

new/usr/src/Makefile.smatch

1

```
*****
1182 Mon Jul 15 07:12:19 2019
new/usr/src/Makefile.smatch
11462 enable smatch by default
*****
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 # Copyright 2019 Joyent, Inc.
12 #
13 #
14 #
15 # smatch/sparse checks we always disable, due to too many false positives (or
16 # simply too much legacy).
17 #
18 #
19 SMATCH_ARGS = --disable=uninitialized,check_check_deref,unreachable
20 #
21 # VLAs are OK by us
22 SMATCH_ARGS += -Wno-vla
23 # don't care
24 SMATCH_ARGS += -Wno-one-bit-signed-bitfield
25 # there are lots of "extern void myfunc() { ... }" around
26 SMATCH_ARGS += -Wno-external-function-has-definition
27 # we have lots of legacy "void foo();" in headers
28 SMATCH_ARGS += -Wno-old-style-definition
29 SMATCH_ARGS += -Wno-strict-prototypes
30 SMATCH_ARGS += --fatal-checks
31 SMATCH_ARGS += --timeout=120
32 #
33 CERRWARN += $(SMATCH_ARGS:%=-_smatch=%)
34 #
35 CERRWARN += $(SMOFF:%=-_smatch=--disable=%)
36 #
37 SMATCH_ =
38 SMATCH_on =
39 SMATCH_off = -_smatch=off
40 #
41 CERRWARN += $(SMATCH_$(SMATCH))
```

new/usr/src/tools/env/illumos.sh

1

```
*****
10845 Mon Jul 15 07:12:20 2019
new/usr/src/tools/env/illumos.sh
11462 enable smatch by default
*****
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 (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
22 # Copyright 2015 Nexenta Systems, Inc. All rights reserved.
23 # Copyright 2012 Joshua M. Clulow <josh@sysmgr.org>
24 # Copyright 2015, OmniTI Computer Consulting, Inc. All rights reserved.
25 # Copyright 2018 OmniOS Community Edition (OmniOSce) Association.
26 # Copyright 2019, Joyent, Inc.
27 #
28 # - This file is sourced by "bldenv.sh" and "nightly.sh" and should not
29 #   be executed directly.
30 # - This script is only interpreted by ksh93 and explicitly allows the
31 #   use of ksh93 language extensions.
32 #
33 # -----
34 # Parameters you are likely to want to change
35 # -----
36 #
37 #
38 #     DEBUG build only (-D, -F)
39 #     do not bringover from the parent (-n)
40 #     runs 'make check' (-C)
41 #     checks for new interfaces in libraries (-A)
42 #     sends mail on completion (-m and the MAILTO variable)
43 #     creates packages for PIT/RE (-p)
44 #     checks for changes in ELF runpaths (-r)
45 #     build and use this workspace's tools in $SRC/tools (-t)
46 export NIGHTLY_OPTIONS='-FnCDAmprt'
47 #
48 # Some scripts optionally send mail messages to MAILTO.
49 #export MAILTO=
50 #
51 # CODEMGR_WS - where is your workspace at
52 export CODEMGR_WS="'git rev-parse --show-toplevel'"
53 #
54 # Compilers may be specified using the following variables:
55 # PRIMARY_CC      - primary C compiler
56 # PRIMARY_CCC     - primary C++ compiler
57 #
58 # SHADOW_CCS      - list of shadow C compilers
59 # SHADOW_CCCS     - list of shadow C++ compilers
60 #
61 # Each entry has the form <name>,<path to binary>,<style> where name is a
```

new/usr/src/tools/env/illumos.sh

2

```
62 # free-form name (possibly used in the makefiles to guard options), path is
63 # the path to the executable. style is the 'style' of command line taken by
64 # the compiler, currently either gnu (or gcc) or sun (or cc), which is also
65 # used by Makefiles to guard options.
66 #
67 # __SUNC and __GNUC must still be set to reflect the style of the primary
68 # compiler (and to influence the default primary, otherwise)
69 #
70 # for example:
71 # export PRIMARY_CC=gcc4,/opt/gcc/4.4.4/bin/gcc,gnu
72 # export PRIMARY_CCC=gcc4,/opt/gcc/4.4.4/bin/g++,gnu
73 # export SHADOW_CCS=studio12,/opt/SUNWspro/bin/cc,sun
74 # export SHADOW_CCCS=studio12,/opt/SUNWspro/bin/CC,sun
75 #
76 # There can be several space-separated entries in SHADOW_* to run multiple
77 # shadow compilers.
78 #
79 # To disable shadow compilation, unset SHADOW_* or set them to the empty string.
80 #
81 export GNUC_ROOT=/usr/gcc/7
82 export PRIMARY_CCS=gcc7,$GNUC_ROOT/bin/gcc,gnu
83 export PRIMARY_CCCS=gcc7,$GNUC_ROOT/bin/g++,gnu
84 export SHADOW_CCS=gcc4,/opt/gcc/4.4.4/bin/gcc,gnu
85 export SHADOW_CCCS=gcc4,/opt/gcc/4.4.4/bin/g++,gnu
86 #
87 # comment to disable smatch
88 export ENABLE_SMATCH=1
89 # uncomment to enable smatch
90 #export ENABLE_SMATCH=1
91 #
92 # Comment this out to disable support for SMB printing, i.e. if you
93 # don't want to bother providing the CUPS headers this needs.
94 export ENABLE_SMB_PRINTING=
95 #
96 # If your distro uses certain versions of Perl, make sure either Makefile.master
97 # contains your new defaults OR your .env file sets them.
98 # These are how you would override for building on OmniOS r151028, for example.
99 #export PERL_VERSION=5.28
100 #export PERL_ARCH=i86pc-solaris-thread-multi-64int
101 #export PERL_PKGVERS=
102 #
103 # If your distro uses certain versions of Python, make sure either
104 # Makefile.master contains your new defaults OR your .env file sets them.
105 #export PYTHON_VERSION=2.7
106 #export PYTHON_PKGVERS=-27
107 #export PYTHON_SUFFIX=
108 #export PYTHON3_VERSION=3.5
109 #export PYTHON3_PKGVERS=-35
110 #export PYTHON3_SUFFIX=m
111 #
112 # To disable building with either Python2 or Python 3 (or both), uncomment
113 # these lines:
114 #export BUILD_PY2='#'
115 #export BUILD_PY3='#'
116 #
117 # Set console color scheme either by build type:
118 #
119 #export RELEASE_CONSOLE_COLOR="-DDEFAULT_ANSI_FOREGROUND=ANSI_COLOR_BLACK \
120 -DDEFAULT_ANSI_BACKGROUND=ANSI_COLOR_WHITE"
121 #
122 #export DEBUG_CONSOLE_COLOR="-DDEFAULT_ANSI_FOREGROUND=ANSI_COLOR_RED \
123 -DDEFAULT_ANSI_BACKGROUND=ANSI_COLOR_WHITE"
124 #
125 # or just one for any build type:
126 #export DEFAULT_CONSOLE_COLOR="-DDEFAULT_ANSI_FOREGROUND=ANSI_COLOR_BLACK \
```

```
126 # -DDEFAULT_ANSI_BACKGROUND=ANSI_COLOR_WHITE"
128 # Set if your distribution has different package versioning
129 #export PKGVERS_BRANCH=2018.0.0.17900
131 # Skip Java 8 builds on distributions that don't support it
132 #export BLD_JAVA_8=
134 # POST_NIGHTLY can be any command to be run at the end of nightly. See
135 # nightly(1) for interactions between environment variables and this command.
136 #POST_NIGHTLY=
138 # -----
139 # You are less likely to need to modify parameters below.
140 # -----
142 # Maximum number of dmake jobs. The recommended number is 2 + NCPUS,
143 # where NCPUS is the number of logical CPUs on your build system.
144 function maxjobs
145 {
146     nameref maxjobs=$1
147     integer ncpu
148     integer -r min_mem_per_job=512 # minimum amount of memory for a job
150     ncpu=$(builtin getconf ; getconf 'NPROCESSORS_ONLN')
151     (( maxjobs=ncpu + 2 ))
153     # Throttle number of parallel jobs launched by dmake to a value which
154     # gurantees that all jobs have enough memory. This was added to avoid
155     # excessive paging/swapping in cases of virtual machine installations
156     # which have lots of CPUs but not enough memory assigned to handle
157     # that many parallel jobs
158     if [[ $(/usr/sbin/prtconf 2>/dev/null) == ~(E)Memory\ size:\ ([:digit
159         integer max_jobs_per_memory # parallel jobs which fit into physi
160         integer physical_memory # physical memory installed
162         # The array ".sh.match" contains the contents of capturing
163         # brackets in the last regex, .sh.match[1] will contain
164         # the value matched by ([:digit:])+), i.e. the amount of
165         # memory installed
166         physical_memory="10${.sh.match[1]}"
168         ((
169             max_jobs_per_memory=round(physical_memory/min_mem_per_jo
170             maxjobs=fmax(2, fmin(maxjobs, max_jobs_per_memory))
171         ))
172     fi
174     return 0
175 }
_____unchanged_portion_omitted_
```

```

*****
5200 Mon Jul 15 07:12:20 2019
new/usr/src/tools/smacth/Makefile
11462 enable smacth by default
*****
1 #
2 # This file and its contents are supplied under the terms of the
3 # Common Development and Distribution License ("CDDL"), version 1.0.
4 # You may only use this file in accordance with the terms of version
5 # 1.0 of the CDDL.
6 #
7 # A full copy of the text of the CDDL should have accompanied this
8 # source. A copy of the CDDL is also available via the Internet at
9 # http://www.illumos.org/license/CDDL.
10 #
11 # Copyright (c) 2019, Joyent, Inc.
12 #
13 #
14 #
15 # The src/ sub-directory is un-modified copy of
16 # https://github.com/illumos/smacth/tree/0.5.1-il-3
17 # https://github.com/illumos/smacth/tree/0.5.1-il-1
18 # This Makefile installs just enough for us to be able to run smacth
19 # locally.
20 #
21 #
22 PROG = smacth
23 SPARSE_VERSION = 0.5.1-il-3
24 SPARSE_VERSION = 0.5.1-il-1
25
26 include ../Makefile.tools
27
28 # We have to build smacth before we can use cw
29 i386_CC = $(GNUCC_ROOT)/bin/gcc
30 sparc_CC = $(GNUCC_ROOT)/bin/gcc
31
32 CFLAGS = -O -D_sun -Wall -Wno-unknown-pragmas -std=gnu99 -nodefaultlibs
33
34 SMATCHDATADIR = $(ROOTONBLDSHARE)/smacth
35
36 CFLAGS += -DSMATCHDATADIR="'$(SMATCHDATADIR)'"
37 CFLAGS += -DGCC_BASE="'/no/such/dir'"
38 CFLAGS += -DMULTIARCH_TRIPLET=NULL
39
40 LDLIBS += -lsqllite3 -lcrypto -lm -lgcc -lc
41 LDFLAGS = $(MAPFILE.NES:%=-Wl,-M%)
42 LDFLAGS += -L$(NATIVE_ADJUNCT)/lib -R$(NATIVE_ADJUNCT)/lib
43
44 CPPFLAGS += -nostdinc
45 CPPFLAGS += -Isrc/
46 CPPFLAGS += -I$(NATIVE_ADJUNCT)/include
47
48 # no install.bin
49 INS.file = $(RM) $@; $(CP) $< $(@D); $(CHMOD) $(FILEMODE) $@
50 INS.dir = mkdir -p $@; $(CHMOD) $(DIRMODE) $@
51
52 SMATCH_CHECK_OBJS=sh=ls src/check_*.c | sed -e 's+\.c+.o+s+src/++;'
53
54 OBJS = smacth.o $(SMATCH_CHECK_OBJS)
55
56 OBJS += smacth_flow.o smacth_conditions.o smacth_slist.o smacth_states.o \
57 smacth_helper.o smacth_type.o smacth_hooks.o smacth_function_hooks.o \
58 smacth_modification_hooks.o smacth_extra.o smacth_estate.o smacth_math.o
59 smacth_sval.o smacth_ranges.o smacth_implied.o smacth_ignore.o smacth_pr
60 smacth_var_sym.o smacth_tracker.o smacth_files.o smacth_expression_stack

```

```

60 smacth_equiv.o smacth_buf_size.o smacth_strlen.o smacth_capped.o smacth_
61 smacth_expressions.o smacth_returns.o smacth_parse_call_math.o \
62 smacth_param_limit.o smacth_param_filter.o \
63 smacth_param_set.o smacth_comparison.o smacth_param_compare_limit.o smat
64 smacth_function_ptrs.o smacth_annotate.o smacth_string_list.o \
65 smacth_param_cleared.o smacth_start_states.o \
66 smacth_recurse.o smacth_data_source.o smacth_type_val.o \
67 smacth_common_functions.o smacth_struct_assignment.o \
68 smacth_unknown_value.o smacth_stored_conditions.o avl.o \
69 smacth_function_info.o smacth_links.o smacth_auto_copy.o \
70 smacth_type_links.o smacth_untracked_param.o smacth_impossible.o \
71 smacth_strings.o smacth_param_used.o smacth_container_of.o smacth_addr
72 smacth_buf_comparison.o smacth_real_absolute.o smacth_scope.o \
73 smacth_imaginary_absolute.o smacth_parameter_names.o \
74 smacth_return_to_param.o smacth_passes_array_size.o \
75 smacth_constraints.o smacth_constraints_required.o \
76 smacth_fn_arg_link.o smacth_about_fn_ptr_arg.o smacth_mtag.o \
77 smacth_mtag_map.o smacth_mtag_data.o \
78 smacth_param_to_mtag_data.o smacth_mem_tracker.o smacth_array_values.o \
79 smacth_nul_terminator.o smacth_assigned_expr.o smacth_kernel_user_data.o
80 smacth_statement_count.o
81
82 OBJS += target.o parse.o tokenize.o pre-process.o symbol.o lib.o scope.o \
83 expression.o show-parse.o evaluate.o expand.o inline.o linearize.o \
84 char.o sort.o allocate.o compat-linux.o ptrlist.o \
85 builtin.o \
86 stats.o \
87 flow.o cse.o simplify.o memops.o liveness.o storage.o unssa.o \
88 dissect.o \
89 macro_table.o token_store.o hashtable.o
90
91 SMATCH_DATA = \
92 illumos_kernel.no_return_funcs \
93 illumos_kernel.skipped_functions \
94 illumos_user.no_return_funcs \
95 illumos_user.skipped_functions
96
97 SMATCH_DB_DATA = \
98 return_states.schema \
99 call_implies.schema \
100 type_value.schema \
101 param_map.schema \
102 function_type_size.schema \
103 parameter_name.schema \
104 fn_ptr_data_link.schema \
105 constraints.schema \
106 mtag_about.schema \
107 type_info.schema \
108 function_type_info.schema \
109 caller_info.schema \
110 function_type_value.schema \
111 return_implies.schema \
112 type_size.schema \
113 constraints_required.schema \
114 fn_data_link.schema \
115 mtag_alias.schema \
116 common_caller_info.schema \
117 data_info.schema \
118 function_type.schema \
119 db.schema \
120 mtag_data.schema \
121 function_ptr.schema \
122 sink_info.schema \
123 local_values.schema \
124 mtag_map.schema

```

new/usr/src/tools/smatch/Makefile

3

```
126 ROOTONBLDDATAFILES = $(SMATCH_DATA:%=$(SMATCHDATADIR)/smatch_data/%)
127 ROOTONBLDDATAFILES += $(SMATCH_DB_DATA:%=$(SMATCHDATADIR)/smatch_data/db/%)

129 BUILT_HEADERS = src/version.h src/check_list_local.h

131 .KEEP_STATE:

133 .PARALLEL: $(OBJS)

135 all: $(PROG)

137 install: all .WAIT $(ROOTONBLDMACHPROG) $(ROOTONBLDDATAFILES)

139 clean:
140     rm -f $(OBJS) $(BUILT_HEADERS)

142 $(ROOTONBLDDATAFILES): $(SMATCHDATADIR)/smatch_data/db

144 $(SMATCHDATADIR)/smatch_data/%: src/smatch_data/%
145     $(INS.file)

147 $(SMATCHDATADIR)/smatch_data/db:
148     $(INS.dir)

150 $(SMATCHDATADIR)/smatch_data:
151     $(INS.dir)

153 $(PROG): $(OBJS)
154     $(LINK.c) $(OBJS) -o $@ $(LDLIBS)
155     $(POST_PROCESS)

157 %.o: src/%.c $(BUILT_HEADERS)
158     $(COMPILE.c) -o $@ $<

160 %.o: src/cwchash/%.c
161     $(COMPILE.c) -o $@ $<

163 src/check_list_local.h:
164     touch src/check_list_local.h

166 src/version.h:
167     echo '#define SPARSE_VERSION "$(SPARSE_VERSION)">' > src/version.h

169 include ../Makefile.targ
```

```

*****
8701 Mon Jul 15 07:12:20 2019
new/usr/src/tools/smatch/src/smatch.c
11462 enable smatch by default
*****
1 /*
2  * Copyright (C) 2006 Dan Carpenter.
3  *
4  * This program is free software; you can redistribute it and/or
5  * modify it under the terms of the GNU General Public License
6  * as published by the Free Software Foundation; either version 2
7  * of the License, or (at your option) any later version.
8  *
9  * This program is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with this program; if not, see http://www.gnu.org/copyleft/gpl.txt
16 *
17 * Copyright 2019 Joyent, Inc.
18 */

20 #include <stdio.h>
21 #include <unistd.h>
22 #include <libgen.h>
23 #include "smatch.h"
24 #include "check_list.h"

26 char *option_debug_check = (char *)"";
27 char *option_project_str = (char *)"smatch_generic";
28 static char *option_db_file = (char *)"smatch_db.sqlite";
29 enum project_type option_project = PROJ_NONE;
30 char *bin_dir;
31 char *data_dir;
32 int option_no_data = 0;
33 int option_spammy = 0;
34 int option_info = 0;
35 int option_full_path = 0;
36 int option_param_mapper = 0;
37 int option_call_tree = 0;
38 int option_no_db = 0;
39 int option_enable = 0;
40 int option_disable = 0;
41 int option_debug_related;
42 int option_file_output;
43 int option_time;
44 int option_mem;
45 char *option_datadir_str;
46 int option_fatal_checks;
47 int option_succeed;
48 int option_timeout = 60;

50 FILE *sm_outfd;
51 FILE *sql_outfd;
52 FILE *caller_info_fd;

54 int sm_nr_errors;
55 int sm_nr_checks;

57 bool __silence_warnings_for_stmt;

59 const char *progname;

61 typedef void (*reg_func) (int id);

```

```

62 #define CK(_x) {.name = #_x, .func = &_amp;_x, .enabled = 0},
63 static struct reg_func_info {
64     const char *name;
65     reg_func func;
66     int enabled;
67 } reg_funcs[] = {
    unchanged_portion_omitted

178 #define OPTION(_x) do { \
179     if (match_option((*argvp)[i], #_x)) { \
176     if (match_option((*argvp)[1], #_x)) { \
180         option_##_x = 1; \
181     } \
182 } while (0)

184 void parse_args(int *argc, char ***argv)
185 {
186     int i;

188     for (i = 1; i < *argc; i++) {
189         if (!strcmp((*argv)[i], "--help"))
190             help();

192         if (!strcmp((*argv)[i], "--show-checks"))
193             show_checks();

195         if (!strncmp((*argv)[i], "--project=", 10))
196             option_project_str = (*argv)[i] + 10;

198         if (!strcmp((*argv)[i], "-p=", 3))
199             option_project_str = (*argv)[i] + 3;

201         if (!strncmp((*argv)[i], "--db-file=", 10))
202             option_db_file = (*argv)[i] + 10;

204         if (!strcmp((*argv)[i], "--data=", 7))
205             option_datadir_str = (*argv)[i] + 7;

207         if (!strcmp((*argv)[i], "--debug=", 8))
208             option_debug_check = (*argv)[i] + 8;

210         if (strcmp((*argv)[i], "--trace=", 8) == 0)
211             trace_variable = (*argv)[i] + 8;

213         if (strcmp((*argv)[i], "--enable=", 9) == 0) {
214             enable_disable_checks((*argv)[i] + 9, 1);
215             option_enable = 1;
216         }

218         if (strcmp((*argv)[i], "--disable=", 10) == 0) {
219             enable_disable_checks((*argv)[i] + 10, 0);
220             option_enable = 1;
221             option_disable = 1;
222         }

224         if (!strcmp((*argv)[i], "--timeout=", 10)) {
225             if (sscanf((*argv)[i] + 10, "%d",
226                 &option_timeout) != 1)
227                 sm_fatal("invalid option %s", (*argv)[i]);
228         }

230     OPTION(fatal_checks);
231     OPTION(spammy);
232     OPTION(info);
233     OPTION(debug);
234     OPTION(debug_implied);

```

```
235         OPTION(debug_related);
236         OPTION(assume_loops);
237         OPTION(no_data);
238         OPTION(two_passes);
239         OPTION(full_path);
240         OPTION(param_mapper);
241         OPTION(call_tree);
242         OPTION(file_output);
243         OPTION(time);
244         OPTION(mem);
245         OPTION(no_db);
246         OPTION(succeed);
247     }
249     if (strcmp(option_project_str, "smatch_generic") != 0)
250         option_project = PROJ_UNKNOWN;
252     if (strcmp(option_project_str, "kernel") == 0)
253         option_project = PROJ_KERNEL;
254     else if (strcmp(option_project_str, "wine") == 0)
255         option_project = PROJ_WINE;
256     else if (strcmp(option_project_str, "illumos_kernel") == 0)
257         option_project = PROJ_ILLUMOS_KERNEL;
258     else if (strcmp(option_project_str, "illumos_user") == 0)
259         option_project = PROJ_ILLUMOS_USER;
260 }
_____unchanged_portion_omitted_
```

new/usr/src/tools/smacth/src/smacth.h

1

```
*****
46054 Mon Jul 15 07:12:21 2019
new/usr/src/tools/smacth/src/smacth.h
11462 enable smacth by default
*****
1 /*
2  * Copyright (C) 2006 Dan Carpenter.
3  *
4  * This program is free software; you can redistribute it and/or
5  * modify it under the terms of the GNU General Public License
6  * as published by the Free Software Foundation; either version 2
7  * of the License, or (at your option) any later version.
8  *
9  * This program is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with this program; if not, see http://www.gnu.org/copyleft/gpl.txt
16 *
17 * Copyright 2019 Joyent, Inc.
18 */

20 #ifndef SMATCH_H_
21 #define SMATCH_H_

23 #include <stdio.h>
24 #include <string.h>
25 #include <limits.h>
26 #include <sys/time.h>
27 #include <sqlite3.h>
28 #include "lib.h"
29 #include "allocate.h"
30 #include "scope.h"
31 #include "parse.h"
32 #include "expression.h"
33 #include "avl.h"

35 typedef struct {
36     struct symbol *type;
37     union {
38         long long value;
39         unsigned long long uvalue;
40     };
41 } sval_t;
    unchanged portion omitted

189 typedef void (modification_hook)(struct sm_state *sm, struct expression *mod_exp
190 void add_modification_hook(int owner, modification_hook *call_back);
191 void add_modification_hook_late(int owner, modification_hook *call_back);
192 struct smatch_state *get_modification_state(struct expression *expr);

194 int outside_of_function(void);
195 const char *get_filename(void);
196 const char *get_base_file(void);
197 char *get_function(void);
198 int get_lineno(void);
199 extern int final_pass;
200 extern struct symbol *cur_func_sym;
201 extern int option_debug;
202 extern int local_debug;
203 extern int option_info;
204 extern int option_spammy;
205 extern int option_timeout;
206 extern char *trace_variable;
```

new/usr/src/tools/smacth/src/smacth.h

2

```
207 extern struct stree *global_states;
208 int is_skipped_function(void);
209 int is_silenced_function(void);

211 /* smacth_impossible.c */
212 int is_impossible_path(void);
213 void set_path_impossible(void);

215 extern FILE *sm_outfd;
216 extern FILE *sql_outfd;
217 extern FILE *caller_info_fd;
218 extern int sm_nr_checks;
219 extern int sm_nr_errors;
220 extern const char *progname;

222 /*
223  * How to use these routines:
224  *
225  * sm_fatal(): an internal error of some kind that should immediately exit
226  * sm_ierror(): an internal error
227  * sm_perror(): an internal error from parsing input source
228  * sm_error(): an error from input source
229  * sm_warning(): a warning from input source
230  * sm_info(): info message (from option_info)
231  * sm_debug(): debug message
232  * sm_msg(): other message (please avoid using this)
233  */

235 #define sm_printf(msg...) do { if (final_pass || option_debug || local_debug) fp
237 static inline void sm_prefix(void)
238 {
239     sm_printf("%s: %s:%d %s() ", progname, get_filename(), get_lineno(), get
240 }
    unchanged portion omitted
```



```

*****
30904 Mon Jul 15 07:12:21 2019
new/usr/src/tools/smatch/src/smatch_implied.c
11462 enable smatch by default
*****
1 /*
2  * Copyright (C) 2008 Dan Carpenter.
3  *
4  * This program is free software; you can redistribute it and/or
5  * modify it under the terms of the GNU General Public License
6  * as published by the Free Software Foundation; either version 2
7  * of the License, or (at your option) any later version.
8  *
9  * This program is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13 *
14 * You should have received a copy of the GNU General Public License
15 * along with this program; if not, see http://www.gnu.org/copyleft/gpl.txt
16 *
17 * Copyright 2019 Joyent, Inc.
18 */

20 /*
21 * Imagine we have this code:
22 * foo = 1;
23 * if (bar)
24 *     foo = 99;
25 * else
26 *     frob();
27 *
28 * // <-- point #1
29 * if (foo == 99) // <-- point #2
30 *     bar->baz; // <-- point #3
31 *
32 * At point #3 bar is non null and can be dereferenced.
33 *
34 * It's smatch_implied.c which sets bar to non null at point #2.
35 *
36 * At point #1 merge_slist() stores the list of states from both
37 * the true and false paths. On the true path foo == 99 and on
38 * the false path foo == 1. merge_slist() sets their pool
39 * list to show the other states which were there when foo == 99.
40 *
41 * When it comes to the if (foo == 99) the smatch implied hook
42 * looks for all the pools where foo was not 99. It makes a list
43 * of those.
44 *
45 * Then for bar (and all the other states) it says, ok bar is a
46 * merged state that came from these previous states. We'll
47 * chop out all the states where it came from a pool where
48 * foo != 99 and merge it all back together.
49 *
50 * That is the implied state of bar.
51 *
52 * merge_slist() sets up ->pool. An sm_state only has one ->pool and
53 * that is the pool where it was first set. The my pool gets set when
54 * code paths merge. States that have been set since the last merge do
55 * not have a ->pool.
56 * merge_sm_state() sets ->left and ->right. (These are the states which were
57 * merged to form the current state.)
58 * a pool: a pool is an slist that has been merged with another slist.
59 */

61 #include <sys/time.h>

```

```

62 #include <time.h>
63 #include "smatch.h"
64 #include "smatch_slist.h"
65 #include "smatch_extra.h"

67 char *implied_debug_msg;
68 #define DIMPLIED(msg...) do { if (option_debug_implied || option_debug) printf(m

70 int option_debug_implied = 0;

72 /*
73  * tmp_range_list():
74  * It messes things up to free range list allocations. This helper fuction
75  * lets us reuse memory instead of doing new allocations.
76  */
77 static struct range_list *tmp_range_list(struct symbol *type, long long num)
78 {
79     static struct range_list *my_list = NULL;
80     static struct data_range *my_range;

82     __free_ptr_list((struct ptr_list **)&my_list);
83     my_range = alloc_range(ll_to_sval(num), ll_to_sval(num));
84     add_ptr_list(&my_list, my_range);
85     return my_list;
86 }
    unchanged portion omitted

395 static int taking_too_long(void)
396 {
397     static void *printed;

399     if (out_of_memory())
400         return 1;

402     if (time_parsing_function() < option_timeout)
403         if (time_parsing_function() < 60)
404             return 0;

405     if (!__inline_fn && printed != cur_func_sym) {
406         if (!is_skipped_function())
407             sm_perror("turning off implications after 60 seconds");
408         printed = cur_func_sym;
409     }
410     return 1;
411 }
    unchanged portion omitted

555 static void separate_and_filter(struct sm_state *sm, int comparison, struct rang
556     struct stree *pre_stree,
557     struct stree **true_states,
558     struct stree **false_states,
559     int *mixed)
560 {
561     struct state_list *true_stack = NULL;
562     struct state_list *false_stack = NULL;
563     struct timeval time_before;
564     struct timeval time_after;
565     int sec;

567     gettimeofday(&time_before, NULL);

569     if (!is_merged(sm)) {
570         DIMPLIED("%d '%s' is not merged.\n", get_lineno(), sm->name);
571         return;
572     }

```

```
574     if (option_debug_implied || option_debug) {
575         sm_msg("checking implications: (%s %s %s)",
576             sm->name, show_special(comparison), show_rl(rl));
577     }
579     separate_pools(sm, comparison, rl, &true_stack, &>false_stack, NULL, mixe
581     DIMPLIED("filtering true stack.\n");
582     *true_states = filter_stack(sm, pre_stree, false_stack, true_stack);
583     DIMPLIED("filtering false stack.\n");
584     *false_states = filter_stack(sm, pre_stree, true_stack, false_stack);
585     free_slist(&true_stack);
586     free_slist(&>false_stack);
587     if (option_debug_implied || option_debug) {
588         printf("These are the implied states for the true path: (%s %s %
589             sm->name, show_special(comparison), show_rl(rl));
590         __print_stree(*true_states);
591         printf("These are the implied states for the false path: (%s %s
592             sm->name, show_special(comparison), show_rl(rl));
593         __print_stree(*false_states);
594     }
596     gettimeofday(&time_after, NULL);
597     sec = time_after.tv_sec - time_before.tv_sec;
598     if (sec > option_timeout) {
599         if (sec > 20) {
600             sm->nr_children = 4000;
601             sm_perror("Function too hairy. Ignoring implications after %d s
602     }
```

unchanged_portion_omitted