

```

*****
4076 Fri Aug 1 15:57:43 2014
new/usr/src/lib/libdisasm/Makefile.com
patch fix-lint
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 # Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
25 # Copyright 2014 Nexenta Systems, Inc. All rights reserved.
26 #endif /* ! codereview */
27 #

29 #
30 # The build process for libdisasm is slightly different from that used by other
31 # libraries, because libdisasm must be built in two flavors - as a standalone
32 # for use by kmdb and as a normal library. We use $(CURTYPE) to indicate the
33 # current flavor being built.
34 #

36 LIBRARY= libdisasm.a
37 STANDLIBRARY= libstanddisasm.so
38 VERS= .1

40 # By default, we build the shared library. Construction of the standalone
41 # is specifically requested by architecture-specific Makefiles.
42 TYPES= library
43 CURTYPE= library

45 COMDIR= $(SRC)/lib/libdisasm/common

47 #
48 # Architecture-independent files
49 #
50 SRCS_common= $(COMDIR)/libdisasm.c
51 OBJECTS_common= libdisasm.o

53 #
54 # Architecture-dependent disassembly files
55 #
56 SRCS_i386= $(COMDIR)/dis_i386.c \
57 $(SRC)/common/dis/i386/dis_tables.c
58 SRCS_sparc= $(COMDIR)/dis_sparc.c \
59 $(COMDIR)/dis_sparc_fmt.c \
60 $(COMDIR)/dis_sparc_instr.c

```

```

62 OBJECTS_i386= dis_i386.o \
63 dis_tables.o
64 OBJECTS_sparc= dis_sparc.o \
65 dis_sparc_fmt.o \
66 dis_sparc_instr.o

68 #
69 # We build the regular shared library with support for all architectures.
70 # The standalone version should only contain code for the native
71 # architecture to reduce the memory footprint of kmdb.
72 #
73 OBJECTS_library= $(OBJECTS_common) \
74 $(OBJECTS_i386) \
75 $(OBJECTS_sparc)
76 OBJECTS_standalone= $(OBJECTS_common) \
77 $(OBJECTS_$(MACH))
78 OBJECTS= $(OBJECTS_$(CURTYPE))

80 include $(SRC)/lib/Makefile.lib

82 SRCS_library= $(SRCS_common) \
83 $(SRCS_i386) \
84 $(SRCS_sparc)
85 SRCS_standalone= $(SRCS_common) \
86 $(SRCS_$(MACH))
87 SRCS= $(SRCS_$(CURTYPE))

89 #
90 # Used to verify that the standalone doesn't have any unexpected external
91 # dependencies.
92 #
93 LINKTEST_OBJ = objs/linktest_stand.o

95 CLOBBERFILES_standalone = $(LINKTEST_OBJ)
96 CLOBBERFILES += $(CLOBBERFILES_$(CURTYPE))

98 LIBS_standalone = $(STANDLIBRARY)
99 LIBS_library = $(DYNLIB) $(LINTLIB)
100 LIBS = $(LIBS_$(CURTYPE))

102 MAPFILES = $(COMDIR)/mapfile-vers

104 LDLIBS += -lc

106 LDFLAGS_standalone = $(ZNOVERSION) $(BREDUCE) -dy -r
107 LDFLAGS = $(LDFLAGS_$(CURTYPE))

109 ASFLAGS_standalone = -DDIS_STANDALONE
110 ASFLAGS_library =
111 ASFLAGS += -P $(ASFLAGS_$(CURTYPE)) -D_ASM

113 $(LINTLIB) := SRCS = $(COMDIR)/$(LINTSRC)

115 CERRWARN += -_gcc=-Wno-parentheses
116 CERRWARN += -_gcc=-Wno-uninitialized

118 # We want the thread-specific errno in the library, but we don't want it in
119 # the standalone. $(DTS_ERRNO) is designed to add -D_TS_ERRNO to $(CPPFLAGS),
120 # in order to enable this feature. Conveniently, -D_REENTRANT does the same
121 # thing. As such, we null out $(DTS_ERRNO) to ensure that the standalone
122 # doesn't get it.
123 DTS_ERRNO=

125 CPPFLAGS_standalone = -DDIS_STANDALONE -I$(SRC)/cmd/mdb/common
25 # We need to rename some standard functions so we can easily implement them
26 # in consumers.

```

```
27 STAND_RENAMED_FUNCS= \
28     snprintf

30 CPPFLAGS_standalone = -DDIS_STANDALONE $(STAND_RENAMED_FUNCS:%=-D%=mdb_%) \
31     -Dvsprintf=mdb_io_b_vsnprintf -I$(SRC)/cmd/mdb/common
126 CPPFLAGS_library = -D_REENTRANT
127 CPPFLAGS += -I$(COMDIR) $(CPPFLAGS_$(CURTYPE))

129 # For the x86 disassembler we have to include sources from usr/src/common
130 CPPFLAGS += -I$(SRC)/common/dis/i386 -DDIS_TEXT

132 CFLAGS_standalone = $(STAND_FLAGS_32)
133 CFLAGS_common =
134 CFLAGS += $(CFLAGS_$(CURTYPE)) $(CFLAGS_common)

136 CFLAGS64_standalone = $(STAND_FLAGS_64)
137 CFLAGS64 += $(CVERBOSE) $(CFLAGS64_$(CURTYPE)) $(CFLAGS64_common)

139 DYNFLAGS += $(ZINTERPOSE)

141 .KEEP_STATE:
```

new/usr/src/lib/libdisasm/common/dis_i386.c

1

```
*****
6000 Fri Aug 1 15:57:43 2014
new/usr/src/lib/libdisasm/common/dis_i386.c
patch fix-lint
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
26  * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
27 #endif /* ! codereview */
28 */

30 #include <libdisasm.h>
26 #include <stdlib.h>
27 #include <stdio.h>

32 #include "dis_tables.h"
33 #include "libdisasm_impl.h"

35 typedef struct dis_handle_i386 {
36     int          dhx_mode;
37     dis86_t      dhx_dis;
38     uint64_t     dhx_end;
39 } dis_handle_i386_t;
_____unchanged_portion_omitted_____

93 static int
94 dis_i386_handle_attach(dis_handle_t *dhp)
95 {
96     dis_handle_i386_t *dhx;

98     /*
99     * Validate architecture flags
100    */
101    if (dhp->dh_flags & ~(DIS_X86_SIZE16 | DIS_X86_SIZE32 | DIS_X86_SIZE64 |
102        DIS_OCTAL | DIS_NOIMMSYM)) {
103        (void) dis_seterrno(E_DIS_INVALIDFLAG);
104        return (-1);
105    }

107    /*
108    * Create and initialize the internal structure
109    */
110    if ((dhx = dis_zalloc(sizeof (dis_handle_i386_t))) == NULL) {
```

new/usr/src/lib/libdisasm/common/dis_i386.c

2

```
111         (void) dis_seterrno(E_DIS_NOMEM);
112         return (-1);
113     }
114     dhp->dh_arch_private = dhx;

116     /*
117     * Initialize x86-specific architecture structure
118     */
119     if (dhp->dh_flags & DIS_X86_SIZE16)
120         dhx->dhx_mode = SIZE16;
121     else if (dhp->dh_flags & DIS_X86_SIZE64)
122         dhx->dhx_mode = SIZE64;
123     else
124         dhx->dhx_mode = SIZE32;

126     if (dhp->dh_flags & DIS_OCTAL)
127         dhx->dhx_dis.d86_flags = DIS_F_OCTAL;

129     dhx->dhx_dis.d86_sprintf_func = dis_snprintf;
126     dhx->dhx_dis.d86_sprintf_func = snprintf;
130     dhx->dhx_dis.d86_get_byte = get_byte;
131     dhx->dhx_dis.d86_sym_lookup = do_lookup;
132     dhx->dhx_dis.d86_check_func = check_func;

134     dhx->dhx_dis.d86_data = dhp;

136     return (0);
137 }
_____unchanged_portion_omitted_____

173 #define MIN(a, b)      ((a) < (b) ? (a) : (b))

176 /*
177  * Return the previous instruction. On x86, we have no choice except to
178  * disassemble everything from the start of the symbol, and stop when we have
179  * reached our instruction address. If we're not in the middle of a known
180  * symbol, then we return the same address to indicate failure.
181  */
182 static uint64_t
183 dis_i386_previnstr(dis_handle_t *dhp, uint64_t pc, int n)
184 {
185     uint64_t *hist, addr, start;
186     int cur, nseen;
187     uint64_t res = pc;

189     if (n <= 0)
190         return (pc);

192     if (dhp->dh_lookup(dhp->dh_data, pc, NULL, 0, &start, NULL) != 0 ||
193         start == pc)
194         return (res);

196     hist = dis_zalloc(sizeof (uint64_t) * n);

198     for (cur = 0, nseen = 0, addr = start; addr < pc; addr = dhp->dh_addr) {
199         hist[cur] = addr;
200         cur = (cur + 1) % n;
201         nseen++;

203         /* if we cannot make forward progress, give up */
204         if (dis_disassemble(dhp, addr, NULL, 0) != 0)
205             goto done;
206     }

208     if (addr != pc) {
209         /*
```

```
210             * We scanned past %pc, but didn't find an instruction that
211             * started at %pc. This means that either the caller specified
212             * an invalid address, or we ran into something other than code
213             * during our scan. Virtually any combination of bytes can be
214             * construed as a valid Intel instruction, so any non-code bytes
215             * we encounter will have thrown off the scan.
216             */
217             goto done;
218         }

220         res = hist[(cur + n - MIN(n, nseen)) % n];

222 done:
223     dis_free(hist, sizeof (uint64_t) * n);
224     return (res);
225 }
unchanged_portion_omitted
```

new/usr/src/lib/libdisasm/common/dis_sparc.c

1

8995 Fri Aug 1 15:57:44 2014

new/usr/src/lib/libdisasm/common/dis_sparc.c

patch fix-lint

unchanged_portion_omitted

```
225 static int
226 dis_sparc_disassemble(dis_handle_t *dhp, uint64_t addr, char *buf,
227     size_t buflen)
228 {
229     dis_handle_sparc_t *dhx = dhp->dh_arch_private;
230     const table_t *tp = &initial_table;
231     const inst_t *inp = NULL;
232
233     uint32_t instr;
234     uint32_t idx = 0;
235
236     if (dhp->dh_read(dhp->dh_data, addr, &instr, sizeof (instr)) !=
237         sizeof (instr))
238         return (-1);
239
240     dhx->dhx_buf = buf;
241     dhx->dhx_buflen = buflen;
242     dhp->dh_addr = addr;
243
244     buf[0] = '\0';
245
246     /* this allows sparc code to be tested on x86 */
247     #if !defined(DIS_STANDALONE)
248     instr = BE_32(instr);
249     #endif /* DIS_STANDALONE */
250
251     #if !defined(DIS_STANDALONE)
252     if ((dhx->dhx_debug & DIS_DEBUG_PRTBIN) != 0)
253         do_binary(instr);
254     #endif /* DIS_STANDALONE */
255
256     /* CONSTCOND */
257     while (1) {
258         idx = dis_get_bits(instr, tp->tbl_field, tp->tbl_len);
259         inp = &tp->tbl_inp[idx];
260
261         inp = dis_get_overlay(dhp, tp, idx);
262
263         if ((inp->in_type == INST_NONE) ||
264             ((inp->in_arch & dhp->dh_flags) == 0))
265             goto error;
266
267         if (inp->in_type == INST_TBL) {
268             tp = inp->in_data.in_tbl;
269             continue;
270         }
271
272         break;
273     }
274
275     if (tp->tbl_fmt(dhp, instr, inp, idx) == 0)
276         return (0);
277
278 error:
279
280     (void) dis_snprintf(buf, buflen,
281         (void) snprintf(buf, buflen,
282             ((dhp->dh_flags & DIS_OCTAL) != 0) ? "0%0111o" : "0x%08lx",
283             instr);
```

new/usr/src/lib/libdisasm/common/dis_sparc.c

2

```
284     return (0);
285 }
unchanged_portion_omitted
```

new/usr/src/lib/libdisasm/common/dis_sparc_fmt.c

1

```
*****
60085 Fri Aug 1 15:57:44 2014
new/usr/src/lib/libdisasm/common/dis_sparc_fmt.c
patch fix-lint
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21
22 /*
23  * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */
26
27 /*
28  * Copyright 2009 Jason King. All rights reserved.
29  * Use is subject to license terms.
30  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
31  */
32
33 #include <sys/byteorder.h>
34 #include <stdarg.h>
35
36 #if !defined(DIS_STANDALONE)
37 #include <stdio.h>
38 #endif /* DIS_STANDALONE */
39
40 #include "libdisasm.h"
41 #include "libdisasm_impl.h"
42 #include "dis_sparc.h"
43 #include "dis_sparc_fmt.h"
44
45 extern char *strncpy(char *, const char *, size_t);
46 extern size_t strlen(const char *);
47 extern int strcmp(const char *, const char *);
48 extern int strncmp(const char *, const char *, size_t);
49 extern size_t strlen(const char *, const char *, size_t);
50 extern size_t strlcat(char *, const char *, size_t);
51 extern size_t strlcpy(char *, const char *, size_t);
52 extern int snprintf(char *, size_t, const char *, ...);
53 extern int vsnprintf(char *, size_t, const char *, va_list);
54
55 /*
56  * This file has the functions that do all the dirty work of outputting the
57  * disassembled instruction
58  * All the non-static functions follow the format_fcn (in dis_sparc.h):
59  * Input:
60  *     disassembler handle/context
```

new/usr/src/lib/libdisasm/common/dis_sparc_fmt.c

2

```
60 *     instruction to disassemble
61 *     instruction definition pointer (inst_t *)
62 *     index in the table of the instruction
63 * Return:
64 *     0 Success
65 *     !0 Invalid instruction
66 *
67 * Generally, instructions found in the same table use the same output format
68 * or have a few minor differences (which are described in the 'flags' field
69 * of the instruction definition. In some cases, certain instructions differ
70 * radically enough from those in the same table, that their own format
71 * function is used.
72 *
73 * Typically each table has a unique format function defined in this file. In
74 * some cases (such as branches) a common one for all the tables is used.
75 *
76 * When adding support for new instructions, it is largely a judgement call
77 * as to when a new format function is defined.
78 */
79
80 /* The various instruction formats of a sparac instruction */
81
82 #if defined(_BIT_FIELDS_HTOL)
83 typedef struct format1 {
84     uint32_t op:2;
85     uint32_t disp30:30;
86 } format1_t;
87 unchanged_portion_omitted
88
89 /* ARGUSED3 */
90 int
91 fmt_branch(dis_handle_t *dhp, uint32_t instr, const inst_t *inp, int idx)
92 {
93     dis_handle_sparc_t *dhx = dhp->dh_arch_private;
94     const char *name = inp->in_data.in_def.in_name;
95     const char *r = NULL;
96     const char *annul = "";
97     const char *pred = "";
98
99     char buf[15];
100
101     ifmt_t *f = (ifmt_t *)&instr;
102
103     size_t curlen;
104     int32_t disp;
105     uint32_t flags = inp->in_data.in_def.in_flags;
106     int octal = ((dhp->dh_flags & DIS_OCTAL) != 0);
107
108     if ((dhx->dhx_debug & DIS_DEBUG_PRTFMT) != 0) {
109         prt_field("op", f->f2.op, 2);
110         prt_field("op2", f->f2.op2, 3);
111
112         switch (FLG_DISP_VAL(flags)) {
113             case DISP22:
114                 prt_field("cond", f->f2a.cond, 4);
115                 prt_field("a", f->f2a.a, 1);
116                 prt_field("p", f->f2b.p, 1);
117                 prt_field("disp22", f->f2a.disp22, 22);
118                 break;
119
120             case DISP19:
121                 prt_field("cond", f->f2a.cond, 4);
122                 prt_field("a", f->f2a.a, 1);
123                 prt_field("p", f->f2b.p, 1);
124                 prt_field("cc", f->f2b.cc, 2);
125                 prt_field("disp19", f->f2b.disp19, 19);
126                 break;
127         }
128     }
129 }
```

```

809         case DISP16:
810             prt_field("bit 28", ((instr & (1L << 28)) >> 28), 1);
811             prt_field("rcond", f->f2c.cond, 3);
812             prt_field("p", f->f2c.p, 1);
813             prt_field("rs1", f->f2c.rs1, 5);
814             prt_field("dl6hi", f->f2c.dl6hi, 2);
815             prt_field("dl6lo", f->f2c.dl6lo, 14);
816             break;
817         }
818     }

820     if (f->f2b.op2 == 0x01 && idx == 0x00 && f->f2b.p == 1 &&
821         f->f2b.cc == 0x02 && ((dhx->dhx_debug & DIS_DEBUG_SYN_ALL) != 0)) {
822         name = "iprefetch";
823         flags = FLG_RS1(REG_NONE)|FLG_DISP(DISP19);
824     }

827     switch (FLG_DISP_VAL(flags)) {
828     case DISP22:
829         disp = sign_extend(f->f2a.disp22, 22);
830         break;

832     case DISP19:
833         disp = sign_extend(f->f2b.disp19, 19);
834         break;

836     case DISP16:
837         disp = sign_extend((f->f2c.dl6hi << 14)|f->f2c.dl6lo, 16);
838         break;

840     }

842     disp *= 4;

844     if ((FLG_RS1_VAL(flags) == REG_ICC) || (FLG_RS1_VAL(flags) == REG_FCC))
845         r = get_regname(dhp, FLG_RS1_VAL(flags), f->f2b.cc);
846     else
847         r = get_regname(dhp, FLG_RS1_VAL(flags), f->f2c.rs1);

849     if (r == NULL)
850         return (-1);

852     if (f->f2a.a == 1)
853         annul = ",a";

855     if ((flags & FLG_PRED) != 0) {
856         if (f->f2b.p == 0) {
857             pred = ",pn";
858         } else {
859             if ((dhx->dhx_debug & DIS_DEBUG_COMPAT) != 0)
860                 pred = ",pt";
861         }
862     }

864     (void) dis_snprintf(buf, sizeof (buf), "%s%s", name, annul, pred);
866     (void) snprintf(buf, sizeof (buf), "%s%s", name, annul, pred);
865     prt_name(dhp, buf, 1);

868     switch (FLG_DISP_VAL(flags)) {
869     case DISP22:
870         bprintf(dhp,
871             (octal != 0) ? "%s0%-11lo <" : "%s0x%-10lx <",
872             (disp < 0) ? "-" : "+",

```

```

873             (disp < 0) ? (-disp) : disp);
874         break;

876     case DISP19:
877         bprintf(dhp,
878             (octal != 0) ? "%s, %s0%-5lo <" :
879             "%s, %s0x%-04lx <", r,
880             (disp < 0) ? "-" : "+",
881             (disp < 0) ? (-disp) : disp);
882         break;

884     case DISP16:
885         bprintf(dhp,
886             (octal != 0) ? "%s, %s0%-6lo <" : "%s, %s0x%-5lx <",
887             r,
888             (disp < 0) ? "-" : "+",
889             (disp < 0) ? (-disp) : disp);
890         break;
891     }

893     curlen = strlen(dhx->dhx_buf);
894     dhp->dh_lookup(dhp->dh_data, dhp->dh_addr + (int64_t)disp,
895                 dhx->dhx_buf + curlen, dhx->dhx_buflen - curlen - 1, NULL, NULL);

897     (void) strcat(dhx->dhx_buf, ">", dhx->dhx_buflen);

899     return (0);
900 }
unchanged_portion_omitted_

2751 /*
2752  * just a handy function that takes care of managing the buffer length
2753  * w/ printf
2754  */

2756 /*
2757  * PRINTF LIKE 1
2758  */
2759 static void
2760 bprintf(dis_handle_t *dhp, const char *fmt, ...)
2761 {
2762     dis_handle_sparc_t *dhx = dhp->dh_arch_private;
2763     size_t curlen;
2764     va_list ap;

2766     curlen = strlen(dhx->dhx_buf);

2768     va_start(ap, fmt);
2769     (void) dis_vsnprintf(dhx->dhx_buf + curlen, dhx->dhx_buflen -
2770                        curlen, fmt, ap);
2771     (void) vsnprintf(dhx->dhx_buf + curlen, dhx->dhx_buflen - curlen, fmt,
2772                    ap);
2771     va_end(ap);
2772 }
unchanged_portion_omitted_

```

```

*****
4934 Fri Aug 1 15:57:44 2014
new/usr/src/lib/libdisasm/common/libdisasm.c
patch fix-lint
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
26  * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
27 #endif /* ! codereview */
28 */

30 #include <libdisasm.h>
31 #include <stdlib.h>
32 #ifdef DIS_STANDALONE
33 #include <mdb/mdb_modapi.h>
34 #define _MDB
35 #include <mdb/mdb_io.h>
36 #else
37 #include <stdio.h>
38 #endif /* ! codereview */
39 #endif

41 #include "libdisasm_impl.h"

43 static int _dis_errno;

45 /*
46  * If we're building the standalone library, then we only want to
47  * include support for disassembly of the native architecture.
48  * The regular shared library should include support for all
49  * architectures.
50  */
51 #if !defined(DIS_STANDALONE) || defined(__i386) || defined(__amd64)
52 extern dis_arch_t dis_arch_i386;
53 #endif
54 #if !defined(DIS_STANDALONE) || defined(__sparc)
55 extern dis_arch_t dis_arch_sparc;
56 #endif

58 static dis_arch_t *dis_archs[] = {
59 #if !defined(DIS_STANDALONE) || defined(__i386) || defined(__amd64)
60     &dis_arch_i386,
61 #endif

```

```

62 #if !defined(DIS_STANDALONE) || defined(__sparc)
63     &dis_arch_sparc,
64 #endif
65     NULL
66 };

68 /*
69  * For the standalone library, we need to link against mdb's malloc/free.
70  * Otherwise, use the standard malloc/free.
71  */
72 #ifndef DIS_STANDALONE
73 void *
74 dis_zalloc(size_t bytes)
75 {
76     return (mdb_zalloc(bytes, UM_SLEEP));
77 }

79 void
80 dis_free(void *ptr, size_t bytes)
81 {
82     mdb_free(ptr, bytes);
83 }
84 #else
85 void *
86 dis_zalloc(size_t bytes)
87 {
88     return (calloc(1, bytes));
89 }

91 /*ARGSUSED*/
92 void
93 dis_free(void *ptr, size_t bytes)
94 {
95     free(ptr);
96 }
97 #endif

99 int
100 dis_seterrno(int error)
101 {
102     _dis_errno = error;
103     return (-1);
104 }

106 int
107 dis_errno(void)
108 {
109     return (_dis_errno);
110 }

112 const char *
113 dis_strerror(int error)
114 {
115     switch (error) {
116     case E_DIS_NOMEM:
117         return ("out of memory");
118     case E_DIS_INVALIDFLAG:
119         return ("invalid flags for this architecture");
120     case E_DIS_UNSUPARCH:
121         return ("unsupported machine architecture");
122     default:
123         return ("unknown error");
124     }
125 }

127 void

```



```

128 dis_set_data(dis_handle_t *dhp, void *data)
129 {
130     dhp->dh_data = data;
131 }

133 void
134 dis_flags_set(dis_handle_t *dhp, int f)
135 {
136     dhp->dh_flags |= f;
137 }

139 void
140 dis_flags_clear(dis_handle_t *dhp, int f)
141 {
142     dhp->dh_flags &= ~f;
143 }

145 void
146 dis_handle_destroy(dis_handle_t *dhp)
147 {
148     dhp->dh_arch->da_handle_detach(dhp);
149     dis_free(dhp, sizeof (dis_handle_t));
150 }

152 dis_handle_t *
153 dis_handle_create(int flags, void *data, dis_lookup_f lookup_func,
154                 dis_read_f read_func)
155 {
156     dis_handle_t *dhp;
157     dis_arch_t *arch = NULL;
158     int i;

160     /* Select an architecture based on flags */
161     for (i = 0; dis_archs[i] != NULL; i++) {
162         if (dis_archs[i]->da_supports_flags(flags)) {
163             arch = dis_archs[i];
164             break;
165         }
166     }
167     if (arch == NULL) {
168         (void) dis_seterrno(E_DIS_UNSUPARCH);
169         return (NULL);
170     }

172     if ((dhp = dis_zalloc(sizeof (dis_handle_t))) == NULL) {
173         (void) dis_seterrno(E_DIS_NOMEM);
174         return (NULL);
175     }
176     dhp->dh_arch = arch;
177     dhp->dh_lookup = lookup_func;
178     dhp->dh_read = read_func;
179     dhp->dh_flags = flags;
180     dhp->dh_data = data;

182     /*
183      * Allow the architecture-specific code to allocate
184      * its private data.
185      */
186     if (arch->da_handle_attach(dhp) != 0) {
187         dis_free(dhp, sizeof (dis_handle_t));
188         /* dis errno already set */
189         return (NULL);
190     }

192     return (dhp);
193 }

```

```

195 int
196 dis_disassemble(dis_handle_t *dhp, uint64_t addr, char *buf, size_t buflen)
197 {
198     return (dhp->dh_arch->da_disassemble(dhp, addr, buf, buflen));
199 }

201 uint64_t
202 dis_previnstr(dis_handle_t *dhp, uint64_t pc, int n)
203 {
204     return (dhp->dh_arch->da_previnstr(dhp, pc, n));
205 }

207 int
208 dis_min_instrlen(dis_handle_t *dhp)
209 {
210     return (dhp->dh_arch->da_min_instrlen(dhp));
211 }

213 int
214 dis_max_instrlen(dis_handle_t *dhp)
215 {
216     return (dhp->dh_arch->da_max_instrlen(dhp));
217 }

219 int
220 dis_instrlen(dis_handle_t *dhp, uint64_t pc)
221 {
222     return (dhp->dh_arch->da_instrlen(dhp, pc));
223 }

225 int
226 dis_vsnprintf(char *restrict s, size_t n, const char *restrict format,
227              va_list args)
228 {
229     #ifdef DIS_STANDALONE
230         return (mdb_iob_vsnprintf(s, n, format, args));
231     #else
232         return (vsnprintf(s, n, format, args));
233     #endif
234 }

236 int
237 dis_snprintf(char *restrict s, size_t n, const char *restrict format, ...)
238 {
239     va_list args;

241     va_start(args, format);
242     n = dis_vsnprintf(s, n, format, args);
243     va_end(args);

245     return (n);
246 }
247 #endif /* ! codereview */

```

```

*****
2087 Fri Aug 1 15:57:44 2014
new/usr/src/lib/libdisasm/common/libdisasm_impl.h
patch fix-lint
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright 2006 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  * Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
26  * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
27 #endif /* ! codereview */
28 */

30 #ifndef LIBDISASM_IMPL_H
31 #define LIBDISASM_IMPL_H

33 #include <stdarg.h>
34 #include <sys/sysmacros.h>

36 #endif /* ! codereview */
37 #ifdef __cplusplus
38 extern "C" {
39 #endif

41 typedef struct dis_arch {
42     int (*da_supports_flags)(int);
43     int (*da_handle_attach)(dis_handle_t *);
44     void (*da_handle_detach)(dis_handle_t *);
45     int (*da_disassemble)(dis_handle_t *, uint64_t, char *, size_t);
46     uint64_t (*da_previnstr)(dis_handle_t *, uint64_t, int n);
47     int (*da_min_instrlen)(dis_handle_t *);
48     int (*da_max_instrlen)(dis_handle_t *);
49     int (*da_instrlen)(dis_handle_t *, uint64_t);
50 } dis_arch_t;

52 struct dis_handle {
53     void *dh_data;
54     int dh_flags;
55     dis_lookup_f dh_lookup;
56     dis_read_f dh_read;
57     uint64_t dh_addr;

59     dis_arch_t dh_arch;
60     void *dh_arch_private;
61 };

```

```

63 extern int dis_seterrno(int);

65 extern void *dis_zalloc(size_t);
66 extern void dis_free(void *, size_t);
67 extern int dis_vsnprintf(char *restrict, size_t, const char *restrict, va_list);
68 extern int dis_snprintf(char *restrict, size_t, const char *restrict, ...);
69 #endif /* ! codereview */

71 #ifdef __cplusplus
72 }
73 #endif

75 #endif /* LIBDISASM_IMPL_H */

```

new/usr/src/lib/libdisasm/common/linktest_stand.c

1

1331 Fri Aug 1 15:57:44 2014

new/usr/src/lib/libdisasm/common/linktest_stand.c

patch fix-lint

```
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright 2007 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */

27 #pragma ident "%Z%M% %I% %E% SMI"

27 /*
28  * This file is used to verify that the standalone's external dependencies
29  * haven't changed in a way that'll break things that use it.
30 */

32 void mdb_free(void) {}
33 void mdb_snprintf(void) {}
34 void mdb_iob_vsnprintf(void) {}
35 void snprintf(void) {}
36 void vsnprintf(void) {}
35 void mdb_zalloc(void) {}
36 void strcmp(void) {}
37 void strlen(void) {}
38 void strlcat(void) {}
39 void strncpy(void) {}
40 void strncmp(void) {}
41 void memcpy(void) {}
42 void _memcpy(void) {}
```