(1) HL1 mit fehler

```
/************************************************************************
*************************************************************************

        W E G A - Quelle        (C) FSU Jena RZ Bereich Systemprogrammierun

        KERN 3.2           Modul:  datem.c

        Bearbeiter:        J. Richter
        Datum:             $D$
        Version:           $R$

*************************************************************************
*************************************************************************

char *xxxvers = "§(#)datem.c     1.3";

#include <stdio.h>
#include <signal.h>
#include <utmp.h>
#include <time.h>
#include <fcntl.h>
#include <ctype.h>

#define YES       1
#define NO        0
#define WTMP      "/usr/adm/wtmp"
#define UTMP      "/etc/utmp"
#define LSTR      20
#define max(a,b) (a>b?a:b)
#define T88       (long)567990000 /* time on 1.1.88 */
#define MAXVOR  (long)24

static long int ldate, newdate;
static int year, month, day, hour, minute;
struct tm *gmtime();
char *ctime(), *cmdname();
static int daypermonthÄÜ = ä31,28,31,30,31,30,31,31,30,31,30,31ü;
static int check = YES;
static int checkchange = NO;

main(argc, argv)
char ** argv;
ä
        char buÄ512Ü;
        int rb, fdc, aus();
        int cvdate(), cvtime();
        long int ls, lseek();
        struct utmp utmp;
        struct tm *tm;
        int mknocheck();
#ifndef D
        if(getuid())ä
                pstr(stdout, cmdname(argvÄ0Ü));
                pstr(stdout, ": Permission deniedön");
                exit(1);
        ü
        signal(SIGINT, SIG_IGN);
        signal(SIGTERM, SIG_IGN);
#endif
        signal(SIGQUIT, mknocheck);
        if(argc > 0 && strcmp(argvÄ1Ü, "-s") == 0) check = NO;
        ldate = 0l;
        if((fdc = open(UTMP, O_RDONLY)) < 0)ä
                pstr(stdout, "önLast date not availableön");
                check = NO;
        ü else ä
```

(12)

```
                for(;;)ä
                        if((rb = read(fdc, &utmp, sizeof(struct utmp))) < 1
                        if(rb < sizeof(struct utmp))ä
                                pstr(stdout, "Önread error ");
                                pstr(stdout, UTMP);
                                check = NO;
                        ü else
                                ldate = max(utmp.ut_time, ldate);
                ü
e0:
                close(fdc);
        ü
        newdate = ldate + 60l;
        if(newdate < T88)ä
                pstr(stdout, "can't accept time from ");
                pstr(stdout, UTMP);
                pstr(stdout, "Ön");
                check = NO;
        ü
        tm = gmtime(&ldate);
        hour = tm->tm_hour;
        minute = tm->tm_min;
        signal(SIGALRM, aus);
        beep();
d0:
        if(! check) goto d3;
        pstr(stdout, "ÖnLast known date and time: ");
        pstr(stdout, ctime(&ldate));
d3:
        pstr(stdout, "Ön");
d1:
        month = tm->tm_mon + 1;
        year = tm->tm_year;
        day = tm->tm_mday;
        alarm(300);
        if(! check) pstr(stdout, "! ");
        pstr(stdout, "Enter date (MM/DD/YY or <cr>)      : ");
        fgets(bu, sizeof(bu), stdin);
        if(checkchange)ä
                pstr(stdout, "Ön");
                checkchange = NO;
                goto d1;
        ü
        alarm(0);
        if(*bu == 'Ön') goto d2;
        if(cvdate(bu)) goto d1;
        if(checkkal()) goto d1;
d2:
        if(! check) pstr(stdout, "! ");
        pstr(stdout, "Enter time (HH:MM)                 : ");
        fgets(bu, sizeof(bu), stdin);
        if(checkchange)ä
                pstr(stdout, "Ön");
                checkchange = NO;
                goto d2;
        ü
        if(*bu == 'Ön') ä
                pstr(stdout, "A time must be specifiedÖn");
                goto d2;
        ü
        if(cvtime(bu)) goto d2;
        if(checktim()) goto d2;
        mknewtim();
        if(check)ä
                if(ldate > newdate)ä
                        pstr(stdout, "Do not set back clockÖn");
```

```
                                goto d0;
                        ü
                        else
                                if(newdate > ldate + MAXVOR*24l*60l*60l)ä
                                        pstr(stdout, "can't accept (too far away)ön
                                        goto d0;
                                ü
        ü
        aus();
ü

cvdate(b)
char *b;
ä
        register char *t = b;
        char *fields();
        register int r = 3;
        for(; *t ; t++)
                if(*t == 'ön') *t = 'ö0';
        strcat(b, "///");
        r -= stoi  (fields(b,1,'/'), "%d", &month);
        r -= stoi  (fields(b,2,'/'), "%d", &day  );
        r -= stoi  (fields(b,3,'/'), "%d", &year );
        if(r) pstr(stdout, "Bad date format. Try againön");
        return(r);
ü

cvtime(b)
char *b;
ä
        register char *t = b;
        char *fields();
        register int r = 2;
        for(; *t ; t++)
                if(*t == 'ön') *t = 'ö0';
        strcat(b, ":::");
        r -= stoi  (fields(b,1,':'), "%d", &hour  );
        r -= stoi  (fields(b,2,':'), "%d", &minute);
        if(r) pstr(stdout, "Bad time format. Try againön");
        return(r);
ü

char *fields(buff, no, ch)
char ch, *buff;
ä
        static char fÄLSTRÜ;
        register int i;
        register char *s, *ptrf = f;

        s = buff;
        for(i=1; i<no; i++)ä
                while(*s != ch) s++;
                s++;
        ü
        while(*s!=ch) *ptrf++ = *s++;
        *ptrf = 'ö0';
        return(f);
ü

aus()
ä
        char sÄ16Ü;
        spstr(s,"%02d%02d%02d%02d%02d",
                month, day, hour, minute, year);
#ifdef D
        pstr(stdout, "Date:");
```

(4)

```
        pstr(stdout, s);
        pstr(stdout, "ön");
#else
        execl("/bin/date", "date", s, 0);
        pstr(stdout, "/bin/date command missing in system!ön");
#endif
        exit(0);
ü

checktim()
ä
        if(hour < 0 öö hour > 23
                öö minute < 0 öö minute > 59)ä
                pstr(stdout, "Bad time. Try againön");
                return(1);
        ü
        return(0);
ü

checkkal()
ä
        daypermonthÄ1Ü = (schaltjahr((long)year)? 29:28);
        if( month < 1 öö month > 12
                        öö day < 1 öö day > daypermonthÄmonth-1Ü
                        öö year < 0 öö year > 99)ä
                pstr(stdout, "Bad date. Try againön");
                return(1);
        ü
        return(0);
ü

mknewtim()
ä
        extern long int ktime();
        long int lday = (long)day;
        long int lmon = (long)month;
        long int lyear = (long)year;
        long int lhour = (long)hour;
        long int lmin = (long)minute;
        newdate = ktime(lday, lmon, lyear, lhour, lmin);
        return(0);
ü

stoi(s, dumm, i)
char *dumm, *s;
int *i;
ä
        register char *t;
        if(*s == '00') return(0);
        *i = 0;
        for(t=s; *t; t++)ä
                if(!isdigit(*t)) return(0);
                *i *= 10;
                *i += (*t - '0');
        ü
        return(1);
ü

char *ito2s(i)
ä
        static char fÄÜ = "00";
        register di = i / 10;
        fÄ0Ü = di + '0';
        fÄ1Ü = (i - di * 10) + '0';
        return(f);
```

(5)

```
spstr(s, dumm, i1, i2, i3, i4, i5)
char *s, *dumm;
ä
        strcpy(s, ito2s(i1));
        strcat(s, ito2s(i2));
        strcat(s, ito2s(i3));
        strcat(s, ito2s(i4));
        strcat(s, ito2s(i5));
ü

mknocheck()
ä
        check = NO;
        checkchange = YES;
ü
```

alt mit Fehler

```
/*  K T I M E  -- wandelt Datum in long-Wert um (sek)                  */

#include <stdio.h>

#define T88     (long)567990000 /* time on 1.1.88 */
#define TYEAR   (long)31536000  /* seconds in a common year */
#define TDAY    (long)86400     /* seconds in a day */
#define THOUR   (long)3600      /* seconds in an hour */
#define TMIN    (long)60        /* seconds in a minute */
static long int daypermonth AÜ =
ä
        31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30ü;

long ktime (day, month, year, hour, min)
long int day, month, hour, year, min;
ä
        long int t;
        register int i;

        if(year < 88l) year += (long)100;
        t = T88;
        for (i=88; i<year; i++)          /* zyklus ueber jahre ab 1988 */
        ä
                t += TYEAR;
                if(schaltjahr((long)i)) /* korrektur fuer schaltjahre */
                        t += TDAY;
        ü

        if(schaltjahr(year))             /* korrektur wenn schaltjahr */
                daypermonthÄ1Ü = 29;

        month -= 2;                 /* zyklus ueber abgeschlossenen monate */
        for(i=0; i<=month; i++)
                t += (daypermonthÄiÜ * TDAY);

        for(i=1; i<day; i++)        /* zyklus ueber tage des laufenden monats
                t += TDAY;          /* (abgeschlossene tage!) */

        t += hour * THOUR;                  /* die laufende stunde */
        t += min * TMIN;                    /* und minute */

        return(t - THOUR);          /*eine stunde korrektur wegen gmt-mez*/
ü

schaltjahr(i)
long int i;
ä
        int i1 = i;
        /*
         * ein jahr ist schaltjahr, wenn die jahreszahl durch 4,
         *      aber nicht durch 100 teilbar ist.
         */
        if(((i1%4) == 0)  && ((i1%100) != 0)) return(1);
        return(0);
ü
```

Fehler