

AD706104

11

THE UNIVERSITY OF MICHIGAN



Memorandum 25 ✓

CONCOMP

February 1970

MODIFICATIONS TO THE SEL EXECUTIVE SYSTEM

James H. Jackson
James F. Blinn

This document has been approved for public release and sale; its distribution is unlimited.

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

DDDC
RECEIVED
MAY 22 1970
G

THE UNIVERSITY OF MICHIGAN

Memorandum 25

MODIFICATIONS TO THE SEL EXECUTIVE SYSTEM

**James H. Jackson
James F. Blinn**

**CONCOMP: Research in Conversational Use of Computers
ORA Project 07449
F. H. Westervelt, Director**

supported by:

**DEPARTMENT OF DEFENSE
ADVANCED RESEARCH PROJECTS AGENCY
WASHINGTON, D. C.**

**CONTRACT NO. DA-49-083 OSA-3050
ARPA ORDER NO. 716**

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

February 1970

ABSTRACT

Some modifications to the SEL Executive System are described which improve the performance of the system, but which do not change the set of system subroutines, the entry point to any system subroutine, or the calling sequence for any system subroutine. User programs which run with the original version of the system need not be changed in order to run with the new version of the system.

TABLE OF CONTENTS

ABSTRACT	iii
1. Introduction	1
2. Alphanumeric Tape Format	2
3. Response to Empty Task Queue	4
4. Display Structure Modification	5

BLANK PAGE

1. INTRODUCTION

This paper describes several modifications which have been made to the SFL Executive System since it was documented [1]. These modifications were intended to improve the performance of the system, rather than to extend the set of operations which it performs. Consequently, neither the vector of system subroutine entry points nor the calling sequence to any system subroutine has been changed. User programs which run with the original version of the system (Version 1) need not be changed in order to run with the new version of the system (Version 2).

2. ALPHANUMERIC TAPE FORMAT

Each 8-bit alphanumeric tape image for use with the system originally consisted of two control bits (11_2) followed by a 6-bit character code [1 pp. 23-24]. An alphanumeric record was a sequence of these tape images, followed by an end-of-record character (000_8). This format was adopted because, through examination of the control bits, a program (e.g., a copy function in the idle-time task) could easily determine whether it was reading an alphanumeric record or a binary record. In this way, the system was protected against the user's inadvertent attempt to load an alphanumeric record. With this protection provided, interspersing descriptive text with binary records was practical.

Since the ability to list alphanumeric tapes on an off-line teletype is desirable, the alphanumeric tape format has been changed from the 6-bit code format to an ASCII code format. The only changes to the system which were required by this change of format were in the copy functions of the idle-time task which involve reading or punching of alphanumeric tape. Each record in the new alphanumeric tape format consists of a sequence of ASCII characters with the parity bit punched, followed by a form feed (214_8), a rubout (377_8), and ten blank tape images. If each alphanumeric record is considered to represent one page of text, this format is suitable for listing on a model ASR 35 teletype with one form feed stop.

(The rubout at the end of each record facilitates recognition of the last record read as being alphanumeric in order to prevent concatenation of alphanumeric records when copying paper tape with the idle-time task.)

A record can still be identified as being alphanumeric or binary before it is used, even with this modification in format. The parity bit of each type image in an alphanumeric record is set to one. However, since the first three tape images in a binary record must represent an origin, the first tape image in a binary record has this bit set to zero. Consequently, whether a record is alphanumeric or binary can be determined by examining the parity bit of its first tape image.

3. RESPONSE TO EMPTY TASK QUEUE

Whenever all user tasks in the original system terminated and all output buffers became empty, the system was reinitialized and the idle-time task was scheduled [1 pp. 5-6]. Since any IBM 360 program which was running in conjunction with the terminal was interrupted by this process, this program had to be restarted after each normal termination of a set of user tasks. Furthermore, since the data-phone is a half-duplex device, a record inbound from the IBM 360 could easily be lost when the interrupt was sent to that machine. In addition, this reinitialization process required the user to wait until all output buffers were empty before he could begin typing commands to the idle-time task.

Since an empty task queue does not normally represent a programming or operating error, the system is intact when this condition is detected. Consequently, there is no need to reinitialize the system when the task queue becomes empty. With this in mind, the restrictions mentioned above have been removed via a system modification which causes the idle-time task to be started without reinitializing the system when the task queue becomes empty.

4. DISPLAY STRUCTURE MODIFICATION

Since the display structure is a program which is executed asynchronously with the PDP-9 program by the display control, the system must insure that the display control is not executing a level when certain changes are made to that level. If no such precautions were taken, two problems could occur. First, if the coordinates of a level were changed while the display control was executing that level, the coordinates of the rest of the picture would be offset for the remainder of that frame. This problem would occur because the display control would use the old coordinates from the head of the level to position the beam, but would restore the beam by using the new coordinates from the tail of the level. Secondly, if a section of the display structure was removed from a level and the storage which it occupied was released while the display control was still executing it, subsequent use of that storage by some other part of the PDP-9 program could destroy this section of the display structure while the display control was still executing it. In the original system, these problems were avoided by waiting until the end of the frame to change the coordinates of a level or to return from a system subroutine which removed part of the display structure from a level. However, this delay was usually longer than necessary for active levels, and it was unnecessary for non-active levels.

A new procedure, which allows these modifications to be made to a level whenever the display control is not executing it, has been employed in the new version of the system. This procedure determines whether or not a level is being executed by the display control by examining the push-down stack. If the address of a call to the level is not found in the push-down stack, the level is not being executed; otherwise, the level is being executed. However, because the display control operates asynchronously with the PDP-9, it can call the level after the stack has been examined, but before the level is modified. For this reason, the following combination of an interruptible program procedure and an interrupt service is used to modify the level:

- (1) Store a display internal stop instruction at the beginning of the level to be modified.
- (2) Determine whether or not the interrupt service has modified the level. If it has, terminate the operation. Otherwise, proceed with Step 3.
- (3) Examine the push-down stack. If no call to the level is found, modify the level, remove the internal stop instruction, and terminate the operation. Otherwise, proceed with Step 2.

Internal Stop Interrupt Service:

- (1) Modify the level and remove the internal stop instruction from it.

(2) Restart the display at the beginning of the level.

With this procedure, either the interruptable program or the interrupt service modifies the level. The reader should note that non-active levels are modified almost immediately.

REFERENCE

1. Jackson, J. H., An Executive System for a DEC 339 Computer Display Terminal, Technical Report 15, Concomp Project; also SEL-32, Systems Engineering Laboratory, University of Michigan, Ann Arbor, Michigan, December 1968.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract, and inclusive dates must be of the same classification as the overall report to which classified)

1. ORIGINATING ACTIVITY (Corporate author) THE UNIVERSITY OF MICHIGAN COMCOMP PROJECT		2a. REPORT SECURITY CLASSIFICATION Unclassified	
		2b. GROUP	
3. REPORT TITLE MODIFICATIONS TO THE SEL EXECUTIVE SYSTEM			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Memorandum 25			
5. AUTHOR(S) (First name, middle initial, last name) JAMES H. JACKSON and JAMES F. BLINN			
6. REPORT DATE February 1970		7a. TOTAL NO. OF PAGES 8	7b. NO. OF REFS 1
8a. CONTRACT OR GRANT NO. DA-49-083 OSA-3050		8b. ORIGINATOR'S REPORT NUMBER(S) Memorandum 25	
9a. PROJECT NO.		9b. OTHER REPORT OR(S) (Any other numbers that may be assigned this report)	
10. DISTRIBUTION STATEMENT Qualified requesters may obtain copies of this report from DDC.			
11. SUPPLEMENTARY NOTES		12. SPONSORING MILITARY ACTIVITY Advanced Research Projects Agency	
13. ABSTRACT Some modifications to the SEL Executive System are described which improve the performance of the system, but which do not change the set of system subroutines, the entry point to any system subroutine, or the calling sequence for any system subroutine. User programs which run with the original version of the system need not be changed in order to run with the new version of the system.			

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
SEL Executive System interactive computer graphics remote display terminal DEC 339						