file: uapi/v4l/keytable.c

```
/* keytable.c - This program allows checking/replacing keys at IR

   Copyright (C) 2006-2009 Mauro Carvalho Chehab <mchehab@kernel.org>

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation, version 2 of the License.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.
 */

#include <ctype.h>
#include <errno.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <linux/input.h>
#include <sys/ioctl.h>

#include "parse.h"

void prtcode (int *codes)
{
        struct parse_key *p;

        for (p=keynames;p->name!=NULL;p++) {
                if (p->value == (unsigned)codes[1]) {
                        printf("scancode 0x%04x = %s (0x%02x)\\n", codes[0], p->name, codes[1]);
                        return;
                }
        }

        if (isprint (codes[1]))
                printf("scancode %d = '%c' (0x%02x)\\n", codes[0], codes[1], codes[1]);
        else
                printf("scancode %d = 0x%02x\\n", codes[0], codes[1]);
}

int parse_code(char *string)
{
        struct parse_key *p;

        for (p=keynames;p->name!=NULL;p++) {
                if (!strcasecmp(p->name, string)) {
                        return p->value;
                }
        }
        return -1;
}

int main (int argc, char *argv[])
{
        int fd;
        unsigned int i, j;
        int codes[2];

        if (argc<2 || argc>4) {
                printf ("usage: %s <device> to get table; or\\n"
                        "       %s <device> <scancode> <keycode>\\n"
                        "       %s <device> <keycode_file>n",*argv,*argv,*argv);
                return -1;
        }

        if ((fd = open(argv[1], O_RDONLY)) < 0) {
                perror("Couldn't open input device");
                return(-1);
        }

        if (argc==4) {
                int value;

                value=parse_code(argv[3]);

                if (value==-1) {
                        value = strtol(argv[3], NULL, 0);
                        if (errno)
                                perror("value");
                }
```

```
                codes [0] = (unsigned) strtol(argv[2], NULL, 0);
                codes [1] = (unsigned) value;

                if(ioctl(fd, EVIOCSKEYCODE, codes))
                        perror ("EVIOCSKEYCODE");

                if(ioctl(fd, EVIOCGKEYCODE, codes)==0)
                        prtcode(codes);
                return 0;
        }

        if (argc==3) {
                FILE *fin;
                int value;
                char *scancode, *keycode, s[2048];

                fin=fopen(argv[2],"r");
                if (fin==NULL) {
                        perror ("opening keycode file");
                        return -1;
                }

                /* Clears old table */
                for (j = 0; j < 256; j++) {
                        for (i = 0; i < 256; i++) {
                                codes[0] = (j << 8) | i;
                                codes[1] = KEY_RESERVED;
                                ioctl(fd, EVIOCSKEYCODE, codes);
                        }
                }

                while (fgets(s,sizeof(s),fin)) {
                        scancode=strtok(s,"\\n\\t =:");
                        if (!scancode) {
                                perror ("parsing input file scancode");
                                return -1;
                        }
                        if (!strcasecmp(scancode, "scancode")) {
                                scancode = strtok(NULL,"\\n\\t =:");
                                if (!scancode) {
                                        perror ("parsing input file scancode");
                                        return -1;
                                }
                        }

                        keycode=strtok(NULL,"\\n\\t =:(");
                        if (!keycode) {
                                perror ("parsing input file keycode");
                                return -1;
                        }

                        // printf ("parsing %s=%s:", scancode, keycode);
                        value=parse_code(keycode);
                        // printf ("\\tvalue=%d\\n",value);

                        if (value==-1) {
                                value = strtol(keycode, NULL, 0);
                                if (errno)
                                        perror("value");
                        }

                        codes [0] = (unsigned) strtol(scancode, NULL, 0);
                        codes [1] = (unsigned) value;

                        // printf("\\t%04x=%04x\\n",codes[0], codes[1]);
                        if(ioctl(fd, EVIOCSKEYCODE, codes)) {
                                fprintf(stderr, "Setting scancode 0x%04x with 0x%04x via ",codes[0], codes[
                                perror ("EVIOCSKEYCODE");
                        }

                        if(ioctl(fd, EVIOCGKEYCODE, codes)==0)
                                prtcode(codes);
                }
                return 0;
        }

        /* Get scancode table */
        for (j = 0; j < 256; j++) {
                for (i = 0; i < 256; i++) {
                        codes[0] = (j << 8) | i;
                        if (!ioctl(fd, EVIOCGKEYCODE, codes) && codes[1] != KEY_RESERVED)
                                prtcode(codes);
                }
        }
        return 0;
}
```