

---

# VCS, VCSa

## Mémoires des consoles virtuelles

`/dev/vcs0` est un périphérique caractère de numéro majeur 7 et mineur 0, avec les permissions 0644 et un propriétaire root :tty. Il correspond à la mémoire d'affichage de la console virtuelle en cours d'utilisation.

`/dev/vcs[1-63]` sont des périphériques caractères représentant la mémoire d'affichage des consoles virtuelles. Ils ont un numéro majeur 7 et mineurs de 1 à 63. Ils ont généralement un mode d'accès 0644 et un propriétaire root :tty. Les `/dev/vcsa[0-63]` sont équivalents mais utilisent des entiers non-signés court (dans l'ordre des octets de l'hôte) qui incluent les attributs, et sont préfixés avec 4 octets indiquant les dimensions de l'écran et la position du curseur : lines, columns, x, x, (x = y = 0 en haut à gauche).

Lorsqu'une police à 512 caractères est chargée, la position du 9e bit peut être extraite en appliquant l'opération `ioctl(2)` `VT_GETHIFONTMASK` sur `/dev/tty[1-63]`; la valeur est renvoyée dans l'entier de type non-signé court pointé par le troisième paramètre de `ioctl(2)`.

Ces périphériques remplacent les opérations `ioctl(2)` `screedump` des console(4), ainsi l'administrateur peut contrôler les accès en utilisant les permissions du système de fichiers.

Les périphériques pour les 8 premières consoles virtuelles peuvent être créés ainsi :

```
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*
```

## Exemple

On peut faire un screendump de la console vt3 en basculant sur vt1 et en tapant :

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

Noter que la sortie ne contient pas de caractères de retour-chariot, quelques manipulations peuvent être nécessaires comme :

```
old -w 81 /dev/vcs3 | lpr
```

ou

```
xetterm -dump 3 -file /proc/self/fd/1
```

Ce programme affiche le caractère et l'attribut d'écran sous le curseur de la seconde console virtuelle, puis y change la couleur de fond :

```
#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 ch, attrib;

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), 0);
(void) read(fd, &s, 2);
ch = s & 0xff;
if (attrib & mask)
    ch |= 0x100;
attrib = ((s & ~mask) » 8);
printf("ch='%c' attrib=0x%02x\n", ch, attrib);
attrib ^= 0x10;
(void) lseek(fd, -1, 1);
(void) write(fd, &attrib, 1);
exit(EXIT_SUCCESS);
}
```