MOSN - 11.3

To: Distribution

From: Bill Silver

Date: November 13, 1973

Subject: Changes to the Operation of the Operator's Console

When the new operator's console and syserr software is installed the mechanics of operating the operator's console will change. The major areas of change are:

1. Use of the operator's console will no longer interfere with the normal functioning of Multics.

2. The console timer switch should always be ENABLED.

3. The use of the console in Multics has changed.

4. The use of the console in BOS has changed.

5. Some changes have been made to the format of syserr messages.

6. The conversion of characters from BCD to ASCII and from ASCII to BCD has changed.

EFFECTS OF CHANGE ON SYSTEM PERFORMANCE

Previously any use of the operator's console required the dedicated services of one processor. Its use on service meant that for many on-line users the system would appear stopped. There was even the real danger that the system would crash because it was waiting for the operator's console to complete an input or output operation.

This is no longer true. Except when an unusual burst of syserr messages must be typed the system will not wait for the operator's console to complete an operation. Under normal circumstances the operator's console can be used on service without any adverse affect on system performance or system reliability.

USE OF CONSOLE THIRTY-SECOND TIMER

The operator's console has a 30 second timer mechanism. When reading input if no character is typed within 30 seconds the read operation will be terminated. This mecahnism is controlled by a switch in the maintenance panel of the operator's console. Currently this switch is set to the disabled position and thus the 30 second timer is not used. With the installation of the new operator's console software this switch should be set to the enabled position during the operation of both Multics and BOS. The effects which the use of the 30 second timer will have on the operation of the console are discussed in detail below.

USE OF THE CONSOLE IN MULTICS

This discussion of the operation of the console applies only when Multics is running. Except where specifically stated the operation of the console while BOS is running has not changed. In order to help describe the new method of using the console we will define two modes in which the console will operate:

LOCK MODE - The console keyboard is locked and cannot be used to type input. The console remains in lock mode until the user hits the Request button.

UNLOCK MODE - The console is either typing output or the keyboard is unlocked and ready to accept input.

When the console keyboard is unlocked the user may start typing input. He still must hit the End of Message button to signal that he has completed an input line. When the system has processed that input line and is ready to accept another the console keyboard will be automatically unlocked. The user can type the next input line without first hitting the Request button. In unlock mode the Request button has another use. Output generated as a result of an input line (for example a "list" command) may be stopped by hitting the Request button.

There are two important points to note about using the Request button to suppress unwanted output:

1. syserr messages cannot be suppressed.

2. An idiosyncrasy of the operator's console results in the following phenomenon. When the Request button is pushed the line being typed will finish being typed. In addition one more line of output (if there is one) will also be typed.

Once the console is in unlock mode the keyboard will continue to be unlocked after each read operation has completed and the system is ready to accept another input line. Since it is not desirable (expecially on service) to have the console keyboard unlocked for a long period of time there must be a way to return the console to lock mode once the user has no more input lines to type. This can be accomplished by inputting a null line, i.e., hit the End of Message button without typing any other character.

There are other ways in which the console may be taken out of unlock mode and put back in lock mode. One involves the 30 second timer. Once the console keyboard is unlocked and ready to accept input, if the user does not type at least one character every 30 seconds, the 30 second timer will go off. The read operation will be terminated, any input already typed will be lost, and the console will be placed in lock mode. The same actions may be taken if the user does any of the following:

1. types an illegal character (for example an upper case A, this only works sometimes),
2. turns the ON/OFF button off and then on again (unless you do it very fast), or
3. turns the POWER button in the onsole maintenance panel off and then on again (unless you do it very fast.)

The maximum number of input characters that can be typed in one line is 84. If this number is exceeded or if the Operator Input Error button is pushed the read operation will be terminated. All input typed in the current line will be discarded. The console will remain in the unlock mode - the keyboard will be unlocked.

A summary of the new features involved in using the operator console is given in Table I.

USE OF THE CONSOLE IN BOS

Note, when running in BOS if the 30 second timer goes off the console keyboard will be locked. It will remain locked until the Request button is pushed. This is the same way the 30 second timer works when Multics is running. This is the only change to the way the console is operated when running in BOS.

SYSERR MESSAGE FORMAT

The format of all syserr messages has changed slightly. The current time that is prefixed to each syserr message now includes tenths of a minute. The new format of the time field is:

"hhmm.t"

where: hh = hours
mm = minutes
t = tenths of a minute.

The new operator's console and syserr software provides a mechanism which can maintain a log of all syserr messages. It is possible for the wired syserr log buffer to become full and for some syserr messages to be lost from the log. syserr messages which are lost from the log may be typed with the following data prefixed before the time field:

"lost xxxxxx, z"

where xxxxxx is the sequence number of the syserr message, and
z is the syserr code of this message.

TRANSLATION and CANONICALIZATION

The current operator's console reads and writes only GEBCD characters. The standard format of all characters in Multics is ASCII. The new operator's console software provides a slightly different translation between these two character sets. In addition it provides some canonicalization of input.

The rules of translation are:

1. The following ASCII characters correspond to a specific BCD character:

   a) all lower case letters,

   b) all numeric digits,

   c) the characters: blank ! " $ & ' ( ) * + , - . ? : ; < = > ? [ ] _ |

2. The following ASCII characters correspond to a specific BCD character preceded by the BCD escape character "\":

   a) all upper case letters,

   b) the characters: tab # @ \ ` { } ~ ^

3. All other ASCII characters must be expressed in BCD in octal format: "\xxx" where xxx represents the octal value of the ASCII character.

4. Any ASCII character can be input in octal format.

5. For all but a few exceptions the representation of an ASCII character as output will be exactly the same as the standard way (nonoctal) it is represented as input. The exceptions

are:

ASCII CHAR	BCD INPUT	BCD OUTPUT
octal 006	\006	ignored
octal 007	\007	ignored
tab	\blank	tab
RRS	\016	ignored
BRS	\017	ignored
#	\#	#
@	\@	@
octal 177	\177	ignored

6. Table II shows the BCD input needed to express each ASCII character and it shows the BCD output representation of each ASCII character.

The rules of canonicalization:

1. A new_line character is appended to every input line.

2. All blanks and tabs are stripped off the end of the input line. Note, in order to input a line that contains just a new_line character you must type a line containing just blanks and tabs. One blank is sufficient. Hitting the End of Message button before typing any characters will produce no input line at all. This action will place the console in lock mode.

3. NO canonicalization of back_space characters is performed.

4. The character "#" is used as the erase character. The previous character will be deleted from the input line. For example "###" will cause the previous three characters to be deleted.

5. The character "@" is used as the kill character. All the characters typed so far for this line will be deleted.

6. The maximum size of an output line is 80 characters. If a line to be typed is greater than 80 characters it will be continued on the next line. The first character printed on the continued line will be a "b̸". Note, the "b̸" character is also printed automatically when the End of Message button is pushed to signal the end of input. Other than the two cases just mentioned there is no way the "b̸" character can be typed. It is not part of either the BCD or ASCII character set.

TABLE I

Summary of How to Use the Operator's Console

Use of the Request Button:

MODE	RESULT from HITTING REQUEST
lock	console enters unlock mode
unlock	output is suppressed

Ways to Get Out of Unlock Mode:

1. Input a null line - hit End of Message button before typing any other character.

2. Wait - 30 second timer will go off.

3. Input an illegal character.

4. Turn off and then on the ON/OFF button.

5. Turn off and then on the POWER button in the console maintenance panel.

How to Input a Line that Has Just a New_Line Character:

Type a line of just blanks or tabs.

Table II

BCD → ASCII → BCD

	BCD INPUT	ASCII	BCD OUTPUT		BCD INPUT	ASCII	BCD OUTPUT
000	\000		\000	040	blank	blank	blank
001	\001		\001	041	!	!	!
002	\002		\002	042	"	"	"
003	\003		\003	043	\#	#	#
004	\004		\004	044	$	$	$
005	\005		\005	045	%	%	%
006	\006		ignored	046	&	&	&
007	\007	BEL	ignored	047	'	'	'
010	\010	BS	\010	050	(	(	(
011	\blank	HT	HT	051	)	)	)
012	\012	NL	CR	052	*	*	*
013	\013	VT	\013	053	+	+	+
014	\014	NP	\014	054	,	,	,
015	\015	CR	\015	055	-	-	-
016	\016	RRS	ignored	056	.	.	.
017	\017	BRS	ignored	057	/	/	/
020	\020		\020	060	0	0	0
021	\021		\021	061	1	1	1
022	\022	HLF	\022	062	2	2	2
023	\023		\023	063	3	3	3
024	\024	HLR	\024	064	4	4	4
025	\025		\025	065	5	5	5
026	\026		\026	066	6	6	6
027	\027		\027	067	7	7	7
030	\030		\030	070	8	8	8
031	\031		\031	071	9	9	9
032	\032		\032	072	:	:	:
033	\033	mc	\033	073	;	;	;
034	\034		\034	074	<	<	<
035	\035		\035	075	=	=	=
036	\036		\036	076	>	>	>
037	\037	EGM	\037	077	?	?	?

	BCD INPUT	ASCII	BCD OUTPUT
100	\@	@	\@
101	\A	A	\A
102	\B	B	\B
103	\C	C	\C
104	\D	D	\D
105	\E	E	\E
106	\F	F	\F
107	\G	G	\G
110	\H	H	\H
111	\I	I	\I
112	\J	J	\J
113	\K	K	\K
114	\L	L	\L
115	\M	M	\M
116	\N	N	\N
117	\O	O	\O
120	\P	P	\P
121	\Q	Q	\Q
122	\R	R	\R
123	\S	S	\S
124	\T	T	\T
125	\U	U	\U
126	\V	V	\V
127	\W	W	\W
130	\X	X	\X
131	\Y	Y	\Y
132	\Z	Z	\Z
133	[	[	[
134	\\	\	\\
135	]	]	]
136	\↑	^	\↑
137	←	_	←

	BCD INPUT	ASCII	BCD OUTPUT
140	\'	`	\'
141	A	a	A
142	B	b	B
143	C	c	C
144	D	d	D
145	E	e	E
146	F	f	F
147	G	g	G
150	H	h	H
151	I	i	I
152	J	j	J
153	K	k	K
154	L	l	L
155	M	m	M
156	N	n	N
157	O	o	O
160	P	p	P
161	Q	q	Q
162	R	r	R
163	S	s	S
164	T	t	T
165	U	u	U
166	V	v	V
167	W	w	W
170	X	x	X
171	Y	y	Y
172	Z	z	Z
173	\(	{	\(
174	↑	|	↑
175	\)	}	\)
176	\"	~	\"
177	\177 PAD ignored		