

To: BBN TEN-SYS Group  
From: TRS  
Subject: The TEN-SYS EXECUTIVE  
Date: 24 March 1969

---

### Introduction

The EXECUTIVE is the user's handle on the time sharing system. It is a program which translates conversational user commands into monitor calls. The TEN-SYS EXECUTIVE will be designed and implemented to run at multiple places in a job-fork structure which will permit the user to have several EXEC's running for him. This will permit the user to have quite general control over several processes from the same teletype terminal.

#### 1. COMMAND AND FILE NAME RECOGNITION

A modified "command recognition" scheme will be used in TEN-SYS for anticipating a full name when a unique subset of the name is presented. The scheme will permit over-typing and will allow the user to select whether he wants the remainder of the name (the first part of which he presented) to be typed out.

The user will type as much of the name as he thinks is unique. He will then type an explicit terminator which will be either alt-mode, Space, EOT or carriage return.

#### ALT-MODE Termination

The alt-mode terminator means to determine if the presented string is unique. If it is, the system will supply the remainder of the string. If it is not unique, the system types G<sup>c</sup> (BELL) and waits for the user to present more of the string.

#### SPACE, EOT, or CARRIAGE RETURN TERMINATION

The above terminators result in the echo of the character (except EOT) as a format affector and tell the system to determine if the presented string is unique. If it is, the system does not echo the remainder of the string. If it is not unique, the system types G<sup>c</sup> (BELL) and waits for the user to present more of the string.

Examples:

COP ⊕ Y FROM AB ⊕ DCEF TO TERM ⊕ INAL ↘

or

COP<sub>Δ</sub> FROM AB<sub>Δ</sub> TO TERM ↘

or

COP (EOT) FROM AB (EOT) TO TERM ↘

## 2. SUBSYSTEM SPECIFICATION

The names of all subsystems will exist in the file directory of SUBSYS. When the user presents a string at the beginning of a line to the EXEC, it is potentially either a command name or a call to a subsystem. If the string is recognized as a unique command name or a subset of several command names, the system knows the string is an EXEC command. If the string is not recognized as a command name or a subset of several command names, the directory of SUBSYS will be searched for a possible subsystem name match. If such a name match (or unique subset of a name) is found, the subsystem will be called by the EXEC.

## 3. NAME UNIQUENESS

The system will not permit the user to create a file name which is a subset of an existing name in a file directory. This policy will eliminate many system and user problems associated with specifying such names. It will also be illegal to create an EXEC command name which is the same as or a subset of a subsystem name.

## 4. COMMANDS FROM A FILE AND BATCH PROCESSING

The EXECUTIVE will have the capability of accepting commands from any file and of putting its normal terminal output on any file. This and other capabilities will be used for pre-specifying in a file a list of operations to be performed.

While the EXECUTIVE is accepting commands from a non-conversational (i.e., not the TERMINAL) file, it will be transmitting the appropriate arguments from this file to all subsystems and programs which the file calls. The EXECUTIVE will retain control over reading this input file by opening it as a "subroutine file" (c.f. 940 subroutine files) which means every call for input from this

file will trap to a subroutine in the EXEC. This subroutine will "echo" every character from the input file to the specified file for output. This will cause the output file to contain an "intact" dialogue of the run as though the user had typed in all the commands from his terminal.

The EXECUTIVE will take special action when errors are detected in the job run when input is from a non-conversational file. Anytime the file specifies an action which results in a detected error the job will be stopped, a message will be typed on either the job attached terminal or the operator's console if the job is detached. The job will be resumed only after corrective action or an override is specified. When a subsystem or program detects an error which should stop a non-conversational run, it will execute a particular monitor call which will invoke the EXEC and cause the above action if input was from a non-conversational file.

The commands from any file and jobs detached from terminal features will supply most of the requirements of batch processing. The remaining requirements are the necessity for specifying job priority (for scheduling) and the necessity of specifying limitations of the job's use of system resources. The EXEC will provide commands for specifying these parameters to the system through its accounting features described in part 6 of this memo.

## 5. CONCISE COMMAND LANGUAGE

The concise command language features like the DEC 10/50 monitor CCL will not be implemented in the initial MINI-SYSTEM but will be available in TEN-SYS. These features will be implemented much like the DEC CCL implementation except that the command compiler will be a part of the EXEC. The command compiler will call subsystems with the input file specified as a subroutine file so that calls for input will be trapped to the EXEC which will supply the composed strings as arguments. This will be superior to the DEC implementation which writes the composed strings into specific named files for each subsystem. The DEC implementation requires a lot of extraneous I/O and opening/closing of files.

A general, macro EXEC command capability is not needed in the system and will not be implemented. It will not be possible for users to define EXEC commands or concise commands.

## 6. ACCOUNTING FEATURES

The EXEC will perform all system accounting and will maintain the user to system access file.

As users enter the system, exit the system, specify new job numbers, use system resources, etc., the EXEC will make entries in a system LOGFILE. This file will be processed by an accounting subsystem for charge purpose. The EXEC also periodically will sample Monitor maintained statistics (such as AP time used, Core page time used, DP and HP time used) and record these numbers in this LOGFILE. This periodic sampling will be initiated by a process set up when the user enters the system. This process will also be activated whenever a user attempts to exceed his system access rights—to permit operator intervention or to save the user's environment and then exit the user from the system (e.g., if the user attempts to exceed allocated AP time).

The user to system access information will be in the user directory. There will be a master set and a working set of access information for each user. Generally the working set will be identical to the master set, but it will be possible for authorized users to restrict the working access to a subset of the master access permits. This capability is provided to allow a user to restrict users of his account to time dependent access to system resources. When a user enters the system, the working set access rights are copied into the TS block for rapid checks of user to system access. If an authorized user alters the working access rights of an on-line user, these new rights are compared (for compatibility) with the current system access of the on-line user and, where compatible, will replace his current access permits.

## 7. COMMAND CATEGORIES

The categories of commands will include:

- a) commands which permit users to enter or leave the system.
- b) commands which permit users to manipulate files by copying them from one form to another, deleting files, backing up files, renaming files, changing file sharing permits, etc.
- c) File directory commands for listing entire or subsets of directories, attaching to another file directory, etc.
- d) Subsystem and program calling commands.

- e) Commands for saving and/or restoring all or a portion of an environment.
- f) Commands for manipulating the job terminal to attach and detach it from the job, link its output to other terminals, change the terminal "type" specification.
- g) Commands which change the users' access rights to the system.
- h) Commands for initiating "batch" jobs specifying non-conversational command input.
- i) Concise commands for abbreviating command input to the EXEC by specifying one command which invokes the action of several commands.
- j) System maintenance commands which permit examining and/or changing the Monitor or EXEC, printing the state of crucial tables for the running system or for the "saved environment" file of a previously running (crashed or misbehaving) system, etc.

Many commands (or certain options of commands) are not accessible to the average user.

## 8. PANICS

When a super panic is typed or when a job is re-attached to a teletype, the entire job/fork structure is suspended or frozen. All running forks are stopped. Any initiated I/O action will proceed to completion but will not cause any fork of this job to resume. During the freeze, the user will be typing to the top level EXEC. Since the top level EXEC will in general have some state information important to the "frozen" fork structure, the previous state will be "pushed." The user may initiate other forks, kill the entire fork structure, or even resume the structure.

Since the super panic procedure described above is likely to result in proliferation of forks and memory pages, the system will impose some restrictions on the allocation of memory space and forks. The restrictions will have upper bounds stated in the user to system access specs but will permit the user to specify lower bounds to eliminate "run-away" structure like the 940 MACHINE SIZE commands. Unlike the 940 system, however, when the user attempts to exceed one of his preset lower bounds, the system will ask him to indicate at his terminal if it should override or even reset this bound or abort. The system will never let the user exceed any of the user to system access bounds unless systems personnel intervene.