

new/usr/src/cmd/mdb/common/kmdb/kmdb_kvm.c

1

62997 Fri Jan 25 18:07:27 2019

new/usr/src/cmd/mdb/common/kmdb/kmdb_kvm.c

10132 smatch fixes for MDB

Reviewed by: Andy Fiddaman <andy@omniosce.org>

unchanged portion omitted

```
2373 static void
2374 kmt_destroy(mdb_tgt_t *t)
2375 {
2376     kmt_data_t *kmt = t->t_data;
2377     kmt_module_t *km, *pkm;

2379     mdb_nv_destroy(&kmt->kmt_modules);
2380     for (km = mdb_list_prev(&kmt->kmt_modlist); km != NULL; km = pkm) {
2381         pkm = mdb_list_prev(km);
2382         mdb_free(km, sizeof (kmt_module_t));
2383     }

2385     if (!kmt_defbp_lock)
2386         kmt_defbp_destroy_all();

2388     if (kmt->kmt_trapmap != NULL)
2389         mdb_free(kmt->kmt_trapmap, BT_SIZEOFMAP(kmt->kmt_trapmap));

2391     if (kmt != NULL)
2391         mdb_free(kmt, sizeof (kmt_data_t));
2392 }
```

unchanged portion omitted

new/usr/src/cmd/mdb/common/mdb/mdb.c

1

```
*****
35962 Fri Jan 25 18:07:27 2019
new/usr/src/cmd/mdb/common/mdb/mdb.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */
25 /*
26  * Copyright (c) 2012 by Delphix. All rights reserved.
27  * Copyright (c) 2018, Joyent, Inc.
28  * Copyright (c) 2012 Joyent, Inc. All rights reserved.
29 */
30 /*
31  * Modular Debugger (MDB)
32  *
33  * Refer to the white paper "A Modular Debugger for Solaris" for information
34  * on the design, features, and goals of MDB. See /shared/sac/PSARC/1999/169
35  * for copies of the paper and related documentation.
36  *
37  * This file provides the basic construction and destruction of the debugger's
38  * global state, as well as the main execution loop, mdb_run().  MDB maintains
39  * a stack of execution frames (mdb_frame_t's) that keep track of its current
40  * state, including a stack of input and output buffers, walk and memory
41  * garbage collect lists, and a list of commands (mdb_cmd_t's).  As the
42  * parser consumes input, it fills in a list of commands to execute, and then
43  * invokes mdb_call(), below.  A command consists of a dcmd, telling us
44  * what function to execute, and a list of arguments and other invocation-
45  * specific data.  Each frame may have more than one command, kept on a list,
46  * when multiple commands are separated by | operators.  New frames may be
47  * stacked on old ones by nested calls to mdb_run: this occurs when, for
48  * example, in the middle of processing one input source (such as a file
49  * or the terminal), we invoke a dcmd that in turn calls mdb_eval().  mdb_eval
50  * will construct a new frame whose input source is the string passed to
51  * the eval function, and then execute this frame to completion.
52  */
53
54 #include <sys/param.h>
55 #include <stropts.h>
56
57 #define _MDB_PRIVATE
58 #include <mdb/mdb.h>
```

new/usr/src/cmd/mdb/common/mdb/mdb.c

2

```
60 #include <mdb/mdb_context.h>
61 #include <mdb/mdb_argvec.h>
62 #include <mdb/mdb_signal.h>
63 #include <mdb/mdb_macalias.h>
64 #include <mdb/mdb_module.h>
65 #include <mdb/mdb_modapi.h>
66 #include <mdb/mdb_string.h>
67 #include <mdb/mdb_callb.h>
68 #include <mdb/mdb_debug.h>
69 #include <mdb/mdb_frame.h>
70 #include <mdb/mdb_conf.h>
71 #include <mdb/mdb_err.h>
72 #include <mdb/mdb_lex.h>
73 #include <mdb/mdb_io.h>
74 #include <mdb/mdb_ctf.h>
75 #ifdef _KMDB
76 #include <kmdb/kmdb_module.h>
77 #endif
78
79 /*
80  * Macro for testing if a dcmd's return status (x) indicates that we should
81  * abort the current loop or pipeline.
82  */
83 #define DCMD_ABORTED(x) ((x) == DCMD_USAGE || (x) == DCMD_ABORT)
84
85 extern const mdb_dcmd_t mdb_dcmd_builtins[];
86 extern mdb_dis_ctor_f *const mdb_dis_builtins[];
87
88 /*
89  * Variable discipline for toggling MDB_FL_PSYM based on the value of the
90  * undocumented '_' variable.  Once adb(1) has been removed from the system,
91  * we should just remove this functionality and always disable PSYM for macros.
92  */
93 static uintmax_t
94 psym_disc_get(const mdb_var_t *v)
95 {
96     int i = (mdb.m_flags & MDB_FL_PSYM) ? 1 : 0;
97     int j = (MDB_NV_VALUE(v) != 0) ? 1 : 0;
98
99     if ((i ^ j) == 0)
100         MDB_NV_VALUE((mdb_var_t *)v) = j ^ 1;
101
102     return (MDB_NV_VALUE(v));
103 }
104
105 _____ unchanged_portion_omitted _____
106
107 void
108 mdb_call_tab(mdb_idcmd_t *idcp, mdb_tab_cookie_t *mcp, uint_t flags,
109             uintmax_t argc, mdb_arg_t *argv)
110 {
111     if (idcp->idc_tabp == NULL)
112         return;
113
114     (void) idcp->idc_tabp(mcp, flags, argc, argv);
115     idcp->idc_tabp(mcp, flags, argc, argv);
116 }
117
118 _____ unchanged_portion_omitted _____
```

```

*****
84294 Fri Jan 25 18:07:27 2019
new/usr/src/cmd/mdb/common/mdb/mdb_cmds.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
_____unchanged_portion_omitted_____

```

```

1509 static const char *
1510 map_name(const mdb_map_t *map, const char *name)
1511 {
1512     if (map->map_flags & MDB_TGT_MAP_HEAP)
1513         return ("[ heap ]");
1514     if (name != NULL && name[0] != 0)
1515         return (name);
1516
1517     if (map->map_flags & MDB_TGT_MAP_SHMEM)
1518         return ("[ shmem ]");
1519     if (map->map_flags & MDB_TGT_MAP_STACK)
1520         return ("[ stack ]");
1521     if (map->map_flags & MDB_TGT_MAP_ANON)
1522         return ("[ anon ]");
1523     if (map->map_name[0] == '\0')
1524         return ("[ unknown ]");
1525     if (map->map_name != NULL)
1526         return (map->map_name);
1527     return ("[ unknown ]");
1528 }
_____unchanged_portion_omitted_____

```

```

2338 /*ARGSUSED*/
2339 static int
2340 cmd_head(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
2341 {
2342     uint64_t cnt = 10;
2343     const char *c;
2344     mdb_pipe_t p;
2345
2346     if (!(flags & DCMD_PIPE))
2347         return (DCMD_USAGE);
2348
2349     if (argc == 1 || argc == 2) {
2350         const char *num;
2351
2352         if (argc == 1) {
2353             if (argv[0].a_type != MDB_TYPE_STRING ||
2354                 *argv[0].a_un.a_str != '-')
2355                 return (DCMD_USAGE);
2356
2357             num = argv[0].a_un.a_str + 1;
2358
2359         } else {
2360             if (argv[0].a_type != MDB_TYPE_STRING ||
2361                 strcmp(argv[0].a_un.a_str, "-n") != 0)
2362                 return (DCMD_USAGE);
2363
2364             num = argv[1].a_un.a_str;
2365         }
2366
2367         for (cnt = 0, c = num; *c != '\0' && isdigit(*c); c++)
2368             cnt = cnt * 10 + (*c - '0');
2369
2370         if (*c != '\0')
2371             return (DCMD_USAGE);

```

```

2373     } else if (argc != 0) {
2374         return (DCMD_USAGE);
2375     }
2376
2377     mdb_get_pipe(&p);
2378
2379     if (p.pipe_data == NULL)
2380         return (DCMD_OK);
2381     p.pipe_len = MIN(p.pipe_len, cnt);
2382
2383     if (flags & DCMD_PIPE_OUT) {
2384         mdb_set_pipe(&p);
2385     } else {
2386         while (p.pipe_len-- > 0)
2387             mdb_printf("%lx\n", *p.pipe_data++);
2388     }
2389
2390     return (DCMD_OK);
2391 }
_____unchanged_portion_omitted_____

```

```

*****
22003 Fri Jan 25 18:07:27 2019
new/usr/src/cmd/mdb/common/mdb/mdb_nm.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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 2008 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */
25
26 /*
27  * Copyright (c) 2018, Joyent, Inc.
28 */
29
30 #include <sys/elf.h>
31 #include <sys/elf_SPARC.h>
32
33 #include <libproc.h>
34 #include <libctf.h>
35 #include <stdlib.h>
36 #include <string.h>
37 #include <fcntl.h>
38 #include <errno.h>
39
40 #include <mdb/mdb_string.h>
41 #include <mdb/mdb_argvec.h>
42 #include <mdb/mdb_nv.h>
43 #include <mdb/mdb_fmt.h>
44 #include <mdb/mdb_target.h>
45 #include <mdb/mdb_err.h>
46 #include <mdb/mdb_debug.h>
47 #include <mdb/mdb_conf.h>
48 #include <mdb/mdb_module.h>
49 #include <mdb/mdb_modapi.h>
50 #include <mdb/mdb_stdlib.h>
51 #include <mdb/mdb_lex.h>
52 #include <mdb/mdb_io_impl.h>
53 #include <mdb/mdb_help.h>
54 #include <mdb/mdb_disasm.h>
55 #include <mdb/mdb_frame.h>
56 #include <mdb/mdb_evset.h>
57 #include <mdb/mdb_print.h>
58 #include <mdb/mdb_nm.h>
59 #include <mdb/mdb_set.h>
60 #include <mdb/mdb_demangle.h>

```

```

61 #include <mdb/mdb.h>
62
63 enum {
64     NM_FMT_INDEX      = 0x0001,      /* -f ndx */
65     NM_FMT_VALUE     = 0x0002,      /* -f val */
66     NM_FMT_SIZE      = 0x0004,      /* -f size */
67     NM_FMT_TYPE      = 0x0008,      /* -f type */
68     NM_FMT_BIND      = 0x0010,      /* -f bind */
69     NM_FMT_OTHER     = 0x0020,      /* -f oth */
70     NM_FMT_SHNDX     = 0x0040,      /* -f shndx */
71     NM_FMT_NAME      = 0x0080,      /* -f name */
72     NM_FMT_CTYPE     = 0x0100,      /* -f ctype */
73     NM_FMT_OBJECT    = 0x0200,      /* -f obj */
74
75     NM_FMT_CTFID     = 0x1000      /* -f ctfid */
76 };
77
78 unchanged portion omitted
79
80 /*ARGSUSED*/
81 int
82 cmd_nm(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
83 {
84     enum {
85         NM_DYNSYM      = 0x0001,      /* -D (use dynsym) */
86         NM_DEC         = 0x0002,      /* -d (decimal output) */
87         NM_GLOBAL      = 0x0004,      /* -g (globals only) */
88         NM_NOHDRS     = 0x0008,      /* -h (suppress header) */
89         NM_OCT         = 0x0010,      /* -o (octal output) */
90         NM_UNDEF      = 0x0020,      /* -u (undefs only) */
91         NM_HEX         = 0x0040,      /* -x (hex output) */
92         NM_SORT_NAME   = 0x0080,      /* -n (sort by name) */
93         NM_SORT_VALUE  = 0x0100,      /* -v (sort by value) */
94         NM_PRVSYM     = 0x0200,      /* -P (use private symtab) */
95         NM_PRTASGN    = 0x0400      /* -p (print in asgn syntax) */
96     };
97
98     mdb_subopt_t opt_fmt_opts[] = {
99         { NM_FMT_INDEX, "ndx" },
100        { NM_FMT_VALUE, "val" },
101        { NM_FMT_SIZE, "sz" },
102        { NM_FMT_TYPE, "type" },
103        { NM_FMT_BIND, "bind" },
104        { NM_FMT_OTHER, "oth" },
105        { NM_FMT_SHNDX, "shndx" },
106        { NM_FMT_NAME, "name" },
107        { NM_FMT_CTYPE, "ctype" },
108        { NM_FMT_OBJECT, "obj" },
109        { NM_FMT_CTFID, "ctfid" },
110        { 0, NULL }
111    };
112
113     mdb_subopt_t opt_type_opts[] = {
114         { NM_TYPE_NOTY, "noty" },
115         { NM_TYPE_OBJT, "objt" },
116         { NM_TYPE_FUNC, "func" },
117         { NM_TYPE_SECT, "sect" },
118         { NM_TYPE_FILE, "file" },
119         { NM_TYPE_COMM, "comm" },
120         { NM_TYPE_TLS, "tls" },
121         { NM_TYPE_REGI, "regi" },
122         { 0, NULL }
123    };
124
125     uint_t optf = 0;
126     uint_t opt_fmt;
127     uint_t opt_types;

```

```

557     int i;

559     mdb_tgt_sym_f *callback;
560     uint_t which, type;

562     char *object = (char *)MDB_TGT_OBJ_EVERY;
563     int hwidth;
564     size_t nsyms = 0;

566     nm_sym_t *syms, *symp;

568     nm_iter_info_t nii;

570     /* default output columns */
571     opt_fmt = NM_FMT_VALUE | NM_FMT_SIZE | NM_FMT_TYPE | NM_FMT_BIND |
572             NM_FMT_OTHER | NM_FMT_SHNDX | NM_FMT_NAME;

574     /* default output types */
575     opt_types = NM_TYPE_NOTY | NM_TYPE_OBJT | NM_TYPE_FUNC | NM_TYPE_SECT |
576             NM_TYPE_FILE | NM_TYPE_COMM | NM_TYPE_TLS | NM_TYPE_REGI;

578     i = mdb_getopts(argc, argv,
579         'D', MDB_OPT_SETBITS, NM_DYNSYM, &optf,
580         'P', MDB_OPT_SETBITS, NM_PRVSYM, &optf,
581         'd', MDB_OPT_SETBITS, NM_DEC, &optf,
582         'g', MDB_OPT_SETBITS, NM_GLOBAL, &optf,
583         'h', MDB_OPT_SETBITS, NM_NOHDRS, &optf,
584         'n', MDB_OPT_SETBITS, NM_SORT_NAME, &optf,
585         'o', MDB_OPT_SETBITS, NM_OCT, &optf,
586         'p', MDB_OPT_SETBITS, NM_PRTASGN | NM_NOHDRS, &optf,
587         'u', MDB_OPT_SETBITS, NM_UNDEF, &optf,
588         'v', MDB_OPT_SETBITS, NM_SORT_VALUE, &optf,
589         'x', MDB_OPT_SETBITS, NM_HEX, &optf,
590         'f', MDB_OPT_SUBOPTS, opt_fmt_opts, &opt_fmt,
591         't', MDB_OPT_SUBOPTS, opt_type_opts, &opt_types,
592         NULL);

594     if (i != argc) {
595         if (flags & DCMD_ADDRSPEC)
596             return (DCMD_USAGE);

598         if (argc != 0 && (argc - i) == 1) {
599             if (argv[i].a_type != MDB_TYPE_STRING ||
600                 argv[i].a_un.a_str[0] == '-')
601                 return (DCMD_USAGE);
602             else
603                 object = (char *)argv[i].a_un.a_str;
604         } else
605             return (DCMD_USAGE);
606     }

608     if ((optf & (NM_DEC | NM_HEX | NM_OCT)) == 0) {
609         switch (mdb.m_radix) {
610             case 8:
611                 optf |= NM_OCT;
612                 break;
613             case 10:
614                 optf |= NM_DEC;
615                 break;
616             default:
617                 optf |= NM_HEX;
618         }
619     }

621     switch (optf & (NM_DEC | NM_HEX | NM_OCT)) {
622     case NM_DEC:

```

```

623 #ifdef _LP64
624     nii.nii_pfmt = "%-20llu";
625     nii.nii_ofmt = "%-5u";
626     hwidth = 20;
627 #else
628     nii.nii_pfmt = "%-10llu";
629     nii.nii_ofmt = "%-5u";
630     hwidth = 10;
631 #endif
632     break;
633     case NM_HEX:
634 #ifdef _LP64
635     nii.nii_pfmt = "0x%016llx";
636     nii.nii_ofmt = "0x%-3x";
637     hwidth = 18;
638 #else
639     nii.nii_pfmt = "0x%08llx";
640     nii.nii_ofmt = "0x%-3x";
641     hwidth = 10;
642 #endif
643     break;
644     case NM_OCT:
645 #ifdef _LP64
646     nii.nii_pfmt = "%-22llo";
647     nii.nii_ofmt = "%-5o";
648     hwidth = 22;
649 #else
650     nii.nii_pfmt = "%-11llo";
651     nii.nii_ofmt = "%-5o";
652     hwidth = 11;
653 #endif
654     break;
655     default:
656     mdb_warn("-d/-o/-x options are mutually exclusive\n");
657     return (DCMD_USAGE);
658     }

660     if (object != MDB_TGT_OBJ_EVERY && (optf & NM_PRVSYM)) {
661     mdb_warn("-P/object options are mutually exclusive\n");
662     return (DCMD_USAGE);
663     }

665     if ((flags & DCMD_ADDRSPEC) && (optf & NM_PRVSYM)) {
666     mdb_warn("-P/address options are mutually exclusive\n");
667     return (DCMD_USAGE);
668     }

670     if (!(optf & NM_NOHDRS)) {
671     mdb_printf("%<u>");
672     mdb_table_print(opt_fmt, " ",
673         MDB_TBL_PRINT, NM_FMT_INDEX, "Index",
674         MDB_TBL_PRINT, NM_FMT_OBJECT, "%-15s", "Object",
675         MDB_TBL_PRINT, NM_FMT_VALUE, "%-*s", hwidth, "Value",
676         MDB_TBL_PRINT, NM_FMT_SIZE, "%-*s", hwidth, "Size",
677         MDB_TBL_PRINT, NM_FMT_TYPE, "%-5s", "Type",
678         MDB_TBL_PRINT, NM_FMT_BIND, "%-5s", "Bind",
679         MDB_TBL_PRINT, NM_FMT_OTHER, "%-5s", "Other",
680         MDB_TBL_PRINT, NM_FMT_SHNDX, "%-8s", "Shndx",
681         MDB_TBL_PRINT, NM_FMT_CTFID, "%-9s", "CTF ID",
682         MDB_TBL_PRINT, NM_FMT_CTYPE, "%-50s", "C Type",
683         MDB_TBL_PRINT, NM_FMT_NAME, "%s", "Name",
684         MDB_TBL_DONE);

686     mdb_printf("%</u>\n");
687     }

```

```

689     nii.nii_flags = opt_fmt;
690     nii.nii_types = opt_types;

692     if (optf & NM_DYNSYM)
693         which = MDB_TGT_DYNSYM;
694     else
695         which = MDB_TGT_SYMTAB;

697     if (optf & NM_GLOBAL)
698         type = MDB_TGT_BIND_GLOBAL | MDB_TGT_TYPE_ANY;
699     else
700         type = MDB_TGT_BIND_ANY | MDB_TGT_TYPE_ANY;

702     if (flags & DCMD_ADDRSPEC)
703         optf |= NM_SORT_NAME; /* use sorting path if only one symbol */

705     if (optf & (NM_SORT_NAME | NM_SORT_VALUE)) {
706         char name[MDB_SYM_NAMLEN];
707         GElf_Sym sym;
708         mdb_syminfo_t si;

710         if (optf & NM_UNDEF)
711             callback = nm_cnt_undef;
712         else
713             callback = nm_cnt_any;

715         if (flags & DCMD_ADDRSPEC) {
716             const mdb_map_t *mp;
717             /* gather relevant data for the specified addr */

719             nii.nii_fp = mdb_tgt_addr_to_ctf(mdb.m_target, addr);

721             if (mdb_tgt_lookup_by_addr(mdb.m_target, addr,
722                                     MDB_SYM_FUZZY, name, sizeof(name), &sym,
723                                     &si) == -1) {
724                 mdb_warn("%lr", addr);
725                 return (DCMD_ERR);
726             }

728             if ((mp = mdb_tgt_addr_to_map(mdb.m_target, addr))
729                 != NULL) {
730                 object = mdb_alloc(strlen(mp->map_name) + 1,
731                                  UM_SLEEP | UM_GC);

733                 (void) strcpy(object, mp->map_name);

735                 /*
736                  * Try to find a better match for the syminfo.
737                  */
738                 (void) mdb_tgt_lookup_by_name(mdb.m_target,
739                                               object, name, &sym, &si);
740             }

742             (void) callback(&nsyms, &sym, name, &si, object);

744         } else if (optf & NM_PRVSYM) {
745             nsyms = mdb_gelf_syntab_size(mdb.m_prsym);
746         } else {
747             (void) mdb_tgt_symbol_iter(mdb.m_target, object,
748                                       which, type, callback, &nsyms);
749         }

751         if (nsyms == 0)
752             return (DCMD_OK);

754         syms = symp = mdb_alloc(sizeof(nm_sym_t) * nsyms,

```

```

755             UM_SLEEP | UM_GC);

757         nii.nii_symp = &symp;

759         if (optf & NM_UNDEF)
760             callback = nm_get_undef;
761         else
762             callback = nm_get_any;

764         if (flags & DCMD_ADDRSPEC) {
765             (void) callback(&nii, &symp, name, &si, object);
766         } else if (optf & NM_PRVSYM) {
767             nm_gelf_syntab_iter(mdb.m_prsym, object, MDB_TGT_PRVSYM,
768                               callback, &nii);
769         } else if (nm_symbol_iter(object, which, type, callback,
770                                  &nii) == -1) {
771             mdb_warn("failed to iterate over symbols");
772             return (DCMD_ERR);
773         }

775         if (optf & NM_SORT_NAME)
776             qsort(syms, nsyms, sizeof(nm_sym_t), nm_compare_name);
777         else
778             qsort(syms, nsyms, sizeof(nm_sym_t), nm_compare_val);
779     }

781     if ((optf & (NM_PRVSYM | NM_PRTASGN)) == (NM_PRVSYM | NM_PRTASGN))
782         callback = nm_asgn;
783     else if (optf & NM_UNDEF)
784         callback = nm_undef;
785     else
786         callback = nm_any;

788     if (optf & (NM_SORT_NAME | NM_SORT_VALUE)) {
789         for (symp = syms; nsyms-- != 0; symp++) {
790             nii.nii_fp = symp->nm_fp;

792             (void) callback(&nii, &symp->nm_sym, symp->nm_name,
793                           callback(&nii, &symp->nm_sym, symp->nm_name,
794                                   &symp->nm_si, symp->nm_object));

796         } else {
797             if (optf & NM_PRVSYM) {
798                 nm_gelf_syntab_iter(mdb.m_prsym, object, MDB_TGT_PRVSYM,
799                                   callback, &nii);

801             } else if (nm_symbol_iter(object, which, type, callback, &nii)
802                       == -1) {
803                 mdb_warn("failed to iterate over symbols");
804                 return (DCMD_ERR);
805             }
806         }

808         return (DCMD_OK);
809     }

```

_____unchanged_portion_omitted_____

148678 Fri Jan 25 18:07:28 2019

new/usr/src/cmd/mdb/common/mdb/mdb_proc.c

10132 smatch fixes for MDB

Reviewed by: Andy Fiddaman <andy@omniosce.org>

unchanged_portion_omitted

```
1943 /*
1944  * Function to set a variable in the internal environment, which is used when
1945  * creating new processes. Note that it is possible that 'nameval' can refer to
1946  * read-only memory, if mdb calls putenv() on an existing value before calling
1947  * this function. While we should avoid this situation, this function is
1948  * designed to be robust in the face of such changes.
1949  */
1950 static void
1951 pt_env_set(pt_data_t *pt, const char *nameval)
1952 {
1953     mdb_var_t *v;
1954     char *equals, *val;
1955     const char *name;
1956     size_t len;
1957
1958     if ((equals = strchr(nameval, '=') != NULL) {
1959         val = strdup(nameval);
1960         equals = val + (equals - nameval);
1961     } else {
1962         /*
1963          * nameval doesn't contain an equals character. Convert this to
1964          * be 'nameval='.
1965          */
1966         len = strlen(nameval);
1967         val = mdb_alloc(len + 2, UM_SLEEP);
1968         (void) mdb_snprintf(val, len + 2, "%s=", nameval);
1969         equals = val + len;
1970     }
1971
1972     /* temporary truncate the string for lookup/insert */
1973     *equals = '\0';
1974     v = mdb_nv_lookup(&pt->p_env, val);
1975
1976     if (v != NULL) {
1977         char *old = mdb_nv_get_cookie(v);
1978         mdb_free(old, strlen(old) + 1);
1979         name = mdb_nv_get_name(v);
1980     } else {
1981         /*
1982          * The environment is created using MDB_NV_EXTNAME, so we must
1983          * provide external storage for the variable names.
1984          */
1985         name = strdup(val);
1986     }
1987
1988     *equals = '=';
1989
1990     (void) mdb_nv_insert(&pt->p_env, name, NULL, (uintptr_t)val,
1991         MDB_NV_EXTNAME);
1992
1993     if (equals)
1994         *equals = '=';
1995 }
1996
1997 unchanged_portion_omitted
```

new/usr/src/cmd/mdb/common/mdb/mdb_tab.c

1

```
*****
14767 Fri Jan 25 18:07:28 2019
new/usr/src/cmd/mdb/common/mdb/mdb_tab.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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 (c) 2013 by Delphix. All rights reserved.
23  * Copyright (c) 2018, Joyent, Inc.
24  * Copyright (c) 2012 Joyent, Inc. All rights reserved.
25  * Copyright (c) 2013 Josef 'Jeff' Sipek <jeffpc@josefsipek.net>
26 */
27 * This file contains all of the interfaces for mdb's tab completion engine.
28 * Currently some interfaces are private to mdb and its internal implementation,
29 * those are in mdb_tab.h. Other pieces are public interfaces. Those are in
30 * mdb_modapi.h.
31 *
32 * Memory allocations in tab completion context have to be done very carefully.
33 * We need to think of ourselves as the same as any other command that is being
34 * executed by the user, which means we must use UM_GC to handle being
35 * interrupted.
36 */
37
38 #include <mdb/mdb_modapi.h>
39 #include <mdb/mdb_ctf.h>
40 #include <mdb/mdb_ctf_impl.h>
41 #include <mdb/mdb_string.h>
42 #include <mdb/mdb_module.h>
43 #include <mdb/mdb_debug.h>
44 #include <mdb/mdb_print.h>
45 #include <mdb/mdb_nv.h>
46 #include <mdb/mdb_tab.h>
47 #include <mdb/mdb_target.h>
48 #include <mdb/mdb.h>
49
50 #include <ctype.h>
51
52 /*
53  * There may be another way to do this, but this works well enough.
54  */
55 #define COMMAND_SEPARATOR ":"
56
57 /*
58  * find_command_start --
59  */
```

new/usr/src/cmd/mdb/common/mdb/mdb_tab.c

2

```
60 *      Given a buffer find the start of the last command.
61 */
62 static char *
63 tab_find_command_start(char *buf)
64 {
65     char *offset = strstr(buf, COMMAND_SEPARATOR);
66
67     if (offset == NULL)
68         return (NULL);
69
70     for (;;) {
71         char *next = strstr(offset + strlen(COMMAND_SEPARATOR),
72                             COMMAND_SEPARATOR);
73
74         if (next == NULL) {
75             return (offset);
76         }
77
78         offset = next;
79     }
80 }
81
82 _____ unchanged_portion_omitted _____
83
391 /*
392  * Determine whether the specified name is a valid tab completion for
393  * the given command. If the name is a valid tab completion then
394  * it will be saved in the mdb_tab_cookie_t.
395  */
396 void
397 mdb_tab_insert(mdb_tab_cookie_t *mcp, const char *name)
398 {
399     size_t matches, index;
400     mdb_var_t *v;
401
402     /*
403      * If we have a match set, then we want to verify that we actually match
404      * it.
405      */
406     if (strncmp(name, mcp->mtc_base, strlen(mcp->mtc_base)) != 0)
407         if (mcp->mtc_base != NULL &&
408             strncmp(name, mcp->mtc_base, strlen(mcp->mtc_base)) != 0)
409             return;
410
411     v = mdb_nv_lookup(&mcp->mtc_nv, name);
412     if (v != NULL)
413         return;
414
415     (void) mdb_nv_insert(&mcp->mtc_nv, name, NULL, 0, MDB_NV_RDONLY);
416
417     matches = mdb_tab_size(mcp);
418     if (matches == 1) {
419         (void) strcpy(mcp->mtc_match, name, MDB_SYM_NAMLEN);
420     } else {
421         index = 0;
422         while (mcp->mtc_match[index] &&
423             mcp->mtc_match[index] == name[index])
424             index++;
425
426         mcp->mtc_match[index] = '\0';
427     }
428 }
429
430 _____ unchanged_portion_omitted _____
431
442 const char *
443 mdb_tab_match(mdb_tab_cookie_t *mcp)
444 {
```



```
445     return (mcp->mtc_match + strlen(mcp->mtc_base));
446     size_t blen;

448     if (mcp->mtc_base == NULL)
449         blen = 0;
450     else
451         blen = strlen(mcp->mtc_base);
452     return (mcp->mtc_match + blen);
446 }
```

_____unchanged_portion_omitted_____

```

*****
10879 Fri Jan 25 18:07:28 2019
new/usr/src/cmd/mdb/common/modules/crypto/spi.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24  */
25
26 /*
27  * Copyright (c) 2018, Joyent, Inc.
28  */
29
30 /*
31  * mdb dcmds for selected structures from
32  * usr/src/uts/common/sys/crypto/spi.h
33  *
34  * Also the mdb module housekeeping
35  */
36
37 #include <sys/mdb_modapi.h>
38 #include <sys/modctl.h>
39 #include <sys/crypto/api.h>
40 #include <sys/crypto/common.h>
41 #include <sys/crypto/spi.h>
42 #include <sys/crypto/impl.h>
43 #include "crypto_cmds.h"
44
45
46 const mdb_bitmask_t extf_flags[] = {
47     { "NIL", (ulong_t)-1, 0L },
48     { "CRYPTO_EXTF_RNG", CRYPTO_EXTF_RNG, CRYPTO_EXTF_RNG },
49     { "CRYPTO_EXTF_WRITE_PROTECTED", CRYPTO_EXTF_WRITE_PROTECTED,
50       CRYPTO_EXTF_WRITE_PROTECTED },
51     { "CRYPTO_EXTF_LOGIN_REQUIRED", CRYPTO_EXTF_LOGIN_REQUIRED,
52       CRYPTO_EXTF_LOGIN_REQUIRED },
53     { "CRYPTO_EXTF_USER_PIN_INITIALIZED", CRYPTO_EXTF_USER_PIN_INITIALIZED,
54       CRYPTO_EXTF_USER_PIN_INITIALIZED },
55     { "CRYPTO_EXTF_CLOCK_ON_TOKEN", CRYPTO_EXTF_CLOCK_ON_TOKEN,
56       CRYPTO_EXTF_CLOCK_ON_TOKEN },
57     { "CRYPTO_EXTF_PROTECTED_AUTHENTICATION_PATH",
58       CRYPTO_EXTF_PROTECTED_AUTHENTICATION_PATH,
59       CRYPTO_EXTF_PROTECTED_AUTHENTICATION_PATH },
60     { "CRYPTO_EXTF_DUAL_CRYPT_OPERATIONS",

```

```

61     CRYPTO_EXTF_DUAL_CRYPT_OPERATIONS,
62     CRYPTO_EXTF_DUAL_CRYPT_OPERATIONS },
63     { "CRYPTO_EXTF_TOKEN_INITIALIZED", CRYPTO_EXTF_TOKEN_INITIALIZED,
64       CRYPTO_EXTF_TOKEN_INITIALIZED },
65     { "CRYPTO_EXTF_USER_PIN_COUNT_LOW", CRYPTO_EXTF_USER_PIN_COUNT_LOW,
66       CRYPTO_EXTF_USER_PIN_COUNT_LOW },
67     { "CRYPTO_EXTF_USER_PIN_FINAL_TRY", CRYPTO_EXTF_USER_PIN_FINAL_TRY,
68       CRYPTO_EXTF_USER_PIN_FINAL_TRY },
69     { "CRYPTO_EXTF_USER_PIN_LOCKED", CRYPTO_EXTF_USER_PIN_LOCKED,
70       CRYPTO_EXTF_USER_PIN_LOCKED },
71     { "CRYPTO_EXTF_USER_PIN_TO_BE_CHANGED",
72       CRYPTO_EXTF_USER_PIN_TO_BE_CHANGED,
73       CRYPTO_EXTF_USER_PIN_TO_BE_CHANGED },
74     { "CRYPTO_EXTF_SO_PIN_COUNT_LOW", CRYPTO_EXTF_SO_PIN_COUNT_LOW,
75       CRYPTO_EXTF_SO_PIN_COUNT_LOW },
76     { "CRYPTO_EXTF_SO_PIN_FINAL_TRY", CRYPTO_EXTF_SO_PIN_FINAL_TRY,
77       CRYPTO_EXTF_SO_PIN_FINAL_TRY },
78     { "CRYPTO_EXTF_SO_PIN_LOCKED", CRYPTO_EXTF_SO_PIN_LOCKED,
79       CRYPTO_EXTF_SO_PIN_LOCKED },
80     { "CRYPTO_EXTF_SO_PIN_TO_BE_CHANGED", CRYPTO_EXTF_SO_PIN_TO_BE_CHANGED,
81       CRYPTO_EXTF_SO_PIN_TO_BE_CHANGED },
82     { NULL, 0, 0 }
83 };
84
85 unchanged portion omitted
86
87
88
89
90 /*ARGSUSED*/
91 int
92 crypto_mech_info(uintptr_t addr, uint_t flags, int argc,
93                 const mdb_arg_t *argv)
94 {
95     crypto_mech_info_t minfo;
96     const char *unit = "bits";
97
98     if (!(flags & DCMD_ADDRSPEC))
99         if (!flags & DCMD_ADDRSPEC)
100             return (DCMD_USAGE);
101
102     if (mdb_vread(&minfo, sizeof (crypto_mech_info_t), addr)
103         == -1) {
104         mdb_warn("cannot read addr %p", addr);
105         return (DCMD_ERR);
106     }
107     mdb_printf("cm_mech_name\t\t%s\n", minfo.cm_mech_name);
108     mdb_printf("cm_mech_number\t\t%lld\n", minfo.cm_mech_number);
109     mdb_printf("cm_func_group_mask\t\t0x%x:\t<%b>\n",
110               minfo.cm_func_group_mask, minfo.cm_func_group_mask, mech_bits);
111     if (minfo.cm_keysize_unit & CRYPTO_KEYSIZE_UNIT_IN_BYTES)
112         unit = "bytes";
113     mdb_printf("cm_min_key_length\t\t%lu %s\n", minfo.cm_min_key_length,
114               unit);
115     mdb_printf("cm_max_key_length\t\t%lu %s\n", minfo.cm_max_key_length,
116               unit);
117
118     return (DCMD_OK);
119 }
120
121 unchanged portion omitted

```

```

*****
14118 Fri Jan 25 18:07:28 2019
new/usr/src/cmd/mdb/common/modules/fcp/fcp.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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 2008 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24  *
25  * Copyright (c) 2018, Joyent, Inc.
26  * FCP mdb module
27 */

29 #include <sys/mdb_modapi.h>
30 #include <sys/mutex.h>
31 #include <sys/modctl.h>
32 #include <sys/scsi/scsi.h>
33 #include <sys/sunndi.h>
34 #include <sys/fibre-channel/fc.h>
35 #include <sys/fibre-channel/ulp/fcpvar.h>

37 static struct fcp_port port;
38 static struct fcp_tgt tgt;
39 static struct fcp_lun lun;
40 static uint32_t tgt_hash_index;

43 /*
44  * Leadville fcp walker/dcmd code
45  */

47 static int
48 fcp_walk_i(mdb_walk_state_t *wsp)
49 {
50     if (wsp->walk_addr == NULL &&
51         mdb_readvar(&wsp->walk_addr, "fcp_port_head") == -1) {
52         mdb_warn("failed to read 'fcp_port_head'");
53         return (WALK_ERR);
54     }

56     wsp->walk_data = mdb_alloc(sizeof (struct fcp_port), UM_SLEEP);
57     return (WALK_NEXT);
58 }

```

```

306 /*
307  * Leadville targets walker/dcmd code
308  */

310 static int
311 targets_walk_i(mdb_walk_state_t *wsp)
312 {
313     if (wsp->walk_addr == NULL) {
314         mdb_warn("Can not perform global walk\n");
315         return (WALK_ERR);
316     }

318     /*
319      * Input should be a fcp_port, so read it to get the port_tgt
320      * table's head
321      */

323     if (mdb_vread(&port, sizeof (struct fcp_port), wsp->walk_addr) !=
324         sizeof (struct fcp_port)) {
325         mdb_warn("Unable to read in the port structure address\n");
326         return (WALK_ERR);
327     }

329     tgt_hash_index = 0;

331     while (tgt_hash_index < FCP_NUM_HASH &&
332           port.port_tgt_hash_table[tgt_hash_index] == NULL) {
333         while ((port.port_tgt_hash_table[tgt_hash_index] == NULL) &&
334              (tgt_hash_index < FCP_NUM_HASH)) {
335             tgt_hash_index++;
336         }

338         wsp->walk_addr = (uintptr_t)(port.port_tgt_hash_table[tgt_hash_index]);

340         wsp->walk_data = mdb_alloc(sizeof (struct fcp_tgt), UM_SLEEP);

342         return (WALK_NEXT);
343     }

344 static int
345 targets_walk_s(mdb_walk_state_t *wsp)
346 {
347     int status;

349     if ((wsp->walk_addr == NULL) &&
350         (tgt_hash_index >= (FCP_NUM_HASH - 1))) {
351         return (WALK_DONE);
352     }

354     if (mdb_vread(wsp->walk_data, sizeof (struct fcp_tgt),
355                 wsp->walk_addr) == -1) {
356         mdb_warn("failed to read fcp_tgt at %p", wsp->walk_addr);
357         return (WALK_DONE);
358     }

360     status = wsp->walk_callback(wsp->walk_addr, wsp->walk_data,
361                               wsp->walk_cbdata);

363     wsp->walk_addr =
364         (uintptr_t)(((struct fcp_tgt *)wsp->walk_data)->tgt_next);

366     if (wsp->walk_addr == NULL) {
367         /*
368          * locate the next hash list

```

unchanged portion omitted

```
368         */
370         tgt_hash_index++;
372         while (tgt_hash_index < FCP_NUM_HASH &&
373                port.port_tgt_hash_table[tgt_hash_index] == NULL)
372         while ((port.port_tgt_hash_table[tgt_hash_index] == NULL) &&
373                (tgt_hash_index < FCP_NUM_HASH)) {
374             tgt_hash_index++;
375         }
376         if (tgt_hash_index == FCP_NUM_HASH) {
377             /* You're done */
378             return (status);
379         }
381         wsp->walk_addr =
382             (uintptr_t)(port.port_tgt_hash_table[tgt_hash_index]);
383     }
385     return (status);
386 }
unchanged_portion_omitted
```

new/usr/src/cmd/mdb/common/modules/fctl/fctl.c

1

```
*****
33551 Fri Jan 25 18:07:28 2019
new/usr/src/cmd/mdb/common/modules/fctl/fctl.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */
25
26 /*
27  * Copyright (c) 2018, Joyent, Inc.
28 */
29
30 #include <sys/mdb_modapi.h>
31 #include <sys/mutex.h>
32 #include <sys/modctl.h>
33 #include <time.h>
34 #include <sys/fibre-channel/fc.h>
35 #include <sys/fibre-channel/impl/fctl_private.h>
36 #include <sys/fibre-channel/impl/fc_ulpif.h>
37 #include <sys/fibre-channel/impl/fc_portif.h>
38 #include <sys/fibre-channel/impl/fc_fcaif.h>
39
40
41 /*
42  * If we #include <string.h> then other definitions fail. This is
43  * the easiest way of getting access to the function
44  */
45 extern char *strtok(char *string, const char *sepset);
46
47 /* we need 26 bytes for the cftime() call */
48 #define TIMESTAMPSIZE 26 * sizeof (char)
49
50 /* for backward compatibility */
51 typedef struct fc_trace_dmsgv1 {
52     int         id_size;
53     int         id_flag;
54     time_t      id_time;
55     caddr_t     id_buf;
56     struct fc_trace_dmsgv1 *id_next;
57 } fc_trace_dmsgv1_t;
58
59 unchanged portion omitted
```

1089 int

new/usr/src/cmd/mdb/common/modules/fctl/fctl.c

2

```
1090 fc_trace_dump(uintptr_t addr, uint_t flags, int argc, const mdb_arg_t *argv)
1091 {
1092     fc_trace_logq_t logq;
1093     uint_t         pktnum = 0;
1094     uint_t         printed = 0; /* have we printed anything? */
1095
1096     uintptr_t      pktstart = 0;
1097     uintptr_t      pktend = UINT_MAX;
1098     int            rval = DCMD_OK;
1099
1100     if (mdb_vread(&logq, sizeof (logq), addr) != sizeof (logq)) {
1101         mdb_warn("Failed to read log queue in kernel");
1102         return (DCMD_ERR);
1103     }
1104
1105     if (mdb_getopts(argc, argv,
1106         's', MDB_OPT_UINTPTR, &pktstart,
1107         'e', MDB_OPT_UINTPTR, &pktend) != argc) {
1108         return (DCMD_USAGE);
1109     }
1110
1111     if (pktstart > pktend) {
1112         return (DCMD_USAGE);
1113     }
1114
1115     if ((logq.il_flags & FC_TRACE_LOGQ_V2) != 0) {
1116         if (logq.il_flags & FC_TRACE_LOGQ_V2 != 0) {
1117             rval = fc_dump_logmsg((fc_trace_dmsg_t *)logq.il_msgh, pktstart,
1118                 pktend, &printed);
1119         } else {
1120             rval = fc_dump_old_logmsg((fc_trace_dmsgv1_t *)logq.il_msgh,
1121                 pktstart, pktend, &printed);
1122         }
1123     }
1124
1125     if (rval != DCMD_OK) {
1126         return (rval);
1127     }
1128
1129     if (printed == 0) {
1130         mdb_printf("No packets in the buffer match the"
1131             " criteria given");
1132     }
1133
1134     return (rval);
1135 }
1136
1137 unchanged portion omitted
```

new/usr/src/cmd/mdb/common/modules/genunix/ndievents.c

1

```
*****
6344 Fri Jan 25 18:07:28 2019
new/usr/src/cmd/mdb/common/modules/genunix/ndievents.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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, Version 1.0 only
6  * (the "License").  You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23 * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
24 * Use is subject to license terms.
25 */
27 /*
28 * Copyright (c) 2018, Joyent, Inc.
29 */
30 #pragma ident      "%Z%M% %I%      %E% SMI"
31 #include "ndievents.h"
32 #include <sys/sunndi.h>
33 #include <sys/ndi_impldefs.h>
34 #include <sys/dditypes.h>
35 #include <sys/ddi_impldefs.h>
36 #include <sys/sunddi.h>
37 #include <sys/param.h>
```

```
40 int
41 dip_to_pathname(struct dev_info *device, char *path, int buflen) {
42
43     char *bp;
44     char *addr;
45     char addr_str[32];
46     char nodename[MAXNAMELEN];
47     struct dev_info devi_parent;
48
49     if (!device) {
50         mdb_warn("Unable to access devinfo.");
51         return (-1);
52     }
53
54     if (device->devi_parent == NULL) {
55         if (mdb_readstr(nodename, sizeof (nodename),
56             (uintptr_t)device->devi_node_name) == -1) {
57             return (-1);
58         }
59     }
```

new/usr/src/cmd/mdb/common/modules/genunix/ndievents.c

2

```
60         if (sizeof (nodename) > (buflen - strlen(path))) {
61             return (-1);
62         }
63
64         strncpy(path, nodename, sizeof (nodename));
65         return (0);
66     }
67
68     if (mdb_vread(&devi_parent, sizeof (struct dev_info),
69         (uintptr_t)device->devi_parent) == -1) {
70         mdb_warn("Unable to access devi_parent at %p",
71             (uintptr_t)device->devi_parent);
72         return (-1);
73     }
74
75     if (dip_to_pathname(&devi_parent, path, buflen) == -1) {
76         return (-1);
77     }
78
79     if (mdb_readstr(nodename, sizeof (nodename),
80         (uintptr_t)device->devi_node_name) == -1) {
81         return (-1);
82     }
83
84     if (device->devi_node_state < DS_INITIALIZED) {
85         addr_str[0] = '\0';
86         strncpy(addr_str, '\0', sizeof ('\0'));
87     } else {
88         addr = device->devi_addr;
89         if (mdb_readstr(addr_str, sizeof (addr_str),
90             (uintptr_t)addr) == -1) {
91             return (-1);
92         }
93     }
94
95     bp = path + strlen(path);
96
97     if (addr_str[0] == '\0') {
98         (void) mdb_snprintf(bp, buflen - strlen(path), "%s", nodename);
99     } else {
100         (void) mdb_snprintf(bp, buflen - strlen(path), "%s@%s",
101             nodename, addr_str);
102     }
103     return (0);
104 }
```

unchanged_portion_omitted

```

*****
105981 Fri Jan 25 18:07:28 2019
new/usr/src/cmd/mdb/common/modules/libumem/umem.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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 2009 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24  */
25
26 /*
27  * Copyright (c) 2018, Joyent, Inc.
28  * Copyright 2012 Joyent, Inc. All rights reserved.
29  * Copyright (c) 2013, 2015 by Delphix. All rights reserved.
30  */
31 #include "umem.h"
32
33 #include <sys/vmem_impl_user.h>
34 #include <umem_impl.h>
35
36 #include <alloca.h>
37 #include <limits.h>
38 #include <mdb/mdb_whatinfo.h>
39 #include <thr_uberdata.h>
40
41 #include "misc.h"
42 #include "leaky.h"
43 #include "dist.h"
44
45 #include "umem_pagesize.h"
46
47 #define UM_ALLOCATED      0x1
48 #define UM_FREE          0x2
49 #define UM_BUFCTL        0x4
50 #define UM_HASH          0x8
51
52 int umem_ready;
53
54 static int umem_stack_depth_warned;
55 static uint32_t umem_max_ncpus;
56 uint32_t umem_stack_depth;
57
58 size_t umem_pagesize;

```

```

60 #define UMEM_READVAR(var) \
61     (umem_readvar(&(var), #var) == -1 && \
62     (mdb_warn("failed to read "#var), 1))
63
64 int
65 umem_update_variables(void)
66 {
67     size_t pagesize;
68
69     /*
70      * Figure out which type of umem is being used; if it's not there
71      * yet, succeed quietly.
72      */
73     if (umem_set_standalone() == -1) {
74         umem_ready = 0;
75         return (0); /* umem not there yet */
76     }
77
78     /*
79      * Solaris 9 used a different name for umem_max_ncpus. It's
80      * cheap backwards compatibility to check for both names.
81      */
82     if (umem_readvar(&umem_max_ncpus, "umem_max_ncpus") == -1 &&
83         umem_readvar(&umem_max_ncpus, "max_ncpus") == -1) {
84         mdb_warn("unable to read umem_max_ncpus or max_ncpus");
85         return (-1);
86     }
87     if (UMEM_READVAR(umem_ready))
88         return (-1);
89     if (UMEM_READVAR(umem_stack_depth))
90         return (-1);
91     if (UMEM_READVAR(pagesize))
92         return (-1);
93
94     if (umem_stack_depth > UMEM_MAX_STACK_DEPTH) {
95         if (umem_stack_depth_warned == 0) {
96             mdb_warn("umem_stack_depth corrupted (%d > %d)\n",
97                 umem_stack_depth, UMEM_MAX_STACK_DEPTH);
98             umem_stack_depth_warned = 1;
99         }
100         umem_stack_depth = 0;
101     }
102
103     umem_pagesize = pagesize;
104
105     return (0);
106 }

```

unchanged portion omitted

```

2090 static int
2091 whatis_walk_seg(uintptr_t addr, const vmem_seg_t *vs, whatis_info_t *wi)
2092 {
2093     mdb_whatinfo_t *w = wi->wi_w;
2094
2095     size_t size = vs->vs_end - vs->vs_start;
2096     uintptr_t cur;
2097
2098     /* We're not interested in anything but alloc and free segments */
2099     if (vs->vs_type != VMEM_ALLOC && vs->vs_type != VMEM_FREE)
2100         return (WALK_NEXT);
2101
2102     while (mdb_whatinfo_match(w, vs->vs_start, size, &cur)) {
2103         mdb_whatinfo_report_object(w, cur, vs->vs_start, "");
2104     }
2105     /*

```

```
2106         * If we're not printing it separately, provide the vmem_seg
2107         * pointer if it has a stack trace.
2108         */
2109         if ((mdb_what_is_flags(w) & WHATIS_QUIET) &&
2110             ((mdb_what_is_flags(w) & WHATIS_BUFCTL) != 0 ||
2111              (vs->vs_type == VMEM_ALLOC && vs->vs_depth != 0))) {
2112             mdb_printf("vmem_seg %p ", addr);
2113         }
2114
2115         mdb_printf("%s from %s vmem arena",
2116                  (vs->vs_type == VMEM_ALLOC) ? "allocated" : "freed",
2117                  wi->wi_vmem->vm_name);
2118
2119         if (!(mdb_what_is_flags(w) & WHATIS_QUIET))
2120             if (!(mdb_what_is_flags(w) & WHATIS_QUIET))
2121                 whatis_call_printer(vmem_seg, addr);
2122         else
2123             mdb_printf("\n");
2124
2125         return (WHATIS_WALKRET(w));
2126     }
2127     unchanged_portion_omitted
```



```

*****
19068 Fri Jan 25 18:07:29 2019
new/usr/src/cmd/mdb/intel/mdb/proc_amd64dep.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
_____unchanged_portion_omitted_____

```

```

279 static const char *
280 fpcw2str(uint32_t cw, char *buf, size_t nbytes)
281 {
282     char *end = buf + nbytes;
283     char *p = buf;
284
285     buf[0] = '\0';
286
287     /*
288      * Decode all masks in the 80387 control word.
289      */
290     if (cw & FPIM)
291         p += mdb_snprintf(p, (size_t)(end - p), "|IM");
292     if (cw & FPDN)
293         p += mdb_snprintf(p, (size_t)(end - p), "|DM");
294     if (cw & FPZM)
295         p += mdb_snprintf(p, (size_t)(end - p), "|ZM");
296     if (cw & FPOM)
297         p += mdb_snprintf(p, (size_t)(end - p), "|OM");
298     if (cw & FPUM)
299         p += mdb_snprintf(p, (size_t)(end - p), "|UM");
300     if (cw & FPPM)
301         p += mdb_snprintf(p, (size_t)(end - p), "|PM");
302     if (cw & FPPC)
303         p += mdb_snprintf(p, (size_t)(end - p), "|PC");
304     if (cw & FPRC)
305         p += mdb_snprintf(p, (size_t)(end - p), "|RC");
306     if (cw & FPIC)
307         p += mdb_snprintf(p, (size_t)(end - p), "|IC");
308
309     /*
310      * Decode precision, rounding, and infinity options in control word.
311      */
312     if (cw & FPSIG24)
313         p += mdb_snprintf(p, (size_t)(end - p), "|SIG24");
314     if (cw & FPSIG53)
315         p += mdb_snprintf(p, (size_t)(end - p), "|SIG53");
316     if (cw & FPSIG64)
317         p += mdb_snprintf(p, (size_t)(end - p), "|SIG64");
318
319     if ((cw & FPRC) == (FPRD|FPRU))
320         p += mdb_snprintf(p, (size_t)(end - p), "|RTZ");
321     else if (cw & FPRD)
322         p += mdb_snprintf(p, (size_t)(end - p), "|RD");
323     else if (cw & FPRU)
324         p += mdb_snprintf(p, (size_t)(end - p), "|RU");
325     else
326         p += mdb_snprintf(p, (size_t)(end - p), "|RTN");
327
328     if (cw & FPA)
329         p += mdb_snprintf(p, (size_t)(end - p), "|A");
330     else
331         p += mdb_snprintf(p, (size_t)(end - p), "|P");
332     if (cw & WFPB17)
333         p += mdb_snprintf(p, (size_t)(end - p), "|WFPB17");
334     if (cw & WFPB24)
335         p += mdb_snprintf(p, (size_t)(end - p), "|WFPB24");

```

```

335         if (buf[0] == '|')
336             return (buf + 1);
337
338     return ("0");
339 }
_____unchanged_portion_omitted_____

```

```

*****
16441 Fri Jan 25 18:07:29 2019
new/usr/src/cmd/mdb/intel/mdb/proc_ia32dep.c
10132 smatch fixes for MDB
Reviewed by: Andy Fiddaman <andy@omniosce.org>
*****
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, Version 1.0 only
6  * (the "License").  You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23 * Copyright 2004 Sun Microsystems, Inc.  All rights reserved.
24 * Use is subject to license terms.
25 */
26 /*
27 * Copyright (c) 2018, Joyent, Inc.
28 * Copyright 2015 Joyent, Inc.
29 */
30 /*
31 * User Process Target Intel 32-bit component
32 *
33 * This file provides the ISA-dependent portion of the user process target.
34 * For more details on the implementation refer to mdb_proc.c.
35 */
36
37 #include <mdb/mdb_proc.h>
38 #include <mdb/mdb_kreg.h>
39 #include <mdb/mdb_err.h>
40 #include <mdb/mdb_ia32util.h>
41 #include <mdb/mdb.h>
42
43 #include <sys/ucontext.h>
44 #include <sys/frame.h>
45 #include <libproc.h>
46 #include <sys/fp.h>
47 #include <ieeeefp.h>
48
49 #include <stddef.h>
50
51 const mdb_tgt_regdesc_t pt_regdesc[] = {
52     { "gs", GS, MDB_TGT_R_EXPORT },
53     { "fs", FS, MDB_TGT_R_EXPORT },
54     { "es", ES, MDB_TGT_R_EXPORT },
55     { "ds", DS, MDB_TGT_R_EXPORT },
56     { "edi", EDI, MDB_TGT_R_EXPORT },
57     { "di", EDI, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
58     { "esi", ESI, MDB_TGT_R_EXPORT },
59     { "si", ESI, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },

```

```

60     { "ebp", EBP, MDB_TGT_R_EXPORT },
61     { "bp", EBP, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
62     { "kesp", ESP, MDB_TGT_R_EXPORT },
63     { "ksp", ESP, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
64     { "ebx", EBX, MDB_TGT_R_EXPORT },
65     { "bx", EBX, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
66     { "bh", EBX, MDB_TGT_R_EXPORT | MDB_TGT_R_8H },
67     { "bl", EBX, MDB_TGT_R_EXPORT | MDB_TGT_R_8L },
68     { "edx", EDX, MDB_TGT_R_EXPORT },
69     { "dx", EDX, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
70     { "dh", EDX, MDB_TGT_R_EXPORT | MDB_TGT_R_8H },
71     { "dl", EDX, MDB_TGT_R_EXPORT | MDB_TGT_R_8L },
72     { "ecx", ECX, MDB_TGT_R_EXPORT },
73     { "cx", ECX, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
74     { "ch", ECX, MDB_TGT_R_EXPORT | MDB_TGT_R_8H },
75     { "cl", ECX, MDB_TGT_R_EXPORT | MDB_TGT_R_8L },
76     { "eax", EAX, MDB_TGT_R_EXPORT },
77     { "ax", EAX, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
78     { "ah", EAX, MDB_TGT_R_EXPORT | MDB_TGT_R_8H },
79     { "al", EAX, MDB_TGT_R_EXPORT | MDB_TGT_R_8L },
80     { "trapno", TRAPNO, MDB_TGT_R_EXPORT },
81     { "err", ERR, MDB_TGT_R_EXPORT },
82     { "eip", EIP, MDB_TGT_R_EXPORT },
83     { "cs", CS, MDB_TGT_R_EXPORT },
84     { "eflags", EFL, MDB_TGT_R_EXPORT },
85     { "esp", UESP, MDB_TGT_R_EXPORT },
86     { "sp", UESP, MDB_TGT_R_EXPORT | MDB_TGT_R_16 },
87     { "ss", SS, MDB_TGT_R_EXPORT },
88     { NULL, 0, 0 }
89 };
90
91 unchanged_portion_omitted
92
93 static const char *
94 fpcw2str(uint32_t cw, char *buf, size_t nbytes)
95 {
96     char *end = buf + nbytes;
97     char *p = buf;
98
99     buf[0] = '\0';
100
101     /*
102      * Decode all masks in the 80387 control word.
103      */
104     if (cw & FPIM)
105         p += mdb_snprintf(p, (size_t)(end - p), "|IM");
106     if (cw & FPDM)
107         p += mdb_snprintf(p, (size_t)(end - p), "|DM");
108     if (cw & FPZM)
109         p += mdb_snprintf(p, (size_t)(end - p), "|ZM");
110     if (cw & FPOM)
111         p += mdb_snprintf(p, (size_t)(end - p), "|OM");
112     if (cw & FPUM)
113         p += mdb_snprintf(p, (size_t)(end - p), "|UM");
114     if (cw & FPPM)
115         p += mdb_snprintf(p, (size_t)(end - p), "|PM");
116     if (cw & FPPC)
117         p += mdb_snprintf(p, (size_t)(end - p), "|PC");
118     if (cw & FPRC)
119         p += mdb_snprintf(p, (size_t)(end - p), "|RC");
120     if (cw & FPIC)
121         p += mdb_snprintf(p, (size_t)(end - p), "|IC");
122
123     /*
124      * Decode precision, rounding, and infinity options in control word.
125      */
126     if (cw & FPSIG24)

```

```
254     p += mdb_snprintf(p, (size_t)(end - p), "|SIG24");
253     if (cw & FPSIG53)
254         p += mdb_snprintf(p, (size_t)(end - p), "|SIG53");
255     if (cw & FPSIG64)
256         p += mdb_snprintf(p, (size_t)(end - p), "|SIG64");

258     if ((cw & FPRC) == (FPRD|FPRU))
259         p += mdb_snprintf(p, (size_t)(end - p), "|RTZ");
260     else if (cw & FPRD)
261         p += mdb_snprintf(p, (size_t)(end - p), "|RD");
262     else if (cw & FPRU)
263         p += mdb_snprintf(p, (size_t)(end - p), "|RU");
264     else
265         p += mdb_snprintf(p, (size_t)(end - p), "|RTN");

267     if (cw & FPA)
268         p += mdb_snprintf(p, (size_t)(end - p), "|A");
269     else
270         p += mdb_snprintf(p, (size_t)(end - p), "|P");
271     if (cw & WFPB17)
272         p += mdb_snprintf(p, (size_t)(end - p), "|WFPB17");
273     if (cw & WFPB24)
274         p += mdb_snprintf(p, (size_t)(end - p), "|WFPB24");

276     if (buf[0] == '|')
277         return (buf + 1);

279     return ("0");
280 }
```

unchanged portion omitted