Time-Sharing System Note 2 Dec. 12, 1963

TO: Time-Sharing Users

FROM: Jessica Hellwig

A Warning about the Current Implementation of the Command RESTOR, and Some Incidental Information about System Commands

When the command restor (or resume, which is a chain of restor and start) is issued, a number of user conditions or options are not restored to normal status. In particular, any of the user's active disk files remain active, including those being used by command programs as well as by the user's own program; the input mode (automatic generation of line numbers) remains in effect; and the status of the switch which specifies that the user is in the acknowledge mode is preserved along with certain other conditions. It seems clear that this should not be the case, since the setting of any pertinent switches should be preserved along with the saved file being restored, and restor should reset all options to normal before restoring the saved status.

In general this (temporarily) incorrect implementation will not cause much grief, because most of the conditions in question are normally reset at the conclusion of the program which set them. Thus the file command causes the input mode to be reset, and commands and user programs which reference the disk (with or without the TSS file switch on) normally close out any active files. On other than normal exits, however, these options may not be reset; the most common sources of trouble are a user program which reaches an unexpected exit to the supervisor, leaving disk files open, and any program which is terminated by the quit sequence.

For example, if the user gives a quit while in the input command and then issues a restor of some saved file, the input mode is not reset; if he now puts the restored program into execution, he is very likely to find that the supervisor continues to generate line numbers. Again, suppose he has a program which reads a data file, and an unexpected stop has caused the program to leave the file open. If he restarts the program by a resume, the data file will not be read from the beginning, even if the first thing the resumed program does is to seek the file, since the active status may take precedence over the fresh start.

The moral is that until some forthcoming changes are made in the system, the user must take precautions to safeguard against confusion. Certain commands which are of a simple and self-contained sort can be guaranteed to reset all the options in question. If one of these is issued just prior to the restor, no problems should ensue. It remains to choose the easiest and shortest command to use: listf works, but may be time-consuming; in lieu of some command which performs a task the user actually wants at the moment, the best candidate is delete with no arguments, which is very quick and doesn't even produce an error comment (?!).

Additional Remarks

By way of illuminating some of the problems implicit in the remarks above, a few details about types of commands may be of interest. At present there are three types of system commands, commonly called internal, external, and disk commands. A more useful set of terms might be: A-core transfer, B-core transfer, and disk-loaded commands. The current commands fall into these groups as follows:

1. A-core transfer commands:

   logout, login, start, octlk, octpat, save, restor, resume;

2. B-core transfer commands:

   use, pm, stopat, tra, patch, fapdbg;

3. disk-loaded commands:

   input, edit, load, fap, listf, printf, file, mad, madtrn, chmode, delete, rename, combin, split, memo, modify, ditto.

When a command is interpreted by the supervisor, a command directory is searched for the command name; a code word associated with each entry in the directory describes into which of the three categories the command falls. On the first type, the supervisor effects a transfer to a subroutine within itself. On the second, the supervisor effects a transfer to a specified location in B-core (the user's program). On the third type, the command program is loaded from the disk into B-core, overwriting the user's current memory area.

Before starting up a disk-loaded command, the supervisor resets the following user conditions:

switch to save machine register conditions;

disk status (active files);

input mode;

acknowledge mode;

break level.

In other words, the disk-loaded commands make a fresh start. Consideration of the nature of the B-core transfer commands reveals that they depend on preservation of all user conditions; the same is clearly true, among the A-core transfer type, of start, octlk, octpat, and save. Login and logout are special, and provide their own initialization and terminating precedures. The generalization cannot be made that all in-core commands expect user status to be preserved; such commands must be programmed with the understanding that automatic resetting does not take place as it does with the disk-loaded commands, and each in-core command must reset where necessary. Clearly this should apply to restor and resume.

What will be done about it

A method has been devised for the preservation of disk status along with a saved file. This will be put into effect in the not-too-distant future, and at the same time other pertinent information will also be preserved, including the options mentioned above. At the same time that this procedure is implemented, restor will be revised so that it will reset old conditions before restoring those preserved in the saved file. At this time the warnings mentioned here will be obsolete.

CAS