MULTICS STAFF BULLETIN- 85

To: MSB Distribution

From: Robert S. Coren

Date: January 2, 1973

Subject: I/O Daemon Changes

A new I/O Daemon is being installed soon. This document details the operational changes involved.

I. Changes visible to users

None except that multi-copy requests should be slightly cheaper owing to more efficient code.

II. Changes visible to operators

A. Console Printout

1. The following messages have been eliminated:

"Printer/Punch attached" (after *init* request)
Date and time of request
Resources used by request
"io_daemon goes blocked"

2. The format of the message describing the current request is now as follows:

"Request nnn.*q*:$\left\{\begin{matrix}\text{Printing}\\ \text{Punching}\end{matrix}\right\}$ *pathname* for *user-pgid*"

where *nnn* is a sequentially-assigned identifier, and *q* is the number of the queue in which the request was placed e.g.

"Request 352.3:Printing >udd >Multics >cp >glop.list for Gintell. Multics.m"

3. If an error makes printing/punching of the request impossible (e.g. improper access, segment not found, etc.) the error message will include the line: "Request not processed."

4. Error messages will be written on "error_output"; informational messages will be written on a stream called "daemon_output". In the present implementation, "daemon_output" is made synonymous with "user_output" at init time, but in future, when the daemon no longer has its own console, "daemon_output" may be sent to a file.

B. New requests

The Waltham implementation of the "kill" and "repr" requests has been incorporated. In order to abort or restart a printing or punching operation, the operator hits QUIT, and, when daemon_exec_ is entered, types "kill" or "repr" (or "reprint" if he likes) respectively, thereby signalling the appropriate condition. The daemon does not acknowledge a successful "kill" or "repr" on "daemon_output"; however, it will print a message on "error_output" if there was no active request.

C. A daemon can now be told to attach a punch but not a printer. An init request with a first argument of "null" will not attach a printer (but some null 2nd argument must be provided, since it will use args 3 and 4 for the punch as heretofore). For example:

```
init null null pun21 punb42
```

will result in the attachment of punch DIM pun21 and punch channel punb42, but will not attach a printer.

III. Internal Changes

An attempt has been made to rationalize the logic of the io_daemon module. In particular, the following modifications have been made:

A. Error messages are printed on both "error_output" and "print_stream" by a central internal procedure called "error_write", which uses system status codes wherever possible. It also attempts to make sure that the print header has been printed; if there is not enough information to put out a header, the printing on "print_stream" is suppressed.

B. The header is printed by an external procedure called head_sheet_. This makes it possible to change the header format without recompiling io_daemon.

C. When multiple copies are requested, unnecessary repetition is avoided. In particular, the segment being printed is not terminated and then initiated anew for each copy. This also means that in the case of "fatal" errors (such as missing segments) the daemon will only go around once.

D. Multi-segment files are now processed using msf_manager_.

E. Deletions are now done by delete_, freeing the io_daemon from worry about how directory access is implemented on a given file at a given moment.

IV. What the future holds

A. By means of the new message-routing features, the daemon will eventually cease to print anything but error messages (if that) on a console. It will also have a default source for its input arguments (so the operator will not have to type them in), including the information as to whether it is the first, second, ...etc. daemon to be initialized.

B. The daemon queues will be converted from links to message segments. This will allow us to leave space for the easy addition of new features.

C. The issue of multi-tasking the daemon (in some sense) will have to be studied in detail, and some particular implementation decided on.

D. A new MOSN on I/O Daemon Operation, superseding MOSN 227, is forthcoming.