NAME

vcs, vcsa − virtual console memory

DESCRIPTION

/dev/vcs0 is a character device with major number 7 and minor number 0, usually with mode 0644 and ownership root:tty. It refers to the memory of the currently displayed virtual console terminal.

/dev/vcs[1−63] are character devices for virtual console terminals, they have major number 7 and minor number 1 to 63, usually mode 0644 and ownership root:tty. */dev/vcsa[0−63]* are the same, but using *unsigned short*s (in host byte order) that include attributes, and prefixed with four bytes giving the screen dimensions and cursor position: *lines*, *columns*, *x*, *y*. (*x* = *y* = 0 at the top left corner of the screen.)

When a 512-character font is loaded, the 9th bit position can be fetched by applying the **ioctl**(2) **VT_GETHIFONTMASK** operation (available since Linux 2.6.18) on */dev/tty[1−63]*; the value is returned in the *unsigned short* pointed to by the third **ioctl**(2) argument.

These devices replace the screendump **ioctl**(2) operations of **ioctl_console**(2), so the system administrator can control access using filesystem permissions.

The devices for the first eight virtual consoles may be created by:

```
for x in 0 1 2 3 4 5 6 7 8; do
    mknod −m 644 /dev/vcs$x c 7 $x;
    mknod −m 644 /dev/vcsa$x c 7 $[$x+128];
done
chown root:tty /dev/vcs*
```

No **ioctl**(2) requests are supported.

FILES

/dev/vcs[0−63]
/dev/vcsa[0−63]

VERSIONS

Introduced with Linux 1.1.92.

EXAMPLES

You may do a screendump on vt3 by switching to vt1 and typing

```
cat /dev/vcs3 >foo
```

Note that the output does not contain newline characters, so some processing may be required, like in

```
fold −w 81 /dev/vcs3 | lpr
```

or (horrors)

```
setterm −dump 3 −file /proc/self/fd/1
```

The */dev/vcsa0* device is used for Braille support.

This program displays the character and screen attributes under the cursor of the second virtual console, then changes the background color there:

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/ioctl.h>
#include <linux/vt.h>

int
main(void)
{
    int fd;
    char *device = "/dev/vcsa2";
    char *console = "/dev/tty2";
    struct {unsigned char lines, cols, x, y;} scrn;
    unsigned short s;
    unsigned short mask;
    unsigned char attrib;
    int ch;

    fd = open(console, O_RDWR);
    if (fd < 0) {
        perror(console);
        exit(EXIT_FAILURE);
    }
    if (ioctl(fd, VT_GETHIFONTMASK, &mask) < 0) {
        perror("VT_GETHIFONTMASK");
        exit(EXIT_FAILURE);
    }
    (void) close(fd);
    fd = open(device, O_RDWR);
    if (fd < 0) {
        perror(device);
        exit(EXIT_FAILURE);
    }
    (void) read(fd, &scrn, 4);
    (void) lseek(fd, 4 + 2*(scrn.y*scrn.cols + scrn.x), SEEK_SET);
    (void) read(fd, &s, 2);
    ch = s & 0xff;
    if (s & mask)
        ch |= 0x100;
    attrib = ((s & ~mask) >> 8);
    printf("ch=%#03x attrib=%#02x\n", ch, attrib);
    s ^= 0x1000;
    (void) lseek(fd, −2, SEEK_CUR);
    (void) write(fd, &s, 2);
    exit(EXIT_SUCCESS);
}
```

SEE ALSO

ioctl_console(2), **tty**(4), **ttyS**(4), **gpm**(8)