***********************************************************
***2013 Sun Nov 30 18:43:47 2014***
***new/usr/src/lib/libc/port/locale/mbstowcs.c***
***5366 strcoll_l may destroy its arguments, then crash***
***********************************************************

```
/*
 * Copyright 2013 Garrett D'Amore <garrett@damore.org>
 * Copyright 2010 Nexenta Systems, Inc.  All rights reserved.
 * Copyright (c) 2002-2004 Tim J. Robbins.
 * All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. 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.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR 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 AUTHOR 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 "lint.h"
#include <limits.h>
#include <stdlib.h>
#include <wchar.h>
#include <locale.h>
#include "mblocal.h"
#include "localeimpl.h"
#include "lctype.h"

size_t
mbstowcs_l(wchar_t *_RESTRICT_KYWD pwcs, const char *_RESTRICT_KYWD s,
    size_t n, locale_t loc)
{
        mbstate_t mbs = { 0 };
        static const mbstate_t initial = { 0 };
        mbstate_t mbs;
        const char *sp;

        mbs = initial;
        sp = s;
        return (loc->ctype->lc_mbsnrtowcs(pwcs, &sp, ULONG_MAX, n, &mbs));
}
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    3318 Sun Nov 30 18:43:47 2014
new/usr/src/lib/libc/port/locale/strcoll.c
5366 strcoll_l may destroy its arguments, then crash
**********************************************************
```

```
/*
 * Copyright 2013 Garrett D'Amore <garrett@damore.org>
 * Copyright 2010 Nexenta Systems, Inc.  All rights reserved.
 * Copyright (c) 1995 Alex Tatmanjants <alex@elvisti.kiev.ua>
 *              at Electronni Visti IA, Kiev, Ukraine.
 *                      All rights reserved.
 *
 * Redistribution and use in source and binary forms, with or without
 * modification, are permitted provided that the following conditions
 * are met:
 * 1. Redistributions of source code must retain the above copyright
 *    notice, this list of conditions and the following disclaimer.
 * 2. 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.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ''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 AUTHOR 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 "lint.h"
#include "file64.h"
#include <alloca.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <wchar.h>
#include <xlocale.h>
#include "localeimpl.h"
#include "collate.h"

#define ALLOCA_LIMIT    16

/*
 * In order to properly handle multibyte locales, its easiet to just
 * convert to wide characters and then use wcscoll.  However if an
 * error occurs, we gracefully fall back to simple strcmp.  Caller
 * should check errno.
 */
int
strcoll_l(const char *s1, const char *s2, locale_t loc)
{
        int ret;
        wchar_t *t1 = NULL, *t2 = NULL;
        wchar_t *w1 = NULL, *w2 = NULL;
        size_t sz1, sz2;
        const struct lc_collate *lcc = loc->collate;

        mbstate_t mbs1 = { 0 }; /* initial states */
        mbstate_t mbs2 = { 0 };

        if (lcc->lc_is_posix)
```



```
                return (strcmp(s1, s2));

        sz1 = strlen(s1) + 1;
        sz2 = strlen(s2) + 1;

        /*
         * Simple assumption: conversion to wide format is strictly
         * reducing, i.e. a single byte (or multibyte character)
         * cannot result in multiple wide characters.
         *
         * We gain a bit of performance by giving preference to alloca
         * for small string allocations.
         */
        if (sz1 > ALLOCA_LIMIT) {
                if ((t1 = malloc(sz1 * sizeof (wchar_t))) == NULL)
                        goto error;
                w1 = t1;
        } else {
                if ((w1 = alloca(sz1 * sizeof (wchar_t))) == NULL)
                        goto error;
        }
        if (sz2 > ALLOCA_LIMIT) {
                if ((t2 = malloc(sz2 * sizeof (wchar_t))) == NULL)
                        goto error;
                w2 = t2;
        } else {
                if ((w2 = alloca(sz2 * sizeof (wchar_t))) == NULL)
                        goto error;
        }

        if ((mbstowcs_l(w1, s1, sz1, loc)) == (size_t)-1)
        if ((mbsrtowcs_l(w1, &s1, sz1, &mbs1, loc)) == (size_t)-1)
                goto error;

        if ((mbstowcs_l(w2, s2, sz2, loc)) == (size_t)-1)
        if ((mbsrtowcs_l(w2, &s2, sz2, &mbs2, loc)) == (size_t)-1)
                goto error;

        ret = wcscoll_l(w1, w2, loc);
        if (t1)
                free(t1);
        if (t2)
                free(t2);

        return (ret);

error:
        if (t1)
                free(t1);
        if (t2)
                free(t2);
        return (strcmp(s1, s2));
}
```

**_____unchanged_portion_omitted_**