If All Else Fails

The following people from Multics should be contacted if problems arise which you can't deal with: Ken Pogran, Doug Wells, Mike Padlipsky. At least one of them should be available in Region I during all periods when the room is open. If you can't find any of them, call Ed Meyer (617-253-6006) or Raj Kanodia (617-253-6007).

TIP Settings and Escapes

TIP commands of particular significance to Multics are:

a. @E A
which sets the echo mode to local (pre-login)

b. @T O L
which causes line-at-a-time transmission (pre-login)

c. @I L
which insures that a line-feed will be sent even if carriage return is struck (pre-login)

d. @@
which causes the TIP to transmit a @ (the Multics "kill the whole line to this point" character)

e. @S S
which causes the TIP to send the Telnet Protocol "synch", in order to get Multics to "quit" (see Output Control, below)

Apparently Bad TIP Statuses

Two statuses reported by the TIP frequently require special interpretation:

a. DEAD
The TIP will say "DEAD" on incomplete transmissions as well as when the foreign host is actually dead. If this happens while someone is logged in, Multics will not know about the incomplete transmission case and will still show the person logged in. If you try to login again, the TIP reuses the old link and makes the Multics NCP unhappy (you'll see T OPEN R OPEN T CLOSED R CLOSED in response to the @L 6). The TIP console reset command may cure this. If it doesn't, apologize and try another console if the attendee is still interested in Multics. If pressed for an explanation, grudgingly admit that it's the TIP's error, not Multics'. Also, please tell one of the Multics people so that he can do a host reset at a time when no one is logged in from elsewhere.

b. DEAD REFUSED
If this happens right after a @L 6, try again a time or two as it may be caused by our Logger's being busy. If it persists, please tell one of us so we can try to find out why we're down. If it occurs after login, please tell one of us.

Preferred Consoles

Case is significant on Multics, therefore it will be much more convenient for all concerned if you encourage attendees to use full ASCII consoles for logging in to us. (See Typing Conventions, below, if only an upper-case only console is available). Also, as Multics is line-at-atime, it prefers large allocations; therefore, if there is a choice of consoles and if the TIP buffers turn out to be different, choose the larger buffer.

Typing Conventions

a. "Erase and Kill"
Note that the Multics "erase" character (#) erases the previous print position (i.e., a single # erases an overstruck characater); also, #'s may be used consecutively for consecutive print positions (i.e. abcd###z would be interpreted as az). Further, the "kill" character (@) wipes out the entire line up to that point. Be sure, though, that you type it as @@ from the TIP.
b. New Line
Please remember that Multics won't lock at a line until the line-feed character comes through. Therefore, it's imperative that the TIP command @I L be given so as to avoid problems on consoles which don't clearly cause the TIP to send a line-feed at the end of a line (especially those consoles which have a separate carriage-return key). So to be safe, do the @I L.
c. Case
As noted above, full-ASCI, upper and lower case consoles are preferred. If it becomes inescapable to use an upper-case only console, the following should be working by the time of the demo (although not yet in as of this writing): before issuing the login command, give the command "MAP" (quotation marks not necessary); this will cause Multics to map upper case input to lower case except for letters preceded by a \ character (the left-slant or backslash, octal 134) which will be treated as upper case. As command names are in lower case, this mapping must be done so as to get the login command through; however, as names (particularly project names) contain capital letters, the mapping can't be inescapably to lower case. For more information, do help net_case after login.

About Logins

You'll probably be asked to explain Multics' message about load, which appears in response to the @I 6. Briefly, for any particular hardware configuration (usually either 1 cpu, 256k memory or 2 cpu, 384k memory) there is an administratively-set maximum system load. The load is not, however, in terms of users, but in terms of "load units". That is, users who are logged in to limited subsystems are rated at .5 load unit while certain system "daemon" processes which are particularly heavy users can be rated at 1.5. Another likely point of questioning is why the login command is "enter" and doesn't ask for a

password. The explanation is that for convenience at the demo we're taking advantage of a Multics mechanism which permits projects to establish their own "login responders" to avoid the password, and of another mechanism which allows "anonymous users" to share a common working directory which is invoked by the enter command.

Output Control

a. Line Length
Default Multics line-length is 2741-sized. If you're at a narrow console, do line_length n (where n is the desired max length).
b. "Quits"
To stop Multics output, give the TIP command @S S. On some (all?) consoles, you'll have to hit the Attn/Inrpt button first to get the TIP's attention.
c. Tabs
Default Multics Network code assumption is that the actual device being used doesn't support tabs (we tried it the other way and got a lot of static), so if you want to turn tabbing on, give the following command: iocall order user_i/o tabs . To turn tabbing off, it's the same command except you say ^tabs. As tabbing off means that spaces are inserted at the Multics end, this shouldn't be a big problem unless somebody complains about things not lining up.

Command Environment

There are few useful points to be made about the Multics command environment in regard to likely questions and problems:
a. "Ready Messages"
The lines after commands finish which look something like "r 1148 6.437 2+321" are Multics ready messages. You're at command level when you've just gotten one typed out. The numbers are, respectively, time of day, cpu seconds since last ready message, pre-paged pages, and dynamically-paged pages.
b. Editor Modes
If attendees follow the browsing scenario closely, they may have to edit the files they create. It's probably already a familiar problem, but if you're in "Edit" mode in the "eam" editor you can get back to "Input" by giving a line consisting only of a period (.), and from Input to Edit in the same fashion; as all requests (e.g., to write out the file) must come from Edit mode, it's a good idea to do the following if an attendee seems to be "stuck" in the editor: try a linefeed first, as the last request might not have been taken; if that doesn't work, try shifting modes with a period to make sure you weren't in input. Feel free to ask one of us if the editing notes in the scenario are obscure to you.
c. On-line Consulting
We'll try to have somebody logged in at all times to respond to on-line consulting attempts. To contact him, send a line which begins with an apostrophe ('). Note, this only works when you're

at command level.
d. IMPORTANT REMINDER
Please remember Multics is not a character-at-a-time system, so be sure to end all commands with a newline (line-feed or, if you've done a @I L, carriage return).

Remember the Demos

If any of the antendees you're helping expresses an interest in finding out more about Multics than you know, please push the prepared demos which we'll be giving. Thanks.