DESIGN NOTEBOOK

APPENDIX M

April 9, 1965

TO: Professor F.J. Corbató

FROM: J. H. Saltzer

SUBJ: Allocation of Time-Sharing System Resources.

This note attempts to pin down in writing and in one place a number of ideas which have been suggested by many people about one phase of the allocation problem. It offers little new except perhaps some nomenclature. The problem considered is that of "charging" a user for his use of system facilities and, if necessary, limiting such use. This area involves both administrative (policy) decisions, and questions of structure which may affect the design of at least the time-accounting sections of the system, We hope here to avoid questions of policy and construct a structure permitting a wide range of policy decisions as possible.

The allocation problem on the 7094 CTSS is handled by allotments of four (now five) kinds of computer time, according to time, day, and priority; and, as originally proposed, two kinds of disk storage, distinguished by permanancy. Thus a particular user may be allocated a maximum of 1 hour of 7094 time during prime shift, 2 hours on weekends, and 150 tracks of disk storage space. We may call such a system a multi-class allocation system. This multi-class allocation technique, while attempting to force particular patterns of usage (it is harder or "more expensive" to obtain prime shift allotments) is guaranteed to produce the following effects:

1. Breakage. Some users will use up one class of allotment first while others will use up a different class; if for any user work is impeded, probably the rest of his allotment will go unused. When

allotments go unused, there is danger that the system will go unused.

2. Overallocation. To avoid the possibility of parts of the system going unused, each of the separate classes of usage must be overallocated. While overallocation is a sensible approach to obtaining full utilization, an attempt to precisely overallocate each of a half-a-dozen classes of resources is very likely to result in actual running out of some of them, a prospect to be avoided unless it can be handled gracefully.

3. Unreliability. In estimating the amount of resources needed to solve a problem, a user is forced to sit down in advance and decide roughly how many hours or minutes of first, second, and third shift time he is likely to utilize. It is bad enough that he has to predict his total strain on the system; to force him to break down his usage into the classes prescribed by the system will likely produce an intended overestimate of each class, for safety, and an unintended underestimate of some classes, since the problem may not be thought out well enough to predict how its solution will go.

To avoid the evils inherent in the multiclass allocation system, a single class allocation system is proposed. With this system, a user is given an allocation of "Information Processing Credits" (IPC's). He may redeem his IPC's according to some published rate scale, such as the following:

One IPC will buy 1 minute of computer time between 9 and 5.
One IPC will buy 1.5 minutes of computer time from 5 to 12 and on weekends.
One IPC will buy 2 minutes of computer time at night.
One IPC will buy $2 \cdot 10^6$ bits of primary bulk (disc) storage for a month.
One IPC will buy 10^7 bits of secondary bulk (tape) storage for six months.

The important difference between this system and the multi-class allocation system is that the various separate classes of usage have been submerged below the allocation level; decisions as to which class of service to use can be made on a continuing basis by a user in the light of his present problems and the rate of using up of his allocation of IPC's.

The rates, of course, are set by consideration of the relative desirability of usage of the various classes of service. If there is unused third shift time available when all other shifts are being used to capacity, the third shift rate may be lowered to 3 minutes per IPC to encourage some users to move to that shift.

The single class allocation system is very adaptable and flexible. For example, suppose that a new high-capacity disk and channel have been added to the system to replace an older. slower, smaller one. If the system resources would be better balanced by heavier usage of this new disk, the IPC rate of disk storage can be lowered. Similarly, if disk storage should become scarce, and for technical reasons it is impossible to add more disk storage, the bulk storage IPC rate can be increased, to encourage users to go to tape storage for their larger, less frequently used files, or use more computer time to regenerate some files rather than storing them. If such a system were in effect on the Project MAC 7094 system, probably the present crisis in disk storage space would have been easily averted, or at least taken care of quickly.

An extremely important feature of the single-class system is that when a change of rates is made, a user is not forced to instantly change his habits; he may take his time if he is willing to accept the higher cost temporarily. Compare this situation with that of a user in the present system whose track quota has been suddenly cut in half.

At least the following machinery should be available to the system administrator and to any user:

1. The system administrator can allocate any number of IPC's to any user; he may overallocate if he desires.

2. Any user can sub-allocate part or all of his own quota of IPC's to other users in any of the following ways:
   a. Allocation of a fixed quota of IPC's to each user.
   b. Allocation of a pool of IPC's to a group of users in the spirit of a joint checking account.

c. Allocation as in (b.) but with a limit on the amount any individual can draw from the pool. (This technique may be named overallocation, if you like.)

d. Any combination of the above possibilities, with any set of users.

3. Any user may obtain part of his allocation from one allocator, another part of his allocation from another. In such a case the user should be permitted to decide at any time which portion of his account should be charged, or that his IPC usage should be shared in specific proportions. It would be desirable if he could change his mind about the decision for a reasonable period of time after the IPC's have been used.

4. Any user may transfer any part of his allocation to any other user, if this transferral is permitted by the person providing his original allocation.

5. By a console command, any user should be able to determine how many IPC's he has used (since a given time, if desired) how many IPC's he has left, and how many his group has left, if he is a member of a group. A user who has allocated his IPC's to other users should be able to ask the same questions of any of the other users' allotments and of his own unallocated allotment.

We must emphasize that the proposed single class allocation technique does not solve all problems; in fact it raises some very difficult ones which we must surmount if the advantages of the scheme are to be obtained. The most important difficulty is that demand for computer use is subject to wide fluctuations, with only the rate structure available to damp out such fluctuations. There is nothing in the single class allocation principle which prevents all users from attempting to cash in all their IPC's in a single afternoon. Although such an extreme possibility is unlikely, periodic fluctuations, such as during the period in the spring just before thesis deadlines at any university are well known.