TO: Multics Distribution

FROM: C. Marceau

SUBJECT: System Control in Limited Initial Multics

DATE: 6/28/68

DRAFT

date_____

copies: Corby
Daley
Saltzer
Kriloff

"For considerable periods, the four oboe players had nothing to do. The numbers should be reduced and the work spread more evenly over the whole of the concert, thus eliminating peaks of activity.

"All the twelve violins were playing identical notes; this seems unnecessary duplication. The staff of this section should be drastically cut. If a larger volume of sound is required, it could be obtained by electronic apparatus.

"Much effort was absorbed in the playing of demi-semi-quavers; this seems to be an unnecessary refinement. It is recommended that all notes should be rounded up to the nearest semi-quaver. If this were done, it would be possible to use trainees and lower-grade operatives more extensively.

"There seems to be too much repetition of some musical passages. Scores should be drastically pruned. No useful purpose is served by repeating on the horns a passage which has already been handled by the strings. It is estimated that if all redundant passages were eliminated, the whole concert time of two hours could be reduced to twenty minutes and there would be no need for an intermission.

"The conductor agrees generally with these recommendations, but expressed the opinion that there might be some falling off in box-office receipts. In that unlikely event, it should be possible to close sections of the auditorium

entirely, with a consequential saving of overhead expenses, lighting, attendants, etc. If the worst came to the worst, the whole thing could be abandoned and the public could go to the Albert Hall instead."

Identification

Interim System Control

C. Marceau

Purpose

Some restrictions of Limited Initial Multics impose a simplified form on initial system control. These restrictions include:

1) no separate device manager process,
2) long time needed for process creation ,
3) large amount of wired-down core needed to keep a process loaded,
4) long time needed for process loading.

The resulting "system control" bears little resemblance to that described in BQ.1.01. Experience with interim system control will determine the future development of system control.

Discussion

In Limited Initial Multics, after system initialization, the initialization process (or possibly a daughter process) begins to execute the user control code to log in a user from the operator's line (the line identification is a system constant). User control attempts to read a login line from the console. When the "operator" (probably a system programmer) dials up and types a login line, user control logs him in. He now types the command answer (see BX.15) giving as arguments the names of lines over which the system should accept dialups.

The answer command creates a process group for each line, giving it the name

.user_control.xx

where

xx=aa,ab,ac, etc.

Answer creates a process in each process group ~~and~~ In the process initialization table for each process, it specifies that the process should execute user_control (see BQ.2.03) after initialization and, it provides as data the name of the line which the process-group is created to serve.

The operator's process is now ready to progress to other work. As users dial up, they find processes awaiting their login lines. Note that if process A is waiting to read a login line from a console, when the user dials up and begins to type, the resulting interrupt to process A is enough to rouse it and cause it to log in the user.

When a user logs in, user control in his process changes the name of his group from

.user_control.xx

to

Joe_Doe.Multics.aa (or whatever).

When a user logs out, user control creates a new user control process-group (with name .user_control.xx) and causes it to begin executing in user_control$ cleanup. This entry of user control destroys ("cleans up") the old process group before giong on to log in a new user over the same line.

Evaluation

Let's see how this scheme lies within the restrictions mentioned in the purpose.

1) No separate Device Manager Process. There is one process waiting to read on each line over which users may dial up. Since that process is dedicated to that line, no process is kept from doing useful work while waiting for a user to dial up or to type.

2) Long time needed for process creation. When a use logs in he does not have to wait for a process to be created before he can work. The process is waiting for him. (On the other hand, the ready message after the answer command should be a dilly).

3 and 4) Long time needed for process loading, but loaded processes use lots of core. The Answering Service Process and the System Control Process have disappeared. The purpose of the Answering Service was to respond to user dialups (no longer necessary) and the System Control Process was to mediate between the operator and the system (it is not yet clear whether this function will be necessary in successors to Limited Initial Multics.) The processes waiting for user logins may be unloaded until users log in.