October 11, 1969

Progress Report on 6400 CAL Time-Sharing System

Jim Gray

In order to understand the status of TSS, it is necessary to understand its architecture. TSS is built up in three layers called the ECS layer, the disk layer and the executive layer. The ECS layer is the core of the system. Its design has strong implications for all higher layers. Its function is to create, manage and destroy objects in ECS and to provide protection walls and and communication paths between processes and other TSS objects. It also includes the process scheduler and the ECS-CM swapper. The disk layer reflects ECS files up into the disk store. It provides facilities for creating, managing and destroying disk files as well as opening and closing them. The executive consists of a command processor, log in-log out procedures, accounting routines and a directory system. Its duties are comparable to those at SCOPE except that the objects that it manipulates are the disk/ECS objects created by the low-level systems. Compilers, interpreters, editors and user-constructed subsystems run "on top of" the exec just as the exec runs "on top of" the disk system.

Currently the ECS system is operative. About four months of work and an equal amount of documentation remain to be done on it. There is a provisional executive program running on top of the ECS system allowing TSS to be written on itself (see Figure 1). Currently TSS has enough CPU to support 60 systems programmers (or about 150 ordinary users). However, there is only enough ECS for about 10 active processes. There are 6 teletypes connected to TSS. We are confident that TSS will gracefully support 100 student users when it is complete.

The design of the disk system is almost complete. Implementation has begun recently and should be complete by Feb. 1. This project is in series with a disk driver program which will be available in mid-December. With the advent of the disk system, a new porivisional executive will be written. At that point TSS will be able to support many (~ 60) users. We plan to offer TSS to persons who can provide their own teletypes and who are developing subsystems for TSS (e.g., Basic, CAL, APL, FORTRAN, ...). A manual on the system is being prepared for this eventuality.

The executive is in the preliminary design stages. A reasonable guess of its delivery date is mid-summer 1970.

A background batch system is in development. It will run simple SCOPE jobs (no tapes) and will be SCOPE-compatible. It requires routines to drive card readers and printers, a display driver and a dayfile generator. Almost all other work to interface SCOPE with TSS is done in the SCOPE simulator now running.

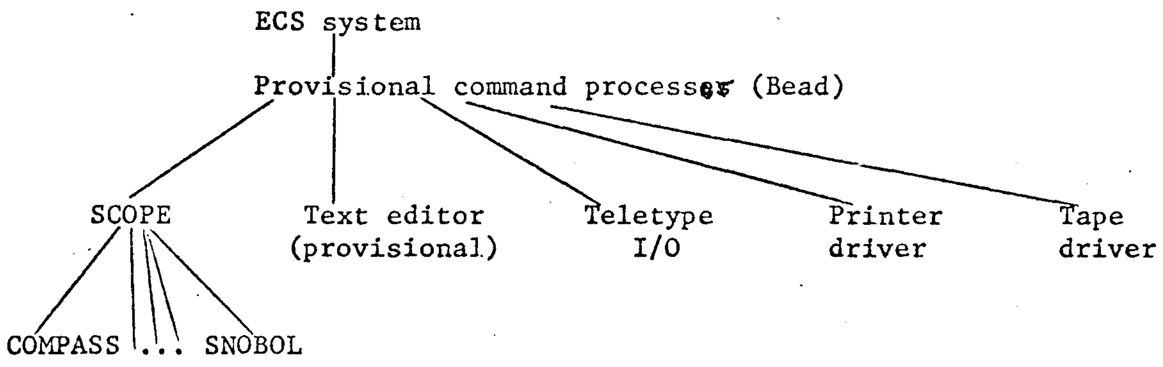
To facilitate systems programming one software subsystem (not part of TSS) is being implemented. It is an assembler/debugger called Cool Aid. The assembler has an Algol syntax and an elegant macro-facility. It is designed to be very fast (~ 10 times faster than Compass) and compact, and is re-entrant. It will feed a loader which is SCOPE-compatible. There will be a run time interactive debugger which will allow the teletype to examine and modify (symbolically) a running program without complete reassembly.

Also in development is a sophisticated editor. Members of the CS and EECS departments are supervising the development of a BASIC and an APL.

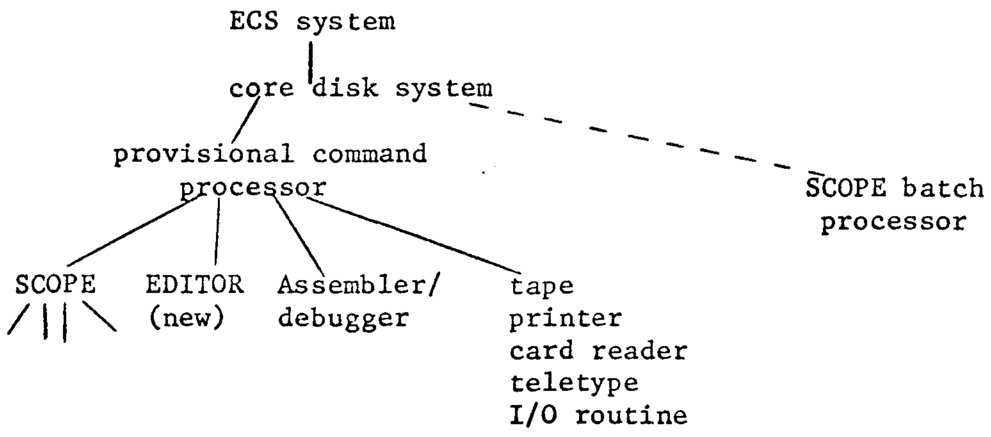
I plan to implement a JOSS-like language next spring (with the help of CS undergraduates) and to supervise FORTRAN and ALGOL syntax checkers at that time.

The developers of Cool Aid have expressed an interest in producing an interactive SNOBOL 4.

Current Status (October 10, 1969)



February 28, 1970



The current personnel allocation is

Malbrain McJones	Debugger/assembler (Feb. 28)
Redell Bentley Vaughan	Complete and document ECS system (Jan. 1)
Lampson Lindsey Morris Redell Sturgis	Design executive system (mid-summer)
Lindsey Redell	Implement core disk (Feb. 1)
Sturgis	Implement disk driver (Jan. 1)
Gray	Design and Implement editor (Jan. 1)
Standiford	Implement batch system (April 1)