

A NOTE ON CHARGING

INTRODUCTION

Within the next few weeks it will be possible to begin charging for the use of TSS. In fact we will probably begin testing the charging procedures on live customers within a week. This note will explore various rationales for setting rates and the resulting charges for a few standard tasks.

RATIONALES

There are a number of different ways to set rates. The method used in this note is to set a nominal rate for the machine as a whole. This rate is then partitioned among the various resources for which we can account. The actual rate for the machine as a whole will then depend on the percentage of utilization of each of these accountable resources. The nominal rate for the machine as a whole is then adjusted on an ad hoc basis to obtain the desired income.

Another method, not used in this note, is to attempt to set the rate for each piece of equipment so as to pay its costs. Thus, for example, if disk storage becomes full there would be enough income from the disk itself to pay for another. This method has a number of problems. Not all components can be extended at will and all components are not accounted. Moreover, this method makes no allowance for various overhead costs, such as system programmers. It might be worthwhile to examine this method later.

Once a nominal overall rate has been set, it must be partitioned among the accountable components. The obvious method that presents itself is to divide the rate proportional to the cost of each component. This leads to similar problems to the other method, i.e., not all components can be accounted, and not all of each accounted component can be accounted. Another problem is that one of the accountable items, connect time, is not a component.

We can charge for part of each of the major components: CPU, ECS and the disk. One procedure is to just ignore all other components, and that is what is done in this note. Another question is whether to weight each component by its full cost and require the accountable portion to pay for the full weight, or to weight each component by the cost of its accountable portion. This is a real problem for ECS since we can only account for half of it. Finally, there is the question of how to determine the cost for a component, by the original cost to the Computer Center, or by a probable replacement cost. These last two questions lead to four possibilities, and in later computations all of them are examined.

STANDARD TASKS

There are a number of standard tasks considered in this note. The first is the null task. The cost for just logging in to the system and doing nothing, the so-called connect charge. This must be distinguished from a charge for connect time alone, since while a TTY is logged in it is sitting on some resources that are accounted. The total connect charge would be the sum of the charge for connect time alone and the charge for the resources reserved to the TTY.

It will turn out later, under the charging assumptions made in this note, that the charge for the resources reserved to a logged in TTY are quite high, due to a number of inefficiencies in the current system. It is hoped that they will be reduced in the next six months. In the meantime we propose a negative charge for connect time, so as to reduce the total connect charge to a reasonable amount. This amounts to simply permitting the free use of a minimum amount of ECS, and then charging for any additional.

The total resources used by a logged in TTY are 7.5K (decimal) ECS and some disk space. The ECS space comes from two types of overhead. The first is system overhead of 4K of fixed ECS space, which should reduce over the next six months. The second is due to a crude algorithm for the control of swapped ECS space, which also should reduce to near zero under the forced swapped procedures to be installed late this year. The cost of the disk space is ignored, since it is at most \$.10.

The next tasks considered will involve the use of the SCOPE Simulator. All estimates of charging assume that ECS will be used for the same length of time as is the CPU. This would be approximately true if only one user is on the system. Since ECS costs will be low compared with CPU costs, the errors can be ignored under low load. Under high load the increase in cost for ECS will become significant, which may lead users to avoid the system during high load.

All of these SCOPE tasks will assume a field length of 45K (octal), sufficient for large assemblies. Under these conditions, the SCOPE Simulator requires 100K (octal) of swapped ECS. In order to run large assemblies more disk space than the nominal amount for a logged in TTY must be reserved. Compared to the cost for CPU this will be small, on the order of \$10.00 an hour, and therefore will be neglected.

The particular SCOPE tasks will be: null, null assembly, and large assembly. Under the current system the null use of the SCOPE Simulator, call and return, takes about 12 seconds. Running a null NOMPASS assembly raises the total to about 20 seconds.

The large assembly is one of the decks used in the system itself. The cost on the A machine for assembling the deck under MOMPASS is 1 minute of CPU time and 80 seconds of PPU time. At \$400 per hour on the A machine the charge would be about \$6.70. Under the current TSS system the cost seems to be about 3 minutes, including the overhead for calling the SCOPE Simulator.

The final task to be considered is that of permanent file storage. We will attempt to compute the costs on a per month basis. Since the system accumulates charges for disk space only while the system is actually running, we need an estimate of the number of hours the system will be up in a month. Since a month has about 700 hours, it is unlikely that the system will be up more than 350 hours.

Next we need estimates of the amount of permanent disk space needed. These are given for three classes of users: small, medium and large. The small user is a student with a very small program to save, say about 1 or 2 pages. We estimate that about 800 words or 12 sectors will be sufficient. The medium user has maybe a 50 page program, about 20 thousand words or 300 sectors. Finally, the large user has maybe 120 thousand words, or about 1800 sectors. (The system cannot support many large users.)

#### RATE SETTING

In order to set a rate by the method chosen, we need a nominal rate for the whole machine. For this purpose we have chosen \$400 per hour so as to compare with the A machine, which is slightly more than \$400 an hour. The B machine is probably somewhat cheaper than the A machine since it does not have as much central memory, nor as many printers or tape drives. The operator overhead for running TSS is considerably less than the SCOPE system, at present requiring an operator only at start up and shutdown, plus an occasional tape mount. The programming staff for TSS is also somewhat smaller than for the A machine.

The TSS should give a much higher rate of utilization for the accounted components than does the SCOPE system, thus inherently producing a higher rate of income for a fully loaded system. (\$400 per hour, 24 hours per day, 365 days per year amounts to  $\$3.5 \cdot 10^6$ .)

In view of the results that follow, we suggest starting with a basic rate of less than \$400 per hour if the desire is to produce a competitively priced system.

Having chosen a basic rate, we need the comparative costs of the various components in order to divide up the rate. The following table contains the basic information used in the subsequent calculations. These values were obtained from Ken Hebert on August 12, 1971.

<u>component</u>	<u>original cost</u>	<u>probable replacement cost</u>	<u>standard unit</u>	<u>total standard units (NSU)</u>	<u>accountable units (NASU)</u>
CPU + 32K CM	.69*10 <sup>6</sup>	.5*10 <sup>6</sup>	1 hour	1	.9
300K ECS	.48*10 <sup>6</sup>	.6*10 <sup>6</sup>	1 K-words-hr	300	160
1/2 disk	.26*10 <sup>6</sup>	.10*10 <sup>6</sup>	1 K-sector-hr	1020	765
other	.08*10 <sup>6</sup>	.10*10 <sup>6</sup>	--	--	--

The following tables contain the computation of the rates per standard unit under different assumptions. The following symbols are used:

R = total rate to be distributed (using \$400 per hour)  
 cc = component cost  
 tc = total cost for all components  
 nsu = number of standard units in the component  
 nasu = number of accountable standard units in the component  
 ac = cost of accountable portion of a component = cc × nasu/nsu  
 tac = total cost of accountable portions of components

Next we give the formulas for the two methods:

Method A

$$\text{rate/s.u.} = R \times (\text{cc/tc}) \times (1/\text{nasu})$$

Method B

$$\text{rate/s.u.} = R \times (\text{ac/tac}) \times (1/\text{nasu})$$

Finally we give the tables containing the computations:

Method A

component	(original costs)			(replacement costs)		
	cc	cc/tc	rate/su	cc	cc/tc	rate/su
CPU + 32K CM	.69	.46	\$ 205.	.5	.38	\$ 170.
300K ECS	.48	.32	.80	.6	.46	1.15
1/2 disk	.26	.17	.089	.1	.08	.042
other	.08	.05		.1	.08	
tc =	1.51			1.3		

Method B

component	ac			ac/tac		
	ac	ac/tac	rate/su	ac	ac/tac	rate/su
CPU + 32K CM	.62	.59	\$ 263.	.45	.53	\$ 236.
300K ECS	.25	.23	.58	.32	.38	.95
1/2 disk	.19	.18	.094	.08	.09	.047
	1.06			.85		

COSTS PER TASK

The following table gives the charges for the tasks described above in the 4 cases of rate division considered.

	<u>original costs</u>		<u>replacement costs</u>	
	<u>Method A</u>	<u>Method B</u>	<u>Method A</u>	<u>Method B</u>
connect time per hour	\$ 5.24	\$ 4.05	\$ 10.50	\$ 6.75
SCOPE Simulator per hour for CPU and ECS	230.00	281.00	207.00	266.00
null call	.77	.94	.69	.89
null NOMPASS assembly	1.28	1.56	1.15	1.48
large NOMPASS assembly (\$6.70 on A machine)	11.50	14.00	10.04	13.30
Disk storage/month (at 350 hours/month)				
small (800 wds)	.37	.40	.18	.20
medium (20K wds)	9.40	9.90	4.40	5.00
large (120K wds)	56.00	59.00	26.00	30.00

FINAL

After consideration of the above results, we propose that:

- 1) the basic charge rate be \$300 per hour.
- 2) Method B, based on original costs, be used to calculate the rates.
- 3) a negative connect charge be used so that the basic charge for connect time is about 1 or 2 dollars an hour.

These proposals result in the following rates:

CPU	\$200.00 per hour
ECS	0.43 per K-hour
disk	0.07 per K-sector-hour

and the following charges for the tasks considered above:

connect time per hour	\$ 3.01 (not including negative connect fee)
SCOPE Simulator per hour	210.00
null call	.70
null NOMPASS assembly	1.17
large NOMPASS assembly	10.50
Disk storage/month	
small	.30
medium	7.40
large	44.00