.A., SELETKOV, S.N., Avtomatizirovannye banki dannykh, (Computerized data banks), Finansy i statistika, Moscow (1982).

[2] KUZNETSOV, M.V., NEMTSOV, L.F., RUBANOV, V.O., A system of preparing data for a computer centre without punched tape or cards [in Russian], Upravlyayushchie sistemy i mashiny 3 (1978) 30-35.

[3] VOROJSKIJ, F.S., TITOV, V.A., Direct document input technology in an integrated data system [in Russian], in: Problemy avtomatizirovannoj obrabotki nauchnotekhnicheskoj informatsii (Problems of computer processing of scientific and technical information) (Proc. Inter-Disciplinary Scientific Conferences, Conventions, Seminars and Exhibitions), VIMI, Moscow (1982) 123-130.

[4] PAVLOV, A.N., Experience with the use of a data preparation system for the input of textual information [in Russian], ibid. (1983) 26-28.

[5] BUD'KO, N.S., GRIGOR'EVA, K.A., Possible improvements in the operation of computerized systems for scienfitic and technical information for the automation of preformatting [in Russian], ibid. pp. 54-58.

[6] GRAJFER, R.S., Mathematical models for evaluating the work of keyboard operators [in Russian], Upravlyayushchie sistemy i mashiny 2 (1983) 44-48.

[7] LIPAEV, V.V., KALINICHENKO, I.M., Reliability of "ASOU" computerized data processing systems [in Russian], ibid. 1 (1979) 3-6.

[8] TSYBULEVSKIJ, I.E., Incorrect responses by the human operator in a control system (Review of research abroad) [in Russian], Avtomatika i telemekhanika 6 (1977) 112-143.

[9] BELONOGOV, G.G., KUZNETSOV, B.A., Yazykovye sredstva avtomatizirovannykh informatsionnykh sistem (Linguistic resources of computerized information systems), Nauka, Moscow (1983).

[10] Generator program vvoda dannykh dlya ES EhVM (A data input program generator for the Unified System (ES) computer), Statistika, Moscow (1976).

[11] KETKOV, Yu. L., Software for dialogue with an application program controlled by a Unified System (ES) DOS Computer [in Russian], in: Optimizatsiya i MO SAPR, Moscow (1980) 126-140.

[12] LYULINETSKIJ, V.P., Aspects of improving the effectiveness of computerized information system development [in Russian], in: Programmnoe obespechenie upravlyayushchikh kompleksov sistem malykh EhVM (Software for process control with minicomputer systems), Proc. Institute of Electronic Control Machines (IEhUM), 80 Moscow (1980).

[13] NDS EXFOR Manual, Documentation Series of the IAEA Nuclear Data Section, IAEA-NDS-3, IAEA, Vienna (1984).

[14] Vychislitel'naya mashina 1010B: Spravochnik (The 1010B computer: A handbook), Videoton, Budapest (1973).

[15] ZHURAVLEVA, G.M., CHUKREEV, F.E., Updating and using the international non-neutron experimental nuclear data base in "Generalized EXFOR" format, this issue.

[16] ROTSHTEIJN, A.P., Some models for estimating the reliability of the operator in the "operator/VDU" system [in Russian], Upravlyayushchie sistemy i mashiny 6 (1978) 19-25.

Manuscript [Russian version] submitted to Editors: 30 April 1984

Table 1

Structure of the list of information identifiers (file V2)

| Information identifier | Availability flag | Presence flag | Information identifier | Availability flag | Presence flag |
|---|---|---|---|---|---|
| ADD-RES | 02 | 20 | HISTORY | 02 | 20 |
| ANALYSIS | 00 | 20 | INC-SPECT | 00 | 20 |
| ASSUMED | 00 | 20 | INSTITUTE | 0I | 20 |
| AUTHOR | 0I | 20 | METHOD | 00 | 20 |
| COMMENT | 02 | 20 | MISC-COL | 02 | 20 |
| CORRECTION | 02 | 20 | MONITOR | 00 | 20 |
| COVARIANCE | 02 | 20 | MONIT-REF | 00 | 20 |
| CRITIQUE | 02 | 20 | N-SOURCE | 00 | 20 |
| DECAY-DATA | 02 | 20 | PART-DET | 00 | 20 |
| DECAY-MON | 00 | 20 | RAD-DET | 00 | 20 |
| DETECTOR | 00 | 20 | REACTION | 03 | 20 |
| EN-SEC | 02 | 20 | REFERENCE | 0I | 20 |
| ERR-ANALYS | 02 | 20 | REL-REF | 02 | 20 |
| EXP-YEAR | 0I | 20 | SAMPLE | 02 | 20 |
| FACILITY | 0I | 20 | STATUS | 02 | 20 |
| FLAG | 02 | 20 | TITLE | 0I | 20 |
| HALF-LIFE | 02 | 20 | ENDBIB | 02 | 20 |

Table 2

Calculated operator speeds for performing operations with different VDUs

| Operation and characteristics | VDU type | |
|---|---|---|
| | with MARKER LEFT key | without MARKER LEFT key |
| Time taken to place marker in appropriate position ($T_{pm}$), s: | | |
| minimum | 0,55 | 0,55 |
| maximum | 3,I | 4,3 |
| Time taken to key in one character ($T_{ch}$), s: | | |
| with self-checking | 0,62 | I,I |
| without self-checking | 0,5 | 0,5 |

## Table 3

Comparison of data input times for assembling the SID for ENTRY A0118 at international centres

| Centre | $T_{ch}$ | $T_{pm}$ | Number of characters keyed in per line | Number of lines in the SID | Number of fields | Input time |
|---|---|---|---|---|---|---|
| BNL*/ | 0,5 | 0,55 | 80 | 27 | 3 | I423I,25 s = 3,95 h |
| | | | | 27 | 2 | |
| NDS | 0,62 | 0,55 | | 9 | I | I7504,85 s = 4,86 h |
| | | | | 203 | 2 | |
| | | | | 75 | 7 | |
| CJD | 0,62 | 0,55 | 33 | 27 | 2 | I0500,28 s = 2,92 h |
| | | | 22 | 27 | I | |
| | | | II | 9 | - | |
| | | | 50 | 203 | I | |
| | | | 66 | 75 | - | |
| CAJaD | 0,62 | - | 4 | I2 | - | 8058,76 s = 2,24 h |
| | | | 50 | I96 | - | |
| | | | 42 | 75 | - | |

*/ The calculated characteristics for punched card readers are approximated to those for VDUs without a MARKER LEFT key and without self-checking.

## Table 4

Comparison of the accuracy of data input in assembling the SID of ENTRY A0118 at international centres

| Centre | $\beta_{ch}$ | Number of characters keyed in | $\beta_{key}$ | Expected number of errors in text |
|---|---|---|---|---|
| BNL | 0,9950 | 27280 | $0{,}4I\cdot I0^{-59}$ | I36,4 |
| NDS | 0,9994 | 27280 | $0{,}76\cdot I0^{-7}$ | I6,4 |
| CJD | 0,9994 | I6684 | $0{,}44\cdot I0^{-4}$ | I0,0 |
| CAJaD | 0,9994 | I2998 | $0{,}40\cdot I0^{-3}$ | 7,8 |