MULTICS STAFF BULLETIN - 106

To: Distribution

From: Robert S. Coran

Date: July 3, 1973

Subject: Operation of Remote Peripheral Devices

The capability to run remote peripheral devices is being added to Multics. In particular, a Mohawk Data 2400 is expected at 575 Tech Square sometime in the fall of 1973, including a printer and a card reader. The Mohawk 2400 can emulate either a G115 or an IBM 2780; at the moment we expect to run it as a 115. A document giving the functional specifications of the 2400 is forthcoming.

The present document is intended to describe, from an operational point of view, the proposed method of running this remote device. The discussion assumes the possibility of running such a device with both a printer and a card punch, although we are not getting a punch at present. (The 2400 is also capable of running a tape drive, but there no plans at present to implement this feature.)

The remote printer will be controlled by the I/O daemon, which is being redesigned to operate as a group of processes (one per device) coordinated by a single master process. The details of the new I/O daemon design will be discussed in a future document. Communication between the operator of the device and the system software will be primarily by means of control cards read in through the device's card reader. The software will check for the presence of control cards in the reader at certain well-defined intervals, as described below.

Bringing up the remote device

The high-speed line connecting the remote device to the system will be attached by the initializer when the system comes up, like a tty line. When the remote device dials up the system, the Answering Service reads a card from the reader, which is expected to be of the following form:

```
$*$DLOGIN device_id password
```

(Note: all control cards are expected by the GRTS115 software to begin with the characters "$*$".)

where DLOGIN ("device login") is a special command to create a process for running a device through the I/O daemon, and "device_id" is a character string identifying the particular device. (1) The Answering Service, having validated the device_id

---

(1) The "dlogin" command will also be used to bring up other

and password, creates a process to run the device, and notifies the coordinating process of its existence. The process thus created determines how to proceed according to a "function request card" read from the reader.

Function Request Cards

The operator of the remote device will feed a card into the reader to request either printed output, punched output, or the acceptance of card input. The function request cards are:

$*$START_PRINT

The I/O daemon will start processing requests from the appropriate print queue, directing output to the remote printer.

$*$START_PUNCH

The I/O daemon will process requests from the appropriate punch queue, sending output to the remote punch. These two requests function similarly to the "start" request given to the present I/O daemon.

$*$START_PRINT_PUNCH

The I/O daemon will check for the existence of print requests, and start working on them if there are any. When the print queues are empty it will start on punch requests; if the print queues are empty to begin with it will check the punch queues right away.

$*$START_PUNCH_PRINT

is exactly like $*$START_PRINT_PUNCH except that the punch queues are checked first.

$*$READ_CARDS

The daemon will accept an appropriately-formatted input deck from the card reader (as with a "read_cards" request to the present I/O daemon).

After completion of each output request (print or punch), the daemon will check to see if there is a further function request card. If there is none, it will proceed to the next request in the queue on which it was working.

---

devices; it may be typed in from the initializer console or whatever console is attached to the master I/O daemon process, or read in from an exec_com.

If after reading a $*$START_PUNCH card (for example) the daemon finds that there are no punch requests pending, a message to this effect will appear on the remote printer(if it exists and is working); the daemon will then wait until either a punch request comes in from dpunch or a new function request card is read in. If there is no punch attached, an error message will be printed.

Special Requests

If for some reason (e.g. as a result of device malfunction) it is necessary to interrupt the activity of the remote printer or punch in the middle of performing a request, the operator should hit the ABORT button on the device's control panel; this will act essentially as a QUIT to the daemon process running the device. Printing or punching will be suspended, and the daemon will expect a "special request" control card to be fed into the reader. This request can be any of the following:

$*$KILL

requests the daemon to abort the current request. The request will be kept available for restarting for a definite period (tentatively fixed at 3(minutes). (2) The daemon will then check for a function request card, and, if there is none, proceed to the next request.

$*$CANCEL

requests the daemon to abort the current request and not keep it available for restarting.

$*$RESTART -n-

where n is an integer. This card instructs the daemon to repeat all requests that it has performed for that device, starting with the request whose sequential identifier is n. If n is omitted, only the current request is restarted.

Other requests

$*$STOP

If this card is read when a function request card is expected, activity on the device will cease until another function request card is read.

---

(2) The new I/O daemon design includes a provision for saving all requests for a fixed period after printing or punching, in case it is necessary to restart them as described below.

$*$DLOGOUT

This request is used to terminate the process that is running the device.

Messages

Error messages relating to the functioning of the remote printer/punch, or, in some cases, the process driving it, will, in general, appear on the printer (if possible). In addition, these messages, as well as messages describing the normal functioning of the daemon (e.g. "Request 26.3: Printing >udd>m>Smith>foo", etc.) will be routed through the message coordinator. The operator may decide to send them to a file, or to a dialed-up terminal, or both; alternatively, normal-functioning messages may be sent to a file while error messages are typed out. In addition, certain serious error conditions (such as inconsistent data bases) may be recorded in the system log.