

\$SPAD/src/lib cursor.c

Stephen Watt, James Wen

July 31, 2014

Abstract

Contents

1	License	3
---	---------	---

1 License

/*

Copyright (c) 1991-2002, The Numerical ALgorithms Group Ltd.
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:

- Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in
the documentation and/or other materials provided with the
distribution.
- Neither the name of The Numerical ALgorithms Group Ltd. nor the
names of its contributors may be used to endorse or promote products
derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS
IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER
OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

*/

— * —

#include <stdlib.h>

#include "cursor.h1"

/*

* This routine changes the shape of the cursor. it is a modified version of
* a program by SMWatt, called cursor.c. JMW 6/22/89
*/

/* this stuff can only be done on AIX <AND> the right terminal (aixterm,hft) */
#if (defined(RIOSplatform) || defined(RTplatform)) && !defined(_AIX41)

```

#include "edible.h"
/* the HFT stuff requires ioctl's and termio's */
#include <termio.h>
#include <stdio.h>
#include <fcntl.h>
#include <sys/hft.h>

int
Cursor_shape(int shape)
{
    int hftfd;
    char hftpath[16], s[100];
    int chno;
    int i;
    struct termio oldterm, newterm;
    struct hftgetid hftgid;
    char *termVal;

    termVal = (char *) getenv("TERM");
    if (strcmp("hft", termVal) && strcmp("aixterm", termVal, 7))
        return;

    /* determine the desired shape */
    if (shape < 0 || shape > 5) {
        fprintf(stderr, "%d - Invalid cursor number\n");
        return (-1);
    }
    /* change the shape */
    s[0] = 033;                /* hf_intro.hf_esc      */
    s[1] = '[';                /* hf_intro.hf_lbr     */
    s[2] = 'x';                /* hf_intro.hf_ex       */
    s[3] = 0;                  /* hf_intro.hf_len[0]   */
    s[4] = 0;                  /* hf_intro.hf_len[1]   */
    s[5] = 0;                  /* hf_intro.hf_len[2]   */
    s[6] = 10;                 /* hf_intro.hf_len[3]   */
    s[7] = 2;                  /* hf_intro.hf_typehi   */
    s[8] = 8;                  /* hf_intro.hf_typelo   */
    s[9] = 2;                  /* hf_sublen            */
    s[10] = 0;                 /* hf_subtype           */
    s[11] = 0;                 /* hf_rsvd              */
    s[12] = shape;             /* hf_shape             */

    if (ioctl(0, HFTGETID, &hftgid) < 0) {
        /* perror("ioctl: HFTGETID"); */
        chno = -1;
    }
    else
        chno = hftgid.hf_chan;
}

```

```

if (chno == -1) {
    /** try being moronic and just writing what I want to
        standard output                                     *****/

    if (((ioctl(2, TCGETA, &oldterm)) == -1) ||
        ((ioctl(2, TCGETA, &newterm)) == -1)) {
        perror("Getting termio");
        exit(0);
    }
    newterm.c_oflag = newterm.c_lflag = newterm.c_iflag = 0;
    newterm.c_cc[0] = -1;
    for (i = 1; i <= 5; i++)
        newterm.c_cc[i] = 0;
    if ((ioctl(2, TCSETAF, &newterm)) == -1) {
        perror("Setting to raw mode");
        exit(0);
    }
    write(2, s, 13);
    read(0, s, 1024);
    if ((ioctl(2, TCSETAF, &oldterm)) == -1) {
        perror("Resetting terminal");
        exit(0);
    }
}
else {
    /* open the currently active virtual terminal on the hft */
    sprintf(hftpath, "/dev/hft/%d", chno);
    if ((hftfd = open(hftpath, O_RDWR)) == -1) {
        perror("Could not open hft channel\n");
        exit(0);
    }
    write(hftfd, s, 13);
}
}
#else

int
Cursor_shape(int shape)
{
    return shape;
}
#endif

```

References

- [1] nothing