INTERDEPARTMENTAL

MASSACHUSETTS INSTITUTE OF TECHNOLOGY CAMBRIDGE, MASS. 02139

from the office of

July 3, 1969

TO: H. E. Brammer, Electrical Engineering Department

FROM: J. H. Saltzer, M. I. T. Project MAC

SUBJECT: Cost of Multics for Block Student Use

This memo is an addendum to the memo "A Proposed Pricing Policy for Multics at M. I. T.". It provides two lines of discussion not given in that memo:

- Consideration of the effect of block purchase of machine access on prices. In general, block purchase reduces utilization uncertainty, and therefore reduces prices.

- Estimation of the amount of resources which would be used by a student, both at the present state of system development and those states projected over the next 12 months.

Block Purchase

All of the prices given in the original pricing policy memo were based on demand utilization of the system. That is, it was presumed that the system has available sufficient equipment to meet peaks (and steady growth) of usage which could, in general, be precisely predicted or scheduled. As a result, it was necessary to propose prices which recover the cost of the system when it is only 2/3 utilized. We consider here the first-order effects of block purchase of typewriter ports, paying a lump sum in advance for utilization of those ports.

Discussion parallels the other memorandum.

1. Central Processor. On a base of 450 hours/month, one has a price of $67/hr. for block usage. The non-block price remains at $100/hr. since both the remaining amount of cpu time (of which 2/3 is sold) and the amount of cost to be recovered are reduced in the same ratio by block purchases at $67/hr.

2. Pageable Core Memory. The Multics algorithm for using core memory tends to use as much as is available at all times regardless of load, so its price of $1.40 per page-hour is not adjusted to account for a predictable load.

3. Secondary Storage. On a base of 34M words rented, one has a block purchase price of $0.70/page-month. As with cpu usage, non-block purchase price remains unchanged.

4. Teletypewriter Communication Equipment. This computation originally assumed only 1/3 usage, and did not presume that many light users would help provide economies of scale in the typewriter adapters. For block purchase, we should have

GE equipment rental	$2340
64 data sets (not 48)	1600
	$3940

$$\frac{(3940)\ (2.38)}{64} = \$147\text{/console-month*}$$

If a console is used for 300 hours/month, one has an hourly cost of $.50/console-hour.

5. ARDS Equipment. This option is presumed not to be interesting because of the high price of the ARDS terminal. However, based on block purchases, a price of $460/console-month would result.*

---

* Note that this figure includes rental of equipment at the computer end only. It does not cover the terminal itself or its associated data set, which would, for example, run about $120/month for model 37 teletype consoles with data sets.

Estimation of Student Resource Usage

The attached copy of Multics Performance Log 20 shows the performance of the system today for a programmer writing and debugging small FORTRAN programs, an application probably similar to usage a student would make. This measurement was taken by having our PDP-8 computer simulate a real user by dialing up Multics, logging in, typing in a program, etc., all with human reaction times. The test shown was performed with a system fully loaded with system programmers. In about 45 minutes, this simulated FORTRAN programmer used about 90 seconds of CPU time and took about 5000 page faults, for a core charge of about 2.6 page hours.* Thus, this particular user would be charged on today's system about

\$1.67	(90 cpu seconds @ \$67/hr.)
3.65	(2.6 page-hr. @ \$1.40/page hr.)
.38	(45 console-minutes @ \$0.50/console hr.)
\$5.70	

in 45 minutes, or about \$7.70/hr.

This brings us to the question of performance improvements which are expected to make a sizeable reduction in those numbers. The most important simple improvement currently planned is a reduction of page-seconds used by a FORTRAN programmer by as much as a factor of four. The first factor of two turns out to be fairly easy to accomplish, by reorganizing the path of control followed by the supervisor in handling the routine situation. These changes permit the light user to get by without involved machinery only needed for special situations; they are scheduled to be done by October, 1969. The second factor of two requires methodical auditing of each of the commands used by the FORTRAN programmer, a technique which has been very successful elsewhere in the system, but which takes time. This same auditing is expected to produce smaller improvements in cpu time used by each command. A reasonable estimate is that by Spring, 1970, this script will require about 60 seconds of cpu time (\$1.10), about .7 page-hours (\$1.00) and an unchanged \$0.38 for terminal charges, for a cost of about \$2.50 in 45 minutes, or \$3.30/hour.

This discussion has not gone into the area of student program storage on the disk, which raises questions about the strategy of usage of the system by the department. If one presumes that a student is allowed to leave behind him files requiring 10 pages (10,000 words, enough room for two or three small programs) of permanent storage, this storage will introduce an additional charge of \$7/month per student.

---

* Each page is presumed to be in core about 2.0 seconds, the average run time per interaction.

This figure can be converted into an equivalent price per console-hour by dividing it by the number of hours per month estimated for each student.

Discussion

The comments given above should be tempered by the observation that all of the estimations are rough, and one is only guessing about the actual load presented to the system by a student. The predictions also depend on a majority (but not all) known performance improvements working as well as anticipated, and also on the willingness of the Information Processing Center to arrange block purchases of machine access.