Short Communications

COMMENT

MOO in Multics

In the summer of 1970, John Larmouth of the University of Cambridge visited me at the Massachusetts Institute of Technology. In talking about Multics, he suggested that we write a simple program to show the various capabilities of the system. Since the game of MOO had proved quite popular at Cambridge, it was, of course, on Dr. Larmouth's mind and we decided to implement a simple version of it on Multics.

After Dr. Larmouth had left, I showed the game to a few friends, and they suggested that I set the access control mechanism so that anyone could get at it and also set up a ladder as had been done at Cambridge. This was promptly accomplished and within two months we had over 75 players and approximately 1000 games had been played.

Interest in MOO was still increasing as people tried to arrive at algorithms and MOO-playing programs, etc. As mentioned in the article in *Software*,[1] there are several characteristics of the Multics system which prevented the simple development of a tamper-proof MOO ladder. It was, of course, possible to make MOO a 'ring-0' procedure (see Graham[2]). However, this was generally against system policy. Similarly, it would have been possible to allow the use of MOO and the associated ladder only when a user was logged-in under a special name. This raised the question of who would pay for this special user I.D. (currently when a user makes use of a program owned by me, he still pays for the computer time to do so—I simply pay for secondary storage charges). This was also deemed infeasible. Therefore, a few detours were inserted in the program to insure that tampering would require a modicum of effort on the part of the tamperer—generally discouraging people from doing this more than a few times (once the novelty had worn off).

Access control in the Multics system is dependent upon the name of the person doing the accessing and the program or data file which is being read, written or executed. However, if a person is executing a program which writes a data base (e.g. MOO writes into the ladder) it is necessary for that person to have access privileges to that data base. In other words there is no way of saying 'a person can have access to a data base only when it is accessed through a particular program'. This is, of course, a difference between Multics and the University of Cambridge system.

The measures taken within the Multics implementation of MOO to thwart (but clearly not prevent) tampering are as follows:

(1) The name of the data base does not explicitly appear in the program but is generated by an algorithm.
(2) The data base is made 'active' for the minimum amount of time necessary to modify it. Therefore, the name and location of the data base does not remain in tables accessible to the user for a long period of time.
(3) While it is necessary to give all users read–write access to the data base, it is not necessary to give them similar privileges on the directory in which it resides (see Daley[3]). This means that if the name of the data base is known or can be generated, it can be accessed, but the directory cannot be searched to see if that data base exists.

Our experience to date has been encouraging. In about six months of usage over 3000 games of MOO were played (for recording on the ladder—many more may have been played without that option) by some 150 players (out of a total population of 500 using the Multics system). Tampering occurred quite a lot in the beginning but has fallen off.

The rate of usage seems to be declining as other games arrive on the scene. 'Life', not really a competitive game, but simply a way of generating patterns based on the theory of cellular automata (see Conway[4]), had a short flourish some time ago as the possibilities for using graphic displays were intriguing. The introduction of the Dartmouth BASIC library into Multics with its approximately 100 different games will probably be enough to keep people amused for a period of time as well. (By the way, the story of how the entire Dartmouth system was encapsulated in Multics in about 4 months is interesting, too.)

REFERENCES

1. $\aleph_0$, 'Computer recreations', *Software—Practice and Experience*, **1**, No. 2, 201–204 (1971).
2. R. M. Graham, 'Protection in an information processing utility', *Comm. ACM*, **11**, No. 5, 365–369 (1968).
3. R. C. Daley and P. G. Neumann, *A General Purpose File System for Secondary Storage*, Proceedings of the Fall Joint Computer Conference, Spartan Books, Washington, D.C., and Macmillan, London, 1965.
4. J. Conway, in 'Mathematical games' (Ed. Martin Gardner), *Scient. Am.* **223**, No. 4, 120–123 (1970).

JERROLD M. GROCHOW
Information Processing Services
Massachusetts Institute of Technology
Cambridge
U.S.A.