## NAME

vt − graphics interface

## DESCRIPTION

OUTPUT: **/dev/vt** (crt − only one user)

Data may be displayed on a vt11 graphics tube by a write system call:

**write (fdo, out_buf, count);**

where *fdo* is an integer file descriptor, *out_buf* is an integer array containing the display list and *count* is an integer containing the number of bytes in the display list and must be even. The display list is a sequence of octal numbers that define the image to be drawn. (These octal numbers are a mixture of control words and data that are given to the vt11 microprocessor.)

Prior to the write, the device must have been opened by

**fdo = open ("/dev/vt11", 1);**

and a seek must have been made to the proper frame

**lseek (fdo,** *n*, **0);**

where *n* is long and indicates the frame number (0 thru 9). A frame is an independently modifiable overlay which when overlayed with other frames complete the image.

The following is an example of a user program that will draw a 0200 by 0200 unit box at location 0500,0500 on the screen:

```
main()
{
        char *file;
        int fd;
        static int buf[]{0117124, 0500, 0500, 0110000, 040200,
                0, 040000, 0200, 060200, 0, 040000, 020200};

        file = "/dev/vt";
        fd = open(file, 1);
        if(fd < 0) {
                printf("failed to open %s0, file);
                exit(0);
        }
        lseek(fd, 0L, 0);
        write(fd, buf, sizeof(buf));
        for(;;)
                sleep(3600);
}
```

INPUT: **/dev/vtlp** (light pen) or **/dev/vtjy** (joy stick)
(only one user each device)

After an open system call: **fdi = open ("/dev/vtlp",0)** or **fdi = open ("/dev/vtjy",0)** ) input data can be obtained by a read system call:

**in_count = read (fdi, in_buf, count);**

where *in_buf* is a 3 element integer array.

If *count* is 0, the process will sleep until input occurs (event 1 or 2).

If *count* is 6, the *read* will return immediately and the 3 integers of *in_buf* contain: **event**, **x**, **y**. Where *x* and *y* are integers and contain the x and y coordinates respectively.

If *event* is 0, there is no unserviced input (event 1 or 2).

If *event* is 1, tracking start or a button is released.

If *event* is 2, tracking is stopped.

SYSTEM: (system proc table)

When a user graphics program is not running, the vtll may be used to display the operating system proc table. A sample of the proc table is shown below:

CB-UNIX Release 2.1 0000:07:46
100 procs 30 texts

| s | fl | wchan | sg | pri | ptm | ctm | clock | group | pid | ppid | size | name |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| s: | u | 22656 | | -100 | 127 | 127 | | 0 | 0 | 0 | 20 | UNIX Scheduler |
| s: | | 24756 | 0 | 40 | 127 | 127 | | 0 | 1 | 0 | 131 | init |
| s: | | 25014 | 0 | 10 | 127 | 3 | 722 | 0 | 4 | 1 | 111 | su |
| *r:* | | *0* | *0* | *10* | *3* | *2* | | *4* | *10* | *4* | *216* | *ls −l* |

FILES

/dev/vt
/dev/vtlp
/dev/vtjy

SEE ALSO

lseek(2), open(2), read(2), write(2)

RESTRICTIONS

Double word vtll instructions must NOT begin at out_buf[i] where i % 254 == 253 or grave disorder will result.