

new/usr/src/cmd/tic/Makefile.common

1

1444 Mon Jan 7 12:52:34 2019

new/usr/src/cmd/tic/Makefile.common

10075 make usr/src/tools smatch clean

```
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 2017 Nexenta Systems, Inc.
26 #
27 # Copyright (c) 2018, Joyent, Inc.
```

```
29 PROG=          tic
30 OBJS=          tic_captab.o \
31               tic_error.o \
32               tic_hash.o \
33               tic_main.o \
34               tic_parse.o \
35               tic_read.o \
36               tic_scan.o
37 SRCS=          $(OBJS:%.o=%.c)
```

```
39 CPPFLAGS +=   -I$(SRC)/lib/libcurses/screen
```

```
41 LDLIBS +=     -lcurses
42 MAPFILES=     $(MAPFILE.INT) $(MAPFILE.NGB)
43 LDFLAGS +=    $(MAPFILES:%=-M%)
```

```
45 CERRWARN +=   -_gcc=-Wno-char-subscripts
46 CERRWARN +=   -_gcc=-Wno-implicit-function-declaration
47 CERRWARN +=   -_gcc=-Wno-parentheses
48 CERRWARN +=   -_gcc=-Wno-uninitialized
```

```
50 # not linted
51 SMATCH=off
```

new/usr/src/cmd/vtfontcvt/Makefile.common

1

729 Mon Jan 7 12:52:36 2019

new/usr/src/cmd/vtfontcvt/Makefile.common

10075 make usr/src/tools smatch clean

```
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 2017 Toomas Soome <tsoome@me.com>
14 #
15 # Copyright (c) 2018, Joyent, Inc.
```

```
17 PROG= vtfontcvt
18 OBJ= vtfontcvt.o
```

```
20 LZ4COMMONDIR= $(SRC)/common/ficl/softcore
21 LZ4_OBJS= lz4.o
22 OBJ += $(LZ4_OBJS)
```

```
24 LDLIBS += -lumem
```

```
26 # not linted
27 SMATCH=off
```

```
29 %.o: $(LZ4COMMONDIR)/%.c
30 $(COMPILE.c) $<
31 $(POST_PROCESS_O)
```

```
33 CSTD = $(CSTD_GNU99)
```

new/usr/src/cmd/zic/Makefile.common

1

305 Mon Jan 7 12:52:38 2019

new/usr/src/cmd/zic/Makefile.common

10075 make usr/src/tools smatch clean

1 #
2 # Copyright 2009 Sun Microsystems, Inc. All rights reserved.
3 # Use is subject to license terms.
4 #
5 # Copyright (c) 2018, Joyent, Inc.

7 PROG= zic

9 OBJS= ialloc.o \
10 scheck.o \
11 zic.o \
12 SRCS= \$(OBJS:%.o=%.c)

14 CERRWARN += _gcc=-Wno-unused-variable
15 CERRWARN += _gcc=-Wno-uninitialized

17 SMOFF += signed

new/usr/src/tools/aw/Makefile

1

1498 Mon Jan 7 12:52:39 2019

new/usr/src/tools/aw/Makefile

10075 make usr/src/tools smatch clean

```
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 # Copyright (c) 2018, Joyent, Inc.

28 PROG      = aw

30 include ../Makefile.tools

32 CFLAGS += $(CCVERBOSE)
33 CERRWARN += _gcc=-Wno-uninitialized

35 # not linted
36 SMATCH=off

38 LINTFLAGS += -ux -Xa -errchk=locfmtchk,parentheses

40 CPPFLAGS += -DDEFAULT_AS_DIR='$(GNU_ROOT)/bin'
41 CPPFLAGS += -DDEFAULT_AS64_DIR='$(GNU_ROOT)/bin'
42 CPPFLAGS += -DDEFAULT_M4_DIR='"/usr/ccs/bin'
43 CPPFLAGS += -DDEFAULT_M4LIB_DIR='"/usr/ccs/lib'
44 CPPFLAGS += -DDEFAULT_CPP_DIR='"/usr/ccs/lib'

46 .KEEP_STATE:

48 all:      $(PROG)

50 install: all .WAIT $(ROOTONBLDMACHPROG)

52 lint:    lint_PROG

54 clean:

56 include ../Makefile.targ
```

new/usr/src/tools/btxld/Makefile

1

```
*****
1357 Mon Jan 7 12:52:40 2019
new/usr/src/tools/btxld/Makefile
10075 make usr/src/tools smatch clean
*****
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 2016 Toomas Soome <tsoome@me.com>
14 #
15 # Copyright (c) 2018, Joyent, Inc.

17 include ../Makefile.tools

19 .KEEP_STATE:

21 COMMON_SRC=    $(SRC)/cmd/boot/common
22 EINFO_SRC=     $(COMMON_SRC)/bblk_einfo.c
23 UTILS_SRC=     $(COMMON_SRC)/boot_utils.c
24 EXTRA_SRC=    $(COMMON_SRC)/mboot_extra.c

26 PROG=         btxld
27 MAN1ONBLDFILES= btxld.1onbld
28 SRCS=         btxld.c elfh.c version.c $(UTILS_SRC) $(EINFO_SRC) $(EXTRA_SRC)
29 OBJS=         btxld.o elfh.o version.o bblk_einfo.o mboot_extra.o boot_utils.o
30 LDLIBS +=     -lmd5
31 CSTD=         $(CSTD_GNU99)
32 CPPFLAGS +=   -I$(SRC)/uts/common -I$(COMMON_SRC)

34 # not linted
35 SMATCH=off

37 $(ROOTONBLDMAN1ONBLDFILES) := FILEMODE= 644

39 all: $(PROG) $(MAN1ONBLDFILES)

41 $(PROG): $(OBJS)
42     $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
43     $(POST_PROCESS)

45 install: all .WAIT $(ROOTONBLDMACHPROG) $(ROOTONBLDMAN1ONBLDFILES)

47 clean:
48     $(RM) $(OBJS)

50 %.o:    %.c
51     $(COMPILE.c) -o $@ $<
52     $(POST_PROCESS_O)

54 %.o:    $(COMMON_SRC)/%.c
55     $(COMPILE.c) -o $@ $<
56     $(POST_PROCESS_O)

58 include ../Makefile.targ
```

new/usr/src/tools/cscope-fast/Makefile

1

```
*****
2169 Mon Jan 7 12:52:42 2019
new/usr/src/tools/cscope-fast/Makefile
10075 make usr/src/tools smatch clean
*****
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 #
24 # Copyright 2004 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # Copyright (c) 2018, Joyent, Inc.

29 PROG= cscope-fast

31 OBJS=  main.o dir.o crossref.o scanner.o lookup.o command.o display.o \
32         find.o edit.o exec.o help.o history.o input.o menu.o alloc.o \
33         cgrep.o compath.o invlib.o logdir.o \
34         mouse.o mygetenv.o mygetwd.o mypopen.o \
35         vpaccess.o vpfopen.o vpinit.o vpopen.o vpstat.o
36 SRCS=  $(OBJS:%.o=%.c)
37 CLEANFILES += $(OBJS) scanner.c
38 TMPDIR= /tmp

40 include ../Makefile.tools

42 # these three are because we cannot seem to redefine the size of YYLMAX,
43 # and thus yytext, in the code because yytext is defined before our code
44 # is seen. YYLMAX is supposed to be STMTMAX+PATLEN+1.
45 PATLEN=      250
46 STMTMAX=     10000
47 YYLMAX=     10251
48 CFLAGS +=    -DPATLEN=$(PATLEN) -DSTMTMAX=$(STMTMAX) -DYYLMAX=$(YYLMAX)
49 CERRWARN +=  -_gcc=-Wno-parentheses
50 CERRWARN +=  -_gcc=-Wno-implicit-function-declaration
51 CERRWARN +=  -_gcc=-Wno-unused
52 CERRWARN +=  -_gcc=-Wno-uninitialized

54 # not linted
55 SMATCH=off

57 CFLAGS +=    $(CCVERBOSE)
58 LDLIBS +=    -lcurses -ll

60 .KEEP_STATE:
```

new/usr/src/tools/cscope-fast/Makefile

2

```
62 .PARALLEL: $(OBJS)

64 all:      $(PROG)

66 # because of goto's in the scanner
67 scanner.o := CCVERBOSE=

69 $(PROG): $(OBJS)
70          $(LINK.c) -o $@ $(OBJS) $(LDLIBS)
71          $(POST_PROCESS)

73 install: all .WAIT $(ROOTONBLDMACHPROG)

75 clean:
76          $(RM) $(CLEANFILES)

78 lint: lint_SRCS

80 include ../Makefile.targ
```

new/usr/src/tools/ctf/Makefile.ctf

1

```
*****
1636 Mon Jan 7 12:52:43 2019
new/usr/src/tools/ctf/Makefile.ctf
10075 make usr/src/tools smatch clean
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # Copyright (c) 2018, Joyent, Inc.

27 include ../../../Makefile.tools

29 #
30 # A 'make install' from the tools directory needs to work, even if the rest of
31 # the tree hasn't been built. As such, we need to tell the ctf builds how
32 # to find the ctf specific headers located outside the tools subtree. We also
33 # want to tell them how to get to the common tools source files.
34 #
35 # For additional details on the ordering of includes via -I, see the comments
36 # in $(SRC)/tools/ctf/common/ctf_headers.h.
37 #

39 HDRDIRS= \
40     -_gcc=-nostdinc \
41     -I../../common \
42     -I$(SRC) \
43     -I/usr/include \
44     -I$(SRC)/uts/common \
45     -I$(NATIVE_ADJUNCT)/include

47 CPPFLAGS += $(HDRDIRS)
48 CFLAGS += $(CCVERBOSE)
49 CERRWARN += -_gcc=-Wno-parentheses

51 SMOFF += all_func_returns
```

```

*****
3277 Mon Jan 7 12:52:43 2019
new/usr/src/tools/ctf/ctfststrip/ctfststrip.c
10075 make usr/src/tools smatch clean
*****
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 2011 Jason King. All rights reserved.
14  * Copyright (c) 2018, Joyent, Inc.
15 */

17 #include <stdlib.h>
18 #include <stdio.h>
19 #include <string.h>
20 #include <unistd.h>
21 #include <sys/types.h>
22 #include <sys/stat.h>
23 #include <sys/wait.h>
24 #include <fcntl.h>
25 #include <err.h>
26 #include <spawn.h>

28 #define MCS      "/usr/bin/mcs"

30 #define ELFLEN 4
31 static const char elf_signature[] = "\177ELF";
32 static posix_spawnattr_t attr;
33 static const char *cmd[] = { MCS, "-d", "-n", ".SUNW_ctf", NULL, NULL };

35 extern char **environ;

37 static boolean_t check_file(const char *, mode_t *);
38 static boolean_t fix_file(const char *, mode_t);
39 static void usage(const char *);

41 int
42 main(int argc, const char **argv)
43 {
44     const char **p;
45     int rc = 0;
46     mode_t mode;

48     if (argc < 2)
49         usage(argv[0]);

51     rc = posix_spawnattr_init(&attr);
52     if (rc != 0) {
53         errx(EXIT_FAILURE, "Spawn attribute initialization failed: %s",
54             strerror(rc));
55     }

57     for (p = argv + 1; *p != NULL; p++) {
58         if (!check_file(*p, &mode))
59             continue;
60         if (!fix_file(*p, mode))
61             rc = 1;

```

```

62     }
64     return (rc);
65 }
_____unchanged_portion_omitted_____

115 static boolean_t
116 fix_file(const char *filename, mode_t mode)
117 {
118     pid_t pid;
119     int i, rc;
120     int stat = 0;

122     if ((mode & S_IWUSR) == 0) {
123         if (chmod(filename, mode | S_IWUSR) == -1) {
124             warn("failed to make %s writable", filename);
125             return (B_FALSE);
126         }
127     }

129     cmd[4] = filename;
130     if ((rc = posix_spawn(&pid, MCS, NULL, &attr,
131         (char *const *)cmd, environ)) != 0) {
132         warnx("could not exec mcs: %s", strerror(rc));
133         return (B_FALSE);
134     }

136     (void) waitpid(pid, &stat, 0);
137     waitpid(pid, &stat, 0);
138     if (!WIFEXITED(stat) || WEXITSTATUS(stat) != 0) {
139         warnx("Removing CTF information from %s failed", filename);
140         return (B_FALSE);
141     }

142     if ((mode & S_IWUSR) == 0) {
143         if (chmod(filename, mode) == -1) {
144             warn("could not reset permissions of %s", filename);
145             return (B_FALSE);
146         }
147     }

149     return (B_TRUE);
150 }
_____unchanged_portion_omitted_____

```


new/usr/src/tools/ctf/cvt/st_parse.c

1

```
*****
24940 Mon Jan 7 12:52:44 2019
new/usr/src/tools/ctf/cvt/st_parse.c
10075 make usr/src/tools smatch clean
*****
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) 2001, 2010, Oracle and/or its affiliates. All rights reserved.
23 */

25 /*
26 * Copyright (c) 2018, Joyent, Inc.
27 */

29 /*
30 * This file is a sewer.
31 */

33 #include <limits.h>
34 #include <stdarg.h>
35 #include <stdio.h>
36 #include <assert.h>
37 #include <strings.h>
38 #include <setjmp.h>
39 #include <ctype.h>
40 #include <uts/common/sys/ctf.h>

42 #include "ctftools.h"
43 #include "memory.h"
44 #include "list.h"

46 #define HASH(NUM)      ((int)(NUM & (BUCKETS - 1)))
47 #define BUCKETS        128

49 #define TYPEPAIRMULT    10000
50 #define MAKETYPEID(file, num)  ((file) * TYPEPAIRMULT + num)
51 #define TYPEFILE(tid)    ((tid) / TYPEPAIRMULT)
52 #define TYPENUM(tid)     ((tid) % TYPEPAIRMULT)

54 #define expected(a, b, c) _expected(a, b, c, __LINE__)

56 static int faketypernumber = 100000000;

58 static tdesc_t *hash_table[BUCKETS];
59 static tdesc_t *name_table[BUCKETS];

61 list_t *typedbitfldmems;
```

new/usr/src/tools/ctf/cvt/st_parse.c

2

```
63 static void reset(void);
64 static jmp_buf  resetbuf;

66 static char *soudef(char *cp, stabtype_t type, tdesc_t **rtdp);
67 static void enumdef(char *cp, tdesc_t **rtdp);
68 static int compute_sum(const char *w);

70 static char *number(char *cp, int *n);
71 static char *name(char *cp, char **w);
72 static char *id(char *cp, int *h);
73 static char *whitespace(char *cp);
74 static void addhash(tdesc_t *tdp, int num);
75 static int tagadd(char *w, int h, tdesc_t *tdp);
76 static char *tdefdecl(char *cp, int h, tdesc_t **rtdp);
77 static char *intrinsic(char *cp, tdesc_t **rtdp);
78 static char *arraydef(char *cp, tdesc_t **rtdp);

80 extern int debug_level;
81 int debug_parse = DEBUG_PARSE;

83 /*PRINTFLIKE3*/
84 static void
85 parse_debug(int level, char *cp, char *fmt, ...)
86 {
87     va_list ap;
88     char buf[1024];
89     char tmp[32];
90     int i;

92     if (level > debug_level || !debug_parse)
93         return;

95     if (cp != NULL) {
96         for (i = 0; i < 30; i++) {
97             if (cp[i] == '\0')
98                 break;
99             if (!isctril(cp[i]))
100                 tmp[i] = cp[i];
101         }
102         tmp[i] = '\0';
103         (void) snprintf(buf, sizeof (buf), "%s [cp='%s']\n", fmt, tmp);
104     } else {
105         strcpy(buf, fmt);
106         strcat(buf, "\n");
107     }

109     va_start(ap, fmt);
110     vdebug(level, buf, ap);
111     va_end(ap);
112 }

    unchanged_portion_omitted

573 static char *
574 tdefdecl(char *cp, int h, tdesc_t **rtdp)
575 {
576     tdesc_t *ntdp;
577     char *w;
578     int c, h2;
579     char type;

581     parse_debug(3, cp, "tdefdecl h=%d", h);

583     /* Type codes */
584     switch (type = *cp) {
585     case 'b': /* integer */
```

```

586     case 'R': /* fp */
587         cp = intrinsic(cp, rtdp);
588         break;
589     case '(': /* equiv to another type */
590         cp = id(cp, &h2);
591         ntdp = lookup(h2);

593         if (ntdp != NULL && *cp == '=') {
594             if (ntdp->t_type == FORWARD && *(cp + 1) == 'x') {
595                 /*
596                  * The 6.2 compiler, and possibly others, will
597                  * sometimes emit the same stab for a forward
598                  * declaration twice. That is, "(1,2)=xsfoo:"
599                  * will sometimes show up in two different
600                  * places. This is, of course, quite fun. We
601                  * want CTF to work in spite of the compiler,
602                  * so we'll let this one through.
603                  */
604                 char *c2 = cp + 2;
605                 char *nm;

607                 if (!strchr("sue", *c2++)) {
608                     expected("tdefdecl/x-redefine", "[sue]",
609                             c2 - 1);
610                 }

612                 c2 = name(c2, &nm);
613                 if (strcmp(nm, ntdp->t_name) != 0) {
614                     terminate("Stabs error: Attempt to "
615                               "redefine type (%d,%d) as "
616                               "something else: %s\n",
617                               TYPEFILE(h2), TYPENUM(h2),
618                               c2 - 1);
619                 }
620                 free(nm);

622                 h2 = faketypernumber++;
623                 ntdp = NULL;
624             } else {
625                 terminate("Stabs error: Attempting to "
626                           "redefine type (%d,%d)\n", TYPEFILE(h2),
627                           TYPENUM(h2));
628             }
629         }

631         if (ntdp == NULL) { /* if that type isn't defined yet */
632             if (*cp != '=') {
633                 /* record it as unresolved */
634                 parse_debug(3, NULL, "tdefdecl unres type %d",
635                             h2);
636                 *rtdp = calloc(1, sizeof (**rtdp));
637                 *rtdp = calloc(sizeof (**rtdp), 1);
638                 (*rtdp)->t_type = TYPEDEF_UNRES;
639                 (*rtdp)->t_id = h2;
640                 break;
641             } else
642                 cp++;

643             /* define a new type */
644             cp = tdefdecl(cp, h2, rtdp);
645             if ((*rtdp)->t_id && (*rtdp)->t_id != h2) {
646                 ntdp = calloc(1, sizeof (*ntdp));
647                 ntdp = calloc(sizeof (*ntdp), 1);
648                 ntdp->t_type = TYPEDEF;
649                 ntdp->t_tdesc = *rtdp;
650                 *rtdp = ntdp;

```

```

651         }
652         addhash(*rtdp, h2);

654     } else { /* that type is already defined */
655         if (ntdp->t_type != TYPEDEF || ntdp->t_name != NULL) {
656             *rtdp = ntdp;
657         } else {
658             parse_debug(3, NULL,
659                         "No duplicate typedef anon for ref");
660             *rtdp = ntdp;
661         }
662     }
663     break;
664 case '**':
665     ntdp = NULL;
666     cp = tdefdecl(cp + 1, h, &ntdp);
667     if (ntdp == NULL)
668         expected("tdefdecl/**", "id", cp);

670     if (!ntdp->t_id)
671         ntdp->t_id = faketypernumber++;

673     *rtdp = xcalloc(sizeof (**rtdp));
674     (*rtdp)->t_type = POINTER;
675     (*rtdp)->t_size = 0;
676     (*rtdp)->t_id = h;
677     (*rtdp)->t_tdesc = ntdp;
678     break;
679 case 'f':
680     cp = tdefdecl(cp + 1, h, &ntdp);
681     *rtdp = xcalloc(sizeof (**rtdp));
682     (*rtdp)->t_type = FUNCTION;
683     (*rtdp)->t_size = 0;
684     (*rtdp)->t_id = h;
685     (*rtdp)->t_fndef = xcalloc(sizeof (fndef_t));
686     /*
687     * The 6.1 compiler will sometimes generate incorrect stabs for
688     * function pointers (it'll get the return type wrong). This
689     * causes merges to fail. We therefore treat function pointers
690     * as if they all point to functions that return int. When
691     * 4432549 is fixed, the lookupname() call below should be
692     * replaced with 'ntdp'.
693     */
694     (*rtdp)->t_fndef->fn_ret = lookupname("int");
695     break;
696 case 'a':
697 case 'z':
698     cp++;
699     if (*cp++ != 'r')
700         expected("tdefdecl/[az]", "r", cp - 1);
701     *rtdp = xcalloc(sizeof (**rtdp));
702     (*rtdp)->t_type = ARRAY;
703     (*rtdp)->t_id = h;
704     cp = arraydef(cp, rtdp);
705     break;
706 case 'x':
707     c = *++cp;
708     if (c != 's' && c != 'u' && c != 'e')
709         expected("tdefdecl/x", "[sue]", cp - 1);
710     cp = name(cp + 1, &w);

712     ntdp = xcalloc(sizeof (*ntdp));
713     ntdp->t_type = FORWARD;
714     ntdp->t_name = w;
715     /*

```

```

716         * We explicitly don't set t_id here - the caller will do it.
717         * The caller may want to use a real type ID, or they may
718         * choose to make one up.
719         */
721         *rtdp = ntdp;
722         break;
724     case 'B': /* volatile */
725         cp = tdefdecl(cp + 1, h, &ntdp);
727         if (!ntdp->t_id)
728             ntdp->t_id = faketypernumber++;
730         *rtdp = xcalloc(sizeof (**rtdp));
731         (*rtdp)->t_type = VOLATILE;
732         (*rtdp)->t_size = 0;
733         (*rtdp)->t_tdesc = ntdp;
734         (*rtdp)->t_id = h;
735         break;
737     case 'k': /* const */
738         cp = tdefdecl(cp + 1, h, &ntdp);
740         if (!ntdp->t_id)
741             ntdp->t_id = faketypernumber++;
743         *rtdp = xcalloc(sizeof (**rtdp));
744         (*rtdp)->t_type = CONST;
745         (*rtdp)->t_size = 0;
746         (*rtdp)->t_tdesc = ntdp;
747         (*rtdp)->t_id = h;
748         break;
750     case 'K': /* restricted */
751         cp = tdefdecl(cp + 1, h, &ntdp);
753         if (!ntdp->t_id)
754             ntdp->t_id = faketypernumber++;
756         *rtdp = xcalloc(sizeof (**rtdp));
757         (*rtdp)->t_type = RESTRICT;
758         (*rtdp)->t_size = 0;
759         (*rtdp)->t_tdesc = ntdp;
760         (*rtdp)->t_id = h;
761         break;
763     case 'u':
764     case 's':
765         cp++;
767         *rtdp = xcalloc(sizeof (**rtdp));
768         (*rtdp)->t_name = NULL;
769         cp = soudef(cp, (type == 'u') ? UNION : STRUCT, rtdp);
770         break;
771     default:
772         expected("tdefdecl", "<type code>", cp);
773     }
774     return (cp);
775 }

```

unchanged_portion_omitted

```

*****
9649 Mon Jan 7 12:52:45 2019
new/usr/src/tools/ctf/cvt/stabs.c
10075 make usr/src/tools smatch clean
*****
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 2006 Sun Microsystems, Inc. All rights reserved.
23 * Use is subject to license terms.
24 */

26 /*
27 * Copyright (c) 2018, Joyent, Inc.
28 */

30 /*
31 * Routines used to read stabs data from a file, and to build a tdata structure
32 * based on the interesting parts of that data.
33 */

35 #include <stdio.h>
36 #include <stdlib.h>
37 #include <fcntl.h>
38 #include <unistd.h>
39 #include <assert.h>
40 #include <string.h>
41 #include <libgen.h>
42 #include <errno.h>
43 #include <sys/types.h>
44 #include <sys/param.h>

46 #include "ctftools.h"
47 #include "list.h"
48 #include "stack.h"
49 #include "memory.h"
50 #include "traverse.h"

52 const char *curhdr;

54 /*
55 * The stabs generator will sometimes reference types before they've been
56 * defined. If this is the case, a TYPEDEF_UNRES tdesc will be generated.
57 * Note that this is different from a forward declaration, in which the
58 * stab is defined, but is defined as something that doesn't exist yet.
59 * When we have read all of the stabs from the file, we can go back and
60 * fix up all of the unresolved types. We should be able to fix all of them.
61 */

```

```

62 /*ARGSUSED2*/
63 static int
64 resolve_tou_node(tdesc_t *node, tdesc_t **nodep, void *private)
65 {
66     tdesc_t *new;

68     debug(3, "Trying to resolve %s (%d)\n", tdesc_name(node), node->t_id);
69     new = lookup(node->t_id);

71     if (new == NULL) {
72         terminate("Couldn't resolve type %d\n", node->t_id);
73     }

75     debug(3, " Resolving to %d\n", new->t_id);

77     *nodep = new;

79     return (1);
80 }

unchanged_portion_omitted

174 /*
175 * Read the stabs from the stab ELF section, and turn them into a tdesc tree,
176 * assembled under an iidesc list.
177 */
178 int
179 stabs_read(tdata_t *td, Elf *elf, const char *file)
180 {
181     Elf_Scn *scn;
182     Elf_Data *data;
183     stab_t *stab;
184     stk_t *file_stack;
185     iidesc_t *iidescp;
186     iidesc_t *curfun = NULL;
187     char curpath[MAXPATHLEN];
188     char *curfile = NULL;
189     char *str;
190     char *fstr = NULL, *ofstr = NULL;
191     int stabidx, stabstridx;
192     int nstabs, rc, i;
193     int scope = 0;

195     if (!((stabidx = findelfsecidx(elf, file, ".stab.excl")) >= 0 &&
196         (stabstridx = findelfsecidx(elf, file, ".stab.exclstr")) >= 0) &&
197         !((stabidx = findelfsecidx(elf, file, ".stab")) >= 0 &&
198         (stabstridx = findelfsecidx(elf, file, ".stabstr")) >= 0)) {
199         errno = ENOENT;
200         return (-1);
201     }

203     file_stack = stack_new(free);

205     stack_push(file_stack, (void *)file);
206     curhdr = file;

208     debug(3, "Found stabs in %d, strings in %d\n", stabidx, stabstridx);

210     scn = elf_getscn(elf, stabidx);
211     data = elf_rawdata(scn, NULL);
212     nstabs = data->d_size / sizeof (stab_t);

214     parse_init(td);
215     for (i = 0; i < nstabs; i++) {
216         stab = &((stab_t *)data->d_buf)[i];

218         /* We don't want any local definitions */

```

```

219     if (stab->n_type == N_LBRAC) {
220         scope++;
221         debug(3, "stab %d: opening scope (%d)\n", i + 1, scope);
222         continue;
223     } else if (stab->n_type == N_RBRAC) {
224         scope--;
225         debug(3, "stab %d: closing scope (%d)\n", i + 1, scope);
226         continue;
227     } else if (stab->n_type == N_EINCL) {
228         /*
229          * There's a bug in the 5.2 (Taz) compilers that causes
230          * them to emit an extra N_EINCL if there's no actual
231          * text in the file being compiled. To work around this
232          * bug, we explicitly check to make sure we're not
233          * trying to pop a stack that only has the outer scope
234          * on it.
235          */
236         if (stack_level(file_stack) != 1) {
237             str = (char *)stack_pop(file_stack);
238             free(str);
239             curhdr = (char *)stack_peek(file_stack);
240         }
241     }
242
243     /* We only care about a subset of the stabs */
244     if (!(stab->n_type == N_FUN || stab->n_type == N_GSYM ||
245         stab->n_type == N_LCSYM || stab->n_type == N_LSYM ||
246         stab->n_type == N_PSYM || stab->n_type == N_ROSYM ||
247         stab->n_type == N_RSYM || stab->n_type == N_STSYM ||
248         stab->n_type == N_BINCL ||
249         stab->n_type == N_SO || stab->n_type == N_OPT))
250         continue;
251
252     if ((str = elf_strptr(elf, stabstridx,
253         (size_t)stab->n_strx)) == NULL) {
254         terminate("%s: Can't find string at %u for stab %d\n",
255             file, stab->n_strx, i);
256     }
257
258     if (stab->n_type == N_BINCL) {
259         curhdr = xstrdup(str);
260         stack_push(file_stack, (void *)curhdr);
261         continue;
262     } else if (stab->n_type == N_SO) {
263         if (str[strlen(str) - 1] != '/') {
264             strcpy(curpath, str);
265             curfile = basename(curpath);
266         }
267         continue;
268     } else if (stab->n_type == N_OPT) {
269         if (strcmp(str, "gcc2_compiled.") == 0) {
270             terminate("%s: GCC-generated stabs are "
271                 "unsupported. Use DWARF instead.\n", file);
272         }
273         continue;
274     }
275
276     if (str[strlen(str) - 1] == '\\') {
277         int offset = 1;
278         /*
279          * There's a bug in the compilers that causes them to
280          * generate \ for continuations with just -g (this is
281          * ok), and \\ for continuations with -g -O (this is
282          * broken). This bug is "fixed" in the 6.2 compilers
283          * via the elimination of continuation stabs.
284          */

```

```

285         if (str[strlen(str) - 2] == '\\')
286             offset = 2;
287         fstr = concat(fstr, str, offset);
288         continue;
289     } else
290         fstr = concat(fstr, str, 0);
291
292     debug(3, "%4d: .stabs \"%s\", %#x, %d, %hd, %d (from %s)\n", i,
293         fstr, stab->n_type, 0, stab->n_desc,
294         stab->n_value, curhdr);
295
296     if (debug_level >= 3)
297         check_hash();
298
299     /*
300      * Sometimes the compiler stutters, and emits the same stab
301      * twice. This is bad for the parser, which will attempt to
302      * redefine the type IDs indicated in the stabs. This is
303      * compiler bug 4433511.
304      */
305     if (ofstr && strcmp(fstr, ofstr) == 0) {
306         debug(3, "Stutter stab\n");
307         free(fstr);
308         fstr = NULL;
309         continue;
310     }
311
312     if (ofstr)
313         free(ofstr);
314     ofstr = fstr;
315
316     iidescp = NULL;
317
318     if ((rc = parse_stab(stab, fstr, &iidescp)) < 0) {
319         terminate("%s: Couldn't parse stab \"%s\" "
320             "(source file %s)\n", file, str, curhdr);
321     }
322
323     if (rc == 0)
324         goto parse_loop_end;
325
326     /* Make sure the scope tracking is working correctly */
327     assert(stab->n_type != N_FUN || (iidescp->ii_type != II_GFUN &&
328         iidescp->ii_type != II_SFUN) || scope == 0);
329
330     /*
331      * The only things we care about that are in local scope are
332      * the N_PSYM stabs.
333      */
334     if (scope && stab->n_type != N_PSYM) {
335         if (iidescp)
336             iidesc_free(iidescp);
337         goto parse_loop_end;
338     }
339
340     switch (iidescp->ii_type) {
341     case II_SFUN:
342         iidescp->ii_owner = xstrdup(curfile);
343         /*FALLTHROUGH*/
344     case II_GFUN:
345         curfun = iidescp;
346         fnarg_free(iidescp);
347         iidesc_add(td->td_iihash, iidescp);
348         break;
349     case II_SVAR:

```

```
350         iidescp->ii_owner = xstrdup(curfile);
351         /*FALLTHROUGH*/
352     case II_GVAR:
353     case II_TYPE:
354     case II_SOU:
355         iidesc_add(td->td_iihash, iidescp);
356         break;
357
358     case II_PSYM:
359         fnarg_add(curfun, iidescp);
360         iidesc_free(iidescp);
361         break;
362     default:
363         aborterr("invalid ii_type %d for stab type %d",
364                 iidescp->ii_type, stab->n_type);
365     }
366
367 parse_loop_end:
368     fstr = NULL;
369 }
370
371 if (ofstr)
372     free(ofstr);
373
374 resolve_nodes(td);
375 resolve_typed_bitfields();
376 parse_finish(td);
377
378 cvt_fixstabs(td);
379 cvt_fixups(td, elf_ptrsz(elf));
380
381 return (0);
382 }
383
384 _____unchanged_portion_omitted_____
```

```

*****
1984 Mon Jan 7 12:52:45 2019
new/usr/src/tools/ctf/dwarf/Makefile.com
10075 make usr/src/tools smatch clean
*****
1 #
2 # Copyright 2004 Sun Microsystems, Inc. All rights reserved.
3 # Use is subject to license terms.
4 #
5 # Copyright 2011, Richard Lowe.
6 #
7 # Copyright (c) 2018, Joyent, Inc.

9 include ../../Makefile.ctf

11 LIBRARY=      libdwarf.a
12 VERS=        .1

14 OBJECTS=dwarf_abbrev.o      \
15 dwarf_addr_finder.o        \
16 dwarf_alloc.o              \
17 dwarf_arange.o             \
18 dwarf_die_deliv.o          \
19 dwarf_elf_access.o         \
20 dwarf_error.o              \
21 dwarf_form.o               \
22 dwarf_frame.o              \
23 dwarf_frame2.o             \
24 dwarf_frame3.o             \
25 dwarf_funcs.o              \
26 dwarf_global.o             \
27 dwarf_harmless.o           \
28 dwarf_init_finish.o        \
29 dwarf_leb.o                \
30 dwarf_line.o               \
31 dwarf_line2.o              \
32 dwarf_loc.o                 \
33 dwarf_macro.o              \
34 dwarf_names.o              \
35 dwarf_original_elf_init.o   \
36 dwarf_print_lines.o        \
37 dwarf_pubtypes.o           \
38 dwarf_query.o              \
39 dwarf_ranges.o             \
40 dwarf_sort_line.o          \
41 dwarf_string.o             \
42 dwarf_stubs.o              \
43 dwarf_types.o              \
44 dwarf_util.o               \
45 dwarf_vars.o               \
46 dwarf_weeks.o              \
47 malloc_check.o            \
48 pro_alloc.o                \
49 pro_arange.o               \
50 pro_die.o                  \
51 pro_encode_nm.o            \
52 pro_error.o                \
53 pro_expr.o                 \
54 pro_finish.o               \
55 pro_forms.o                \
56 pro_frame.o                \
57 pro_funcs.o                \
58 pro_init.o                 \
59 pro_line.o                 \
60 pro_macinfo.o              \
61 pro_pubnames.o            \

```

```

62 pro_reloc.o                \
63 pro_reloc_stream.o         \
64 pro_reloc_symbolic.o      \
65 pro_section.o             \
66 pro_types.o               \
67 pro_vars.o                 \
68 pro_weeks.o                \

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

72 SRCS= $(PICS:%.o=../common/%.c)

74 FILEMODE = 0755

76 SRCDIR = ../common/

78 CPPFLAGS += -I$(SRCDIR) -DELFTARGET_ALL=1
79 CERRWARN += -_gcc=-Wno-unused
80 CERRWARN += -_gcc=-Wno-implicit-function-declaration

82 # libdwarf not clean
83 SMATCH=off

85 LDLIBS = -lelf -lc

87 .KEEP_STATE:
88 .PARALLEL:

90 all: $(DYNLIB)

92 install: all $(ROOTONBLDLIBMACH)/libdwarf.so.1 $(ROOTONBLDLIBMACH)/libdwarf.so

94 $(ROOTONBLDLIBMACH)/%: %
95 $(INS.file)

97 $(ROOTONBLDLIBMACH)/$(LIBLINKS): $(ROOTONBLDLIBMACH)/$(LIBLINKS)$(VERS)
98 $(INS.liblink)

100 FRC:

102 # We can't provide CTF information for libdwarf, as the CTF tools themselves
103 # depend upon it, and so aren't built yet.
104 $(DYNLIB) := CTFMERGE_POST= :
105 CTFCONVERT_O= :

107 include $(SRC)/lib/Makefile.targ

```

new/usr/src/tools/elfextract/elfextract.c

1

```
*****
5713 Mon Jan 7 12:52:46 2019
new/usr/src/tools/elfextract/elfextract.c
10075 make usr/src/tools smatch clean
*****
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 2007 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  */
26
27 /*
28  * Copyright (c) 2018, Joyent, Inc.
29  */
30 #pragma ident "%Z%M% %I% %E% SMI"
31 #include <unistd.h>
32 #include <stdlib.h>
33 #include <fcntl.h>
34 #include <strings.h>
35 #include <stdio.h>
36 #include <errno.h>
37 #include <sys/types.h>
38 #include <sys/inttypes.h>
39 #include <sys/elf.h>
40 #include <sys/elf_notes.h>
41 #include <sys/mman.h>
42 #include <sys/stat.h>
43 #include <sys/statvfs.h>
44
45 static char *pname;
46 static char *fname;
47 static char *image; /* pointer to the ELF file in memory */
48
49 #define ELFSEEK(offset) ((void *) (image + offset))
50
51 /*
52  * Extract the PT_LOAD bits and format them into a .s
53  */
54 static void
55 extract32(Elf32_Ehdr *eh)
56 {
57     Elf32_Phdr *phdr;
58     caddr_t allphdrs;
59     int i;
60     int c;
```

new/usr/src/tools/elfextract/elfextract.c

2

```
61     unsigned char *bytes;
62     uint_t cnt = 10;
63
64     allphdrs = NULL;
65
66     if (eh->e_type != ET_EXEC) {
67         (void) fprintf(stderr, "%s: not ET_EXEC, e_type = 0x%x\n",
68             pname, eh->e_type);
69         exit(1);
70     }
71     if (eh->e_phnum == 0 || eh->e_phoff == 0) {
72         (void) fprintf(stderr, "%s: no program headers\n", pname);
73         exit(1);
74     }
75
76     /*
77      * Get the program headers.
78      */
79     allphdrs = ELFSEEK(eh->e_phoff);
80     if (allphdrs == NULL) {
81         (void) fprintf(stderr, "%s: Failed to get %d program hdrs\n",
82             pname, eh->e_phnum);
83         exit(1);
84     }
85
86     /*
87      * Find the PT_LOAD section
88      */
89     for (i = 0; i < eh->e_phnum; i++) {
90         /*LINTED [ELF program header alignment]*/
91         phdr = (Elf32_Phdr *) (allphdrs + eh->e_phentsize * i);
92
93         if (phdr->p_type != PT_LOAD)
94             continue;
95
96         if (phdr->p_memsz == 0)
97             continue;
98
99         bytes = ELFSEEK(phdr->p_offset);
100         for (c = 0; c < phdr->p_filesz; ++c) {
101             if (c % cnt == 0)
102                 (void) printf("\n      .byte  ");
103             else
104                 (void) printf(",");
105             (void) printf("0x%x", bytes[c]);
106         }
107         for (; c < phdr->p_memsz; ++c) {
108             if (c % cnt == 0) {
109                 (void) printf("\n      .byte  ");
110                 cnt = 20;
111             } else {
112                 (void) printf(", ");
113             }
114             (void) printf("0");
115         }
116         (void) printf("\n");
117         return;
118     }
119
120     (void) fprintf(stderr, "%s: Failed finding PT_LOAD section\n", pname);
121     exit(1);
122 }
_____unchanged_portion_omitted_____
```


new/usr/src/tools/findunref/findunref.c

1

```
*****
13361 Mon Jan 7 12:52:47 2019
new/usr/src/tools/findunref/findunref.c
10075 make usr/src/tools smatch clean
*****
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 * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
22 * Use is subject to license terms.
23 */

25 /*
26  * Copyright (c) 2018, Joyent, Inc.
27 */

29 /*
30  * Finds all unreferenced files in a source tree that do not match a list of
31  * permitted pathnames.
32  */

34 #include <ctype.h>
35 #include <errno.h>
36 #include <fnmatch.h>
37 #include <ftw.h>
38 #include <stdarg.h>
39 #include <stdio.h>
40 #include <stdlib.h>
41 #include <string.h>
42 #include <time.h>
43 #include <unistd.h>
44 #include <sys/param.h>
45 #include <sys/stat.h>
46 #include <sys/types.h>

48 /*
49  * Pathname set: a simple datatype for storing pathname pattern globs and
50  * for checking whether a given pathname is matched by a pattern glob in
51  * the set.
52  */
53 typedef struct {
54     char          **paths;
55     unsigned int   npath;
56     unsigned int   maxpaths;
57 } pnset_t;
58 unchanged portion omitted

195 /*
196  * Load and return a pnset for the manifest for the Mercurial repo at 'hgroot'.
```

new/usr/src/tools/findunref/findunref.c

2

```
197 */
198 static pnset_t *
199 hg_manifest(const char *hgroot)
200 {
201     FILE          *fp = NULL;
202     char          *hgcmd = NULL;
203     char          *newline;
204     pnset_t       *pnsetp;
205     char          path[MAXPATHLEN];

207     pnsetp = calloc(1, sizeof (pnset_t));
208     pnsetp = calloc(sizeof (pnset_t), 1);
209     if (pnsetp == NULL ||
210         asprintf(&hgcmd, "hg manifest -R %s", hgroot) == -1)
211         goto fail;

212     fp = popen(hgcmd, "r");
213     if (fp == NULL)
214         goto fail;

216     while (fgets(path, sizeof (path), fp) != NULL) {
217         newline = strrchr(path, '\n');
218         if (newline != NULL)
219             *newline = '\0';

221         if (pnset_add(pnsetp, path) == 0)
222             goto fail;
223     }

225     (void) pclose(fp);
226     free(hgcmd);
227     return (pnsetp);
228 fail:
229     warn("cannot load hg manifest at %s", hgroot);
230     if (fp != NULL)
231         (void) pclose(fp);
232     free(hgcmd);
233     pnset_free(pnsetp);
234     return (NULL);
235 }

237 /*
238  * Load and return a pnset for the manifest for the Git repo at 'gitroot'.
239  */
240 static pnset_t *
241 git_manifest(const char *gitroot)
242 {
243     FILE          *fp = NULL;
244     char          *gitcmd = NULL;
245     char          *newline;
246     pnset_t       *pnsetp;
247     char          path[MAXPATHLEN];

249     pnsetp = calloc(1, sizeof (pnset_t));
250     pnsetp = calloc(sizeof (pnset_t), 1);
251     if (pnsetp == NULL ||
252         asprintf(&gitcmd, "git --git-dir=%s/.git ls-files", gitroot) == -1)
253         goto fail;

254     fp = popen(gitcmd, "r");
255     if (fp == NULL)
256         goto fail;

258     while (fgets(path, sizeof (path), fp) != NULL) {
259         newline = strrchr(path, '\n');
260         if (newline != NULL)
```

```

261             *newline = '\0';
263             if (pnset_add(pnsetp, path) == 0)
264                 goto fail;
265         }
267         (void) pclose(fp);
268         free(gitcmd);
269         return (pnsetp);
270 fail:
271     warn("cannot load git manifest at %s", gitroot);
272     if (fp != NULL)
273         (void) pclose(fp);
274     free(gitcmd);
275     pnset_free(pnsetp);
276     return (NULL);
277 }

```

unchanged_portion_omitted

```

393 /*
394  * Using 'exceptfile' and a built-in list of exceptions, build and return a
395  * pnset_t consisting of all of the pathnames globs which are allowed to be
396  * unreferenced in the source tree.
397  */
398 static pnset_t *
399 make_exset(const char *exceptfile)
400 {
401     FILE          *fp;
402     char          line[MAXPATHLEN];
403     char          *newline;
404     pnset_t      *pnsetp;
405     unsigned int  i;
407     pnsetp = calloc(1, sizeof (pnset_t));
408     pnsetp = calloc(sizeof (pnset_t), 1);
409     if (pnsetp == NULL)
410         return (NULL);
411     /*
412      * Add any exceptions from the file.
413      */
414     fp = fopen(exceptfile, "r");
415     if (fp == NULL) {
416         warn("cannot open exception file \"%s\"", exceptfile);
417         goto fail;
418     }
420     while (fgets(line, sizeof (line), fp) != NULL) {
421         newline = strrchr(line, '\n');
422         if (newline != NULL)
423             *newline = '\0';
425         for (i = 0; isspace(line[i]); i++)
426             ;
428         if (line[i] == '#' || line[i] == '\0')
429             continue;
431         if (pnset_add(pnsetp, line) == 0) {
432             (void) fclose(fp);
433             goto fail;
434         }
435     }
437     (void) fclose(fp);
438     return (pnsetp);

```

```

439 fail:
440     pnset_free(pnsetp);
441     return (NULL);
442 }

```

unchanged_portion_omitted

new/usr/src/tools/install.bin/install.bin.c

1

```
*****
5085 Mon Jan 7 12:52:48 2019
new/usr/src/tools/install.bin/install.bin.c
10075 make usr/src/tools smatch clean
*****
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) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
23 */
24
25 /*
26 * Copyright (c) 2018, Joyent, Inc.
27 */
28
29 #include <stdio.h>
30 #include <stdlib.h>
31 #include <strings.h>
32 #include <sys/param.h>
33 #include <fcntl.h>
34 #include <sys/errno.h>
35 #include <sys/types.h>
36 #include <sys/uio.h>
37 #include <unistd.h>
38 #include <sys/stat.h>
39 #include <errno.h>
40 #include <libgen.h>
41 #include "stdusers.h"
42
43
44 #define FILE_BUFF 40960
45
46 static int suppress = 0;
47
48 static void usage(void);
49 static void file_copy(char *src_file, char *dest_file);
50 static void chown_file(const char *file, const char *group, const char *owner);
51 static char *find_basename(const char *str);
52 static int creatdir(char *fn);
53
54
55 void
56 usage(void)
57 {
58     (void) fprintf(stderr,
59     "usage: install [-sd][-m mode][-g group][-u owner] "
60     "-f dir file ...\n");
61 }
62
63 unchanged_portion_omitted
```

new/usr/src/tools/install.bin/install.bin.c

2

```
101 void
102 chown_file(const char *file, const char *group, const char *owner)
103 {
104     gid_t  grp = (gid_t)-1;
105     uid_t  own = (uid_t)-1;
106
107     if (group) {
108         grp = stdfind(group, groupnames);
109         if (grp == (gid_t)-1)
110             if (grp < 0)
111                 (void) fprintf(stderr, "unknown group(%s)\n", group);
112     }
113
114     if (owner) {
115         own = stdfind(owner, usernames);
116         if (own == (uid_t)-1) {
117             if (own < 0) {
118                 (void) fprintf(stderr, "unknown owner(%s)\n", owner);
119                 exit(1);
120             }
121         }
122     }
123
124     if (chown(file, own, grp) == -1) {
125         (void) fprintf(stderr, "install:chown_file: failed "
126         "(%d): %s\n", errno, strerror(errno));
127         exit(1);
128     }
129 }
130
131 unchanged_portion_omitted
```

new/usr/src/tools/makesoftcore/Makefile

1

990 Mon Jan 7 12:52:48 2019

new/usr/src/tools/makesoftcore/Makefile

10075 make usr/src/tools smatch clean

```
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 2018 Toomas Soome <tsoome@me.com>
14 #
15 # Copyright (c) 2018, Joyent, Inc.
```

```
17 include      ../Makefile.tools
```

```
19 SOFTCORE=    $(SRC)/common/ficl/softcore
20 PROG=        makesoftcore
```

```
22 CPPFLAGS +=  -I$(SRC)/lib/libficl -I$(SRC)/common/ficl
23 OBJS=        makesoftcore.o lz4.o
24 SRCS=        makesoftcore.c lz4.c
25 LDLIBS=      -lumem
```

```
27 # not linted
28 SMOFF += all_func_returns
```

```
30 .KEEP_STATE:
```

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

```
34 install:    all .WAIT $(ROOTONBLDMACHPROG)
```

```
36 clean:
37             $(RM) $(PROG) $(OBJS)
```

```
39 $(PROG):    $(OBJS)
40             $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
41             $(POST_PROCESS)
```

```
43 %.o:       $(SOFTCORE)/%.c
44             $(COMPILE.c) -o $@ $<
```

```
46 include     ../Makefile.targ
```

new/usr/src/tools/mandoc/Makefile

1

1162 Mon Jan 7 12:52:49 2019

new/usr/src/tools/mandoc/Makefile

10075 make usr/src/tools smatch clean

```
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 2017 Nexenta Systems, Inc.
14 # Copyright 2018 Joyent, Inc.
15 #
```

```
17 CMDDIR=      $(SRC)/cmd/mandoc
18 LCDIR=       $(SRC)/lib/libc/port
```

```
20 include      $(SRC)/tools/Makefile.tools
21 include      $(CMDDIR)/Makefile.common
```

```
23 OBJS +=      fts.o \
24              reallocarray.o \
25              recallocarray.o \
26              strtonum.o
```

```
28 CPPFLAGS +=  -_gcc=-nostdinc
29 CPPFLAGS +=  -I. -include fts.h
30 CPPFLAGS +=  -I$(NATIVE_ADJUNCT)/include
31 LDFLAGS +=    -L$(NATIVE_ADJUNCT)/lib -R$(NATIVE_ADJUNCT)/lib
```

```
33 # 3rd party code
34 SMATCH=off
```

```
36 .KEEP_STATE:
```

```
38 all:         $(PROG)
```

```
40 install:    all $(ROOTONBLDMACHPROG)
```

```
42 clean:
43             $(RM) $(PROG) $(OBJS)
```

```
45 $(PROG):    $(OBJS)
46             $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
47             $(POST_PROCESS)
```

```
49 %.o:        $(CMDDIR)/%.c
50             $(COMPILE.c) -o $@ $<
```

```
52 %.o:        $(LCDIR)/gen/%.c
53             $(COMPILE.c) -o $@ $<
```

```
55 include     $(SRC)/tools/Makefile.targ
```

new/usr/src/tools/protocmp/exception_list.c

1

```
*****
2666 Mon Jan 7 12:52:51 2019
new/usr/src/tools/protocmp/exception_list.c
10075 make usr/src/tools smatch clean
*****
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 2010 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  *
26  * Copyright 2015 PALO, Richard
27  *
28  * Copyright (c) 2018, Joyent, Inc.
29  */
31 #include <stdio.h>
32 #include <stdlib.h>
33 #include <string.h>
35 #include "list.h"
36 #include "exception_list.h"
37 #include "arch.h"
39 /*
40  * This is global so that the protodir reading functions can rely on the
41  * exception list to weed out innocuous problems in the IHV gate.
42  */
43 elem_list exception_list;
45 #define FS      " \t\n"
47 static int
48 parse_exception_line(char *line, elem_list *list)
49 {
50     char    *name, *arch;
51     elem    *e;
53     if ((name = strtok(line, FS)) == NULL) {
54         /* don't complain; this is only a blank line */
55         return (0);
56     }
58     if ((arch = strtok(NULL, FS)) == NULL) {
59         arch = "all";
60     }
```

new/usr/src/tools/protocmp/exception_list.c

2

```
62     e = (elem *) malloc(sizeof (elem));
63     if (e == NULL) {
64         perror("malloc");
65         exit(1);
66     }
68     e->inode = 0;
69     e->perm = 0;
70     e->ref_cnt = 0;
71     e->flag = 0;
72     e->major = 0;
73     e->minor = 0;
74     e->link_parent = NULL;
75     e->link_sib = NULL;
76     e->symsrc = NULL;
77     e->file_type = DIR_T;
79     while ((e->arch = assign_arch(arch)) == 0) {
80         if ((arch = strtok(NULL, FS)) == NULL) {
81             free(e);
82             return (0);
83         }
84     }
86     (void) strcpy(e->name, name);
87     add_elem(list, e);
89     return (1);
90 }
unchanged_portion_omitted
```

new/usr/src/tools/protocmp/proto_list.c

1

6079 Mon Jan 7 12:52:52 2019

new/usr/src/tools/protocmp/proto_list.c

10075 make usr/src/tools smatch clean

```
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 */
```

```
26 /*
27  * Copyright (c) 2018, Joyent, Inc.
28  */
29 #pragma ident "%Z%M% %I% %E% SMI"
```

```
30 #include <stdio.h>
31 #include <fcntl.h>
32 #include <strings.h>
33 #include <ctype.h>
34 #include <stdlib.h>
```

```
36 #include "list.h"
37 #include "proto_list.h"
```

```
39 #define FS      " \t\n"
```

```
41 static void
42 error(const char *msg, int lc)
43 {
44     (void) fprintf(stderr, "warning: line %d - %s\n", lc, msg);
45 }
```

unchanged_portion_omitted