May 4, 1967

This paper is a general description of what I think needs to be done to produce a time`sharing operating system for our CDC 6400 with ECS, that will be commensurate with the capabilities of the machine. I have thought about most of these topics for quite a while. For all except the basic monitor and basic input/output, my work estimates are very tentative.

I. Basic Monitor

It will be a time sharing monitor, in that it will handle many simultaneous programs, say several hundred. These programs will be placed into execution upon demand. The demand may come either from some external device or from some program. Large programs will have a sluggish reaction, say several seconds, but small programs can have reactions on the order of milliseconds. Some of these programs will be in PPU's. In fact, all but one of the PPU's will contain only these programs. A program in a PPU, or the currently executing program in CPU, may request service from the monitor. These services will be simple. They will be such that within about 100 micro-seconds after initiation of service, they may be interrupted to initiate service for a higher priority request from some PPU. The types of service will be to request initiation of some program, or to transmit or receive information to or from some other program.

II. Accounting and Resource Allocation

This package will perform two functions. The first will be to control how much of each resource each user gets. The resources controlled will be ECS space, disk space, tape reels, peripheral I/O devices, CPU compute rate and total CPU time. A user will, from time to time, be able to specify his needs, and this package will attempt to supply them. Otherwise the user will have to wait. Secondly, once a user has been allocated a resource, he will be charged for it on a time basis, even if he does not actually use it, since other users will be prevented from using it. Other services will be accounted for at time of use, e.g., we might charge for each disk access or each tape mount.

III. Basic Input/Output System

As all physical I/O must be done in PPU's, a set of programs for PPU's will be needed to drive the devices tied to the system. I envision 1 PPU driving a multiplexor, 1 PPU driving all unit record equipment (i.e., card reader, punch, 2 line printers), 1 PPU driving the disk and 2 PPU's driving the tapes. The latter is optional and is to take advantage of the dual channel connection to the tapes. These drivers will all be simple, will make no assumptions about the kind of data they are handling and will transmit data to and from other programs in the system via the monitor.

IV. Basic File System

This will give the users objects, called files, which will be sequences of computer words. These will be more or less addressable at random. The files will have names given by the user. There will be some sort of reasonably complicated naming system so that users can pass files from one to another, but protect some files from access by others. System files will be just like user files. There will have to be some kind of insurance shceme to help reduce the probability of loss of files, through errors, either by other users, by the software or by the hardware. Small files will be storable in ECS, and all files on disk or tape.

V. Interface current major components of the Scope system to the basic monitor and file system

1. see the major components as being:

   a. users programs
   b. the Fortran compiler, the assembler, and other language processors
   c. the loader
   d. the control card scanner
   e. the read, print and punch packages

   a/b - The most that should be needed is an I/O interface to the basic file system that looks like CIO.

   c - Unfortunately the current loader is written to run in a PPU. I do not believe this is desirable in a time sharing system, (as each PPU could be loading only one job at a time, so at most 10), hence the whole loader will have to be rewritten.

d - This would be similar to the current one, only in the CPU. The only problem is the many services provided by the Scope system. I would like to see those services completely redefined in terms of the basic file system rather than the Scope files.

e - I believe that these would be rather simple CPU programs that call on the basic I/O system to read and write peripheral equipment and the basic file system.

VI. Interactive Programs

This includes such things as a text editor, incremental compilers, machine language debug packages, desk calculators, etc. In general, the sort of things one thinks of when time sharing is mentioned.

VII. Construction Process

A. Define the monitor

This is a critical step. As I envision it, this will determine how the various parts of the system will interact and what kind of response time the system will have. To define the monitor, one has to look at other parts of the system and estimate their requirements. My guess is several months here.

B. Simultaneously construct various versions of the monitor

1. the real ECS version
2. a simulated ECS version for test on our present machine
3. a simulated ECS version for test under Scope
4. a version for use with my PPU simulator

As I envision the monitor, the actual code for each of these should be fairly short, and have a lot in common.

C. Once the monitor has been defined, then work can start on the design of the three major components, resource allocation, I/O system and file system. My guess is that 3 to 6 months to design these, with 2 or 3 good people. More than three people would probably delay completion. I don't know how much time will be needed to program and debug these, as some programming will probably be done during design. Say another 3 to 6 months.

D. After about 3 months of work on C. the interfaces will probably be well enough known so that work can start on changing the Scope packages. I don't anticipate much trouble here except with the loader, as restructure of monitor call and I/O should be quite straight forward. Three or four months, say, for all except the loader.

E. At the same time that work starts on D., work can also start on real interactive packages. So long as these do not require parts of the system not yet constructed, they can be demonstrated to interested persons. These packages require quality work; however, workers on the more fundamental parts should not be diverted. It must be remembered that the system can not be run for extended periods of time until all of parts A, B, C, and D are finished.

Howard Sturgis