NAME

console, keyboard, log − system console

DESCRIPTION

The TTY device driver manages two devices related to the main user interface, the system screen and the keyboard. These two together are named "the Console".

The Screen

The screen of a PC can be managed by a Monochrome Display Adapter, a Hercules card, a Color Graphics Adapter, an Enhanced Graphics Adapter, or a Video Graphics Array. To the console driver these devices are seen as a block of video memory into which characters can be written to be displayed, an I/O register that sets the video memory origin to the character that is to be displayed on the top-left position of the screen, and an I/O register that sets the position of the hardware cursor. Each character within video memory is a two-byte word. The low byte is the character code, and the high byte is the "attribute byte", a set of bits that controls the way the character is displayed, character and background colours for a colour card, or intensity/underline/reverse video for monochrome.

These are the characteristics of the adapters in text mode:

Adapter	Usable memory	Mono/Colour
MDA	4K	M
Hercules	4K	M
CGA	16K	C
EGA	32K	M or C
VGA	32K	M or C

MDA and Hercules are the same to the console driver, because the graphics mode of the Hercules is of no use to Minix. EGA and VGA are also mostly seen as the same in text mode. An EGA adapter is either a monochrome or a colour device depending on the screen attached to it. A VGA adapter can run in either monochrome or colour (grayscale) mode depending on how the Boot Monitor has initialized it.

The driver uses the video origin to avoid copying the screen contents when scrolling up or down. Instead the origin is simply moved one line. This is named "hardware scrolling", as opposed to copying memory: "software scrolling".

The video origin is also used to implement several virtual consoles inside the video memory of the adapter. Each virtual console gets a segment of video memory. The driver chooses which console to display by moving the video origin. Note that an MDA or Hercules adapter can only support one console. CGA can support up to four 80x25 consoles, and EGA and VGA can have eight. It is best to configure one less console to leave some video memory free so that hardware scrolling has some space to work in.

Character codes are used as indices into a display font that is stored in the adapter. The default font is the IBM character set, which is an ASCII character set in the low 128 codes, and a number of mathematical, greek, silly graphics, and accented characters in the upper 128 codes. This font is fixed in the MDA, Hercules and CGA adapters, but can be replaced by a user selected font for the EGA and VGA adapters.

A number of control characters and escape sequences are implemented by the driver. The result is upward compatible with the ANSI standard terminal. The **termcap**(5) type is **minix**. Normal characters written to the console are displayed at the cursor position and the cursor is advanced one column to the right. If an entire line is filled then the cursor wraps to the first column of the next line when the next character must be displayed. The screen is scrolled up if needed to start a new line. Some characters have special effects when sent to the console. Some even have arguments in the form of comma separated decimal numbers. These numbers default to the lowest possible value when omitted. The top-left character is at position (1, 1). The following control characters and escape sequences are implemented by the console:

Sequence	Name	Function
^@	Null	Ignored (padding character)

ˆG	Bell	Produce a short tone from the speaker
ˆH	Backspace	Move the cursor back one column, wrapping from the left edge up one line to the right edge
ˆI	Horizontal Tab	Move to the next tab stop, with each tab stop at columns 1, 9, 25, etc. Wrap to the next line if necessary.
ˆJ	Line Feed	Move one line down, scrolling the screen up if necessary
ˆK	Vertical Tab	Same as LF
ˆL	Form Feed	Same as LF
ˆM	Carriage Return	Move to column 1
ˆ[	Escape	Start of an escape sequence
ˆ[M	Reverse Index	Move one line up, scrolling the screen down if necessary
ˆ[[*n*A	Cursor Up	Move the cursor up *n* lines
ˆ[[*n*B	Cursor Down	Move the cursor down *n* lines
ˆ[[*n*C	Cursor Forward	Move the cursor right *n* columns
ˆ[[*n*D	Cursor Backward	Move the cursor left *n* columns
ˆ[[*m*;*n*H	Cursor Position	Move the cursor to line *m*, column *n*
ˆ[[*s*J	Erase in Display	Clear characters as follows: *s* = 0: From cursor to end of screen *s* = 1: From start of screen to cursor *s* = 2: Entire screen
ˆ[[*s*K	Erase in Line	Clear characters as follows: *s* = 0: From cursor to end of line *s* = 1: From start of line to cursor *s* = 2: Entire line
ˆ[[*n*L	Insert Lines	Insert *n* blank lines
ˆ[[*n*M	Delete Lines	Delete *n* lines
ˆ[[*n*@	Insert Characters	Insert *n* blank characters
ˆ[[*n*P	Delete Characters	Delete *n* characters
ˆ[[*n*m	Character Attribute	Set character attribute as follows: *n* = 0: Normal (default) attribute *n* = 1: Bold (high intensity fg colour) *n* = 4: Underline (mono) / Cyan (colour) *n* = 5: Blinking *n* = 7: Reverse Video *n* = 30: Black foreground colour *n* = 31: Red *n* = 32: Green *n* = 33: Brown *n* = 34: Blue *n* = 35: Magenta *n* = 36: Cyan *n* = 37: Light Gray *n* = 39: Default fg colour (lt gray) *n* = 40−47: Same for background colour *n* = 49: Default bg colour (black) Note: The "bold" versions of black, brown and lt gray become dark gray, yellow and white.

The console device implements the following ioctl to copy a font into font memory on EGA and VGA adapters:

ioctl(*fd***, TIOCSFON, u8_t** *font***[256][32]);**

Font memory consists of 256 character definitions of 32 lines per character and 8 pixels per line. The first line is the topmost line of the character. The leftmost pixel is lit if the most significant bit of a line is set, etc. How many lines are used depends on the current video mode. The 80x25 video mode used by Minix has an 8x16 character cell, 80x28 has 8x14 characters, and 132x43 or 132x50 has 8x8 characters. The boot variable **console** is used by both the Boot Monitor and the console driver to set the video mode, software scrolling on/off, and VGA screen blank timeout. See **boot**(8).

The Keyboard

The keyboard produces key codes for each key that is pressed. These keys are transformed into character codes or sequences according to the current keyboard translation table. The format of this table is described in **keymap**(5). The character codes can be read from the console device unless they map to special hotkeys. The hotkeys are as follows:

Name	KeyFunction
CTRL−ALT−DEL	Send an abort signal to process 1 (init). Init then halts the system
CTRL−ALT−KP-.	Likewise for keypad period
F1	Process table dump
F2	Show memory map
F3	Toggle software/hardware scrolling
F5	Show network statistics
CTRL−F7	Send a quit signal to all processes connected to the console
CTRL−F8	Send an interrupt signal
CTRL−F9	Send a kill signal. If CTRL−F8 or CTRL−F7 don't get 'em, then this surely will. These keys are for disaster recovery. You would normally use DEL and CTRL−\ to send interrupt and quit signals.
ALT−F1	Select virtual console 0 (/dev/console)
ALT−F2	Select virtual console 1 (/dev/ttyc1)
ALT−F(*n*+1)	Select virtual console *n* (/dev/ttyc*n*)
ALT−Left	Select previous virtual console
ALT−Right	Select next virtual console

The keyboard map is set with the **KIOCSMAP** ioctl whose precise details are currently hidden in the **loadkeys** utility.

Log device

The **log** device can be used by processes to print debug messages onto the console. The console is a terminal type device, so it is taken from processes when a session leader exits. This does not happen with the log device.

SEE ALSO

tty(4), **loadkeys**(1), **keymap**(5), **boot**(8).

NOTES

Output processing turns Line Feeds into CR LF sequences. Don't let this surprise you. Either turn off output processing or use one of the synonyms for LF.

AUTHOR

Kees J. Bot (kjb@cs.vu.nl)