

Published: 8/29/66

Identification

Inter-process-group Communication  
K. J. Martin, B. A. Tague (See note)

Purpose

A process-group is an intercommunicating family of processes created when a Multics user logs in. A user may have more than one process-group at a time. Frequent need arises for one process to communicate with another which is not part of the same process-group. A prime example is the operator, a user operating in a process-group, who must on occasion automatically logout other process-groups. A recognized wakeup to an Overseer process (created at login for each process-group) is sufficient to set the automatic logout of the process-group in motion. Another example is the process-group which wishes to notify a second process-group that certain operations on a block of data have been completed.

Discussion

The only possible method of communication between process-groups is by the wakeup entry of the Traffic Controller. In all cases, the two process-groups involved must have previously agreed on the meaning of a wakeup from one to the other, and must have agreed on any necessary common data bases for passing information. A number of problems present themselves.

1. A process-group must be able to decide from which other process-groups it is willing to receive wakeups.
2. It must be possible to locate a user's process-group knowing only a user identification.
3. If a user is logged in several times simultaneously it must be possible to distinguish between his various process-groups. Refer to MSPM Section B0.1.04 for a discussion of logging in and the creation of a process-group. A common case of being simultaneously logged in more than once is where a user is logged in normally at a console, and has an absentee background job running at the same time.

A user's identification as defined for file system access control purposes has three parts: project, person, and login identification. Project is an administrative grouping of persons. Person is merely the name of a person.

Login id is a two-character ASCII identification used to distinguish between several simultaneous logins of a user. The login identification is useful in access control for testing purposes; a user can prevent a process-group which is testing new procedures from harming other of his process-groups. The login id is not useful in protecting one user from another, so it will usually be \*, meaning anyone.

A record is kept of the login identification of all of a user's process-groups which are currently logged in. When a login occurs a currently unused identification is concatenated with a person-project id to identify the process-group created. A user is free to change any of his login identifications at any time to an unused (by him) two-character identification. Allowing him to change his login identification makes it possible for him to be sure that some process-group has the desired identification.

The three-part name thus derived is used as the name of the process directory of the Overseer process. (Refer to B0.1.04 on logging in and logging out.) One segment, the intercom data base, of the Overseer process is established at the time of logging in to grant permission to other process-groups to wakeup one process of this process-group. Certain permissions are standard, such as permission to the system operator to issue an automatic logout and to send messages which appear between commands. These permissions cannot be deleted or altered by the user. He may add and delete other permissions at will, using commands.

The intercom data base contains three items of information for each permission:

1. The three-part name of a process-group which may wakeup this process-group. Use of the \*-convention to specify a class of process-groups is allowed.
2. The process id of the process within this process-group which should receive the wakeup.
3. An event identification number which must be noted by the process-group initiating the wakeup.

When a user wishes to wakeup a process of a process-group of another user (or of another logged-in process-group of his own) he calls a procedure, `wakeuser`, giving the three-part name of that process-group. The `wakeuser` procedure knows that the process-group name is the name of the process

DIRECTORY FOR THAT PROCESS-group's Overseer and that the intercom data base is in the process directory of the Overseer process. The wakeuser procedure finds the appropriate event identification number in the intercom data base, then notes that event to the Overseer (a normal call to the Wait Coordinator).

The Overseer awakens expecting any one of three occurrences: a quit-event from a Device Manager process, a command completion event from a Working process, or a wakeuser event from some other process-group. The Device Manager and Working process are part of the same process-group. If the event which has been noted is a wakeuser event, the Overseer calls the wakeuser\_reflector, a procedure to reflect the wakeuser event to the appropriate process of the process-group. This involves simply checking that the proper process-group noted the event, and sending a wakeup to the process which wants to know about the event. Wakeups which should go to the Overseer are simply handled within the wakeuser\_reflector.

Note:

The author of this section, K. J. Martin, is being transferred to another subproject. B. A. Tague joined the Central System Control subproject shortly before publication of this section. Although he was not involved in the authorship of this section, he is aware of the issues and will be working on the task. Therefore, comments and criticism of the ideas presented may be directed to him.