# KEIO-TOSBAC Time-Sharing System Input/Output Control and Supervisor

NORIHISA DOI*, FUMITAKA SATO** AND HIDEO KOBORI**

*Introduction*

The ultimate object of the computer utilities is to use the computer as an effective means to process the various problems concerning in the various situations of human life, making full use of the advantageous features of the computer such as the accuracy of recollection, the rapidity of the process and so on. To realize the computer utilities, the time-sharing method is the outstanding one. At present it is not yet possible to realize the computer utilities in a strict sense. However, under the certain restrictions——e. g. the limited capacity of the main and the auxiliary memories and the limited processing speed, and hence the limited number of users processed simultaneously——, the computer utilities can be implemented to some limited extent showing, nevertheless, considerable usefulness. Making effective use of the experience from these systems, we shall be able to approach the ultimate object.

Hereupon, we aim as a subgoal to develop such a system that will be able to process what have not been handled or, if handled, have required a considerable amount of time and expences, and ultimately aim to develop the computer utilities. Now, at KEIO University, the small-sized, general purpose time-sharing system, called the KEIO-TOSBAC Time-Sharing System, is being implemented on the TOSBAC 3400 model 30 as the joint project with Tokyo Shibaura Electric Co., Ltd.

Up to date, on hardware, the terminal device and the Data Transmission Controller (DTC), and on software, conversational FORTRAN, LISP, SNOBOL, KFMS (Keio Formula Manipulation System), Autoplot (Automatic plotting system) and the Desk Calculator system have been developed. All of these software systems are pure-procedures and being operated in conversational mode. The initial system (called Version 0) is operating experimentally since June, 1967. Though this initial system is a pilot model with only two terminals, the supervisor has been programmed as a general purpose and open-ended system and so any supplements and/or modifications of the functions of the supervisor are easy matter.

In this paper, the DTC and the supervisor of the Version 0 are described.

## 1. *The Outline of the System*

Fig. 1 shows the configuration of the KEIO-TOSBAC Time-Sharing System——Version 0.

This paper first appeared in Japanese in Joho Shori (the Journal of the Information Processing Society of Japan) Vol. 9, No. 2 (1968), pp. 80-87.

* Department of Administration Engineering, Faculty of Engineering, KEIO University
** Tokyo Shibaura Electric Co., Ltd.

The functions and the usage of the devices are as follows.

* Magnetic Core Storage
  Capacity : 16 K words with 24 bits/word.
  Cycle time : 800 ns.
  Supervisor, IOCS and the processors of some basic commands are wired down.
* Magnetic Drum
  Capacity : 16 K words.
  Transmission rate : 150 KC.
  Used to swap the user's programs and data.
* Magnetic Tape
  Transmission rate : 28. 8 KC.
  Used for the file of user's information, and holding the system programs.
* Data Transmission Controller
  Control the input/output between each terminal and/or console typewriter and the computer. The controller handles up to eight terminals.
* Terminal
  Each terminal consists of an electric typewriter and some functional keys and lights.

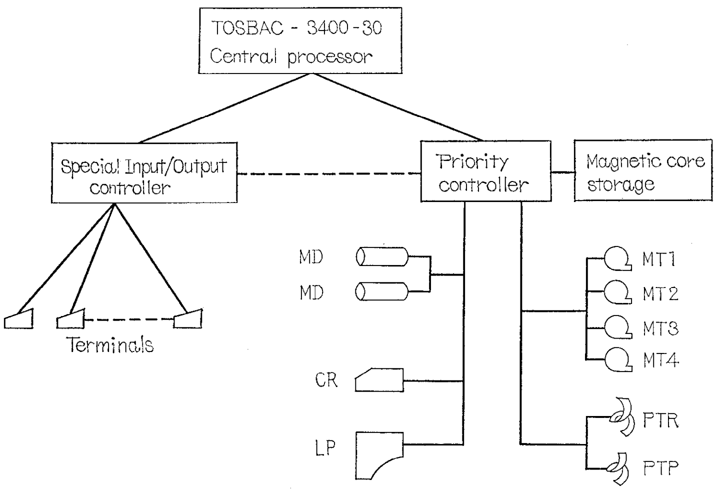



Fig. 1. The configuration of the KEIO-TOSBAC Time-Sharing System (Version 0).

The data transmitted between terminals and the system are handled on a character-by-character basis, and each transmission causes an interrupt. This interrupt signal is generated by DTC and transmitted to the Priority Controller, which, in turn, sends this signal to CPU. The output data are checked by the controller of each terminal and the input data by DTC. All of the controllers of the terminals are controlled collectively by DTC (see 2. 4). The Priority Controller controles all of the interruptions to CPU collectively and concurrently.

## 2. *Additional functions of the TOSBAC 3400*

To implement the time-sharing system, we have added the essential functions to TOSBAC 3400.

### 2.1. *Modes*

There are two modes; master and slave. In master mode, every operations can be executed, and a protection hardware does not perform its function.

* The conditions to enter master mode:
  1. On occurrence of every interrupt and trap, and
  2. when the CONT. CLEAR key on the console is pushed.
* The condition to enter slave mode:
  On occurrence of the transfer to the location in the unprotected area from the protected area.

The mechanism of the interrupt handling is as follows. In advance, the code JTL (Jump and Transfer Link; when this code is executed the content of the instruction location counter is stored in the location, m, specified in the address part, and the control is transfered to the location $m+1$) is stored in the specifically fixed location, the interruption location (IL; depending on the sort of interruption or trap, the location is specified whereto the control is transfered on occurrence of the interrupt or trap). Then, when a certain interrupt or trap occurs, the control is transfered to the entry of the interruption handler, the location of which is specified in the address part of the JTL. At this time, if the location is within the protected area, the interrupt is handled in the master mode, and if the location is in the out of the protected area the mode is changed and handled in the slave mode. To return to the interrupted program at the end of the interruption handler, it may only be well to jump to the location held in the top of the handler.

If the code other than JTL is in IL when the associated interrupt occurs, the mode turns into the master mode and after executing the operation specified by the code in IL the control is returned to the interrupted program being remained in the master mode regardless of the previous mode. But in the case when the operation code is one relating to the transfer, the state of the mode depends on the location specified in the code.

When the interrupt is inhibited by means of the mask, it is needless to say that the interrupt is never initiated nor the mode enters into the master mode, even if the causes of the interrupt are given rise to.

### 2.2. *The privileged operation*

The following operations are executed only in the master mode.

* Input/Output operations
* Operations which change the trapping modes
* Halt operations
* Memory protection operations
* Operations which change the indicator

When these privileged operations are appeared in the slave mode, the operation is regarded as NOP (No Operation) as a rule, and the trap occurs. Some of the privileged operations inhibit the trap or interrupt immediately after the execution of the opration. In the slave mode, however, when one of the operations appears the trap will be raised immediatelty.

2. 3. *Memory Protection*

In the slave mode, each time the core is referred to, the address to be referred to is compared with the contents of the boundary registers. If the location is in the area to be protected, the execution of the operation is terminated and the trap is raised. The area to be protected is specified by the two boundary registers, U register and L register. U register is used to define the upper bound and L register the lower bound of the area not to be protected. The protection can be specified in the units of 256 words.

2. 4. *The Data Transmission Controller*

A terminal is composed of an electric typewriter (in case of the full-scale terminal, called TOSBAC DN-510, it is replaced by the IBM Selectric typewriter) and some functional keys and lights. Each terminal is attached to the system through the Data Transmission Controller (DTC) as a kind of a console input/output equipment. Up to eight terminals can be connected to DTC (the full-scale DTC, called TOSBAC DN-230, is able to manage maximum of 32 terminals concurrently).

The data transmitted between the terminals and the system are handles on a character-by-character basis and are processed by RCS (Read Console) or WCS (Write Console) operation. To permit the concurrent operation of all terminals, the interrupt is raised by each data. DTC finds automatically to which terminal the data processed by RCS or WCS belongs.

When the interrupt occurs, the supervisor's data handling module invokes SII (Set Indicator from Interruption). As the result of SII, the terminal number and one of the interruption signals are set in the indicator register. The interruption signals are as follows.

* ERC ... end of receiving of a character
* ETC ... end of transmission of a character
* CS ... channel signal

On the interruption of ERC the data is read into the accumulator through RCS, and on the ETC one data character is sent through WCS. When the interrupt occurs, if ETM (Enter Trapping Mode) is executed without execution of RCS or WCS, the same interrupt occurs again. Furthermore, if RCS or WCS is executed more than once, the operation on and after the second works on the ordinary console equipments.

For the interruption of CS, the Input Status Codes (ISC) which are the information about the input message up to that time are read through RCS. Some of the major ISC are as follows.

* The operation executed was inconsistent with the state of DTC.
* Some of the data overlapped one another.
* Some of the data were not perfect.
* Carrier cut down.
* The transmission from the terminal was completed.

Parity check for each input data is carried out by DTC and the result is reflected into the parity bit of the data. To accomplish the data transmission, two special operations, SEIP (Select Input) and SEOP (Select Output), are prepared. The functions of these operations toward DTC are as follows.

* SEOP ... Towards the terminal specified in this code, STX (Start of Text) code is transmitted, which makes the state of DTC and the terminal possible to handle the message from the computer.
* SEIP ... After the completion of the data transmission towards the terminal, ETX (End of Text) code is sent to the terminal through this code.

As the result of operation of SEIP, the Output Status Codes (OSC), which are the information about the output data up to that time, are transmitted from the terminal and then the ERC interruption is raised. Some of the major OSC are as follows.

* Some of the data were not perfect.
* Some of the data overlapped one another.
* The terminal is offlined.
* The attention key or the quit key has been hit.
* Though a given time has elapsed after the completion of output, no action was carried out at the terminal.

3. *Supervisor*

The supervisor is the backbone of the time-sharing system and the character of the system greatly depends upon the philosophy of the supervisor.

On developing the supervisor of our system, the following matters were presupposed.

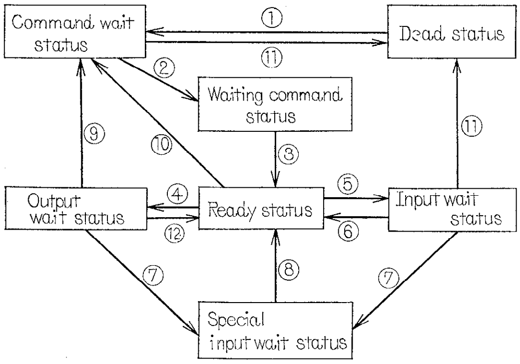
* Receive the messages from the user under any circumstances.
* Treat every user without discrimination.
* Exchange the message between the user and the system in full conversational mode.
* Easy to expand the functions of the system.

At a user's point of view, the levels at which the user works may be devided broadly into following two categories.

* Programming level
* Command level

Within the system, there are seven statuses at which the users are situated. These statuses are as follows and Fig. 2 shows the transition of the users during these statuses.

* Dead status
* Command Wait status
* Waiting Command status
* Ready status
* Input Wait status
* Output Wait status
* Special Input Wait status (This status is raised when the operation executed is inconsistent




① The completion of the HELLO sequence.
② The completion of the receipt of some command.
③ The arrival of the turn for using CPU.
④ Demmand output of supervisor.
⑤ Demmand input of supervisor.
⑥ The completion of the receipt of a message.
⑦ Upon output to a terminal, the terminal is doing input.
⑧ The completion of the receipt of a message.
⑨ The completion of the transmission to the terminal.
⑩ The receipt of the quit signal.
⑪ For hours, the terminal has made no response.
⑫ The completion of the output.

Fig. 2. The transition of the user's status within the system.

with the state of DTC.)

Hereafter, we devide the supervisor into some groups and summarize their functions.

3.1. *Terminal Interruption Handler*

Terminal interruption handler takes charge of the interchanges between the terminals and the system through the interruption. As described before there are three kinds of interruptions belonging to DTC. The actions taken by the handler for these interrupts are as follows.

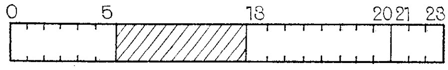
ERC .... Reads an input character. When it is immediately after the execution of SEIP, reads OSC.

ETC .... Converts the code of the data and transmits it. Executes SEIP, when it is immediately after the completion of a message.

CS ...... Reads ISC.

*Input data handling.* The input data (ISC, OSC or normal data character) together with its associated terminal number is packed in a word (see Fig. 3), and stored into the pool-buffer (regarding to the size of the pool-buffer, see 3.2). The pool-buffer is used cyclically.

*Output data handling.* First, the output handling module transfers the output data into the output buffe for each user from user's area, and executes SEOP. From the subsequent ETC interruption, the terminal interruption handler deals with the data on a character-by-




| | |
|---|---|
| 0～5 | Sort code |
| $00_8$ | Data (upper case) |
| $40_8$ | Data (lower case) |
| $20_8$ | OSC |
| $30_8$ | ISC |
| $34_8$ | ISC after OSC |
| 13～20 | Data, OSC or ISC |
| 21～23 | Terminal Number |

Fig. 3. The format of the input information within the pool-buffer.

character basis. As the codes of the terminal equipment are different from those of the computer, upon transmission, the codes are converted.

3. 2. *Clock Trap Handler*

The interval timer is in the location five. Every 20 ms. the content of the timer is decreased by one, and when the timer shifts to $77777777_8$ from zero the trap occurs. This clock is used for accounting and for keeping the time of day clock as well as for time-sharing purpose. At present, the clock trap occurs at intervals of 200 ms. (the size of the pool-buffer depends upon this time interval). When the clock traps, the time of day clock is updated and the accounting is made at first (see Fig. 4 (a)). Then the informations within the pool-buffer are examined (from the point at which the last clock trap happend to the current one). In case of the data character, it is converted and transfered into the input buffer for each user. In case of OSC the associated action is taken after analyzing the corresponding ISC. When both OSC and the corresponding ISC are normal, the output can be considered to have completed successfully and the user turns to an appropriate status. When ISC for the input message is normal, the input can be considered to have completed. For these users, successively, on the basis of the status of the user, it is examined that whether the input is the system command or the subsystem command, or the data for user's programs. When the message is the system command, the user moves to waiting command status, otherwise moves to ready status.

3. 3. *Scheduler*

The scheduler determines to which user's turn should be come round next and takes associated action (Fig. 4 (b)). In our system, no user and/or termenal are given a priority, that is, every user and/or terminal are accessed without discrimination.

Scheduler has three entries: (1) the first, named Entry 1, from the clock trap handler, (2) the second, named Entry 2, from the supervisory routines, and (3) the third, named Entry 3, from some specific command handler (e. g. PROGRAM command; this command will be used to put user in program level).

*Entry* 1. It is examined that whether the clock trapped during the execution of one of the supervisory routines. If so, the control is returned to the routine immediately. Otherwise,

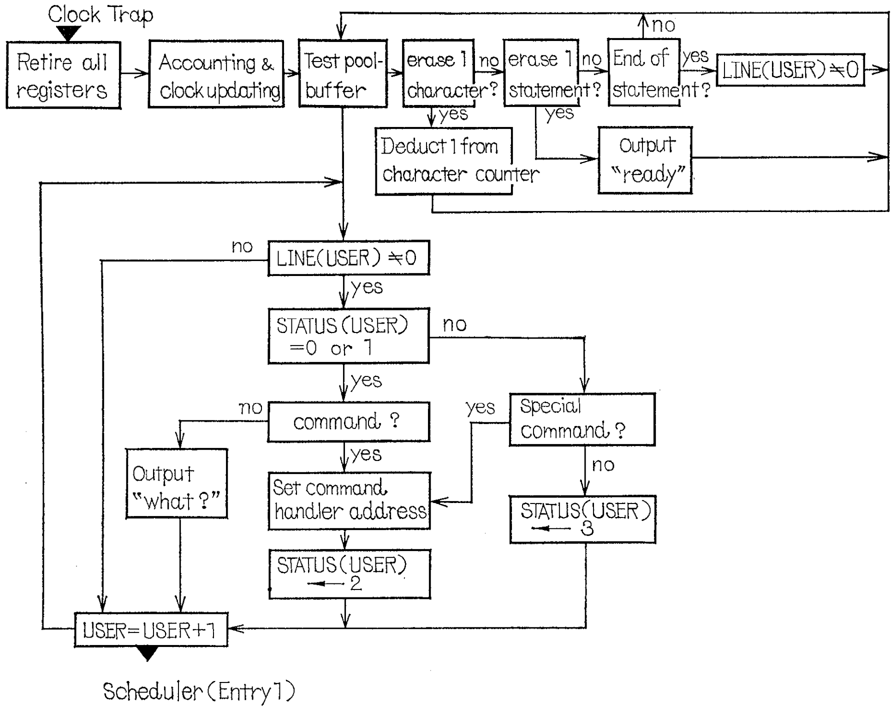



Fig. 4(a). Clock trap handler.

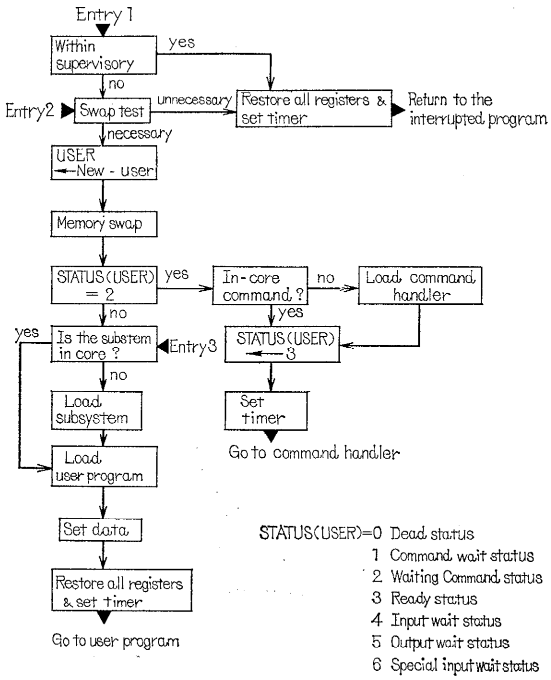



Fig. 4(b). Scheduler.

the control moves to Entry 2.

*Entry* 2. It is examined that whether the time allotted to the current user has elapsed. If not so, the control is retuned to the current user's program. Otherwise or when the current user has moved to the status other than ready, the remaining user's statuses are checked in turn and if a user who is in either ready or waiting command is found, the user is determined to be accessed. If no user is in either ready or waiting command, as for Version 0, the system enters into the waiting loop.

On the occasion of swapping, only user's program becomes the object as all of the subsystems are composed of pure-procedures.

When a user is in waiting command status, the user becomes ready and the control is transfered to the appropriate command handler. Otherwise, the control moves to Entry 3.

*Entry* 3. When the subsystem at present being in core is different from the one to be applied to the next user, the appropriate subsystem is loaded. And then, when the next user's program is already in a drum (even if any program maintained in the file were specified to use, the program would be removed into the drum by this time), the program is swapped in and the data, if any, is transfered into the user's area from this user's input buffer. And the control is transfered to the appropriate point of user's program.

However, when the user is going to make a new program from now on, after setting the interval timer the control is transffered to the subsystem.

3. 4. *Supervisor Call Handler*

When the user's program demands to output the results, to call for the data input or to terminate the execution, it must get into communication with the supervisor through the interruption that is raised with the aid of HJ (Halt and Jump) which is the one of the privileged operations. The Supervisor Call Handler decodes the calling sequences and accedes to the user's demand.

3. 5. *File Management*

As for Version 0, the user's files are maintained on the magnetic tapes. Consequently, the simple mechanism of the file management is used temporarily. Using three directories, Master File Directory (MFD), User File Directory (UFD) and Own File Directory (OFD),

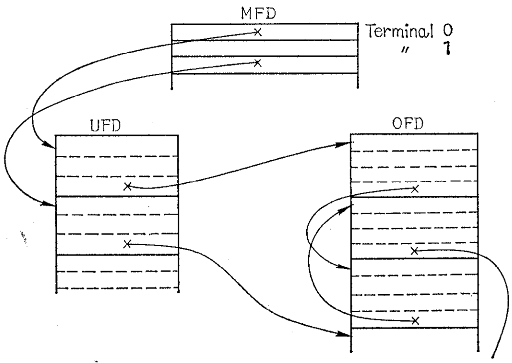



Fig. 5. The interrelation of the directories.

the supervisor manages the files. The interrelation of these directories is shown in Fig. 5.

As for Version 0, only user's programs (regardless of completed or incompleted) are preserved as a file. The file cannot be used until the terminal number, the user number, the program name and the subsystem name are entirely matched.

At present, the file system based on the disc is being developed. On the completion of the system not only data could be saved as a file but also the programs developed by the user could be held as a common file.

## 4. *Conclusion*

The outlines of the systems which have been developed are as follows.

* Conversational FORTRAN system (alias KEIO (Keio Elementary Instructive Operating) system)

This system consists of Monitor, Scanner, Linker, Interpreter and Command Processor, and all of these are pure-procedures. The object program in reverse-polish tree-structure generated by Scanner and Linker is saved as a file. And it is processed by Interpreter at run time. On the demand of listing the source program, LIST Command processor reproduces the FORTRAN statment from the object program. At programming time, Scanner scans and does the syntax check on a statment-by-statement basis.

* Desk calculator system

Given the FORTRAN-like arithmetic statement, the result is obtained immediately by return. The results can be saved up to 100 and these can be referred to by name in the later calculation. Some elementary functions and various commands are prepared and the system is composed of pure-procedures.

* Conversational LISP system
* Conversational SNOBOL system
* KFMS (Keio Formula Manipulation System)

Conversational version of the FORMAC.

Moreover, Autoplot system (JOSS-like Language and has the spacial functions for the curve-plotter) and the file system (including the function for background) are now being implemented. In parallel with these works, we are studing the characteristics of the TSS and its relations to the various environments through simulation.

In next summer, the full-scale devices——TOSBAC DN-230 and TOSBAC DN-510—— shall be implemented and so the system will become more substantial.

*Ackowledgment*

The authors ackowledge Dr. Kohei Amoo and Mr. Sigenori Matsusita (Tokyo Shibaura Electric Co., Ltd.) for development of hardware for the time-sharing system, and Ken'ichi Harada, Masakazu Nakanishi, Mizuo Ohno, Hiroyuki Ishiwata, Yoshio Ohno and Masao Matsumoto (Keio University) for software development, suggestions and discussions. They are grateful to Prof. Shoji Ura and Prof. Jiro Yamauchi for their continued encouragements and edifying suggestions.