

\*\*\*\*\*

5228 Sat Jul 12 19:07:44 2014

new/usr/src/lib/libc/port/locale/lmonetary.c

4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character

4999 libc test suite enhancements

Reviewed by: Jason King <jason.brian.king@gmail.com>

Reviewed by: Dan McDonald <danmcd@omniti.com>

Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>

Approved by: TBD

\*\*\*\*\*

\_\_\_\_\_ unchanged portion omitted \_\_\_\_\_

```

90 struct locdata *
91 __lc_monetary_load(const char *name)
92 {
93     int ret;
94     int clen;
95     struct lc_monetary *lmon;
96     struct locdata *ldata;

98     if ((ldata = __locdata_alloc(name, sizeof (*lmon))) == NULL) {
99         return (NULL);
100     }
101     lmon = ldata->l_data[0];

103     ret = __part_load_locale(name, (char **)&ldata->l_data[1],
104         "LC_MONETARY", LCMONETARY_SIZE_FULL, LCMONETARY_SIZE_MIN,
105         (const char **)lmon);

107     if (ret != _LDP_LOADED) {
108         __locdata_free(ldata);
109         errno = EINVAL;
110         return (NULL);
111     }

113     /* special storage for currency string */
114     clen = strlen(lmon->currency_symbol) + 2;
115     ldata->l_data[2] = libc_malloc(clen);
116     lmon->crncystr = ldata->l_data[2];

118     lmon->mon_grouping = __fix_locale_grouping_str(lmon->mon_grouping);

120 #define M_ASSIGN_CHAR(NAME) \
121     (((char *)lmon->NAME)[0] = cnv(lmon->NAME))

123     M_ASSIGN_CHAR(int_frac_digits);
124     M_ASSIGN_CHAR(frac_digits);
125     M_ASSIGN_CHAR(p_cs_precedes);
126     M_ASSIGN_CHAR(p_sep_by_space);
127     M_ASSIGN_CHAR(n_cs_precedes);
128     M_ASSIGN_CHAR(n_sep_by_space);
129     M_ASSIGN_CHAR(p_sign_posn);
130     M_ASSIGN_CHAR(n_sign_posn);

132     /*
133     * The six additional C99 international monetary formatting
134     * parameters default to the national parameters when
135     * reading FreeBSD LC_MONETARY data files.
136     */
137 #define M_ASSIGN_ICHAR(NAME) \
138     if (lmon->int_##NAME == NULL) \
139         lmon->int_##NAME = lmon->NAME; \
140     else \
141         M_ASSIGN_CHAR(int_##NAME);

143     M_ASSIGN_ICHAR(p_cs_precedes);

```

```

144     M_ASSIGN_ICHAR(n_cs_precedes);
145     M_ASSIGN_ICHAR(p_sep_by_space);
146     M_ASSIGN_ICHAR(n_sep_by_space);
147     M_ASSIGN_ICHAR(p_sign_posn);
148     M_ASSIGN_ICHAR(n_sign_posn);

150     /*
151     * Now calculate the currency string (CRNCYSTR) for nl_langinfo.
152     * This is a legacy SUSv2 interface.
153     */
154     if ((lmon->p_cs_precedes[0] == lmon->n_cs_precedes[0]) &&
155         (lmon->currency_symbol[0] != '\0')) {
156         char sign = '\0';
157         switch (lmon->p_cs_precedes[0]) {
158             case 0:
159                 sign = '+';
160                 sign = '-';
161                 break;
162             case 1:
163                 sign = '-';
164                 sign = '+';
165                 break;
166             case CHAR_MAX:
167                 /*
168                 * Substitute currency string for radix character.
169                 * To the best of my knowledge, no locale uses this.
170                 */
171                 if (strcmp(lmon->mon_decimal_point,
172                     lmon->currency_symbol) == 0)
173                     sign = '.';
174                 break;
175             }
176         (void) snprintf(lmon->crncystr, clen, "%c%s", sign,
177             lmon->currency_symbol);

178     return (ldata);
179 }
_____ unchanged portion omitted _____

```

new/usr/src/pkg/manifests/system-test-libctest.mf

1

\*\*\*\*\*

2128 Sat Jul 12 19:07:44 2014

new/usr/src/pkg/manifests/system-test-libctest.mf

4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character

4999 libc test suite enhancements

Reviewed by: Jason King <jason.brian.king@gmail.com>

Reviewed by: Dan McDonald <danmcd@omniti.com>

Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>

Approved by: TBD

\*\*\*\*\*

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 # Copyright 2014, OmniTI Computer Consulting, Inc. All rights reserved.
15 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
16 #
```

```
18 set name=pkg.fmri value=pkg:/system/test/libctest@$(PKGVERS)
19 set name=pkg.description value="C library Unit Tests"
20 set name=pkg.summary value="C Library Unit Test Suite"
21 set name=info.classification \
22     value=org.opensolaris.category.2008:Development/System
23 set name=variant.arch value=$(ARCH)
24 dir path=opt/libc-tests
25 dir path=opt/libc-tests/bin
26 dir path=opt/libc-tests/runfiles
27 dir path=opt/libc-tests/tests
28 file path=opt/libc-tests/README mode=0444
29 file path=opt/libc-tests/bin/libctest mode=0555
30 file path=opt/libc-tests/runfiles/default.run mode=0444
31 file path=opt/libc-tests/tests/newlocale_test mode=0555
32 $(i386_ONLY)file path=opt/libc-tests/tests/newlocale_test.amd64 mode=0555
33 $(i386_ONLY)file path=opt/libc-tests/tests/newlocale_test.i386 mode=0555
34 $(sparc_ONLY)file path=opt/libc-tests/tests/newlocale_test.sparc mode=0555
35 $(sparc_ONLY)file path=opt/libc-tests/tests/newlocale_test.sparcv9 mode=0555
36 file path=opt/libc-tests/tests/nl_langinfo_test mode=0555
37 $(i386_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.amd64 mode=0555
38 $(i386_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.i386 mode=0555
39 $(sparc_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.sparc mode=0555
40 $(sparc_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.sparcv9 mode=0555
41 license lic_CDDL license=lic_CDDL
42 depend fmri=locale/de type=require
43 depend fmri=locale/en type=require
44 depend fmri=locale/en-extra type=require
45 depend fmri=locale/ja type=require
46 depend fmri=locale/ru type=require
47 depend fmri=system/test/testrunner type=require
```

```
*****
```

```
2455 Sat Jul 12 19:07:44 2014
new/usr/src/test/libc-tests/doc/README
4964 nl_langinfo(CRNCYSTR) returns wrong alignment character
4999 libc test suite enhancements
Reviewed by: Jason King <jason.brian.king@gmail.com>
Reviewed by: Dan McDonald <danmcd@omniti.com>
Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>
Approved by: TBD
```

```
*****
```

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
12 #
13 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
14 # Copyright (c) 2012 by Delphix. All rights reserved.
15 #
```

```
17 libc Unit Test Suite README
```

```
19 1. What this Unit Test Suite tests
20 2. Building and installing this Unit Test Suite
21 3. Running this Unit Test Suite
22 4. Test results
```

```
24 -----
```

```
26 1. What this Unit Test Suite tests
```

```
28 This Unit Test Suite is for testing various libc interfaces. The suite
29 will test both 32 and 64 bit versions, provided that your kernel supports
30 both.
28 This Unit Test Suite is for testing various libc interfaces.
```

```
32 2. Building and installing this Unit Test Suite
```

```
34 This Test Suite runs under the testrunner framework (which can be installed
35 as pkg:/system/test/testrunner). To build both this Unit Test Suite and the
36 testrunner without running a full nightly:
```

```
38 build_machine$ bldenv [-d] <your_env_file>
39 build_machine$ cd $SRC/test
40 build_machine$ dmake install
41 build_machine$ cd $SRC/pkg
42 build_machine$ dmake install
```

```
44 Then set the publisher on the test machine to point to your repository and
45 install the Utils Unit Test Suite.
```

```
47 test_machine# pkg install pkg:/system/test/libctest
```

```
49 Note, the framework will be installed automatically, as this test suite
50 depends on it.
```

```
52 Additionally some text locales are required, specifically, en_US.UTF-8,
53 de_DE.UTF-8, ja_JP.UTF-8, and ru_RU.UTF-8. Again, these are listed as
54 dependencies and will be automatically installed.
```

```
56 3. Running this Unit Test Suite
```

```
58 The pre-requisites for running the this Unit Test Suite are:
```

```
59 - Any user may perform these tests.
60 - The en_US.UTF-8, en_GB.ISO8859-15, ja_JP.UTF-8, de_DE.UTF-8, and
61 ru_RU.UTF-8 locales must be installed.
58 - The en_US.UTF-8, ja_JP.UTF-8, de_DE.UTF-8, and ru_RU.UTF-8 locales
59 must be installed.
```

```
63 Once the pre-requisites are satisfied, simply run the libctest script:
```

```
61 Once the pre-requisites are satisfied, simply run the ostest script:
```

```
65 test_machine$ /opt/libc-tests/bin/libctest
63 test_machine$ /opt/util-tests/bin/libctest
```

```
67 4. Test results
```

```
69 While the Unit Test Suite is running, one informational line is printed at
70 the end of each test, and a results summary is printed at the end of the run.
71 The results summary includes the location of the complete logs, which is of the
72 form /var/tmp/test_results/<ISO 8601 date>.
```

new/usr/src/test/libc-tests/runfiles/default.run

1

\*\*\*\*\*

698 Sat Jul 12 19:07:44 2014

new/usr/src/test/libc-tests/runfiles/default.run

4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character

4999 libc test suite enhancements

Reviewed by: Jason King <jason.brian.king@gmail.com>

Reviewed by: Dan McDonald <danmcd@omniti.com>

Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>

Approved by: TBD

\*\*\*\*\*

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
15 #
```

```
17 [DEFAULT]
18 pre =
19 verbose = False
20 quiet = False
21 timeout = 60
22 post =
23 outputdir = /var/tmp/test_results
24
25 [/opt/libc-tests/tests/newlocale_test]
26
27 [/opt/libc-tests/tests/nl_langinfo_test]
```

new/usr/src/test/libc-tests/tests/Makefile

1

\*\*\*\*\*

613 Sat Jul 12 19:07:44 2014

new/usr/src/test/libc-tests/tests/Makefile

4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character

4999 libc test suite enhancements

Reviewed by: Jason King <jason.brian.king@gmail.com>

Reviewed by: Dan McDonald <danmcd@omniti.com>

Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>

Approved by: TBD

\*\*\*\*\*

1 #

2 # This file and its contents are supplied under the terms of the

3 # Common Development and Distribution License ("CDDL"), version 1.0.

4 # You may only use this file in accordance with the terms of version

5 # 1.0 of the CDDL.

6 #

7 # A full copy of the text of the CDDL should have accompanied this

8 # source. A copy of the CDDL is also available via the Internet at

9 # <http://www.illumos.org/license/CDDL>.

10 #

12 #

13 # Copyright (c) 2012 by Delphix. All rights reserved.

14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>

15 #

17 SUBDIRS = newlocale nl\_langinfo

17 SUBDIRS = newlocale

19 include \$(SRC)/Makefile.master

20 include \$(SRC)/test/Makefile.com

new/usr/src/test/libc-tests/tests/Makefile.com

1

\*\*\*\*\*

1980 Sat Jul 12 19:07:44 2014  
new/usr/src/test/libc-tests/tests/Makefile.com  
4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character  
4999 libc test suite enhancements  
Reviewed by: Jason King <jason.brian.king@gmail.com>  
Reviewed by: Dan McDonald <dancd@omniti.com>  
Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>  
Approved by: TBD

\*\*\*\*\*

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
15 #
16 #
17 include $(SRC)/Makefile.master
18 include $(SRC)/cmd/Makefile.cmd
19 include $(SRC)/test/Makefile.com
20 #
21 OBJ32 = $(PROG:%=%.$(MACH).o)
22 PROG32 = $(PROG).$(MACH)
23 #
24 $(BUILD64) OBJ64 = $(PROG:%=%.$(MACH64).o)
25 $(BUILD64) PROG64 = $(PROG).$(MACH64)
26 #
27 SRCS = $(PROG).c
28 #
29 C99MODE = -xc99=%all
30 CFLAGS_amd64 = -m64
31 CFLAGS_i386 = -m32
32 CFLAGS_sparc = -m32
33 CFLAGS_sparcv9 = -m64
34 LINTFLAGS += -DARCH=\"ARCH\"
35 #
36 ROOTOPTPKG = $(ROOT)/opt/libc-tests
37 TESTDIR = $(ROOTOPTPKG)/tests
38 #
39 CMDS = $(PROG32:%=$(TESTDIR)/%) $(PROG64:%=$(TESTDIR)/%) \
40        $(KSHPROG:%=$(TESTDIR)/%) $(ARCHPROG:%=$(TESTDIR)/%)
41 $(CMDS) := FILEMODE = 0555
42 #
43 all: $(PROG32) $(PROG64) $(KSHPROG) $(ARCHPROG) $(SUBDIRS)
44 #
45 $(PROG32): $(OBJ32)
46         $(LINK.c) $(CFLAGS_$(MACH)) $(OBJ32) -o $@ $(LDLIBS)
47         $(POST_PROCESS)
48 #
49 $(PROG64): $(OBJ64)
50         $(LINK.c) $(CFLAGS_$(MACH64)) $(OBJ64) -o $@ $(LDLIBS)
51         $(POST_PROCESS)
52 #
53 $(KSHPROG): $(KSHPROG).ksh
54         $(RM) $@
55         $(CP) $(KSHPROG).ksh $@
56         $(CHMOD) +x $@
```

new/usr/src/test/libc-tests/tests/Makefile.com

2

```
58 $(ARCHPROG): ../common/run_arch_tests.ksh
59         $(RM) $@
60         $(CP) ../common/run_arch_tests.ksh $@
61         $(CHMOD) +x $@
62 #
63 %.$(MACH).o: %.c
64         $(COMPILE.c) -o $@ $(CFLAGS_$(MACH)) -DARCH=\"$(MACH)\" $<
65 #
66 %.$(MACH64).o: %.c
67         $(COMPILE.c) -o $@ $(CFLAGS_$(MACH64)) -DARCH=\"$(MACH64)\" $<
68 #
69 install: $(SUBDIRS) $(CMDS)
70 #
71 lint: lint_SRCS
72 #
73 clobber: clean
74         -$(RM) $(PROG32) $(PROG64) $(KSHPROG) $(ARCHPROG)
75 #
76 clean:
77         -$(RM) $(OBJ32) $(OBJ64)
78 #
79 $(CMDS): $(TESTDIR) $(PROG32) $(PROG64) $(KSHPROG) $(ARCHPROG)
80 #
81 $(TESTDIR):
82         $(INS.dir)
83 #
84 $(TESTDIR)/%: %
85         $(INS.file)
```

new/usr/src/test/libc-tests/tests/common/run\_arch\_tests.ksh

1

\*\*\*\*\*

```
1387 Sat Jul 12 19:07:45 2014
new/usr/src/test/libc-tests/tests/common/run_arch_tests.ksh
4964 nl_langinfo(CRNCYSTR) returns wrong alignment character
4999 libc test suite enhancements
Reviewed by: Jason King <jason.brian.king@gmail.com>
Reviewed by: Dan McDonald <dancmd@omniti.com>
Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>
Approved by: TBD
```

\*\*\*\*\*

```
1 #!/usr/bin/ksh
2 #
3 #
4 # This file and its contents are supplied under the terms of the
5 # Common Development and Distribution License ("CDDL"), version 1.0.
6 # You may only use this file in accordance with the terms of version
7 # 1.0 of the CDDL.
8 #
9 # A full copy of the text of the CDDL should have accompanied this
10 # source. A copy of the CDDL is also available via the Internet at
11 # http://www.illumos.org/license/CDDL.
12 #
13 #
14 #
15 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
16 #
17 #
18 # First we set $dir to dirname $0, and $prog to basename $0 .ksh,
19 # using efficient ksh builtins.
20 case $0 in
21 */)
22     dir=${0%/*}
23     prog=${0##*/}
24     ;;
25 *)
26     dir=.
27     prog=${0}
28     ;;
29 esac
30 prog=${prog%.ksh}
31 #
32 for a in $*
33 do
34     if [[ $a == "-d" ]]
35     then
36         debug=yes
37     fi
38 done
39 #
40 # We look for architecture specific versions of the program,
41 # searching in several candidate directories. We run each one as
42 # we find it.
43 for f in $(/usr/bin/isainfo)
44 do
45     found=
46     [[ -n $debug ]] && print "Checking for arch $f:"
47     for p in \
48         ${dir}/${prog}.$f \
49         ${dir}/${f}/${prog}.$f \
50         ${dir}/${f}/${prog}
51     do
52         [[ -n $found ]] && continue
53         [[ -n $debug ]] && print -n "    $p"
54         if [[ -f $p ]]; then
55             [[ -n $debug ]] && print " FOUND"
56             [[ -n $debug ]] && print "Executing $p $*"

```

new/usr/src/test/libc-tests/tests/common/run\_arch\_tests.ksh

2

```
57         found=yes
58         $p $* || exit 1
59     else
60         [[ -n $debug ]] && print
61     fi
62 done
63 [[ -z $found ]] && [[ -n $debug ]] && print "NOT PRESENT"
64 done
65 exit 0
```

new/usr/src/test/libc-tests/tests/newlocale/Makefile

1

\*\*\*\*\*

567 Sat Jul 12 19:07:45 2014

new/usr/src/test/libc-tests/tests/newlocale/Makefile

4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character

4999 libc test suite enhancements

Reviewed by: Jason King <jason.brian.king@gmail.com>

Reviewed by: Dan McDonald <dammcd@omniti.com>

Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>

Approved by: TBD

\*\*\*\*\*

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
```

```
12 #
13 # Copyright (c) 2012 by Delphix. All rights reserved.
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
15 #
```

```
16 include $(SRC)/Makefile.master
17 include $(SRC)/cmd/Makefile.cmd
18 include $(SRC)/test/Makefile.com
```

```
18 PROG = newlocale_test
19 ARCHPROG = newlocale_test
21 OBJS = $(PROG:%=%o)
22 SRCS = $(OBJS:%o=%c)
```

```
21 include ../Makefile.com
24 C99MODE = -xc99=%all
```

```
26 ROOTOPTPKG = $(ROOT)/opt/libc-tests
27 TESTDIR = $(ROOTOPTPKG)/tests
```

```
29 CMDS = $(PROG:%=$(TESTDIR)/%)
30 $(CMDS) := FILEMODE = 0555
```

```
32 all: $(PROG)
```

```
34 $(PROG): $(OBJS)
35     $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
36     $(POST_PROCESS)
```

```
38 %.o: ../%.c
39     $(COMPILE.c) $<
```

```
41 install: all $(CMDS)
```

```
43 lint: lint_SRCS
```

```
45 clobber: clean
46     -$(RM) $(PROG)
```

```
48 clean:
49     -$(RM) $(OBJS)
```

```
51 $(CMDS): $(TESTDIR) $(PROG)
```

```
53 $(TESTDIR):
```

new/usr/src/test/libc-tests/tests/newlocale/Makefile

2

```
54     $(INS.dir)
```

```
56 $(TESTDIR)/%: %
```

```
57     $(INS.file)
```



\*\*\*\*\*

6953 Sat Jul 12 19:07:45 2014  
 new/usr/src/test/libc-tests/tests/newlocale/newlocale\_test.c  
 4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character  
 4999 libc test suite enhancements  
 Reviewed by: Jason King <jason.brian.king@gmail.com>  
 Reviewed by: Dan McDonald <danmcd@omniti.com>  
 Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>  
 Approved by: TBD

\*\*\*\*\*

\_\_\_\_\_ unchanged\_portion\_omitted \_\_\_\_\_

```

52 #define NUM_LDATA      5
53 #define NUMTHR        20
54 #define NUMITR        200

56 static void
57 test_start(const char *testName, const char *format, ...)
58 {
59     va_list args;

61     (void) printf("TEST STARTING %s (%s): ", testName, ARCH);
61     (void) printf("TEST STARTING %s: ", testName);

63     va_start(args, format);
64     (void) vprintf(format, args);
65     va_end(args);
66     (void) fflush(stdout);
67 }

69 static void
70 test_failed(const char *testName, const char *format, ...)
71 {
72     va_list args;

74     (void) printf("TEST FAILED %s (%s): ", testName, ARCH);
74     (void) printf("TEST FAILED %s: ", testName);

76     va_start(args, format);
77     (void) vprintf(format, args);
78     va_end(args);
79     (void) printf("\n");

81     (void) exit(-1);
82 }

84 static void
85 test_passed(const char *testName)
86 {
87     (void) printf("TEST PASS: %s (%s)\n", testName, ARCH);
86     (void) printf("TEST PASS: %s\n", testName);
88     (void) fflush(stdout);
89 }

91 void *
92 testlocale_thr_one(void *ptr)
91 testlocale_thr(void *ptr)
93 {
94     locale_t      cloc, loc;
95     struct lconv  *lc;
96     char          *day;
97     char          *tname = ptr;

99     for (int i = 0; i < NUMITR; i++) {
100         struct ldata *l = &ldata[i % NUM_LDATA];
101         cloc = uselocale(NULL);

```

```

103     loc = newlocale(LC_ALL_MASK, l->locale, NULL);
104     if (loc == NULL) {
105         test_failed("newlocale %s failed", l->locale);
106     }
107     day = nl_langinfo_l(DAY_1, loc);
108     if (strcmp(day, l->day1) != 0) {
109         test_failed(tname, "newlocale data mismatch (%s != %s)",
110             day, l->day1);
111     }
112     if (debug)
113         (void) printf("DAY1: %s\n", day);

115     day = nl_langinfo(DAY_1);
116     if (strcmp(day, "Sunday") != 0) {
117         test_failed(tname, "C locale day wrong %s != Sunday",
118             day);
119     }
120     lc = localeconv();
121     if (strcmp(lc->currency_symbol, "") != 0) {
122         test_failed(tname, "C cursym mismatch (%s != %s)",
123             lc->currency_symbol, "");
124     }

126     /* we sleep a random bit to mix it up */
127     (void) usleep(rand() % 10);

129     (void) uselocale(loc);
130     day = nl_langinfo(DAY_1);
131     if (strcmp(day, l->day1) != 0) {
132         test_failed(tname, "uselocale data mismatch (%s != %s)",
133             day, l->day1);
134     }

136     lc = localeconv();
137     if (strcmp(lc->currency_symbol, l->cursym) != 0) {
138         test_failed(tname, "uselocal cursym %s != %s",
139             lc->currency_symbol, l->cursym);
140     }
141     if (debug)
142         (void) printf("CSYM: %s\n", lc->currency_symbol);

144     /* we sleep a random bit to mix it up */
145     (void) usleep(rand() % 10);

147     if (uselocale(cloc) != loc) {
148         test_failed(tname, "revert old locale mismatch");
149     }
150     freelocale(loc);
151     if (uselocale(LC_GLOBAL_LOCALE) != cloc) {
152         test_failed(tname, "revert GLOBAL_LOCALE mismatch");
153     }
154 }
155     return (NULL);
156 }

159 void
160 test_newlocale_threaded(void)
159 testlocale(void)
161 {
162     char          *tname = "newlocale_threaded";
161     char          *tname = "newlocale/uselocale";
163     pthread_t      tid[NUMTHR];

165     test_start(tname, "running %d threads %d iterations\n", NUMTHR, NUMITR);

```

```

167     for (int i = 0; i < NUMTHR; i++) {
168         (void) pthread_create(&tid[i], NULL, testlocale_thr_one, tname);
167         (void) pthread_create(&tid[i], NULL, testlocale_thr, tname);
169     }

171     for (int i = 0; i < NUMTHR; i++) {
172         (void) pthread_join(tid[i], NULL);
173     }
174     test_passed(tname);
175 }

177 void
178 test_newlocale_negative(void)
179 {
180
181     locale_t loc, bad;
182     char *day;
183     char *tname = "newlocale_negative";

185     test_start(tname, "verify proper failure handling\n");
186     loc = newlocale(LC_ALL_MASK, "de_DE.UTF-8", NULL);
187     if (loc == NULL) {
188         test_failed(tname, "cannot set de_DE.UTF-8");
189     }
190     day = nl_langinfo_l(DAY_1, loc);
191     if (strcmp(day, "Sonntag") != 0) {
192         test_failed(tname, "incorrect Sonntag != %s", day);
193     }

195     bad = newlocale(LC_ALL_MASK, "cn_US.BIZRRR", loc);
196     if (bad != NULL) {
197         test_failed(tname, "passed setting bogus locale");
198     }
199     day = nl_langinfo_l(DAY_1, loc);
200     if (strcmp(day, "Sonntag") != 0) {
201         test_failed(tname, "incorrect Sonntag != %s", day);
202     }
203     test_passed(tname);
204 }

206 void
207 test_newlocale_categories(void)
208 {
209     locale_t loc;
210     char *day, *cur, *yes;
211     char *tname = "newlocale_categories";

213     test_start(tname, "verify different category behavior\n");

215     loc = NULL;
216     loc = newlocale(LC_TIME_MASK, "de_DE.UTF-8", loc);
217     loc = newlocale(LC_MESSAGES_MASK, "ru_RU.UTF-8", loc);
218     loc = newlocale(LC_MONETARY_MASK, "en_US.UTF-8", loc);

220     if (loc == NULL) {
221         test_failed(tname, "failed to set locale");
222     }

224     day = nl_langinfo_l(DAY_1, loc);
225     if ((day == NULL) || (strcmp(day, "Sonntag") != 0)) {
226         test_failed(tname, "day1 mismatch %s != %s", day, "Sonntag");
227     }
228     yes = nl_langinfo_l(YESSTR, loc);
229     if ((yes == NULL) || (strcmp(yes, "--°") != 0)) {
230         test_failed(tname, "currency mismatch");

```

```

231     }
232     cur = nl_langinfo_l(CRNCYSTR, loc);
233     if ((cur == NULL) || (strcmp(cur, "-$") != 0)) {
234         test_failed(tname, "currency mismatch [%s] != [%s]", cur, "-$");
235     }

237     test_passed(tname);
238 }

240 void
241 test_newlocale_composite(void)
242 {
243     locale_t loc;
244     char *day, *cur, *yes;
245     char *tname = "newlocale_composite";

247     test_start(tname, "verify composite locale behavior\n");

249     /* order: CTYPE/NUMERIC/TIME/COLLATE/MONETARY/MESSAGES */
250     loc = newlocale(LC_ALL_MASK,
251         "C/C/de_DE.UTF-8/C/en_US.UTF-8/ru_RU.UTF-8", NULL);

253     if (loc == NULL) {
254         test_failed(tname, "failed to set composite locale");
255     }

257     day = nl_langinfo_l(DAY_1, loc);
258     if ((day == NULL) || (strcmp(day, "Sonntag") != 0)) {
259         test_failed(tname, "day1 mismatch %s != %s", day, "Sonntag");
260     }
261     yes = nl_langinfo_l(YESSTR, loc);
262     if ((yes == NULL) || (strcmp(yes, "--°") != 0)) {
263         test_failed(tname, "currency mismatch");
264     }
265     cur = nl_langinfo_l(CRNCYSTR, loc);
266     if ((cur == NULL) || (strcmp(cur, "-$") != 0)) {
267         test_failed(tname, "currency mismatch [%s] != [%s]", cur, "-$");
268     }

270     test_passed(tname);
271 }

273 int
274 main(int argc, char **argv)
275 {
276     int optc;

278     while ((optc = getopt(argc, argv, "d")) != EOF) {
279         switch (optc) {
280             case 'd':
281                 debug++;
282                 break;
283             default:
284                 (void) fprintf(stderr, "Usage: %s [-d]\n", argv[0]);
285                 exit(1);
286         }
287     }

289     test_newlocale_threaded();
290     test_newlocale_negative();
291     test_newlocale_categories();
292     test_newlocale_composite();
293     testlocale();

294     exit(0);
295 }

```

new/usr/src/test/libc-tests/tests/nl\_langinfo/Makefile

1

\*\*\*\*\*

571 Sat Jul 12 19:07:45 2014

new/usr/src/test/libc-tests/tests/nl\_langinfo/Makefile

4964 nl\_langinfo(CRNCYSTR) returns wrong alignment character

4999 libc test suite enhancements

Reviewed by: Jason King <jason.brian.king@gmail.com>

Reviewed by: Dan McDonald <danmcd@omniti.com>

Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>

Approved by: TBD

\*\*\*\*\*

```
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 #
12 #
13 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 include $(SRC)/Makefile.master
17 #
18 PROG = nl_langinfo_test
19 ARCHPROG = nl_langinfo_test
20 #
21 include ../Makefile.com
```

```

*****
7291 Sat Jul 12 19:07:45 2014
new/usr/src/test/libc-tests/tests/nl_langinfo/nl_langinfo_test.c
4964 nl_langinfo(CRNCYSTR) returns wrong alignment character
4999 libc test suite enhancements
Reviewed by: Jason King <jason.brian.king@gmail.com>
Reviewed by: Dan McDonald <danmcd@omniti.com>
Reviewed by: Igor Kozhukhov <ikozhukhov@gmail.com>
Approved by: TBD
*****
1 /*
2  * This file and its contents are supplied under the terms of the
3  * Common Development and Distribution License ("CDDL"), version 1.0.
4  * You may only use this file in accordance with the terms of version
5  * 1.0 of the CDDL.
6  *
7  * A full copy of the text of the CDDL should have accompanied this
8  * source. A copy of the CDDL is also available via the Internet at
9  * http://www.illumos.org/license/CDDL.
10 */

12 /*
13  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
14  */

16 /*
17  * This program tests that newlocale and uselocale work properly in
18  * multi-threaded programs. In order for it to work, it requires that
19  * some additional locales be installed.
20  */

22 #include <stdio.h>
23 #include <stdlib.h>
24 #include <string.h>
25 #include <locale.h>
26 #include <libintl.h>
27 #include <langinfo.h>
28 #include <nl_types.h>
29 #include <err.h>
30 #include <errno.h>
31 #include <unistd.h>

33 int debug = 0;
34 int force = 0;

36 /*
37  * Note that on some platforms, different symbols are used. For example,
38  * MacOS Mavericks uses "Eu" for Euro symbol, instead of â €. If the locale
39  * data changes, then this program will need to update to reflect that.
40  *
41  * Note also that this file is easiest edited with a UTF-8 capable editor,
42  * as there are embedded UTF-8 symbols in some of the strings.
43  */
44 struct langinfo_test {
45     nl_item      param;
46     const char   *value;
47 };

49 struct langinfo_test C_data[] = {
50     { CODESET,      "646" },
51     { D_T_FMT,     "%a %b %e %H:%M:%S %Y" },
52     { D_FMT,       "%m/%d/%y" },
53     { T_FMT,       "%H:%M:%S" },
54     { T_FMT_AMPM, "%I:%M:%S %p" },
55     { AM_STR,      "AM" },
56     { PM_STR,      "PM" },

```

```

57     { ERA,          "" },
58     { ERA_D_FMT,   "" },
59     { ERA_D_T_FMT, "" },
60     { ERA_T_FMT,   "" },
61     { DAY_1,       "Sunday" },
62     { DAY_7,       "Saturday" },
63     { ABDAY_1,     "Sun" },
64     { ABDAY_7,     "Sat" },
65     { MON_1,       "January" },
66     { MON_12,      "December" },
67     { ABMON_1,     "Jan" },
68     { ABMON_12,    "Dec" },
69     { RADIXCHAR,   "." },
70     { THOUSEP,     "" },
71     { YESSTR,      "yes" },
72     { NOSTR,       "no" },
73     { YESEXPR,     "^[yY]" },
74     { NOEXPR,      "^[nN]" },
75     { CRNCYSTR,    "" },
76     { -1,          NULL }
77 };

79 struct langinfo_test en_us_utf8_data[] = {
80     { CODESET,     "UTF-8" },
81     { D_T_FMT,     "%B %e, %Y %I:%M:%S %p %Z" },
82     { D_FMT,       "%m/%e/%y" },
83     { T_FMT,       "%I:%M:%S %p" },
84     { T_FMT_AMPM, "%I:%M:%S %p" },
85     { AM_STR,      "AM" },
86     { PM_STR,      "PM" },
87     { ERA,          "" },
88     { ERA_D_FMT,   "" },
89     { ERA_D_T_FMT, "" },
90     { ERA_T_FMT,   "" },
91     { DAY_1,       "Sunday" },
92     { DAY_7,       "Saturday" },
93     { ABDAY_1,     "Sun" },
94     { ABDAY_7,     "Sat" },
95     { MON_1,       "January" },
96     { MON_12,      "December" },
97     { ABMON_1,     "Jan" },
98     { ABMON_12,    "Dec" },
99     { RADIXCHAR,   "." },
100    { THOUSEP,     "" },
101    { YESSTR,      "yes" },
102    { NOSTR,       "no" },
103    { YESEXPR,     "^[([yY]([eE][sS])?)" },
104    { NOEXPR,      "^[([nN]([oO])?)" },
105    { CRNCYSTR,    "-$" },
106    { -1,          NULL }
107 };

109 struct langinfo_test en_gb_latin15_data[] = {
110     { CODESET,     "ISO8859-15" },
111     { D_T_FMT,     "%e %B %Y %H:%M:%S %Z" },
112     { D_FMT,       "%d/%m/%Y" },
113     { T_FMT,       "%H:%M:%S" },
114     { T_FMT_AMPM, "%I:%M:%S %p" },
115     { AM_STR,      "AM" },
116     { PM_STR,      "PM" },
117     { ERA,          "" },
118     { ERA_D_FMT,   "" },
119     { ERA_D_T_FMT, "" },
120     { ERA_T_FMT,   "" },
121     { DAY_1,       "Sunday" },
122     { DAY_7,       "Saturday" },

```

```

123     { ABDAY_1,      "Sun" },
124     { ABDAY_7,      "Sat" },
125     { MON_1,        "January" },
126     { MON_12,       "December" },
127     { ABMON_1,      "Jan" },
128     { ABMON_12,     "Dec" },
129     { RADIXCHAR,    "." },
130     { THOUSEP,      "" },
131     { YESSTR,        "yes" },
132     { NOSTR,         "no" },
133     { YESEXPR,       "^([yY]([eE][sS])?)" },
134     { NOEXPR,        "^([nN]([oO])?)" },
135     { CRNCYSTR,      "-\\243" },
136     { -1,            NULL },
137 };

139 struct langinfo_test ru_ru_utf8_data[] = {
140     { CODESET,        "UTF-8" },
141     { D_T_FMT,        "%e %B %Y -#. %H:%M:%S %Z" },
142     { D_FMT,          "%d.%m.%y" },
143     { T_FMT,          "%H:%M:%S" },
144     { T_FMT_AMPM,     "%I:%M:%S %p" },
145     { AM_STR,         "--%3/4-%2/4-%N --%N " },
146     { PM_STR,         "-%2/4-%N --%2/4-%N --%N " },
147     { ERA,            "" },
148     { ERA_D_FMT,      "" },
149     { ERA_D_T_FMT,    "" },
150     { ERA_T_FMT,      "" },
151     { DAY_1,          "-+-%N -%N -%N -%N -%N " },
152     { DAY_7,          "%N %N -----%N -° " },
153     { ABDAY_1,        "-+-%N " },
154     { ABDAY_7,        "%N --" },
155     { MON_1,          "%N -%t-%N %N " },
156     { MON_12,         "--%2/4-%2/4-%N %N " },
157     { ABMON_1,        "%N -%t-%N " },
158     { ABMON_12,       "--%2/4-%2/4-%N " },
159     { RADIXCHAR,      "." },
160     { THOUSEP,        "" },
161     { YESSTR,         "--%2/4-%N " },
162     { NOSTR,          "-%t-%N " },
163     { YESEXPR,        "^([[- ]([- ]?)|([yY]([eE][sS])?)" },
164     { NOEXPR,         "^([[- ]([- ]?)|([N ]([oO])?)" },
165     { CRNCYSTR,       "+N %N --." },
166     { -1,             NULL },
167 };

169 struct {
170     const char *locale;
171     struct langinfo_test *loctest;
172 } locales[] = {
173     { "C",                C_data },
174     { "en_US.UTF-8",      en_us_utf8_data },
175     { "en_GB.ISO8859-15", en_gb_latin15_data },
176     { "ru_RU.UTF-8",      ru_ru_utf8_data },
177     { NULL,               NULL },
178 };

180 static void
181 test_start(const char *testName, const char *format, ...)
182 {
183     va_list args;

185     (void) printf("TEST STARTING %s (%s): ", testName, ARCH);

187     va_start(args, format);
188     (void) vprintf(format, args);

```

```

189     va_end(args);
190     (void) fflush(stdout);
191 }

193 static void
194 test_failed(const char *testName, const char *format, ...)
195 {
196     va_list args;

198     (void) printf("TEST FAILED %s (%s): ", testName, ARCH);

200     va_start(args, format);
201     (void) vprintf(format, args);
202     va_end(args);
203     (void) printf("\n");

205     if (!force)
206         (void) exit(-1);
207 }

209 static void
210 test_passed(const char *testName)
211 {
212     (void) printf("TEST PASS: %s (%s)\n", testName, ARCH);
213     (void) fflush(stdout);
214 }

216 void
217 test_nl_langinfo_l(const char *locale, struct langinfo_test *test)
218 {
219     char tname[128];
220     char *v;

222     (void) snprintf(tname, sizeof (tname), "nl_langinfo (locale %s)",
223                     locale);
224     test_start(tname, "\n");

226     v = setlocale(LC_ALL, locale);
227     if (v == NULL) {
228         test_failed(tname, "setlocale failed: %s", strerror(errno));
229     }
230     if (strcmp(v, locale) != 0) {
231         test_failed(tname, "setlocale returned %s instead of %s",
232                     v, locale);
233     }

235     for (int i = 0; test[i].value != NULL; i++) {
236         v = nl_langinfo(test[i].param);
237         if (debug)
238             (void) printf("%d: expect [%s], got [%s]\n",
239                             test[i].param, test[i].value, v);
240         if (strcmp(v, test[i].value) != 0) {
241             test_failed(tname,
242                         "param %d wrong, expected [%s], got [%s]",
243                         test[i].param, test[i].value, v);
244         }
245     }
246     test_passed(tname);
247 }

249 void
250 test_nl_langinfo_l(const char *locale, struct langinfo_test *test)
251 {
252     char tname[128];
253     char *v;
254     locale_t loc;

```

```
256     (void) snprintf(tname, sizeof (tname), "nl_langinfo_1 (locale %s)",
257                    locale);
258     test_start(tname, "\n");

260     v = setlocale(LC_ALL, "C");
261     if (v == NULL) {
262         test_failed(tname, "setlocale failed: %s", strerror(errno));
263     }
264     if (strcmp(v, "C") != 0) {
265         test_failed(tname, "setlocale returned %s instead of %s",
266                    v, "C");
267     }

269     loc = newlocale(LC_ALL_MASK, locale, NULL);
270     if (loc == NULL) {
271         test_failed(tname, "newlocale failed: %s", strerror(errno));
272     }
273
274     for (int i = 0; test[i].value != NULL; i++) {
275         v = nl_langinfo_1(test[i].param, loc);
276         if (debug)
277             (void) printf("%d: expect [%s], got [%s]\n",
278                          test[i].param, test[i].value, v);
279         if (strcmp(v, test[i].value) != 0) {
280             test_failed(tname,
281                        "param %d wrong, expected [%s], got [%s]",
282                        test[i].param, test[i].value, v);
283         }
284     }
285     test_passed(tname);
286 }
287 void
288 test_nl_langinfo(void)
289 {
290     for (int i = 0; locales[i].locale != NULL; i++) {
291         test_nl_langinfo_1(locales[i].locale, locales[i].loctest);
292         test_nl_langinfo_1(locales[i].locale, locales[i].loctest);
293     }
294 }
295 }

297 int
298 main(int argc, char **argv)
299 {
300     int optc;

302     while ((optc = getopt(argc, argv, "df")) != EOF) {
303         switch (optc) {
304             case 'd':
305                 debug++;
306                 break;
307             case 'f':
308                 force++;
309                 break;
310             default:
311                 (void) fprintf(stderr, "Usage: %s [-d]\n", argv[0]);
312                 exit(1);
313         }
314     }

316     test_nl_langinfo();

318     exit(0);
319 }
```