

## PLANS FOR CONSOLELESS OPERATION OF THE NETWORK DAEMON

by Edwin Meyer

The network daemon is managed by a ring 4 program whose purpose it is to respond to network events, to maintain an online log of the network's operation, and to notify the operators of network failure. In addition it provides a mechanism for initialization and manual termination of network operation.

The network daemon on the 645 currently operates on a console in the machine room which is manually logged in by the operators each time Multics is started. The daemon manager program is a command which is reached automatically upon login via a `start_up exec_com`. It initializes the Network Control Program (NCP) and specifies the host numbers of the existing foreign network sites, then settles into normal operation. The program blocks on three event channels, the typewriter channel, the network daemon event channel, and a timer channel. When awakened on the daemon event channel or the timer channel, the network daemon calls the ring 0 NCP to perform service, then blocks again. The timer channel exists to insure that the daemon calls the NCP periodically, as the prolonged absence of it from the NCP is taken as an indication that the daemon has somehow failed. When awakened on the typewriter event channel the daemon calls `cu_$cp` to process a command line. Both the typewriter and the daemon channels are fast ipc event-wait channels, while the timer channel is a regular ipc event-wait channel.

For a number of reasons we do not intend to use this program for consoleless daemon operation on the 6180, but rather to write a new program patterned on the IO daemon's manager, `daemon_exec_`. The chief reasons are that considerable effort would be necessary to convert the present program and that the present interface is less than satisfactory. Using the model provided by `daemon_exec_` we hope to lift most of the code from the current daemon manager and be able to write the new manager `network_exec_` with only modest added effort.

---

This note is an informal working paper of the Project MAC Computer Systems Research Division. It should not be reproduced without the author's permission, and it should not be referenced in other publications.

It would be difficult to convert the network daemon manager procedure `net_driver` to consoleless operation for the following reason. `Net_driver` uses fast ipc event-wait channels for the sake of operating efficiency. The use of event-wait channels necessitates that each call to block specify all three event channels, so that an event on any of them will cause a wakeup. Were a channel to be omitted from the list given to block, that channel would be ignored and an event over it would not cause the process to be awakened. It is for this reason that `net_driver` calls the supervisor level typewriter read entry directly, bypassing the user ring IO system. The IO system read call blocks internally, and to use it would make the network daemon "deaf" to the other channels. However, a consoleless daemon must use the user level IO system because it is at this level that typewriter IO is switched to the proper IO device.

Fortunately, the existing model for the IO daemon manager `daemon_exec_` provides a way out of these difficulties, allowing use of both a fast ipc event-wait channel for the network channel and the user level IO system for typewriter IO. In addition, this model avoids other problems inherent in bypassing the user level IO system, and its use will bring the network daemon's operation more in line with that of other system daemons.

The new daemon manager procedure `network_exec_` is intended to be an initial procedure, i.e., it will be the first procedure to get control of the newly established process. Upon doing so it performs the following functions in sequence:

- 1) Perform normal initialization for the new process, such as attaching i/o streams and specifying quit and default error handlers.
- 2) Call into the supervisor to initialize the NCP.
- 3) Specify to the NCP which network host identifiers represent existing foreign sites.
- 4) Begin normal operation.

The daemon normally is blocked on the network and timer event channels, waking up to perform network service whenever an event is received from either channel. It normally does not "listen" to the command input stream at all. To get the network daemon to listen to operator requests a "quit" signal must be sent to the process. When the daemon receives the quit it

enters a mode called "NETWORK EXEC" in which network service is obeyed and the operator may issue requests to the daemon via the command input stream. Because the daemon can not service the NCP while in NETWORK EXEC, a safety feature is provided which will automatically return the daemon to normal operation if the operator fails to issue a manual return request. If the NCP is in operation, network\_exec\_ will automatically return to normal operation 1 minute following the latest command stream i/o.

The following requests are provided in NETWORK EXEC:

- admin -same as the Answering Service admin request, except no password is required.
- down -shuts down network operation.
- dump -takes a dump of the network data bases for later examination by system programmers.
- hold -disables the automatic return-from-NETWORK EXEC feature. The feature is reactivated by manually returning from NETWORK EXEC.
- host\_off -disables communications with the network host(s) specified as arguments to this request.
- host\_on -enables communications with the network host(s) specified as arguments to this request.
- logout -shuts down network operation and logs out.
- re\_init -reinitializes the network and returns from NETWORK EXEC to begin normal daemon service.
- start -returns from NETWORK EXEC to resume processing network requests.
- survey -prints a brief online survey of the current network activity.
- up -starts up network operation (but remains in NETWORK EXEC).