Identification

Initial Load Control
H. M. Deitel

Purpose

Load Control allows the system to restrict the number of users to some specified amount, and to regulate this number during system operation. This paper describes a straightforward implementation of Load Control for initial Multics. A more sophisticated algorithm will be used in later versions.

Discussion

It is desirable that the load on the system be such that each interactive user is reasonably comfortable with the rate at which he interacts with the system. When the load gets "too heavy" (defined for a particular installation), the interaction rate decreases and system overhead may sharply increase. To prevent such an overloading, Load Control is provided to maintain the number of users at some "satisfactory" level. Load Control is called to perform such functions as:

a) determine if a particular user may log in
b) reset the maximum number of users allowed on the system
c) automatically log out a particular user
d) automatically log out a number of users to forcibly decrease the load

Load Control Table

To aid in performing its functions, Load Control maintains a system-wide data base in a segment with pathname:

">sys_process_reserved_storage>load_control>load_ctl_tbl"

dcl 1 load_ctl_tbl based (lp), /* load control table */
2 max_users fixed bin (17), /* max no. of interactive users */
2 cur_users fixed bin (17), /* current no. of interactive users */
The table contains a fixed length and a variable-length portion. The fixed length portion contains the value for the maximum number of interactive users, the value for the current number of interactive users, and a lock structure used by the Locker facility to lock the table while it is in use. The variable length portion of the table is a list containing one entry per interactive user. Each entry contains the user's process-group id, the process id of the user's User Control Process, and an event channel id of the event channel to be used to signal an automatic logout to this user's User Control Process.

Entries to Load Control

To inquire if a particular user may be allowed to log in (and, if so, to insert an entry for this user into the Load Control Table):

```c
load_control$try_to_insert (username, projid, login_id, reply, auto_logout_channel);
```
dc1 username char (24),
    projid char (24),
    login_id char (2),
    reply bit (1),
    auto_logout_channel bit (70);

username - user's name
projid - user's project identification
login_id - instance tag (always "aa" for initial Multics) returned if the user should be allowed to log in. The instance tag uniquely identifies each of the process-groups for a user-name-projid combination which is logged in more than once.
reply - (return argument)
   "1"b if the user should be allowed to log in
   "0"b if the user should not be allowed to log in
auto_logout_channel - event channel id of the wait-type event channel which may be used to signal an automatic logout to this User Control Process.

To remove the entry for a user from the Load Control Table:
load_control$remove_user (username, projid, login_id, remove_status);

Same declarations as in try_to_insert call except for:

dcl remove_status bit (36),
remove_status - "0"b if entry successfully removed from Load Control Table
               "1"b if entry not found in Load Control Table

To reset the maximum number of users allowed:
load_control$set_max_users (new_max_users);

dcl new_max_users fixed bin (17);
new_max_users - new value of the maximum number of interactive users (max_users)
To automatically log out a specific user:

\[
\text{load\_control\$force\_out\_user (username, projid, login\_id, force\_status);}
\]

dcl force\_status bit (36);

force\_status = "$0"b if entry found in Load Control Table
"1"b if entry not found in Load Control Table

To trim the system load (by automatically logging out a number of users):

\[
\text{load\_control\$trim\_load (n\_users, trim\_status);}
\]

dcl n\_users fixed bin (17),

trim\_status bit (36);

n\_users - number of users to be automatically logged out.

trim\_status - "$0"b if successful
"1"b if caller specified that more users should be logged out than are currently logged in. [Action taken: all users are logged out except for the user represented by the first entry in the Load Control Table.]

**Operation**

**AT LOGIN TIME**

The user\_in procedure in this user's User Control Process calls Load Control to "try-to-insert" this user into the system.

Load Control checks to see that cur\_users is less than max\_users. If not, reply is set to "$0"b and control is returned to the user\_in procedure. If cur\_users is less than max\_users, Load Control searches the user\_list to see if this user is already logged in. If he is, then reply is set to "$0"b and control is returned to user\_in. If he is not already logged in, reply is set to "$1"b, an entry for this user is placed in user\_list, cur\_users is incremented by 1, and control is returned to user\_in.
AT LOGOUT TIME

The user_out procedure in this user's User Control Process calls Load Control to "remove_user" from the system.

Load Control deletes this user's entry from the user_list, decrements cur_users by 1, and returns to user_out.

Note: Load Control calls the Locker to lock the Load Control Table whenever it is in use. The table is unlocked after Load Control has finished referencing it.

DURING SYSTEM OPERATION:

A - resetting maximum system load (and trimming system load by attrition).

The value of max_users is initially determined by the value in the Load Control Table at system initialization time. Once the system is running, this value may be changed by "set_max_users". If the new value of max_users is less than cur_users, the system load is trimmed by attrition, i.e., no further interactive users are allowed to log in until enough users have logged out to make cur_users less than the new value of max_users.

B - forcing a particular user to log out:

System Control calls Load Control to "force_out_user".

Load Control locates this user's entry in the Load Control Table and sends an automatic logout signal over the event channel specified in this entry to this user's User Control Process.

C - trimming the system load (by force):

System Control calls Load Control to "trim_load" by automatically logging out a certain number of users.

Load Control sends automatic logout signals to the specified number of User Control Processes. Under no circumstances is an automatic logout signal sent to the first person who logged in *(presumably the system operator). First, users whose project id is not "sys_operator" or "operator" are logged out (until the specified number of users have been logged out). If more users must be logged out then, and only then, users with project id "sys_operator" or "operator" are logged out until enough users have been logged out to fulfill the request.
*Note: At least for the duration of Initial Multics, the system operator will, in fact, always be the first person to log in. Therefore his entry in the Load Control Table will be the first such entry, and he will not lose this place.
**try_to_insert**

- cur_users < max_users
  - reply = 0
  - remove_user
    - delete the user from load_ctl_tbl
    - decrement cur_users by 1
    - return to caller
  - yes
    - reply = 1
    - return to caller
    - place entry for this user in load_ctl_tbl
    - increment cur_users by 1

**force_out_user**

- locate this user's entry in load_ctl_tbl
- send automatic logout over specified event channel to specified User Control Process
- return to caller

**trim_load**

- logout specified number of users whose prjids are not 'sysoperator' or 'operator'
- have we logged out specified yes number of users?
  - yes
    - decrement max_users by number of logged out users
    - return to caller
  - no
    - log out additional users regardless of project id (but don't log out first user)

**set_max_users**

- reset value of max_users in load_ctl_tbl
- return to caller

---

*Fig. 1 Load Control Processing*