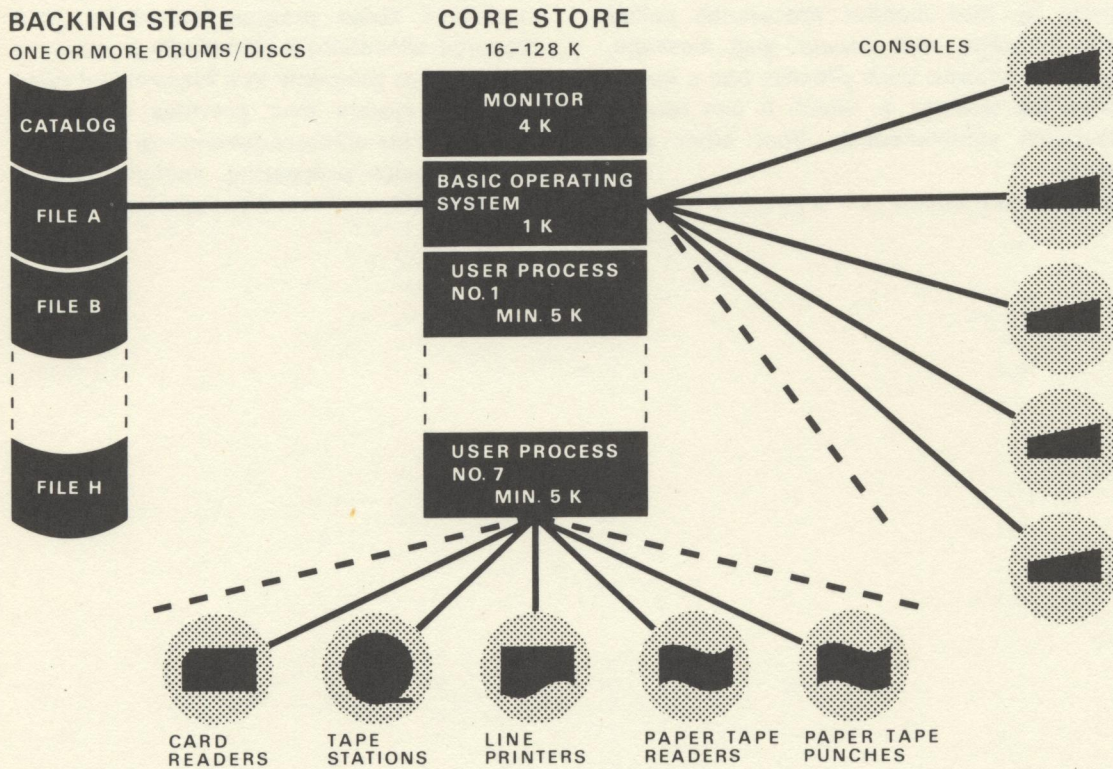


## MONITOR 1



This specification describes the time-sharing system developed for the extended RC 4000 computer with drum and/or disc as backing store. The monitor controls the parallel execution of programs in the core store and the initiation of all input/output.

### Program Configuration

After initial system loading, the core store contains the monitor and a basic operating system, which allows the operator to load and start programs from the backing store.

The running programs, including the basic operating system, are executed in a time-sharing scheme, in which each program in turn is allotted a maximum of 20 milliseconds of computing time.

### Monitor Functions

- The monitor controls the following functions:
- initiation, execution, and termination of parallel programs
  - interprogram communication

```

SOFTWARE
begin
length:= 3
end
end EOF proc;
open (master,
open (new_maste
open (transacti
comment
lurec (master,
lurec (transac
next:
if master (1)
begin comment
newrec (new
new_master (
new_master (
new_master (
lurec (transa
go_to next
end 5;
if master (1)
begin comment
newrec (new
for i:= 1 st
new_master
lurec (maste
go_to next
end 7;
if master (1)
begin comment
master (2);re
lurec (transa
go_to next
end;
comment
close (master,
close (transa
end;

```



- console communication
- reservation and initiation of sequential input/output
- creation, updating, and removal of common data areas on the backing store.

Programs can be written in any of the available programming languages (Slang, Algol, Fortran). The storage protection system guarantees non-interference among 7 parallel user programs, but it is possible to start up to 22 programs provided they are error free. A running program occupies a contiguous storage area with a unique protection key.

Programs and peripheral devices are identified by symbolic names. All communication between parallel processes is handled uniformly by four monitor procedures called send message, wait answer, wait message, and send answer. Each process has a queue within the monitor, in which it can receive messages simultaneously from other processes.

Using these procedures it is possible to trans-

fer parameters from one running program to another. It is also possible for any program to open a conversation with any typewriter console and vice versa. Finally, programs can send messages to sequential input/output devices, which initiate data transfers to and from the core store.

Users can retain programs and data in a semi-permanent manner on the backing store, which is organized as a collection of named data areas. A fixed part of the backing store is set aside for a catalog describing the names and locations of data areas. Each entry in the catalog is supplied with a protection key, which protects the corresponding data area against unauthorized updating.

The time-sharing monitor has no built-in assumptions about program scheduling and resource allocation; it allows any program to initiate other programs in a hierarchical manner. The system thus provides a general framework for different scheduling strategies, such as batch processing, multiple console conversation, and real-time scheduling.