

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

1

5071 Thu Apr 2 18:43:03 2015

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

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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, OmniTI Computer Consulting, Inc. All rights reserved.
15 # Copyright 2015 Garrett D'Amore <garrett@damore.org>
16 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
```

```
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/cfg
27 dir path=opt/libc-tests/cfg/symbols
28 dir path=opt/libc-tests/runfiles
29 dir path=opt/libc-tests/tests
30 dir path=opt/libc-tests/tests
31 dir path=opt/libc-tests/tests/symbols
32 file path=opt/libc-tests/README mode=0444
33 file path=opt/libc-tests/bin/libctest mode=0555
34 file path=opt/libc-tests/cfg/README mode=0444
35 file path=opt/libc-tests/cfg/compilation.cfg mode=0444
36 file path=opt/libc-tests/cfg/symbols/README mode=0444
37 file path=opt/libc-tests/cfg/symbols/ctype_h.cfg mode=0444
38 file path=opt/libc-tests/cfg/symbols/dirent_h.cfg mode=0444
39 file path=opt/libc-tests/cfg/symbols/fcntl_h.cfg mode=0444
40 file path=opt/libc-tests/cfg/symbols/locale_h.cfg mode=0444
41 file path=opt/libc-tests/cfg/symbols/math_h.cfg mode=0444
42 file path=opt/libc-tests/cfg/symbols/netdb_h.cfg mode=0444
43 file path=opt/libc-tests/cfg/symbols/pthread_h.cfg mode=0444
44 file path=opt/libc-tests/cfg/symbols/signal_h.cfg mode=0444
45 file path=opt/libc-tests/cfg/symbols/stdio_h.cfg mode=0444
46 file path=opt/libc-tests/cfg/symbols/stdlib_h.cfg mode=0444
47 file path=opt/libc-tests/cfg/symbols/strings_h.cfg mode=0444
48 file path=opt/libc-tests/cfg/symbols/sys_stat_h.cfg mode=0444
49 file path=opt/libc-tests/cfg/symbols/sys_time_h.cfg mode=0444
50 file path=opt/libc-tests/cfg/symbols/sys_timeb_h.cfg mode=0444
51 file path=opt/libc-tests/cfg/symbols/ucontext_h.cfg mode=0444
52 file path=opt/libc-tests/cfg/symbols/unistd_h.cfg mode=0444
53 file path=opt/libc-tests/cfg/symbols/wchar_h.cfg mode=0444
54 file path=opt/libc-tests/cfg/symbols/wctype_h.cfg mode=0444
55 file path=opt/libc-tests/runfiles/default.run mode=0444
56 file path=opt/libc-tests/tests/fpround_test mode=0555
57 file path=opt/libc-tests/tests/fpround_test.$(ARCH) mode=0555
58 file path=opt/libc-tests/tests/fpround_test.$(ARCH64) mode=0555
```

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

2

```
32 $(i386_ONLY)file path=opt/libc-tests/tests/fpround_test.amd64 mode=0555
33 $(i386_ONLY)file path=opt/libc-tests/tests/fpround_test.i386 mode=0555
34 $(sparc_ONLY)file path=opt/libc-tests/tests/fpround_test.sparc mode=0555
35 $(sparc_ONLY)file path=opt/libc-tests/tests/fpround_test.sparcv9 mode=0555
59 file path=opt/libc-tests/tests/newlocale_test mode=0555
60 file path=opt/libc-tests/tests/newlocale_test.$(ARCH) mode=0555
61 file path=opt/libc-tests/tests/newlocale_test.$(ARCH64) mode=0555
37 $(i386_ONLY)file path=opt/libc-tests/tests/newlocale_test.amd64 mode=0555
38 $(i386_ONLY)file path=opt/libc-tests/tests/newlocale_test.i386 mode=0555
39 $(sparc_ONLY)file path=opt/libc-tests/tests/newlocale_test.sparc mode=0555
40 $(sparc_ONLY)file path=opt/libc-tests/tests/newlocale_test.sparcv9 mode=0555
62 file path=opt/libc-tests/tests/nl_langinfo_test mode=0555
63 file path=opt/libc-tests/tests/nl_langinfo_test.$(ARCH) mode=0555
64 file path=opt/libc-tests/tests/nl_langinfo_test.$(ARCH64) mode=0555
65 file path=opt/libc-tests/tests/symbols/setup mode=0555
66 file path=opt/libc-tests/tests/symbols/symbols_test.$(ARCH) mode=0555
67 file path=opt/libc-tests/tests/symbols/symbols_test.$(ARCH64) mode=0555
42 $(i386_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.amd64 mode=0555
43 $(i386_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.i386 mode=0555
44 $(sparc_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.sparc mode=0555
45 $(sparc_ONLY)file path=opt/libc-tests/tests/nl_langinfo_test.sparcv9 mode=0555
68 file path=opt/libc-tests/tests/wcsrtombs_test mode=0555
69 file path=opt/libc-tests/tests/wcsrtombs_test.$(ARCH) mode=0555
70 file path=opt/libc-tests/tests/wcsrtombs_test.$(ARCH64) mode=0555
47 $(i386_ONLY)file path=opt/libc-tests/tests/wcsrtombs_test.amd64 mode=0555
48 $(i386_ONLY)file path=opt/libc-tests/tests/wcsrtombs_test.i386 mode=0555
49 $(sparc_ONLY)file path=opt/libc-tests/tests/wcsrtombs_test.sparc mode=0555
50 $(sparc_ONLY)file path=opt/libc-tests/tests/wcsrtombs_test.sparcv9 mode=0555
71 file path=opt/libc-tests/tests/wctype_test mode=0555
72 file path=opt/libc-tests/tests/wctype_test.$(ARCH) mode=0555
73 file path=opt/libc-tests/tests/wctype_test.$(ARCH64) mode=0555
74 hardlink path=opt/libc-tests/tests/symbols/ctype_h target=setup
75 hardlink path=opt/libc-tests/tests/symbols/dirent_h target=setup
76 hardlink path=opt/libc-tests/tests/symbols/fcntl_h target=setup
77 hardlink path=opt/libc-tests/tests/symbols/locale_h target=setup
78 hardlink path=opt/libc-tests/tests/symbols/math_h target=setup
79 hardlink path=opt/libc-tests/tests/symbols/netdb_h target=setup
80 hardlink path=opt/libc-tests/tests/symbols/pthread_h target=setup
81 hardlink path=opt/libc-tests/tests/symbols/signal_h target=setup
82 hardlink path=opt/libc-tests/tests/symbols/stdio_h target=setup
83 hardlink path=opt/libc-tests/tests/symbols/stdlib_h target=setup
84 hardlink path=opt/libc-tests/tests/symbols/strings_h target=setup
85 hardlink path=opt/libc-tests/tests/symbols/sys_stat_h target=setup
86 hardlink path=opt/libc-tests/tests/symbols/sys_time_h target=setup
87 hardlink path=opt/libc-tests/tests/symbols/sys_timeb_h target=setup
88 hardlink path=opt/libc-tests/tests/symbols/ucontext_h target=setup
89 hardlink path=opt/libc-tests/tests/symbols/unistd_h target=setup
90 hardlink path=opt/libc-tests/tests/symbols/wchar_h target=setup
91 hardlink path=opt/libc-tests/tests/symbols/wctype_h target=setup
52 $(i386_ONLY)file path=opt/libc-tests/tests/wctype_test.amd64 mode=0555
53 $(i386_ONLY)file path=opt/libc-tests/tests/wctype_test.i386 mode=0555
54 $(sparc_ONLY)file path=opt/libc-tests/tests/wctype_test.sparc mode=0555
55 $(sparc_ONLY)file path=opt/libc-tests/tests/wctype_test.sparcv9 mode=0555
92 license lic_CDDL license=lic_CDDL
93 depend fmri=locale/de type=require
94 depend fmri=locale/en type=require
95 depend fmri=locale/en-extra type=require
96 depend fmri=locale/ja type=require
97 depend fmri=locale/ru type=require
98 depend fmri=system/test/testrunner type=require
```

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

1

610 Thu Apr 2 18:43:04 2015

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

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 .PARALLEL: $(SUBDIRS)
```

```
19 SUBDIRS = cfg cmd runfiles tests doc
```

```
19 SUBDIRS = cmd runfiles tests doc
```

```
21 include $(SRC)/test/Makefile.com
```

```
*****
```

```
1431 Thu Apr 2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/Makefile
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 # Copyright (c) 2012 by Delphix. All rights reserved.
15 #
```

```
17 include $(SRC)/Makefile.master
```

```
19 CFGS = README \
20     compilation.cfg \
21     symbols/README \
22     symbols/ctype_h.cfg \
23     symbols/dirent_h.cfg \
24     symbols/fcntl_h.cfg \
25     symbols/locale_h.cfg \
26     symbols/math_h.cfg \
27     symbols/netdb_h.cfg \
28     symbols/pthread_h.cfg \
29     symbols/signal_h.cfg \
30     symbols/stdio_h.cfg \
31     symbols/stdlib_h.cfg \
32     symbols/strings_h.cfg \
33     symbols/sys_stat_h.cfg \
34     symbols/sys_time_h.cfg \
35     symbols/sys_timeb_h.cfg \
36     symbols/ucontext_h.cfg \
37     symbols/unistd_h.cfg \
38     symbols/wchar_h.cfg \
39     symbols/wctype_h.cfg
```

```
41 ROOTOPTPKG = $(ROOT)/opt/libc-tests
42 ROOTOPTPKGCFG = $(ROOT)/opt/libc-tests/cfg
43 ROOTOPTPKGDIRS = $(ROOTOPTPKG) \
44     $(ROOTOPTPKGCFG) \
45     $(ROOTOPTPKGCFG)/symbols
```

```
47 FILES = $(CFGS:%=$(ROOTOPTPKGCFG)/%)
48 $(FILES) := FILEMODE = 0444
```

```
50 all: $(CFGS)
```

```
52 install: $(ROOTOPTPKG) $(ROOTOPTPKGCFG) $(FILES)
```

```
54 clean lint clobber:
```

```
56 $(ROOTOPTPKGDIRS):
57     $(INS.dir)
```

```
59 $(ROOTOPTPKGCFG)/%: % $(ROOTOPTPKGDIRS)
```

```
60 $(INS.file)
```

```

*****
3377 Thu Apr  2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/README
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 The configuration files in this directory are structured as lines,
17 where each line is made up of fields, separated by "|" characters,
18 possibly surrounded by whitespace.
19 #
20 New lines preceeded by backslashes are ignored, allowing for a continuation
21 of lines, in the usual UNIX way.
22 #
23 A line beginning with a hashmark is a comment, and is ignored, as are lines
24 consisting solely of whitespace.
25 #
26 The first field is always the "keyword", which determines the meaning and
27 presence of any other fields.
28 #
29 These files are parsed using the test_load_config() function. This
30 function has the following prototype:
31 #
32     int test_load_config(test_t, const char *, ...);
33 #
34 The variable arguments are the keywords and handling functions. These
35 must be supplied in pairs and the list is terminated with a NULL, like this:
36 #
37     test_config_load(t, "myfile.cfg", "mykeyword", keywordcb, NULL);
38 #
39 The test_config_load function will search for the named file (provided it
40 is not an absolute path) in a few locations:
41 #
42     * relative to the current directory, exactly as specified
43     * relative to $STF_SUITE/cfg/ (if $STF_SUITE is defined)
44     * relative to ../../cfg/ (if $STF_SUITE is undefined)
45     * relative to cfg/
46 #
47 The handling functions (keywordcb in the example above) have the following
48 typedef:
49 #
50     typedef int (*test_cfg_func_t)(char **fields, int nfields, char **err);
51 #
52 so for example, keywordcb should be declared thusly:
53 #
54     int keywordcb(char **fields, int nfields, char **err);
55 #
56 These functions are called each time a paired keyword is seen in the file.
57 "fields" is an array of fields, pre-split with surrounding whitespace removed,
58 and contains "nfields" items. Internal whitespace is unaffected.

```

```

60 The function should return 0 on successful handling, or -1 on failure. In
61 the event of failure, it should record an error string in "err" using
62 asprintf() or strdup(). ("err" should be unmodified otherwise.)
63 #
64 This parser is rather simplistic, and it lacks support for embedding "|"
65 fields in lines, and also doesn't support escaping, so you can't add "\"
66 at the end of a line (if you need that, leave some trailing whitespace).
67 #
68 There are also some internal limits on the length of lines (1K), and on the
69 number of fields (20). As this is only used for these test suites, this
70 should not be a significant limitation.
71 #
72 Please see ../tests/symbols/symbols_test.c for an example of correct usage.
73 #
74 Aside:
75 #
76 Astute readers may ask why invent a new configuration file, and why use
77 position based parsing instead of name value pairs. These files are
78 optimized for specific needs, and intended to support relatively dense
79 information in a format that is easy for humans to work with. JSON or XML
80 or even YAML could have served, but the overhead of a syntax was more than
81 we wanted to introduce. Test suites are free to use other formats if they
82 choose, but this simple format has the advantage of being built-in and
83 easy to use.

```

```

*****
2850 Thu Apr 2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/compilation.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Compilation environments.
18 #
19 # Each compilation environment is declared using the keyword "env", like
20 # this:
21 #
22 # env | <name> | <std> | <defs>
23 #
24 # <name> is just a symbolic name for environment.
25 # <std> indicates either c89 or c99, i.e. which C standard to compile
26 # under. This influences choice of compiler and switches.
27 # <defs> is a list of CPP style -D or -U flags to define C preprocessor
28 # symbols.
29 #
30 # Groups of compilation environments can be named, using the "env_group"
31 # keyword (this can also be used to create aliases):
32 #
33 # env_group | <name> | <envs>
34 #
35 # <name> is a name for the group or alias
36 # <envs> is a whitespace separated list of previously declared environments
37 # or environment groups (or aliases).
38 #
39 #
40 env | XPG3 | c89 | -D_XOPEN_SOURCE
41 env | XPG4 | c89 | -D_XOPEN_SOURCE -D_XOPEN_VERSION=4
42 env | SUSv1 | c89 | -D_XOPEN_SOURCE -D_XOPEN_SOURCE_EXTENDED=1
43 env | SUSv2 | c89 | -D_XOPEN_SOURCE=500
44 env | SUSv3 | c99 | -D_XOPEN_SOURCE=600
45 env | SUSv4 | c99 | -D_XOPEN_SOURCE=700
46 env | POSIX-1990 | c89 | -D_POSIX_SOURCE
47 env | POSIX-1992 | c89 | -D_POSIX_SOURCE -D_POSIX_C_SOURCE=2
48 env | POSIX-1993 | c89 | -D_POSIX_C_SOURCE=199309L
49 env | POSIX-1995 | c89 | -D_POSIX_C_SOURCE=199506L
50 env | POSIX-2001 | c99 | -D_POSIX_C_SOURCE=200112L
51 env | POSIX-2008 | c99 | -D_POSIX_C_SOURCE=200809L
52 env | C90 | c89 |
53 env | C99 | c99 |
54 #
55 #
56 # These are ordered from less inclusive (most recent) to most inclusive.
57 # This allows for us to "include" by reference.
58 #
59 env_group | POSIX-2008+ | POSIX-2008

```

```

60 env_group | POSIX-2001+ | POSIX-2008+ POSIX-2001
61 env_group | POSIX-1995+ | POSIX-2001+ POSIX-1995
62 env_group | POSIX-1993+ | POSIX-1995+ POSIX-1993
63 env_group | POSIX-1992+ | POSIX-1993+ POSIX-1992
64 env_group | POSIX-1990+ | POSIX-1992+ POSIX-1990
65 env_group | POSIX+ | POSIX-1990+
66 env_group | SUSv4+ | SUSv4 POSIX-2008+
67 env_group | SUSv3+ | SUSv3 SUSv4+ POSIX-2001+
68 env_group | SUSv2+ | SUSv2 SUSv3+
69 env_group | SUSv1+ | SUSv1 SUSv2+
70 env_group | SUS+ | SUSv1+
71 env_group | XPG4+ | XPG4 SUSv1+
72 env_group | XPG3+ | XPG3 XPG4+
73 env_group | C99+ | C99 POSIX-2001+ SUSv3+
74 env_group | C+ | C90 C99 POSIX+ SUS+
75 env_group | ALL | C+
76 #
77 #
78 # Aliases.
79 #
80 env_group | XPG4v2 | SUSv1
81 env_group | XPG4v2+ | SUSv1+
82 env_group | XPG5 | SUSv2
83 env_group | XPG5+ | SUSv2+
84 env_group | XPG6 | SUSv3
85 env_group | XPG6+ | SUSv3+
86 env_group | XPG7 | SUSv4
87 env_group | XPG7+ | SUSv4+

```

```
*****
```

```
3050 Thu Apr 2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/symbols/README
5293 desire symbol visibility test
```

```
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 The configuration files in this directory are structured using the
17 syntax defined in the ../README file. They make use of the compilation
18 environments declared in ../compilation.cfg, and are processed by the
19 symbols test.
20 #
21 We have organized the files by header file, that is the tests for symbols
22 declared in a header file (e.g. <unistd.h> appear in a file based on that
23 header file's name (e.g. unistd_h.cfg.) This is purely for convenience.
24 #
25 Within these various declarations, we have the following field types:
26 #
27 <envs> This is a list of compilation environments where the symbol
28 should be legal. To indicate that the symbol must not be legal
29 an environment group can be prefixed with "-". For example,
30 "SUS+ -SUSv4+" indicates a symbol that is legal in all SUS
31 environments up to SUSv3, and was removed in SUSv4 and subsequent
32 versions of SUS. As you can see, we can list multiple environments
33 or environment groups, and we can add or remove to previous groups
34 with subsequent ones.
35 #
36 <name> This is a symbol name. It follows the rules for C symbol names.
37 #
38 <header> This is a header file, for example, unistd.h. Conventionally,
39 the header files used should match the file where the test is
40 declared.
41 #
42 <type> This is a C type. Function types can be declared without their
43 names, e.g. "void (*)(int)". Structures (e.g. "struct stat") and
44 pointer types (e.g. "pthread_t *") are legal as well.
45 #
46 Here are the types of declarations in these files:
47 #
48 type | <name> | <header> | <envs>
49 #
50 Tests for a C type with <name>. The test verifies that a variable with
51 this type can be declared when the <header> is included.
52 #
53 value | <name> | <type> | <header> | <envs>
54 #
55 Tests for a value named <name>, of type <type>. The test attempts to
56 assign the given value to a scratch variable declared with the given
57 type. The value can be a macro or other C symbol.
58 #
59 func | <name> | <type> | <type> [; <type> ]... | <header> | <envs>
```

```
61 Tests whether a function <name>, returning the first <type>, and
62 taking arguments of following <type> values, is declared. Note that
63 the argument types are separated by semicolons. For varargs style
64 functions, leave out the ... part. For function declarations
65 that have no declared arguments, either void can specified, or
66 the type list can be omitted.
```

```
68 Examples:
```

```
70 type | size_t | sys/types.h | ALL
71 value | NULL | void * | stdlib.h | ALL
72 func strlen | int | const char *; int | string.h | ALL
```

```

*****
2311 Thu Apr  2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/symbols/ctype_h.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 #
17 # Definitions found in ctype.h
18 #
19 #
20 #
21 # Types.
22 #
23 type | locale_t | ctype.h | -ALL +SUSv4+
24 #
25 #
26 # Values.
27 #
28 #
29 #
30 # Functions
31 #
32 func | isalnum | int | int | ctype.h | +ALL
33 func | isalpha | int | int | ctype.h | +ALL
34 func | isascii | int | int | ctype.h | -ALL +XPG3+
35 func | isblank | int | int | ctype.h | -ALL +C99+
36 func | iscntrl | int | int | ctype.h | +ALL
37 func | isdigit | int | int | ctype.h | +ALL
38 func | isgraph | int | int | ctype.h | +ALL
39 func | islower | int | int | ctype.h | +ALL
40 func | isprint | int | int | ctype.h | +ALL
41 func | ispunct | int | int | ctype.h | +ALL
42 func | isspace | int | int | ctype.h | +ALL
43 func | isupper | int | int | ctype.h | +ALL
44 func | isxdigit | int | int | ctype.h | +ALL
45 func | toascii | int | int | ctype.h | -ALL +XPG3+
46 func | tolower | int | int | ctype.h | +ALL
47 func | toupper | int | int | ctype.h | +ALL
48 func | _tolower | int | int | ctype.h | -ALL +XPG3+
49 func | _toupper | int | int | ctype.h | -ALL +XPG3+
50 #
51 func | isalnum_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
52 func | isalpha_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
53 func | isblank_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
54 func | iscntrl_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
55 func | isdigit_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
56 func | isgraph_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
57 func | islower_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
58 func | isprint_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+
59 func | ispunct_1 | int | int; locale_t | ctype.h | -ALL +SUSv4+

```

60	func	isspace_1	int	int; locale_t	ctype.h	-ALL +SUSv4+
61	func	isupper_1	int	int; locale_t	ctype.h	-ALL +SUSv4+
62	func	isxdigit_1	int	int; locale_t	ctype.h	-ALL +SUSv4+
63	func	tolower_1	int	int; locale_t	ctype.h	-ALL +SUSv4+
64	func	toupper_1	int	int; locale_t	ctype.h	-ALL +SUSv4+

```

*****
1545 Thu Apr 2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/symbols/dirent_h.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in dirent.h
18 #
19 # Note that this header is not defined prior to the POSIX & XPG3
20 # specifications. Therefore, applications which #include it in earlier
21 # environments may reasonably obtain their contents, although such use
22 # would not be conformant to those specifications.
23 #
24 # As a result, we only test the newer specifications.
25 #
26 #
27 #
28 # Types.
29 #
30 type | DIR * | dirent.h | XPG3+ POSIX+
31 type | struct dirent | dirent.h | XPG3+ POSIX+
32 #
33 #
34 # Values.
35 #
36 #
37 #
38 # Functions
39 #
40 #
41 func | opendir | | | \
42 | DIR * | | | \
43 | const char * | | | \
44 | dirent.h | POSIX+ XPG3+
45 #
46 func | readdir | | | \
47 | struct dirent * | | | \
48 | DIR * | | | \
49 | dirent.h | POSIX+ XPG3+
50 #
51 # We are not going to test the draft interface.
52 func | readdir_r | | | \
53 | int | | | \
54 | DIR *, struct dirent *, struct dirent ** | | | \
55 | dirent.h | POSIX-1995+
56 #
57 func | rewinddir | | | \
58 | void | | | \
59 | DIR * | dirent.h | POSIX+ XPG3+

```

```

61 func | seekdir | | | \
62 | void | | | \
63 | DIR *, long | | | \
64 | dirent.h | -POSIX+ XPG3+
65 #
66 func | telldir | | | \
67 | long | | | \
68 | DIR * | | | \
69 | dirent.h | -POSIX+ XPG3+

```

```

*****
1587 Thu Apr 2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/symbols/fcntl_h.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in fcntl.h
18 #
19 #
20 #
21 # Types.
22 #
23 type | mode_t | fcntl.h | ALL
24 type | off_t | fcntl.h | ALL
25 type | pid_t | fcntl.h | ALL
26 type | struct flock | fcntl.h | ALL
27 #
28 #
29 # Values.
30 #
31 value | O_ACCMODE | int | fcntl.h | POSIX+ SUS+
32 value | O_APPEND | int | fcntl.h | POSIX+ SUS+
33 value | O_CLOEXEC | int | fcntl.h | POSIX+ SUS+
34 value | O_CREAT | int | fcntl.h | POSIX+ SUS+
35 value | O_DSYNC | int | fcntl.h | XPG3+ POSIX-1993+
36 value | O_EXCL | int | fcntl.h | POSIX+ SUS+
37 value | O_NDELAY | int | fcntl.h | -POSIX-1993+
38 value | O_NOCTTY | int | fcntl.h | POSIX+ SUS+
39 value | O_NONBLOCK | int | fcntl.h | POSIX+ SUS+
40 value | O_RDONLY | int | fcntl.h | POSIX+ SUS+
41 value | O_RDWR | int | fcntl.h | POSIX+ SUS+
42 value | O_RSYNC | int | fcntl.h | XPG3+ POSIX-1993+
43 value | O_SYNC | int | fcntl.h | XPG3+ POSIX-1993+
44 value | O_WRONLY | int | fcntl.h | POSIX+ SUS+
45 #
46 #
47 # Functions
48 #
49 func | creat | | | \
50 | int | | | \
51 | const char *; mode_t | | | \
52 | fcntl.h | ALL | | | \
53 #
54 func | open | | | \
55 | int | | | \
56 | const char *; mode_t | | | \
57 | fcntl.h | ALL | | | \
58 #
59 func | fcntl | | | \

```

```

60 int
61 int; int
62 fcntl.h | ALL

```

```

*****
1440 Thu Apr  2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/symbols/locale_h.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in locale.h
18 #
19 #
20 #
21 # Types.
22 #
23 type | locale_t | locale.h | -ALL SUSv4+
24 type | struct lconv | locale.h | ALL
25 #
26 #
27 # Values.
28 #
29 value | LC_CTYPE | int | locale.h | ALL
30 value | LC_COLLATE | int | locale.h | ALL
31 value | LC_TIME | int | locale.h | ALL
32 value | LC_NUMERIC | int | locale.h | ALL
33 value | LC_MONETARY | int | locale.h | ALL
34 value | LC_MESSAGES | int | locale.h | ALL
35 value | LC_GLOBAL_LOCALE | locale_t | locale.h | -ALL SUSv4+
36 value | NULL | void * | locale.h | ALL
37 #
38 #
39 # Functions
40 #
41 func | duplocale | | | \
42 | locale_t | | | \
43 | locale_t | | | \
44 | locale.h | -ALL SUSv4+ | |
45 #
46 func | newlocale | | | \
47 | locale_t | | | \
48 | int; const char *; locale_t | | | \
49 | locale.h | -ALL SUSv4+ | |
50 #
51 func | freelocale | | | \
52 | void | | | \
53 | locale_t | | | \
54 | locale.h | -ALL SUSv4+ | |
55 #
56 func | localeconv | | | \
57 | struct lconv * | | | \
58 | void | | | \
59 | locale.h | ALL | |

```

```

61 func | setlocale | | | \
62 | char * | | | \
63 | int; const char * | | | \
64 | locale.h | ALL | |

```

new/usr/src/test/libc-tests/cfg/symbols/math_h.cfg

1

797 Thu Apr 2 18:43:04 2015

new/usr/src/test/libc-tests/cfg/symbols/math_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
```

```
16 #
17 # Definitions found in math.h
18 #
```

```
21 #
22 # Types.
23 #
```

```
25 #
26 # Values.
```

```
27 #
28 value | M_PI           | double   | math.h | SUS+
29 value | INFINITY        | float    | math.h | C99+
30 value | NAN              | float    | math.h | C99+
31 value | FP_NAN           | int      | math.h | C99+
```

```
33 #
34 # Functions
```

```
35 #
36 func  | cos              | double   | double  | math.h | ALL
37 func  | cosf             | float    | float   | math.h | -ALL +C99+
```

new/usr/src/test/libc-tests/cfg/symbols/netdb_h.cfg

1

1038 Thu Apr 2 18:43:04 2015

new/usr/src/test/libc-tests/cfg/symbols/netdb_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
```

```
16 #
17 # Definitions found in netdb.h
18 #
```

```
20 #
21 # Types.
22 #
```

```
24 #
25 # Values.
```

```
26 #
27 #value | h_errno | int | netdb.h | -ALL SUS+ -SUSv4+
28 #value | HOST_NOT_FOUND | int | netdb.h | -ALL SUS+ -SUSv4+
29 #value | NO_DATA | int | netdb.h | -ALL SUS+ -SUSv4+
30 #value | NO_RECOVERY | int | netdb.h | -ALL SUS+ -SUSv4+
31 #value | TRY_AGAIN | int | netdb.h | -ALL SUS+ -SUSv4+
```

```
33 #
34 # Functions
35 #
```

```
37 func | endhostent | \
38      | void | \
39      | void | \
40      | netdb.h | \
41      | SUS+ |
```

```
43 func | gethostent | \
44      | struct hostent * | \
45      | void | \
46      | netdb.h | \
47      | SUS+ |
```

```
49 func | sethostent | \
50      | void | \
51      | int | \
52      | netdb.h | \
53      | SUS+ |
```

```

*****
1571 Thu Apr 2 18:43:04 2015
new/usr/src/test/libc-tests/cfg/symbols/pthread_h.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in pthread.h
18 #
19 # Note that this header is not defined prior to the POSIX 1995 & SUSv2
20 # specifications. Therefore, applications which #include it in earlier
21 # environments may reasonably obtain their contents, although such use
22 # would not be conformant to those specifications.
23 #
24 # As a result, we only test the newer specifications.
25 #
26 #
27 #
28 # Types.
29 #
30 type | pthread_t * | pthread.h | POSIX-1995+ SUSv2+
31 #
32 #
33 # Values.
34 #
35 value | PTHREAD_SCOPE_PROCESS | int | pthread.h | POSIX-1995+ SUSv2+
36 value | PTHREAD_SCOPE_SYSTEM | int | pthread.h | POSIX-1995+ SUSv2+
37 #
38 #
39 # Functions
40 #
41 func | pthread_create | \
42 | int | \
43 | pthread_t *; const pthread_attr_t *; void (*)(void *); void * | \
44 | pthread.h | \
45 | POSIX-1995+ SUSv2+ | \
46 #
47 func | pthread_self | \
48 | pthread_t | \
49 | void | \
50 | pthread.h | \
51 | POSIX-1995+ SUSv2+ | \
52 #
53 func | pthread_attr_getscope | \
54 | int | \
55 | const pthread_attr_t *; int * | \
56 | pthread.h | \
57 | POSIX-1995+ SUSv2+ | \
58 #
59 func | pthread_attr_setscope | \

```

```

60 int | \
61 pthread_attr_t *; int | \
62 pthread.h | \
63 POSIX-1995+ SUSv2+ | \

```

new/usr/src/test/libc-tests/cfg/symbols/signal_h.cfg

1

635 Thu Apr 2 18:43:04 2015

new/usr/src/test/libc-tests/cfg/symbols/signal_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in signal.h
18 #
19 #
20 #
21 # Types.
22 #
23 type | ucontext_t | signal.h | -ALL SUS+
24 #
25 #
26 # Values.
27 #
28 value | SIG_DFL | void (*)(int) | signal.h | ALL
29 #
30 #
31 # Functions.
32 #
```

new/usr/src/test/libc-tests/cfg/symbols/stdio_h.cfg

1

1250 Thu Apr 2 18:43:04 2015

new/usr/src/test/libc-tests/cfg/symbols/stdio_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
```

```
16 #
17 # Definitions found in stdio.h
18 #
```

```
20 #
21 # Types.
22 #
23 type | FILE | stdio.h | ALL
24 type | fpos_t | stdio.h | ALL
25 type | size_t | stdio.h | ALL
```

```
27 #
28 # Values.
29 #
30 value | BUFSIZ | int | stdio.h | ALL
31 value | EOF | int | stdio.h | ALL
32 value | NULL | void * | stdio.h | ALL
33 value | stderr | FILE * | stdio.h | ALL
34 value | stdin | FILE * | stdio.h | ALL
35 value | stdout | FILE * | stdio.h | ALL
36 value | optarg | char * | stdio.h | -ALL +XPG3+ -SUSv3+
37 value | optind | int | stdio.h | -ALL +XPG3+ -SUSv3+
38 value | opterr | int | stdio.h | -ALL +XPG3+ -SUSv3+
39 value | optopt | int | stdio.h | -ALL +XPG3+ -SUSv3+
```

```
41 #
42 # Functions
43 #
```

```
45 func | printf | | \
46 | int | | \
47 | const char * | | \
48 | stdio.h | ALL |
49
50 func | rename | | \
51 | int | | \
52 | const char *; const char * | | \
53 | stdio.h | ALL |
```

1045 Thu Apr 2 18:43:04 2015

new/usr/src/test/libc-tests/cfg/symbols/stdlib_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #

```

```

16 #
17 # Definitions found in stdlib.h
18 #

```

```

20 #
21 # Types.
22 #
23 type | size_t | stdlib.h | ALL

```

```

25 #
26 # Values.
27 #
28 value | EXIT_FAILURE | int | stdlib.h | ALL
29 value | EXIT_SUCCESS | int | stdlib.h | ALL
30 value | NULL | void * | stdlib.h | ALL

```

```

32 #
33 # Functions
34 #

```

```

35 func | calloc | \
36 | void * | \
37 | size_t; size_t | \
38 | stdlib.h | \
39 | ALL

```

```

41 func | exit | \
42 | void | \
43 | int | \
44 | stdlib.h | \
45 | ALL

```

```

47 func | free | \
48 | void | \
49 | void * | \
50 | stdlib.h | \
51 | ALL

```

```

53 func | malloc | \
54 | void * | \
55 | size_t | \
56 | stdlib.h | \
57 | ALL

```

new/usr/src/test/libc-tests/cfg/symbols/strings_h.cfg

1

545 Thu Apr 2 18:43:04 2015

new/usr/src/test/libc-tests/cfg/symbols/strings_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in strings.h
18 #
19 #
20 #
21 # Types.
22 #
23 #
24 #
25 # Values.
26 #
27 #
28 #
29 # Functions
30 #
```

new/usr/src/test/libc-tests/cfg/symbols/sys_stat_h.cfg

1

1195 Thu Apr 2 18:43:04 2015

new/usr/src/test/libc-tests/cfg/symbols/sys_stat_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
```

```
16 #
17 # Definitions found in sys/stat.h
18 #
```

```
20 #
21 # Types.
```

```
22 #
23 type | mode_t | sys/stat.h | ALL
24 type | struct stat | sys/stat.h | ALL
```

```
26 #
27 # Values.
```

```
28 #
29 value | S_IFMT | mode_t | sys/stat.h | ALL
30 value | S_IFBLK | mode_t | sys/stat.h | ALL
```

```
32 #
33 # Functions
```

```
34 #
35 func | chmod | \
36 | int | \
37 | const char *; mode_t | \
38 | sys/stat.h | POSIX+ XPG3+
```

```
40 func | fchmod | \
41 | int | \
42 | int; mode_t | \
43 | sys/stat.h | -POSIX+ -XPG3+ POSIX-1993+ SUS+
```

```
45 func | fstat | \
46 | int | \
47 | int; struct stat * | \
48 | sys/stat.h | XPG3+ POSIX+
```

```
50 func | mkdir | \
51 | int | \
52 | const char *; mode_t | \
53 | sys/stat.h | XPG3+ POSIX+
```

```
55 func | mknod | \
56 | int | \
57 | const char *; mode_t; dev_t | \
58 | sys/stat.h | -XPG3+ -POSIX+ SUS+
```

new/usr/src/test/libc-tests/cfg/symbols/sys_time_h.cfg

1

883 Thu Apr 2 18:43:05 2015

new/usr/src/test/libc-tests/cfg/symbols/sys_time_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
```

```
16 #
17 # Definitions found in sys/time.h
18 #
```

```
20 #
21 # Types.
22 #
23 #type | time_t | sys/timeb.h | -ALL SUS+ -SUSv4+
24 #type | struct timeb | sys/timeb.h | -ALL SUS+ -SUSv4+
```

```
26 #
27 # Values.
28 #
```

```
30 #
31 # Functions
32 #
```

```
33 func | futimesat | \
34      | int | \
35      | int; const char *; const struct timeval [2] | \
36      | sys/time.h | \
37      | -SUS+ -POSIX+
```

```
39 func | utimes | \
40      | int | \
41      | const char *; const struct timeval [2] | \
42      | sys/time.h | -POSIX+ -XPG3+ SUS+
```

new/usr/src/test/libc-tests/cfg/symbols/sys_timeb_h.cfg

1

547 Thu Apr 2 18:43:05 2015

new/usr/src/test/libc-tests/cfg/symbols/sys_timeb_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in sys/timeb.h
18 #
19 #
20 #
21 # Types.
22 #
23 #
24 #
25 # Values.
26 #
27 #
28 #
29 # Functions
30 #
```

new/usr/src/test/libc-tests/cfg/symbols/ucontext_h.cfg

1

546 Thu Apr 2 18:43:05 2015

new/usr/src/test/libc-tests/cfg/symbols/ucontext_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in ucontext.h
18 #
19 #
20 #
21 # Types.
22 #
23 #
24 #
25 # Values.
26 #
27 #
28 #
29 # Functions
30 #
```

```

*****
3574 Thu Apr  2 18:43:05 2015
new/usr/src/test/libc-tests/cfg/symbols/unistd.h.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in unistd.h
18 #
19 #
20 #
21 # Types.
22 #
23 type | pid_t | unistd.h | POSIX+ SUS+
24 #
25 #
26 # Values.
27 #
28 # Note that the standard requires the user declare environ.
29 # value | environ | char ** | unistd.h | POSIX+ SUS+
30 value | _CS_PATH | int | unistd.h | SUS+
31 #
32 value | _CS_POSIX_V6_ILP32_OFF32_CFLAGS | int | unistd.h | SUSv3+
33 value | _CS_POSIX_V6_ILP32_OFF32_LDFLAGS | int | unistd.h | SUSv3+
34 value | _CS_POSIX_V6_ILP32_OFF32_LIBS | int | unistd.h | SUSv3+
35 value | _CS_POSIX_V6_ILP32_OFFBIG_CFLAGS | int | unistd.h | SUSv3+
36 value | _CS_POSIX_V6_ILP32_OFFBIG_LDFLAGS | int | unistd.h | SUSv3+
37 value | _CS_POSIX_V6_ILP32_OFFBIG_LIBS | int | unistd.h | SUSv3+
38 value | _CS_POSIX_V6_LP64_OFF64_CFLAGS | int | unistd.h | SUSv3+
39 value | _CS_POSIX_V6_LP64_OFF64_LDFLAGS | int | unistd.h | SUSv3+
40 value | _CS_POSIX_V6_LP64_OFF64_LIBS | int | unistd.h | SUSv3+
41 value | _CS_POSIX_V6_LPBIG_OFFBIG_CFLAGS | int | unistd.h | SUSv3+
42 value | _CS_POSIX_V6_LPBIG_OFFBIG_LDFLAGS | int | unistd.h | SUSv3+
43 value | _CS_POSIX_V6_LPBIG_OFFBIG_LIBS | int | unistd.h | SUSv3+
44 value | _CS_POSIX_V6_WIDTH_RESTRICTED_ENVS | int | unistd.h | SUSv3+
45 #
46 #
47 # Functions
48 #
49 func | access | \
50 | int | \
51 | const char *, int | \
52 | unistd.h | POSIX+ SUS+
53 #
54 func | chown | \
55 | int | \
56 | const char *, uid_t, gid_t | \
57 | unistd.h | POSIX+ SUS+
58 #
59 func | execl | \

```

```

60 | int | \
61 | const char *, const char * | \
62 | unistd.h | POSIX+ SUS+
63 #
64 func | execl | \
65 | int | \
66 | const char *, const char *, char *, char *const [] | \
67 | unistd.h | POSIX+ SUS+
68 #
69 func | execlp | \
70 | int | \
71 | const char *, const char *, char * | \
72 | unistd.h | POSIX+ SUS+
73 #
74 func | execv | \
75 | int | \
76 | const char *, char *const [] | \
77 | unistd.h | POSIX+ SUS+
78 #
79 func | execve | \
80 | int | \
81 | const char *, char *const [], char *const [] | \
82 | unistd.h | POSIX+ SUS+
83 #
84 func | execvp | \
85 | int | \
86 | const char *, char *const [] | \
87 | unistd.h | POSIX+ SUS+
88 #
89 func | fchown | \
90 | int | \
91 | int; uid_t; gid_t | \
92 | unistd.h | -POSIX+ SUS+
93 #
94 func | getlogin | \
95 | char * | \
96 | void | \
97 | unistd.h | POSIX+ SUS+
98 #
99 func | getlogin_r | \
100 | int | \
101 | char *, size_t | \
102 | unistd.h | -POSIX+ -SUS+ +POSIX-1995+ SUSv2+
103 #
104 func | lchown | \
105 | int | \
106 | const char *, uid_t; gid_t | \
107 | unistd.h | -POSIX+ SUS+
108 #
109 func | link | \
110 | int | \
111 | const char *, const char * | \
112 | unistd.h | POSIX+ SUS+
113 #
114 # XPG3 may have put this here incorrectly (Open Group says no..., but...)
115 # Probably this is actually our error, and we should kill it, but we can
116 # do that when kill off XPG3 support altogether.
117 func | rename | \
118 | int | \
119 | const char *, const char * | \
120 | unistd.h | -POSIX+ +XPG3 -XPG4+
121 #
122 func | symlink | \
123 | int | \
124 | const char *, const char * | \
125 | unistd.h | -XPG3+ -POSIX+ SUS+

```

```
127 func | ttyname          | \
128     char *              | \
129     int                  | \
130     unistd.h | POSIX+ SUS+
132 func | ttyname_r            | \
133     int                  | \
134     int; char *; size_t  | \
135     unistd.h | -POSIX+ -SUS+ +POSIX-1995+ SUSv2+
137 func | unlink              | \
138     int                  | \
139     const char *         | \
140     unistd.h | POSIX+ XPG3+
```

```

*****
1764 Thu Apr 2 18:43:05 2015
new/usr/src/test/libc-tests/cfg/symbols/wchar_h.cfg
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
15 #
16 #
17 # Definitions found in wchar.h
18 #
19 #
20 #
21 # Types.
22 #
23 type | locale_t | wchar.h | -ALL SUSv4+
24 type | wctype_t | wchar.h | -ALL XPG3+
25 #
26 #
27 # Values.
28 #
29 value | NULL | void * | wchar.h | ALL
30 #
31 #
32 # Functions
33 #
34 func | wcpncpy | | \
35      | wchar_t * | | \
36      | wchar_t *; const wchar_t * | | \
37      | wchar.h | -ALL SUSv4+
38 #
39 func | wcpncpy | | \
40      | wchar_t * | | \
41      | wchar_t *; const wchar_t *; size_t | | \
42      | wchar.h | -ALL SUSv4+
43 #
44 func | wcsdup | | \
45      | wchar_t * | | \
46      | const wchar_t * | | \
47      | wchar.h | -ALL SUSv4+
48 #
49 func | wcscasecmp | | \
50      | int | | \
51      | const wchar_t *; const wchar_t * | | \
52      | wchar.h | -ALL SUSv4+
53 #
54 func | wcscasecmp_l | | \
55      | int | | \
56      | const wchar_t *; const wchar_t *; locale_t | | \
57      | wchar.h | -ALL SUSv4+
58 #
59 # technically this is new in C90-Amd1, but we don't have a test for it.

```

```

60 func | wcslen | | \
61      | size_t | | \
62      | const wchar_t * | | \
63      | wchar.h | ALL
64 #
65 func | wcsnsecmp | | \
66      | int | | \
67      | const wchar_t *; const wchar_t *; size_t | | \
68      | wchar.h | -ALL SUSv4+
69 #
70 func | wcsnsecmp_l | | \
71      | int | | \
72      | const wchar_t *; const wchar_t *; size_t; locale_t | | \
73      | wchar.h | -ALL SUSv4+
74 #
75 func | wcsnlen | | \
76      | size_t | | \
77      | const wchar_t *; size_t | | \
78      | wchar.h | -ALL SUSv4+
79 #
80 func | wcs wcs | | \
81      | wchar_t * | | \
82      | const wchar_t *; const wchar_t * | | \
83      | wchar.h | -ALL XPG4+

```

new/usr/src/test/libc-tests/cfg/symbols/wctype_h.cfg

1

704 Thu Apr 2 18:43:05 2015

new/usr/src/test/libc-tests/cfg/symbols/wctype_h.cfg

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 #
```

```
16 #
17 # Definitions found in wctype.h
18 #
```

```
20 #
21 # Types.
22 #
23 type | wint_t | wctype.h | ALL
```

```
25 #
26 # Values.
27 #
28 value | WEOF | wint_t | wctype.h | ALL -XPG4+ SUSv2+
```

```
30 #
31 # Functions
32 #
33 func | iswalnum |
34 | int |
35 | wint_t |
36 | wctype.h | ALL
```

```
| \
| \
| \
```

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

1

1109 Thu Apr 2 18:43:05 2015

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

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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/fpround_test]
26 #
27 [/opt/libc-tests/tests/newlocale_test]
28 #
29 [/opt/libc-tests/tests/nl_langinfo_test]
30 #
31 [/opt/libc-tests/tests/wcsrtombs_test]
32 #
33 [/opt/libc-tests/tests/wctype_test]
```

```
35 [/opt/libc-tests/tests/symbols]
36 pre = setup
37 tests = [
38     'ctype_h',
39     'dirent_h',
40     'fcntl_h',
41     'locale_h',
42     'math_h',
43     'netdb_h',
44     'pthread_h',
45     'signal_h',
46     'stdio_h',
47     'stdlib_h',
48     'strings_h',
49     'sys_stat_h',
50     'sys_time_h',
51     'sys_timeb_h',
52     'ucontext_h',
53     'unistd_h',
54     'wchar_h',
55     'wctype_h'
56 ]
```

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

1

646 Thu Apr 2 18:43:05 2015

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

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
15 #
```

```
17 SUBDIRS = fpround newlocale nl_langinfo symbols wcsrtombs wctype
17 SUBDIRS = fpround newlocale nl_langinfo wcsrtombs wctype
```

```
19 include $(SRC)/Makefile.master
20 include $(SRC)/test/Makefile.com
```

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

1

```
*****
2253 Thu Apr  2 18:43:05 2015
new/usr/src/test/libc-tests/tests/Makefile.com
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 include $(SRC)/Makefile.master
18 include $(SRC)/cmd/Makefile.cmd
19 include $(SRC)/test/Makefile.com
21 $(OBJS_OVERRIDE)OBJS = $(PROG).o test_common.o
22 OBJS32 = $(OBJS:%.o=%.$(MACH).o)
23 PROG32 = $(PROG).$(MACH)
25 $(BUILD64) OBJS64 = $(OBJS:%.o=%.$(MACH64).o)
26 $(BUILD64) PROG64= $(PROG).$(MACH64)
28 $(OBJS_OVERRIDE)SRCS = $(PROG).c ../common/test_common.c
30 C99MODE = -xc99=%all
31 LINTFLAGS += -I../common -DARCH=\"ARCH\" -DLINT
32 CPPFLAGS += -I$(ROOT)/usr/include -I../common
33 CPPFLAGS += -I../common
34 ROOTOPTPKG = $(ROOT)/opt/libc-tests
35 TESTDIR = $(ROOTOPTPKG)/tests/$(TESTSUBDIR)
36 TESTDIR = $(ROOTOPTPKG)/tests
37 CMDS = $(PROG32:%=$(TESTDIR)/%) $(PROG64:%=$(TESTDIR)/%) \
38         $(KSHPROG:%=$(TESTDIR)/%) $(ARCHPROG:%=$(TESTDIR)/%) \
39         $(EXTRAPROG:%=$(TESTDIR)/%)
38         $(KSHPROG:%=$(TESTDIR)/%) $(ARCHPROG:%=$(TESTDIR)/%)
41 $(CMDS) := FILEMODE = 0555
43 all: $(PROG32) $(PROG64) $(KSHPROG) $(ARCHPROG) $(SUBDIRS)
45 $(PROG32): $(OBJS32)
46         $(LINK.c) $(OBJS32) -o $@ $(LDLIBS)
47         $(POST_PROCESS)
49 $(PROG64): $(OBJS64)
50         $(LINK64.c) $(OBJS64) -o $@ $(LDLIBS64)
51         $(POST_PROCESS)
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 $@
63 %.$(MACH).o: %.c
64         $(COMPILE.c) -o $@ $(CFLAGS_$(MACH)) -DARCH=\"$(MACH)\" $<
66 %.$(MACH).o: ../common/%.c
67         $(COMPILE.c) -o $@ $(CFLAGS_$(MACH)) -DARCH=\"$(MACH)\" $<
69 %.$(MACH64).o: %.c
70         $(COMPILE64.c) -o $@ $(CFLAGS_$(MACH64)) -DARCH=\"$(MACH64)\" $<
72 %.$(MACH64).o: ../common/%.c
73         $(COMPILE64.c) -o $@ $(CFLAGS_$(MACH64)) -DARCH=\"$(MACH64)\" $<
75 install: $(SUBDIRS) $(CMDS)
77 lint: lint_SRCS
79 clobber: clean
80         -$(RM) $(PROG32) $(PROG64) $(KSHPROG) $(ARCHPROG)
82 clean:
83         -$(RM) $(OBJS32) $(OBJS64)
85 $(CMDS): $(TESTDIR) $(PROG32) $(PROG64) $(KSHPROG) $(ARCHPROG)
87 $(TESTDIR):
88         $(INS.dir)
90 $(TESTDIR)/%: %
91         $(INS.file)
```

new/usr/src/test/libc-tests/tests/common/test_common.c

1

```
*****
7523 Thu Apr  2 18:43:05 2015
new/usr/src/test/libc-tests/tests/common/test_common.c
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14  * Copyright 2014 Garrett D'Amore <garrett@damore.org>
15 */

16 /*
17  * Common handling for test programs.
18 */

20 #include <stdio.h>
21 #include <stdlib.h>
22 #include <stdarg.h>
23 #include <string.h>
24 #include <errno.h>
25 #include <pthread.h>
26 #include <ctype.h>
27 #include <unistd.h>
28 #include <sys/param.h>
29 #include "test_common.h"

31 static int debug = 0;
32 static int force = 0;
33 static pthread_mutex_t lk;

35 static int passes;
36 static int tests;

38 struct test {
39     char          *name;
40     int           ntids;
41     pthread_t     *tids;
42     int           fails;
43     void          *arg;
44     void          (*func)(test_t t, void *);
45 };
    unchanged portion omitted

59 test_t
60 test_start(const char *format, ...)
61 {
62     va_list args;
63     test_t t;
64     char *s;

66     t = calloc(1, sizeof(*t));
67     va_start(args, format);
68     (void) vasprintf(&s, format, args);
69     va_end(args);
```

new/usr/src/test/libc-tests/tests/common/test_common.c

2

```
71     (void) asprintf(&t->name, "%s (%s)", s, ARCH);
72     free(s);

74     (void) pthread_mutex_lock(&lk);
75     (void) printf("TEST STARTING %s:\n", t->name);
76     (void) fflush(stdout);
77     (void) pthread_mutex_unlock(&lk);

79 #ifdef LINT
80     /* We inject references to make avoid name unused warnings */
81     test_run(0, NULL, NULL, NULL);
82     test_debugf(t, NULL);
83     test_failed(t, NULL);
84     test_passed(t);
85     test_set_debug();
86     test_set_force();
87     test_summary();
88     (void) test_load_config(t, NULL, NULL);
89 #endif

91     tests++;
92     return (t);

93 }

95 void
96 test_failed(test_t t, const char *format, ...)
97 {
98     va_list args;

100     (void) pthread_mutex_lock(&lk);
101     if (t == NULL) {
102         (void) printf("FAILURE: ");
103         va_start(args, format);
104         (void) vprintf(format, args);
105         va_end(args);
106         (void) printf("\n");
107         (void) fflush(stdout);
108         (void) pthread_mutex_unlock(&lk);
109         return;
110     }
111     if (force || (t->ntids > 0)) {
112         (void) printf("TEST FAILING %s: ", t->name);
113     } else {
114         (void) printf("TEST FAILED %s: ", t->name);
115     }

117     va_start(args, format);
118     (void) vprintf(format, args);
119     va_end(args);
120     (void) printf("\n");
121     (void) fflush(stdout);
122     (void) pthread_mutex_unlock(&lk);

124     t->fails++;
125     if (!force) {
126         if (t->ntids > 0) {
127             pthread_exit(NULL);
128         } else {
129             (void) exit(EXIT_FAILURE);
130         }
131     }
132 }

134 void
```

```

135 test_passed(test_t t)
136 {
137     if (t == NULL) {
138         return;
139     }
140     if (t->ntids > 0) {
141         if (debug) {
142             (void) pthread_mutex_lock(&lk);
143             (void) printf("TEST PASSING: %s\n", t->name);
144             (void) pthread_mutex_unlock(&lk);
145         }
146         return;
147     }
148     (void) pthread_mutex_lock(&lk);
149     if (t->fails == 0) {
150         passes++;
151         (void) printf("TEST PASS: %s\n", t->name);
152     } else {
153         (void) printf("TEST FAILED: %d failures\n", t->fails);
154     }
155     (void) fflush(stdout);
156     (void) pthread_mutex_unlock(&lk);
157     free(t->name);
158     if (t->tids) {
159         free(t->tids);
160     }
161     free(t);
162 }

164 void
165 test_summary(void)
166 {
167     if (passes == tests) {
168         (void) printf("TEST SUMMARY: %d / %d (ok)\n", passes, tests);
169     } else {
170         (void) printf("TEST SUMMARY: %d / %d (%d failing)\n",
171             passes, tests, tests - passes);
172     }
173 }

175 void
176 test_debugf(test_t t, const char *format, ...)
177 {
178     va_list args;

180     if (!debug)
181         return;

183     (void) pthread_mutex_lock(&lk);
184     if (t) {
185         (void) printf("TEST DEBUG %s: ", t->name);
186     } else {
187         (void) printf("TEST DEBUG: ");
188     }

189     va_start(args, format);
190     (void) vprintf(format, args);
191     va_end(args);
192     (void) printf("\n");
193     (void) fflush(stdout);
194     (void) pthread_mutex_unlock(&lk);
195 }
_____unchanged_portion_omitted_____

205 void
206 test_run(int nthr, void (*func)(test_t, void *), void *arg,

```

```

207     const char *tname, ...)
208 {
209     test_t      t;
210     char        *s;
211     va_list     args;

213     t = calloc(1, sizeof (*t));
214     t->ntids = nthr;
215     t->tids = calloc(nthr, sizeof (pthread_t));
216     t->func = func;
217     t->arg = arg;

219     va_start(args, tname);
220     (void) vasprintf(&s, tname, args);
221     va_end(args);

223     (void) asprintf(&t->name, "%s (%s)", s, ARCH);
224     free(s);

226     (void) pthread_mutex_lock(&lk);
227     (void) printf("TEST STARTING %s:\n", t->name);
228     (void) fflush(stdout);
229     (void) pthread_mutex_unlock(&lk);

231     test_debugf(t, "running %d threads", nthr);

233     for (int i = 0; i < nthr; i++) {
234         test_debugf(t, "started thread %d", i);
235         (void) pthread_create(&t->tids[i], NULL, test_thr_one, t);
236     }

238     for (int i = 0; i < nthr; i++) {
239         (void) pthread_join(t->tids[i], NULL);
240         test_debugf(t, "thread %d joined", i);
241         t->ntids--;
242     }
243     test_passed(t);
244 }

246 void
247 test_trim(char **ptr)
248 {
249     char *p = *ptr;
250     while (isspace(*p)) {
251         p++;
252     }
253     *ptr = p;
254     p += strlen(p);
255     while ((--p >= *ptr) && (isspace(*p))) {
256         *p = 0;
257     }
258 }

260 #define MAXCB      20
261 #define MAXFIELD   20

263 int
264 test_load_config(test_t t, const char *fname, ...)
265 {
266     va_list     va;
267     const char  *keywords[MAXCB];
268     test_cfg_func_t  callbs[MAXCB];
269     char        *fields[MAXFIELD];
270     int         nfields;

272     FILE        *cfg;

```

```

273 char line[1024];
274 char buf[1024];
275 int done;
276 char *ptr;
277 char *tok;
278 char *err;
279 int lineno;
280 int rv;
281 int found;
282 char path[MAXPATHLEN];
283 int i;

285 va_start(va, fname);
286 for (i = 0; i < MAXCB; i++) {
287     keyws[i] = (const char *)va_arg(va, const char *);
288     if (keyws[i] == NULL)
289         break;
290     callbs[i] = (test_cfg_func_t)va_arg(va, test_cfg_func_t);
291 }
292 va_end(va);
293 if (i == MAXCB) {
294     test_debugf(t, "too many arguments to function >= %d", MAXCB);
295 }

297 found = 0;

299 if (access(fname, F_OK) == 0) {
300     found++;
301 }
302 if (!found && fname[0] != '/') {
303     char *stf = getenv("STF_SUITE");
304     if (stf == NULL) {
305         stf = "../..";
306     }
307     (void) snprintf(path, sizeof (path), "%s/cfg/%s", stf, fname);
308     if (access(path, F_OK) == 0) {
309         fname = path;
310         found++;
311     } else {
312         (void) snprintf(path, sizeof (path), "cfg/%s", fname);
313         if (access(path, F_OK) == 0) {
314             fname = path;
315             found++;
316         }
317     }
318 }

320 if ((cfg = fopen(fname, "r")) == NULL) {
321     test_failed(t, "open(%s): %s", fname, strerror(errno));
322     return (-1);
323 }

325 line[0] = 0;
326 done = 0;
327 lineno = 0;

329 while (!done) {
331     lineno++;

333     if (fgets(buf, sizeof (buf), cfg) == NULL) {
334         done++;
335     } else {
336         (void) strtok(buf, "\n");
337         if ((*buf != 0) && (buf[strlen(buf)-1] == '\\')) {
338             /*

```

```

339         * Continuation. This isn't quite right,
340         * as it doesn't allow for a "\" at the
341         * end of line (no escaping).
342         */
343         buf[strlen(buf)-1] = 0;
344         (void) strlcat(line, buf, sizeof (line));
345         continue;
346     }
347     (void) strlcat(line, buf, sizeof (line));
348 }

350 /* got a line */
351 ptr = line;
352 test_trim(&ptr);

354 /* skip comments and empty lines */
355 if (ptr[0] == 0 || ptr[0] == '#') {
356     line[0] = 0;
357     continue;
358 }

360 tok = strsep(&ptr, "|");
361 if (tok == NULL) {
362     break;
363 }
364 test_trim(&tok);

366 for (nfields = 0; nfields < MAXFIELD; nfields++) {
367     fields[nfields] = strsep(&ptr, "|");
368     if (fields[nfields] == NULL) {
369         break;
370     }
371     test_trim(&fields[nfields]);
372 }

374 found = 0;
375 rv = 0;

377 for (int i = 0; keyws[i] != NULL; i++) {
378     if (strcmp(tok, keyws[i]) == 0) {
379         found++;
380         err = NULL;
381         rv = callbs[i](fields, nfields, &err);
382     }
383 }
384 if (!found) {
385     rv = -1;
386     err = NULL;
387     (void) asprintf(&err, "unknown keyword %s", tok);
388 }
389 if (rv != 0) {
390     if (err) {
391         test_failed(t, "%s:%d: %s", fname,
392             lineno, err);
393         free(err);
394     } else {
395         test_failed(t, "%s:%d: unknown error",
396             fname, lineno);
397     }
398     (void) fclose(cfg);
399     return (rv);
400 }

402     line[0] = 0;
403 }
404 (void) fclose(cfg);

```

new/usr/src/test/libc-tests/tests/common/test_common.c

7

```
405     return (0);  
406 }  
_____unchanged_portion_omitted_____
```

new/usr/src/test/libc-tests/tests/common/test_common.h

1

```
*****
1383 Thu Apr  2 18:43:05 2015
new/usr/src/test/libc-tests/tests/common/test_common.h
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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  * Common handling for test programs.
18 */

20 #ifndef _TEST_COMMON_H
21 #define _TEST_COMMON_H

23 #ifdef __cplusplus
24 extern "C" {
25 #endif

27 typedef struct test *test_t;
28 typedef void (*test_func_t)(test_t, void *);

30 extern void test_set_debug(void);
31 extern void test_set_force(void);
32 extern test_t test_start(const char *name, ...);
33 extern void test_failed(test_t, const char *format, ...);
34 extern void test_passed(test_t);
35 extern void test_debugf(test_t, const char *format, ...);
36 extern void test_run(int nthr, test_func_t, void *arg, const char *, ...);
37 extern void test_summary(void);

39 extern void test_trim(char **);

41 typedef int (*test_cfg_func_t)(char **fields, int nfields, char **err);

43 /*
44  * Args list is array of pairs of const char *keyword, test_config_func_t,
45  * terminated by NULL.
46  */
47 extern int test_load_config(test_t, const char *, ...);

49 #ifdef __cplusplus
50 }
    unchanged_portion_omitted_

```

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

1

917 Thu Apr 2 18:43:05 2015

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

5293 desire symbol visibility test

Reviewed by: Robert Mustacchi <rm@joyent.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 include $(SRC)/Makefile.master
```

```
18 TESTSUBDIR = symbols
```

```
19 PROG = symbols_test
```

```
20 KSHPROG = setup
```

```
21 SYMTESTS = \
```

```
22     ctype_h \
```

```
23     dirent_h \
```

```
24     fcntl_h \
```

```
25     locale_h \
```

```
26     math_h \
```

```
27     netdb_h \
```

```
28     pthread_h \
```

```
29     signal_h \
```

```
30     stdio_h \
```

```
31     stdlib_h \
```

```
32     strings_h \
```

```
33     sys_stat_h \
```

```
34     sys_time_h \
```

```
35     sys_timeb_h \
```

```
36     ucontext_h \
```

```
37     unistd_h \
```

```
38     wchar_h \
```

```
39     wctype_h
```

```
41 EXTRAPROG += $(SYMTESTS)
```

```
43 include ../Makefile.com
```

```
45 $(SYMTESTS:%=$(TESTDIR)/%): $(TESTDIR)/setup
```

```
46     -$(RM) $@
```

```
47     $(LN) $(TESTDIR)/setup $@
```

new/usr/src/test/libc-tests/tests/symbols/setup.ksh

1

```
*****
1523 Thu Apr  2 18:43:05 2015
new/usr/src/test/libc-tests/tests/symbols/setup.ksh
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 export STF_SUITE=/opt/libc-tests
19 #
20 # First we set $dir to dirname $0, using efficient ksh builtins.
21 case $0 in
22  /*)
23     dir=${0%/*}
24     prog=${0##*/}
25     ;;
26 *)
27     dir=.
28     prog=${0}
29     ;;
30 esac
31 #
32 cfg=symbols/${prog%.ksh}.cfg
33 #
34 if [[ ! -f ${cfg} ]] && $cfg == symbols/setup.cfg ]]
35 then
36     # compiler check only
37     cfg=-C
38 fi
39 #
40 prog=symbols_test
41 #
42 for a in $*
43 do
44     if [[ $a == "-d" ]]
45     then
46         debug=yes
47     fi
48 done
49 #
50 # We look for architecture specific versions of the program,
51 # searching in several candidate directories. We run each one as
52 # we find it.
53 for f in $(/usr/bin/isainfo)
54 do
55     found=
56     [[ -n $debug ]] && print "Checking for arch $f:"
57     for p in \
58         ${dir}/${prog}.${f} \
59         ${dir}/${f}/${prog}.${f} \
```

new/usr/src/test/libc-tests/tests/symbols/setup.ksh

2

```
60     ${dir}/${f}/${prog}
61 do
62     [[ -n $found ]] && continue
63     [[ -n $debug ]] && print -n "    $p"
64     if [[ -f $p ]]; then
65         [[ -n $debug ]] && print " FOUND"
66         [[ -n $debug ]] && print "Executing $p $* ${cfg}"
67         found=yes
68         $p $* ${cfg} || exit 1
69     else
70         [[ -n $debug ]] && print
71     fi
72 done
73 [[ -z $found ]] && [[ -n $debug ]] && print "NOT PRESENT"
74 done
75 exit 0
```

new/usr/src/test/libc-tests/tests/symbols/symbols_test.c

1

```
*****
18221 Thu Apr  2 18:43:06 2015
new/usr/src/test/libc-tests/tests/symbols/symbols_test.c
5293 desire symbol visibility test
Reviewed by: Robert Mustacchi <rm@joyent.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 2015 Garrett D'Amore <garrett@damore.org>
14  */

16 /*
17  * This program tests symbol visibility using the /usr/bin/c89 and
18  * /usr/bin/c99 programs.
19  */

21 #include <stdio.h>
22 #include <stdlib.h>
23 #include <string.h>
24 #include <errno.h>
25 #include <err.h>
26 #include <unistd.h>
27 #include <sys/types.h>
28 #include <sys/stat.h>
29 #include <note.h>
30 #include <sys/wait.h>
31 #include "test_common.h"

33 char *dname;
34 char *cfile;
35 char *ofile;
36 char *lfile;
37 char *efile;

39 const char *sym = NULL;

41 static int good_count = 0;
42 static int fail_count = 0;
43 static int full_count = 0;
44 static int extra_debug = 0;
45 static char *compilation = "compilation.cfg";

47 #if defined(_LP64)
48 #define MFLAG "-m64"
49 #elif defined(_ILP32)
50 #define MFLAG "-m32"
51 #endif

53 const char *compilers[] = {
54     "cc",
55     "gcc",
56     "/opt/SUNWspro/bin/cc",
57     "/opt/gcc/4.4.4/bin/gcc",
58     "/opt/sunstudio12.1/bin/cc",
59     "/opt/sfw/bin/gcc",
```

new/usr/src/test/libc-tests/tests/symbols/symbols_test.c

2

```
60     "/usr/local/bin/gcc",
61     NULL
62 };

64 char *compiler = NULL;
65 const char *c89flags = NULL;
66 const char *c99flags = NULL;

68 #define MAXENV 64 /* maximum number of environments (bitmask width) */
69 #define MAXHDR 10 /* maximum # headers to require to access symbol */
70 #define MAXARG 20 /* maximum # of arguments */

72 #define WS " \t"

74 static int next_env = 0;

76 struct compile_env {
77     char *ce_name;
78     char *ce_lang;
79     char *ce_defs;
80     int ce_index;
81 };

83 static struct compile_env compile_env[MAXENV];

85 struct env_group {
86     char *eg_name;
87     uint64_t eg_mask;
88     struct env_group *eg_next;
89 };

91 typedef enum { SYM_TYPE, SYM_VALUE, SYM_FUNC } sym_type_t;

93 struct sym_test {
94     char *st_name;
95     sym_type_t st_type;
96     char *st_hdrs[MAXHDR];
97     char *st_rtype;
98     char *st_atypes[MAXARG];
99     uint64_t st_test_mask;
100     uint64_t st_need_mask;
101     char *st_prog;
102     struct sym_test *st_next;
103 };

105 struct env_group *env_groups = NULL;

107 struct sym_test *sym_tests = NULL;
108 struct sym_test **sym_insert = &sym_tests;

110 static char *
111 mystrdup(const char *s)
112 {
113     char *r;
114     if ((r = strdup(s)) == NULL) {
115         perror("strdup");
116         exit(1);
117     }
118     return (r);
119 }

121 static void *
122 myzalloc(size_t sz)
123 {
124     void *buf;
125     if ((buf = calloc(1, sz)) == NULL) {
```

```

126         perror("calloc");
127         exit(1);
128     }
129     return (buf);
130 }

132 static void
133 myasprintf(char **buf, const char *fmt, ...)
134 {
135     int rv;
136     va_list va;
137     va_start(va, fmt);
138     rv = vasprintf(buf, fmt, va);
139     va_end(va);
140     if (rv < 0) {
141         perror("vasprintf");
142         exit(1);
143     }
144 }

146 static void
147 append_sym_test(struct sym_test *st)
148 {
149     *sym_insert = st;
150     sym_insert = &st->st_next;
151 }

153 static int
154 find_env_mask(const char *name, uint64_t *mask)
155 {
156     for (int i = 0; i < MAXENV; i++) {
157         if (compile_env[i].ce_name != NULL &&
158             strcmp(compile_env[i].ce_name, name) == 0) {
159             *mask |= (1ULL << i);
160             return (0);
161         }
162     }

164     for (struct env_group *eg = env_groups; eg != NULL; eg = eg->eg_next) {
165         if (strcmp(name, eg->eg_name) == 0) {
166             *mask |= eg->eg_mask;
167             return (0);
168         }
169     }
170     return (-1);
171 }

174 static int
175 expand_env(char *list, uint64_t *mask, char **erritem)
176 {
177     char *item;
178     for (item = strtok(list, WS); item != NULL; item = strtok(NULL, WS)) {
179         if (find_env_mask(item, mask) < 0) {
180             if (erritem != NULL) {
181                 *erritem = item;
182             }
183             return (-1);
184         }
185     }
186     return (0);
187 }

189 static int
190 expand_env_list(char *list, uint64_t *test, uint64_t *need, char **erritem)
191 {

```

```

192     uint64_t mask = 0;
193     int act;
194     char *item;
195     for (item = strtok(list, WS); item != NULL; item = strtok(NULL, WS)) {
196         switch (item[0]) {
197             case '+':
198                 act = 1;
199                 item++;
200                 break;
201             case '-':
202                 act = 0;
203                 item++;
204                 break;
205             default:
206                 act = 1;
207                 break;
208         }

210         mask = 0;
211         if (find_env_mask(item, &mask) < 0) {
212             if (erritem != NULL) {
213                 *erritem = item;
214             }
215             return (-1);
216         }
217         *test |= mask;
218         if (act) {
219             *need |= mask;
220         } else {
221             *need &= ~(mask);
222         }
223     }
224     return (0);
225 }

227 static int
228 do_env(char **fields, int nfields, char **err)
229 {
230     char *name;
231     char *lang;
232     char *defs;

234     if (nfields != 3) {
235         myasprintf(err, "number of fields (%d) != 3", nfields);
236         return (-1);
237     }

239     if (next_env >= MAXENV) {
240         myasprintf(err, "too many environments");
241         return (-1);
242     }

244     name = fields[0];
245     lang = fields[1];
246     defs = fields[2];

248     compile_env[next_env].ce_name = mystrdup(name);
249     compile_env[next_env].ce_lang = mystrdup(lang);
250     compile_env[next_env].ce_defs = mystrdup(defs);
251     compile_env[next_env].ce_index = next_env;
252     next_env++;
253     return (0);
254 }

256 static int
257 do_env_group(char **fields, int nfields, char **err)

```

```

258 {
259     char *name;
260     char *list;
261     struct env_group *eg;
262     uint64_t mask;
263     char *item;
264
265     if (nfields != 2) {
266         myasprintf(terr, "number of fields (%d) != 2", nfields);
267         return (-1);
268     }
269
270     name = fields[0];
271     list = fields[1];
272     mask = 0;
273
274     if (expand_env(list, &mask, &item) < 0) {
275         myasprintf(terr, "reference to undefined env %s", item);
276         return (-1);
277     }
278
279     eg = myzalloc(sizeof(*eg));
280     eg->eg_name = mystrdup(name);
281     eg->eg_mask = mask;
282     eg->eg_next = env_groups;
283     env_groups = eg;
284     return (0);
285 }
286
287 static char *progbuf = NULL;
288 size_t proglen = 0;
289 size_t progsiz = 0;
290
291 static void
292 addprogch(char c)
293 {
294     while (progsiz <= (proglen + 1)) {
295         progbuf = realloc(progbuf, progsiz + 4096);
296         if (progbuf == NULL) {
297             perror("realloc");
298             exit(1);
299         }
300         progsiz += 1024;
301     }
302     progbuf[proglen++] = c;
303     progbuf[proglen] = 0;
304 }
305
306 static void
307 addprogstr(char *s)
308 {
309     while (*s != NULL) {
310         addprogch(*s);
311         s++;
312     }
313 }
314
315 static void
316 addprogfmt(const char *fmt, ...)
317 {
318     va_list va;
319     char *buf = NULL;
320     va_start(va, fmt);
321     if (vasprintf(&buf, fmt, va) < 0) {
322         perror("vasprintf");
323         exit(1);

```

```

324     }
325     va_end(va);
326     addprogstr(buf);
327     free(buf);
328 }
329
330 static void
331 mkprog(struct sym_test *st)
332 {
333     char *s;
334
335     proglen = 0;
336
337     for (int i = 0; i < MAXHDR && st->st_hdrs[i] != NULL; i++) {
338         addprogfmt("#include <%s>\n", st->st_hdrs[i]);
339     }
340
341     for (s = st->st_rtype; *s; s++) {
342         addprogch(*s);
343         if (*s == '(') {
344             s++;
345             addprogch(*s);
346             s++;
347             break;
348         }
349     }
350     addprogch(' ');
351
352     /* for function pointers, s is closing suffix, otherwise empty */
353
354     switch (st->st_type) {
355     case SYM_TYPE:
356         addprogstr("test_type;");
357         break;
358
359     case SYM_VALUE:
360         addprogfmt("test_value%s;\n", s); /* s usually empty */
361         addprogstr("void\ntest_func(void)\n{\n");
362         addprogfmt("\ttest_value = %s;\n", st->st_name);
363         break;
364
365     case SYM_FUNC:
366         addprogstr("\ntest_func(");
367         for (int i = 0; st->st_atypes[i] != NULL && i < MAXARG; i++) {
368             int didname = 0;
369             if (i > 0) {
370                 addprogstr(", ");
371             }
372             if (strcmp(st->st_atypes[i], "void") == 0) {
373                 didname = 1;
374             }
375             if (strcmp(st->st_atypes[i], "") == 0) {
376                 didname = 1;
377                 addprogstr("void");
378             }
379
380             /* print the argument list */
381             for (char *a = st->st_atypes[i]; *a; a++) {
382                 if (*a == '(' && a[1] == '(' && !didname) {
383                     addprogfmt("(%a%d", i);
384                     didname = 1;
385                     a++;
386                 } else if (*a == '[' && !didname) {
387                     addprogfmt("%a%d["", i);
388                     didname = 1;
389                 } else {

```

```

390         addprogch(*a);
391     }
392     }
393     if (!didname) {
394         addprogfmt(" a%d", i);
395     }
396 }

398 if (st->st_atypes[0] == NULL) {
399     addprogstr("void");
400 }

402 /*
403  * Close argument list, and closing ")" for func ptrs.
404  * Note that for non-function pointers, s will be empty
405  * below, otherwise it points to the trailing argument
406  * list.
407  */
408 addprogfmt("%s\n{\n\t", s);

410 if (strcmp(st->st_rtype, "") != 0 &&
411     strcmp(st->st_rtype, "void") != 0) {
412     addprogstr("return ");
413 }

415 /* add the function call */
416 addprogfmt("%s(", st->st_name);
417 for (int i = 0; st->st_atypes[i] != NULL && i < MAXARG; i++) {
418     if (strcmp(st->st_atypes[i], "") != 0 &&
419         strcmp(st->st_atypes[i], "void") != 0) {
420         addprogfmt("%sa%d", i > 0 ? ", " : "", i);
421     }
422 }

424     addprogstr(");\n}");
425     break;
426 }

428     addprogch('\n');

430     st->st_prog = progbuf;
431 }

433 static int
434 add_envs(struct sym_test *st, char *envs, char **err)
435 {
436     char *item;
437     if (expand_env_list(envs, &st->st_test_mask, &st->st_need_mask,
438         &item) < 0) {
439         myasprintf(err, "bad env action %s", item);
440         return (-1);
441     }
442     return (0);
443 }

445 static int
446 add_headers(struct sym_test *st, char *hdrs, char **err)
447 {
448     int i = 0;

450     for (char *h = strsep(&hdrs, ";"); h != NULL; h = strsep(&hdrs, ";")) {
451         if (i >= MAXHDR) {
452             myasprintf(err, "too many headers");
453             return (-1);
454         }
455         test_trim(&h);

```

```

456         st->st_hdrs[i++] = mystrdup(h);
457     }

459     return (0);
460 }

462 static int
463 add_arg_types(struct sym_test *st, char *atype, char **err)
464 {
465     int i = 0;
466     char *a;
467     for (a = strsep(&atype, ";"); a != NULL; a = strsep(&atype, ";")) {
468         if (i >= MAXARG) {
469             myasprintf(err, "too many arguments");
470             return (-1);
471         }
472         test_trim(&a);
473         st->st_atypes[i++] = mystrdup(a);
474     }

476     return (0);
477 }

479 static int
480 do_type(char **fields, int nfields, char **err)
481 {
482     char *decl;
483     char *hdrs;
484     char *envs;
485     struct sym_test *st;

487     if (nfields != 3) {
488         myasprintf(err, "number of fields (%d) != 3", nfields);
489         return (-1);
490     }
491     decl = fields[0];
492     hdrs = fields[1];
493     envs = fields[2];

495     st = myzalloc(sizeof (*st));
496     st->st_type = SYM_TYPE;
497     st->st_name = mystrdup(decl);
498     st->st_rtype = mystrdup(decl);

500     if ((add_envs(st, envs, err) < 0) ||
501         (add_headers(st, hdrs, err) < 0)) {
502         return (-1);
503     }
504     append_sym_test(st);

506     return (0);
507 }

509 static int
510 do_value(char **fields, int nfields, char **err)
511 {
512     char *name;
513     char *type;
514     char *hdrs;
515     char *envs;
516     struct sym_test *st;

518     if (nfields != 4) {
519         myasprintf(err, "number of fields (%d) != 4", nfields);
520         return (-1);
521     }

```

```

522     name = fields[0];
523     type = fields[1];
524     hdrs = fields[2];
525     envs = fields[3];

527     st = myzalloc(sizeof (*st));
528     st->st_type = SYM_VALUE;
529     st->st_name = mystrdup(name);
530     st->st_rtype = mystrdup(type);

532     if ((add_envs(st, envs, err) < 0) ||
533         (add_headers(st, hdrs, err) < 0)) {
534         return (-1);
535     }
536     append_sym_test(st);

538     return (0);
539 }

541 static int
542 do_func(char **fields, int nfields, char **err)
543 {
544     char *name;
545     char *rtype;
546     char *atype;
547     char *hdrs;
548     char *envs;
549     struct sym_test *st;

551     if (nfields != 5) {
552         myasprintf(err, "number of fields (%d) != 5", nfields);
553         return (-1);
554     }
555     name = fields[0];
556     rtype = fields[1];
557     atype = fields[2];
558     hdrs = fields[3];
559     envs = fields[4];

561     st = myzalloc(sizeof (*st));
562     st->st_type = SYM_FUNC;
563     st->st_name = mystrdup(name);
564     st->st_rtype = mystrdup(rtype);

566     if ((add_envs(st, envs, err) < 0) ||
567         (add_headers(st, hdrs, err) < 0) ||
568         (add_arg_types(st, atype, err) < 0)) {
569         return (-1);
570     }
571     append_sym_test(st);

573     return (0);
574 }

576 struct sym_test *
577 next_sym_test(struct sym_test *st)
578 {
579     return (st == NULL ? sym_tests : st->st_next);
580 }

582 const char *
583 sym_test_prog(struct sym_test *st)
584 {
585     if (st->st_prog == NULL) {
586         mkprog(st);
587     }

```

```

588     return (st->st_prog);
589 }

591 const char *
592 sym_test_name(struct sym_test *st)
593 {
594     return (st->st_name);
595 }

597 /*
598  * Iterate through tests. Pass in NULL for cenv to begin the iteration. For
599  * subsequent iterations, use the return value from the previous iteration.
600  * Returns NULL when there are no more environments.
601  */
602 struct compile_env *
603 sym_test_env(struct sym_test *st, struct compile_env *cenv, int *need)
604 {
605     int i = cenv ? cenv->ce_index + 1 : 0;
606     uint64_t b = 1ULL << i;

608     while ((i < MAXENV) && (b != 0)) {
609         cenv = &compile_env[i];
610         if (b & st->st_test_mask) {
611             *need = (st->st_need_mask & b) ? 1 : 0;
612             return (cenv);
613         }
614         b <<= 1;
615         i++;
616     }
617     return (NULL);
618 }

620 const char *
621 env_name(struct compile_env *cenv)
622 {
623     return (cenv->ce_name);
624 }

626 const char *
627 env_lang(struct compile_env *cenv)
628 {
629     return (cenv->ce_lang);
630 }

632 const char *
633 env_defs(struct compile_env *cenv)
634 {
635     return (cenv->ce_defs);
636 }

638 static void
639 show_file(test_t t, const char *path)
640 {
641     FILE *f;
642     char *buf = NULL;
643     size_t cap = 0;
644     int line = 1;

646     f = fopen(path, "r");
647     if (f == NULL) {
648         test_debugf(t, "fopen(%s): %s", path, strerror(errno));
649         return;
650     }

652     test_debugf(t, "----->> begin (%s) <<-----", path);
653     while (getline(&buf, &cap, f) >= 0) {

```

```

654         (void) strtok(buf, "\r\n");
655         test_debugf(t, "%d: %s", line, buf);
656         line++;
657     }
658     test_debugf(t, "----->> end (%s) <<-----", path);
659     (void) fclose(f);
660 }

662 static void
663 cleanup(void)
664 {
665     if (ofile != NULL) {
666         (void) unlink(ofile);
667         free(ofile);
668         ofile = NULL;
669     }
670     if (lfile != NULL) {
671         (void) unlink(lfile);
672         free(lfile);
673         lfile = NULL;
674     }
675     if (cfile != NULL) {
676         (void) unlink(cfile);
677         free(cfile);
678         cfile = NULL;
679     }
680     if (efile != NULL) {
681         (void) unlink(efile);
682         free(efile);
683         efile = NULL;
684     }
685     if (dname) {
686         (void) rmdir(dname);
687         free(dname);
688         dname = NULL;
689     }
690 }

692 static int
693 mkworkdir(void)
694 {
695     char b[32];
696     char *d;

698     cleanup();

700     (void) strlcpy(b, "/tmp/symbols_testXXXXXX", sizeof(b));
701     if ((d = mkdtemp(b)) == NULL) {
702         perror("mkdtemp");
703         return (-1);
704     }
705     dname = mystrdup(d);
706     myasprintf(&cfile, "%s/compile_test.c", d);
707     myasprintf(&ofile, "%s/compile_test.o", d);
708     myasprintf(&lfile, "%s/compile_test.log", d);
709     myasprintf(&efile, "%s/compile_test.exe", d);
710     return (0);
711 }

713 void
714 find_compiler(void)
715 {
716     test_t t;
717     int i;
718     FILE *cf;

```

```

720     t = test_start("finding compiler");

722     if ((cf = fopen(cfile, "w+") == NULL) {
723         test_failed(t, "Unable to open %s for write: %s", cfile,
724             strerror(errno));
725         return;
726     }
727     (void) fprintf(cf, "#include <stdio.h>\n");
728     (void) fprintf(cf, "int main(int argc, char **argv) {\n");
729     (void) fprintf(cf, "#if defined(__SUNPRO_C)\n");
730     (void) fprintf(cf, "exit(51);\n");
731     (void) fprintf(cf, "#elif defined(__GNUC__)\n");
732     (void) fprintf(cf, "exit(52);\n");
733     (void) fprintf(cf, "#else\n");
734     (void) fprintf(cf, "exit(99);\n");
735     (void) fprintf(cf, "#endif\n");
736     (void) fclose(cf);

738     for (i = 0; compilers[i] != NULL; i++) {
739         char cmd[256];
740         int rv;

742         (void) snprintf(cmd, sizeof(cmd),
743             "%s %s %s -o %s >/dev/null 2>&1",
744             compilers[i], MFLAG, cfile, efile);
745         test_debugf(t, "trying %s", cmd);
746         rv = system(cmd);

748         test_debugf(t, "result: %d", rv);

750         if ((rv < 0) || !WIFEXITED(rv) || WEXITSTATUS(rv) != 0)
751             continue;

753         rv = system(efile);
754         if (rv >= 0 && WIFEXITED(rv)) {
755             rv = WEXITSTATUS(rv);
756         } else {
757             rv = -1;
758         }

760         switch (rv) {
761             case 51: /* STUDIO */
762                 test_debugf(t, "Found Studio C");
763                 c89flags = "-Xc -errwarn=%all -v -xc99=%none " MFLAG;
764                 c99flags = "-Xc -errwarn=%all -v -xc99=%all " MFLAG;
765                 if (extra_debug) {
766                     test_debugf(t, "c89flags: %s", c89flags);
767                     test_debugf(t, "c99flags: %s", c99flags);
768                 }
769                 test_passed(t);
770                 break;
771             case 52: /* GCC */
772                 test_debugf(t, "Found GNU C");
773                 c89flags = "-Wall -Werror -std=c89 " MFLAG;
774                 c99flags = "-Wall -Werror -std=c99 " MFLAG;
775                 if (extra_debug) {
776                     test_debugf(t, "c89flags: %s", c89flags);
777                     test_debugf(t, "c99flags: %s", c99flags);
778                 }
779                 test_passed(t);
780                 break;
781             case 99:
782                 test_debugf(t, "Found unknown (unsupported) compiler");
783                 continue;
784             default:
785                 continue;

```

```

786     }
787     myasprintf(&compiler, "%s", compilers[i]);
788     test_debugf(t, "compiler: %s", compiler);
789     return;
790 }
791 test_failed(t, "No compiler found.");
792 }

794 int
795 do_compile(test_t t, struct sym_test *st, struct compile_env *cenv, int need)
796 {
797     char *cmd;
798     FILE *logf;
799     FILE *dotc;
800     const char *prog;

802     full_count++;

804     if ((dotc = fopen(cfile, "w+")) == NULL) {
805         test_failed(t, "fopen(%s): %s", cfile, strerror(errno));
806         return (-1);
807     }
808     prog = sym_test_prog(st);
809     if (fwrite(prog, 1, strlen(prog), dotc) < strlen(prog)) {
810         test_failed(t, "fwrite: %s", strerror(errno));
811         (void) fclose(dotc);
812         return (-1);
813     }
814     if (fclose(dotc) < 0) {
815         test_failed(t, "fclose: %s", strerror(errno));
816         return (-1);
817     }

819     (void) unlink(ofile);

821     myasprintf(&cmd, "%s %s %s -c %s -o %s >>%s 2>&1",
822               compiler, strcmp(env_lang(cenv), "c99") == 0 ? "c99flags : c89flags",
823               env_defs(cenv), cfile, ofile, lfile);

825     if (extra_debug) {
826         test_debugf(t, "command: %s", cmd);
827     }

829     if ((logf = fopen(lfile, "w+")) == NULL) {
830         test_failed(t, "fopen: %s", strerror(errno));
831         return (-1);
832     }
833     (void) fprintf(logf, "=====\n");
834     (void) fprintf(logf, "PROGRAM:\n%s\n", sym_test_prog(st));
835     (void) fprintf(logf, "COMMAND: %s\n", cmd);
836     (void) fprintf(logf, "EXPECT: %s\n", need ? "OK" : "FAIL");
837     (void) fclose(logf);

839     switch (system(cmd)) {
840     case -1:
841         test_failed(t, "error compiling in %s: %s", env_name(cenv),
842                   strerror(errno));
843         return (-1);
844     case 0:
845         if (!need) {
846             fail_count++;
847             show_file(t, lfile);
848             test_failed(t, "symbol visible in %s", env_name(cenv));
849             return (-1);
850         }
851         break;

```

```

852     default:
853         if (need) {
854             fail_count++;
855             show_file(t, lfile);
856             test_failed(t, "error compiling in %s", env_name(cenv));
857             return (-1);
858         }
859         break;
860     }
861     good_count++;
862     return (0);
863 }

865 void
866 test_compile(void)
867 {
868     struct sym_test *st;
869     struct compile_env *cenv;
870     test_t t;
871     int need;

873     for (st = next_sym_test(NULL); st; st = next_sym_test(st)) {
874         if ((sym != NULL) && strcmp(sym, sym_test_name(st))) {
875             continue;
876         }
877         /* XXX: we really want a sym_test_desc() */
878         for (cenv = sym_test_env(st, NULL, &need);
879              cenv != NULL;
880              cenv = sym_test_env(st, cenv, &need)) {
881             t = test_start("%s : %c%s", sym_test_name(st),
882                           need ? '+' : '-', env_name(cenv));
883             if (do_compile(t, st, cenv, need) == 0) {
884                 test_passed(t);
885             }
886         }
887     }

889     if (full_count > 0) {
890         test_summary();
891     }
892 }

894 int
895 main(int argc, char **argv)
896 {
897     int optc;
898     int optC = 0;

900     while ((optc = getopt(argc, argv, "DdfCs:c:")) != EOF) {
901         switch (optc) {
902         case 'd':
903             test_set_debug();
904             break;
905         case 'f':
906             test_set_force();
907             break;
908         case 'D':
909             test_set_debug();
910             extra_debug++;
911             break;
912         case 'c':
913             compilation = optarg;
914             break;
915         case 'C':
916             optC++;
917             break;

```

```
918         case 's':
919             sym = optarg;
920             break;
921         default:
922             (void) fprintf(stderr, "Usage: %s [-df]\n", argv[0]);
923             exit(1);
924     }
925 }
926
927 if (test_load_config(NULL, compilation,
928     "env", do_env, "env_group", do_env_group, NULL) < 0) {
929     exit(1);
930 }
931
932 while (optind < argc) {
933     if (test_load_config(NULL, argv[optind++],
934         "type", do_type,
935         "value", do_value,
936         "func", do_func,
937         NULL) < 0) {
938         exit(1);
939     }
940 }
941
942 if (atexit(cleanup) != 0) {
943     perror("atexit");
944     exit(1);
945 }
946
947 if (mkworkdir() < 0) {
948     perror("mkdir");
949     exit(1);
950 }
951
952 find_compiler();
953 if (!optC)
954     test_compile();
955
956 exit(0);
957 }
```