Jerry

file - Multics redesign proposals

To: Distribution

From: T. H. Van Vleck

Date: 1/24/72

Subject: Party Groups

Further work on the design of the party group mechanism leads me to the conclusion that we can split the initial proposal into two parts in such a way that the first part is very easy and solves most of our current problems in the load control area.

The first phase will provide the biggest change to the user interface: it introduces "CTSS-style" party groups, and the notion of primary and standby status for users. The party groups can be allocated a quota of primary load units representing a guaranteed number of users; users in excess of the quota will be logged in as standby, and will be subject to bumping if a user who receives primary status logs in; and bumping within a party group across project boundaries will work.

This first phase can be accomplished by the installation of two procedures, and the creation of a single new data segment.

Assume #3 or p-5 is one for illustration

c/c Vic → How hard to divide a project (e.g. Multics) into several line groups?

The second part of the pary-group implementation could provide some or all of the following features:

1) Scheduled service

2) Edit-only service

3) Party group administrators

4) party group control of some project attributes

All of these functions are easy to code but involve more programming than the basic party-group mechanism. Furthermore, these functions may not be necessary at all: I'd like to live with the basic implementation for a month or two before dedicating the resources necessary to provide what may turn out to be nothing but "frills".

There are a few miscellaneous tasks which fall between part one and part two. These tasks include:

1) Modifications to "who" to show group and standby status

2) Administrative reports by group

3) Minimum waiting time after user is bumped before he can login again.

Detailed description of part one

Only one module in the system is involved in the actual load-control decision. Replacing this program, "load_ctl_", with a version that uses a new strategy, is straightforward.

The new version of load_ctl_ will discover the group associated with each project by inspecting a slot in the SAT which is currently ignored. This slot holds an 8-character group identifier, which will be looked up in the "master group table" (MGT) to find the current and maximum number of primary load units for the group.

The other module that must be installed is a simple-minded editor which allows the system administrator to create and edit the MGT. (something like the editor which now exists for the "installation_parms" segment is all that is required.) Group administrators who wish to change parameters of a project in their group and the moving of projects from one group to another will be handled by the system administrators on a request basis. These changes should be infrequent enough so that not much work is involved.

The MGT for the basic party-group mechanism need contain little more than the group identifier, the current and maximum number of primary units, and the current number of standby units for the group. Later, we may want to add pointers to a schedule data base, attribute masks, administrator identifiers, and a count of edit-only users.

The flowchart provided as figure 1 shows the guts of the login decision. The algorithm references the following data items:

1) System maxunits (already in SAT)

2) System current units (already in SAT)

3) System max. users (already in SAT)

4) System current users (already in SAT)

5) Group max. primary units (new, in MGT)

6) Group current primary units (new, in MGT)

7) User "guaranteed_login" attribute (already in pdt)

8) User "prime_ok" attribute (new, in pdt)

9) User "standby_ok" attribute (new, in pdt)

10) User primary flag (new, in answer table)

11) User "bumping" attribute (already in pdt)

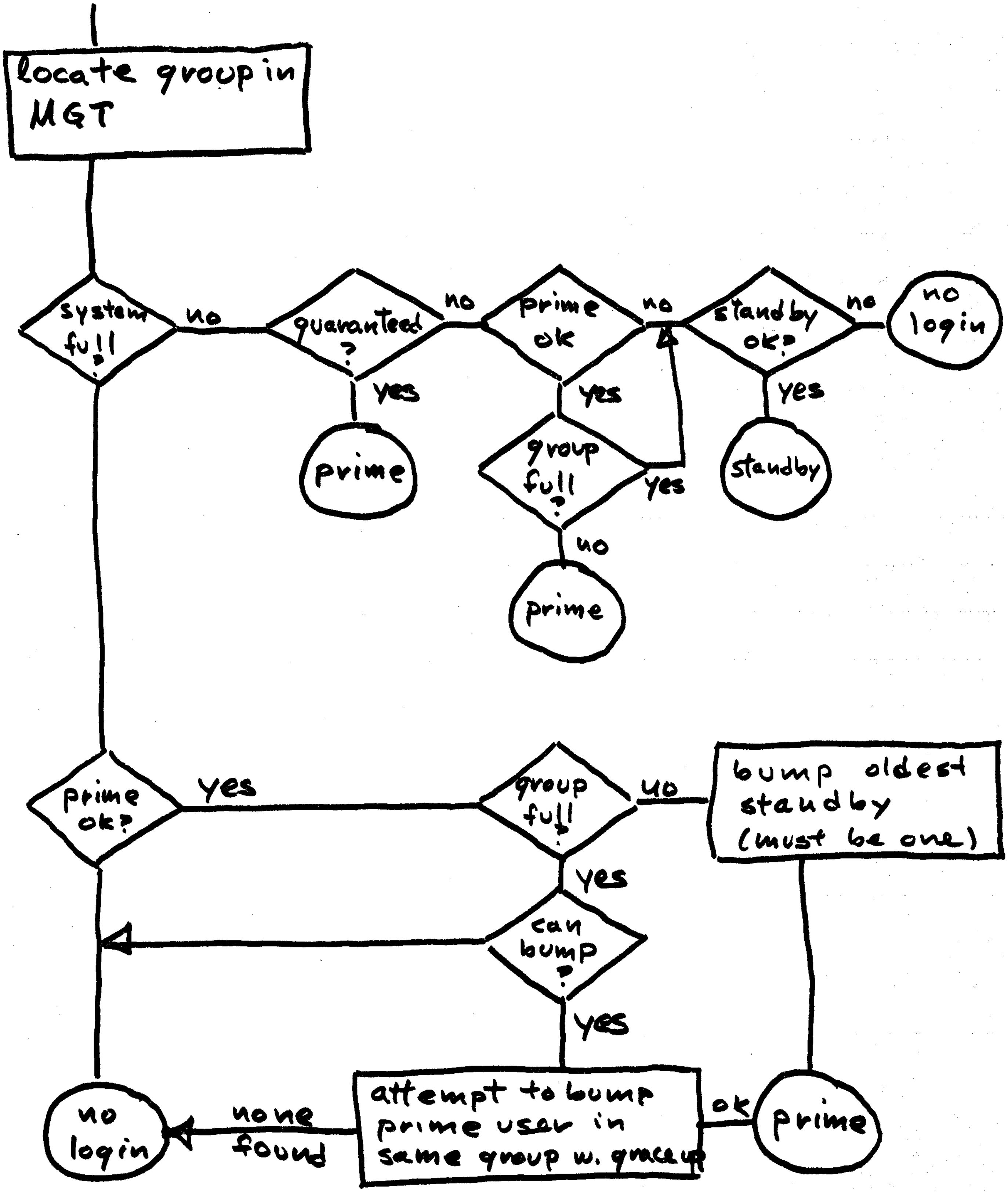
12) User "bump grace" (already in pdt)

13) User login time (already in answer table)

14) User group identifier (new, in answer table, from sat)

Two small complications are not shown on the chart. One is the fact that the system decides whether it is "full" by checking the current number of load units against the maximum, and also by checking the current number of user processes against a maximum. This second check is to insure that we don't run out of APT entries. The other is the case when a user with "guaranteed_login" attempts to log in on a full system. If the system maximum users is not full, he will log in and overload the system. Otherwise, an "emergency preemption" will be sent to the first standby user, or, if there are no standby users, to the first available user (this should never happen).

load_ctl_ - decision at login



Administrative steps

These may be the most difficult. The users must be informed that the mechanism is going in, and what it will do to them. Because we will now make some users standby and subject to bumping, they must be told what the message

You are standby and may be bumped

at login means to them, and why they were made standby. In order to make this explanation, we must publish the number of primary units allocated to the various groups. We will have to negotiate these allocations with the IPC administration.

The users should have at least a week's notice before the machinery is installed. A sample Bulletin article, which could also be used as a "help" file, is attached as an appendix.

Project administrators will not have to do anything, except perhaps re-install their PDT's, unless they want to take advantage of some of the new features of the party group machinery. That part of the Project Administrator's Manual dealing with bumping is attached as an appendix.

APPENDIX 1 Bulletin article

```
***********
From Multics: Standby user preemption.
You will be logged out in 3 minutes
***********
```

The user should then clean up what he is doing and log out. If he has not logged out in three minutes, he will receive an automatic logout. . If the user is being bumped because his primary "grace" ran out, the message will say "Group preemption" instead of "Standby user preemption".

Note that specifying "bumping" and "no_prime" together does not make sense. These attributes may be specified for all users in a project by use of the "Attributes:" statement in the PMF header, as shown above.

Primary users, once they attain primary status, are immune from bumping for a period which you may specify. The length of this period, the "grace", controls how long the users may monopolize a primary line. Your project has a maximum grace, set by the group administrator. To set the grace for a particular user, specify

```
     grace:                    60;
```

where the parameter is the number of minutes' protection desired. To set a default value for all users of a project, specify

```
     Grace:                    120;
```

for example, in the heading of the PMF. This value will be used unless a "grace" statement is found in a particular PMF entry. If you leave the grace completely unspecified, or it you attempt to specify a grace in excess of you project's maximum, the maximum grace set by the group administrator will be used.

When a primary user logs out, a standby user may be "promoted" to primary status. Whenever the system looks for a standby to promote, or for a bumpable user to bump. it picks the user who meets the other criteria (such as groups matching or not matching) and who logged in first.