Kernel Dev Tools
Coccinelle and Sparse

Kernel dev-tools | Quick Summary List			
Tool Name	***Tool Type [Static / Dynamic]***	***Comments***	***Documentation*** ***>= 4.9:*** *Documentation/dev-tools/<foo>.rst]* ***< 4.9 :*** *Documentation/<foo>.txt*
coccinelle	Static	2.6.36-rc1 ; Semantic parser; 'C' only. Very extensible (any # of semantic patches can be developed)	*Documentation/[dev-tools]/coccinelle.txt|[.rst]*
sparse		2.6.12 ; 'C' source static analyser	*Documentation/[dev-tools]/sparse.txt|[.rst]*
kmemleak	Dynamic	2.6.31 ; Kernel memory leakage detector; only slab layer	*Documentation/[dev-tools]/kmemleak.txt|[.rst]*
kmemcheck		2.6.31 ; Reports uninitialized memory accesses (x86[_64] only)	*Documentation/[dev-tools]/kmemcheck.txt|[.rst]*
kasan		4.0 ; Kernel address sanitizer	*Documentation/[dev-tools]/kasan.txt|[.rst]*
ubsan		Undefined behavior sanitizer	*Documentation/[dev-tools]/ubsan.txt|[.rst]*
gdb-kernel-debugging		Using GDB to perform source-level debug on Linux kernel code and kernel modules	*Documentation/[dev-tools]/gdb-kernel-debugging.txt|[.rst]*
kcov		Kernel code coverage	*Documentation/[dev-tools]/kcov.txt|[.rst]*
gcov		2.6.32 ; Kernel profiler; for code-coverage testing/analysis. Requires tooling the Makefile(s) before build (see Documentation)	*Documentation/[dev-tools]/gcov.txt|[.rst]*
Lockdep		2.6.18 ; Kernel 'runtime locking correctness validator'; catches locking violations and reports them; provides rolling mathematical proof that all's ok (or not)	*https://www.kernel.org/doc/html/latest/locking/lockdep-design.html#runtime-locking-correctness-validator*

KCSAN		5.8; x86_64, Aarch64 (5.17). Kernel Concurrency SANitizer - a powerful kernel framework for helping catch data races within the Linux kernel (and modules)	*https://www.kernel.org/doc/html/latest/dev-tools/kcsan.html#the-kernel-concurrency-sanitizer-kcsan* *[https://www.linuxfoundation.org/webinars/the-kernel-concurrency-sanitizer?hsLang=en]*

Also see : Core Infrastructure Initiative best-practices badge, D Wheeler, LWN, June 2016

Coccinelle

Documentation

Offline: within the kernel source tree, see:
Documentation/coccinelle.txt *[here for older kernels, <= 4.8]*
Documentation/dev-tools/coccinelle.rst *[here for newer kernels >= 4.8-rc1 (4.9 onwards)]*

Online *kernel documentation*
https://www.kernel.org/doc/Documentation/dev-tools/coccinelle.rst

[Newer HTML documentation]
https://www.kernel.org/doc/html/latest/dev-tools/index.html
Within this:
https://www.kernel.org/doc/html/latest/dev-tools/coccinelle.html

[Alternatively, use the LXR web tool to browse the kernel source tree including Documentation:
http://elixir.free-electrons.com/linux/latest/source/Documentation]

What is Coccinelle

“Coccinelle” is a French word, roughly translating to “ladybug” in English.

Source
“Coccinelle is a program matching and transformation engine which provides the language SmPL (Semantic Patch Language) for specifying desired matches and transformations in C code.

Coccinelle was initially targeted towards performing *collateral evolutions* in Linux. Such

evolutions comprise the changes that are needed in client code in response to evolutions in library APIs, and may include modifications such as renaming a function, adding a function argument whose value is somehow context-dependent, and reorganizing a data structure.

Beyond collateral evolutions, Coccinelle is successfully used (by us and others) for finding and fixing bugs in systems code."

More information on Coccinelle's Semantic patches here. Has a couple of interesting examples, and many more here: http://coccinelle.lip6.fr/rules/ and here: http://coccinellery.org/

<<
- 'C' source only
- coccinelle "understands" 'C' (it's more than a pattern matcher/editor like grep/sed)

>>

Using Coccinelle for Semantic Analysis of the Linux kernel codebase

- Download / install coccinelle (the package is readily available for most modern Linux distros)

Source [below]

A Coccinelle-specific target is defined in the top level Makefile. This target is named coccicheck and calls the coccicheck front-end in the scripts directory.

Four basic modes are defined: patch, report, context, and org. The mode to use is specified by setting the MODE variable with MODE=<mode>.

• **patch** proposes a fix, when possible.
• **report** generates a list in the following format: file:line:column-column: message
• **context** highlights lines of interest and their context in a diff-like style.Lines of interest are indicated with -.
• **org** generates a report in the Org mode format of Emacs.

Note that not all semantic patches implement all modes. For easy use of Coccinelle, the default mode is "report".

Two other modes provide some common combinations of these modes.
- **chain** tries the previous modes in the order above until one succeeds.
- **rep+ctxt** runs successively the report mode and the context mode. It should be used with the C option (described later) which checks the code on a file basis.
...

In kernel source tree [notes below are for ver 4.4.21]:

'coccicheck' is a Makefile target.

```
$ cd linux-4.4.21
$ grep -A1 "coccicheck\:" Makefile
coccicheck:
	$(Q)$(CONFIG_SHELL) $(srctree)/scripts/$@
$ file scripts/coccicheck
scripts/coccicheck: Bourne-Again shell script, ASCII text executable
$ ls scripts/coccinelle/
api/  free/  iterators/  locks/  misc/  null/  tests/
$ ls -l scripts/coccinelle/locks/
total 16
-rw-rw-r-- 1 seawolf seawolf 2162 Sep 15  2016 call_kern.cocci << these
                                               are coccinelle "semantic patches" >>
-rw-rw-r-- 1 seawolf seawolf 1558 Sep 15  2016 double_lock.cocci
-rw-rw-r-- 1 seawolf seawolf 1572 Sep 15  2016 flags.cocci
-rw-rw-r-- 1 seawolf seawolf 1741 Sep 15  2016 mini_lock.cocci
$
```

Lets try it out

```
$ make coccicheck
spatch is part of the Coccinelle project and is available at http://coccinelle.lip6.fr/
Makefile:1471: recipe for target 'coccicheck' failed
make: *** [coccicheck] Error 1
$ sudo apt install coccinelle
...

$
```

Running coccicheck on the kernel codebase – a few Examples

Eg.1

```
$ nproc                    << get # of cpu cores >>
1
$ make -j2 coccicheck      << run coccinelle on entire codebase using all semantic
scripts; takes a long time ... >>
You have not explicitly specified the mode to use. Using default "report" mode.
Available modes are the following: patch, report, context, org
You can specify the mode with "make coccicheck MODE=<mode>"
Note however that some modes are not implemented by some semantic patches.

Please check for false positives in the output before submitting a patch.
When using "patch" mode, carefully review the patch before submitting it.

./drivers/block/cciss_scsi.c:706:9-41: WARNING: casting value returned by memory allocation
function to (struct cciss_scsi_adapter_data_t *) is useless.

...

./drivers/scsi/hpsa.c:6312:8-32: WARNING: casting value returned by memory allocation function
to (BIG_IOCTL_Command_struct *) is useless.
./sound/pci/emu10k1/emufx.c:1189:19-28: WARNING: casting value returned by memory allocation
function to (u_int32_t __user *) is useless.
```

```
./sound/pci/emu10k1/emufx.c:1827:19-28: WARNING: casting value returned by memory allocation
function to (u_int32_t __user *) is useless.
./fs/proc/inode.c:60:7-24: WARNING: casting value returned by memory allocation function to
(struct proc_inode *) is useless.
./fs/ncpfs/inode.c:55:7-28: WARNING: casting value returned by memory allocation function to
(struct ncp_inode_info *) is useless.
./fs/9p/vfs_inode.c:241:12-29: WARNING: casting value returned by memory allocation function to
(struct v9fs_inode *) is useless.
./drivers/scsi/lpfc/lpfc_els.c:4973:15-22: WARNING: kzalloc should be used for rdp_context,
instead of kmalloc/memset
./drivers/net/ethernet/mellanox/mlx5/core/cmd.c:921:14-15: WARNING: *_pool_zalloc should be used
for mailbox -> buf, instead of *_pool_alloc/memset
./drivers/net/ethernet/mellanox/mlx4/cmd.c:2640:14-15: WARNING: *_pool_zalloc should be used for
mailbox -> buf, instead of *_pool_alloc/memset

...

./arch/arm/mach-zx/zx296702-pm-domain.c:192:3-8: No need to set .owner here. The core will do
it.
./arch/arm/mach-s3c64xx/mach-crag6410-module.c:394:3-8: No need to set .owner here. The core
will do it.
./include/linux/err.h:58:1-3: WARNING: PTR_ERR_OR_ZERO can be used
./mm/page_owner.c:315:1-3: WARNING: PTR_ERR_OR_ZERO can be used

...

$

<<
```

Under the hood, it's the 'spatch' coccinelle utility doing the heavy lifting: eg.

```
spatch -D report --very-quiet --no-show-diff --cocci-file
./scripts/coccinelle/api/alloc/alloc_cast.cocci --no-includes --include-headers --dir .
-I ./arch/x86/include -I arch/x86/include/generated/uapi -I arch/x86/include/generated
-I include -I ./arch/x86/include/uapi -I arch/x86/include/generated/uapi -I
./include/uapi -I include/generated/uapi --include ./include/linux/kconfig.h --max 1 --
index 0
>>
```

Eg.2

```
<< Run coccinelle only on the given branch ('M=...') >>
$ make -j2 M=drivers/net/wireless/realtek/ coccicheck MODE=report

Please check for false positives in the output before submitting a patch.
When using "patch" mode, carefully review the patch before submitting it.

drivers/net/wireless/realtek/rtlwifi/pci.c:368:7-12: ERROR: iterator variable bound on line 366
cannot be NULL
drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c:188:14-17: WARNING: Comparison to bool
drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c:195:13-16: WARNING: Comparison to bool

...

drivers/net/wireless/realtek/rtlwifi/rtl8723be/sw.c:98:1-34: WARNING: Assignment of bool to 0/1
drivers/net/wireless/realtek/rtlwifi/rtl8723be/sw.c:100:1-34: WARNING: Assignment of bool to 0/1
drivers/net/wireless/realtek/rtlwifi/rtl8723be/dm.c:1183:27-47: WARNING: Comparison of bool to
0/1

...

drivers/net/wireless/realtek/rtlwifi/rtl8723ae/hw.c:960:5-13: WARNING: Comparison to bool
drivers/net/wireless/realtek/rtlwifi/rtl8192ee/sw.c:99:1-34: WARNING: Assignment of bool to 0/1
```

```
...

drivers/net/wireless/realtek/rtlwifi/wifi.h:2986:9-24: Move constant to right.
drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c:161:6-20: Move constant to right.
drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c:812:7-26: Move constant to right.

...

drivers/net/wireless/realtek/rtlwifi/pci.c:958:13-16: Unneeded variable: "ret". Return
"IRQ_HANDLED" on line 961
drivers/net/wireless/realtek/rtlwifi/rtl8723be/phy.c:691:2-3: Unneeded semicolon
drivers/net/wireless/realtek/rtlwifi/rtl8723be/phy.c:323:3-4: Unneeded semicolon
drivers/net/wireless/realtek/rtlwifi/rtl8723be/phy.c:281:3-4: Unneeded semicolon
drivers/net/wireless/realtek/rtlwifi/btcoexist/halbtc8821a2ant.c:2841:3-4: Unneeded semicolon
drivers/net/wireless/realtek/rtlwifi/debug.c:40:5-15: duplicated argument to & or |
drivers/net/wireless/realtek/rtlwifi/debug.c:40:18-26: duplicated argument to & or |
drivers/net/wireless/realtek/rtlwifi/debug.c:40:29-42: duplicated argument to & or |
drivers/net/wireless/realtek/rtlwifi/debug.c:40:45-54: duplicated argument to & or |
drivers/net/wireless/realtek/rtlwifi/debug.c:40:57-66: duplicated argument to & or |
$
```

Wow, 470 lines of output (truncated above) for just the drivers/net/wireless/realtek/ branch (kernel ver 4.4.21)! Of course, we must check for false positives.

Eg.3

Lets get coccinelle to generate a patch for the same:

```
$ make -j2 M=drivers/net/wireless/realtek/rtlwifi coccicheck MODE=patch

Please check for false positives in the output before submitting a patch.
When using "patch" mode, carefully review the patch before submitting it.

rule starting on line 17: position variables or mixed modifs interfere with comm_assoc isobool (
(

(
(unknown *to == NULL)    << ignore, coccinelle warnings >>
  >>> IS_ERR(to)

...

<< the patch below is for this 'report'ed issue in the previous run:
drivers/net/wireless/realtek/rtlwifi/pci.c:368:7-12: ERROR: iterator variable bound on line 366
cannot be NULL
>>

diff -u -p a/drivers/net/wireless/realtek/rtlwifi/pci.c
b/drivers/net/wireless/realtek/rtlwifi/pci.c
--- a/drivers/net/wireless/realtek/rtlwifi/pci.c
+++ b/drivers/net/wireless/realtek/rtlwifi/pci.c
@@ -365,7 +365,7 @@ static bool rtl_pci_check_buddy_priv(str
       if (!list_empty(&rtlpriv->glb_var->glb_priv_list)) {
              list_for_each_entry(tpriv, &rtlpriv->glb_var->glb_priv_list,
                                  list) {
-                     if (tpriv) {    << since it cannot be NULL, >>
+                     {               << get rid of the check >>
                             tpcipriv = (struct rtl_pci_priv *)tpriv->priv;
                             RT_TRACE(rtlpriv, COMP_INIT, DBG_LOUD,
                                      "pcipriv->ndis_adapter.funcnumber %x\n",

...

diff -u -p a/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c
```

```
b/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c
--- a/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c
+++ b/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/mac.c
@@ -185,14 +185,14 @@ bool rtl92c_init_llt_table(struct ieee80

       for (i = 0; i < (boundary - 1); i++) {
              rst = rtl92c_llt_write(hw, i , i + 1);
-             if (true != rst) {             << "WARNING: Comparison to bool" >>
+             if (!rst) {
                     pr_err("===> %s #1 fail\n", __func__);
                     return rst;
              }

...

diff -u -p a/drivers/net/wireless/realtek/rtlwifi/rtl8723be/sw.c
b/drivers/net/wireless/realtek/rtlwifi/rtl8723be/sw.c
--- a/drivers/net/wireless/realtek/rtlwifi/rtl8723be/sw.c
+++ b/drivers/net/wireless/realtek/rtlwifi/rtl8723be/sw.c
@@ -95,9 +95,9 @@ int rtl8723be_init_sw_vars(struct ieee80
       rtl8723be_bt_reg_init(hw);
       rtlpriv->btcoexist.btc_ops = rtl_btc_get_ops_pointer();

-      rtlpriv->dm.dm_initialgain_enable = 1;
+      rtlpriv->dm.dm_initialgain_enable = true;
       rtlpriv->dm.dm_flag = 0;
-      rtlpriv->dm.disable_framebursting = 0;
+      rtlpriv->dm.disable_framebursting = false;

...

diff -u -p a/drivers/net/wireless/realtek/rtlwifi/wifi.h
b/drivers/net/wireless/realtek/rtlwifi/wifi.h
--- a/drivers/net/wireless/realtek/rtlwifi/wifi.h
+++ b/drivers/net/wireless/realtek/rtlwifi/wifi.h
@@ -2983,7 +2983,7 @@ static inline void rtl_set_rfreg(struct

 static inline bool is_hal_stop(struct rtl_hal *rtlhal)
 {
-      return (_HAL_STATE_STOP == rtlhal->state);
+      return (rtlhal->state == _HAL_STATE_STOP);    << "Move constant to right." >>
 }

...

diff -u -p a/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/hw.c
b/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/hw.c
--- a/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/hw.c
+++ b/drivers/net/wireless/realtek/rtlwifi/rtl8192cu/hw.c
@@ -780,10 +780,10 @@ static void _rtl92cu_init_chipT_queue_pr
                              HQSEL_HIQ;
              break;
       case 1:
-             if (TX_SELE_LQ == queue_sel) {
+             if (queue_sel == TX_SELE_LQ) {
                     /* map all endpoint to Low queue */
                     hq_sele = 0;
-             } else if (TX_SELE_HQ == queue_sel) {
+             } else if (queue_sel == TX_SELE_HQ) {
                     /* map all endpoint to High queue */

...

diff -u -p a/drivers/net/wireless/realtek/rtlwifi/rtl8723be/phy.c
b/drivers/net/wireless/realtek/rtlwifi/rtl8723be/phy.c
--- a/drivers/net/wireless/realtek/rtlwifi/rtl8723be/phy.c
+++ b/drivers/net/wireless/realtek/rtlwifi/rtl8723be/phy.c
```

```
@@ -278,7 +278,7 @@ static void _rtl8723be_phy_set_txpower_b
                             "Invalid RateSection %d in Band 2.4G, Rf Path %d, %dTx in
PHY_SetTxPowerByRateBase()\n",
                             rate_section, path, txnum);
                        break;
-               };                              << “Unneeded semicolon” >>
+               }

...
```

etc etc

Using Coccinelle with a single semantic patch

```
-------------------------------------------

The optional make variable COCCI can be used to check a single
semantic patch. In that case, the variable must be initialized with
the name of the semantic patch to apply.

For instance::

        make coccicheck COCCI=<my_SP.cocci> MODE=patch

or::

        make coccicheck COCCI=<my_SP.cocci> MODE=report
```

Using Coccinelle for out-of-tree C source

The coccinelle analyser can certainly be used to parse any C source (or header) file(s).
The essential command is the *spatch* one.

Simple eg. from the man page of spatch(1):

```
     ./spatch --sp-file foo.cocci foo.c

       Apply the semantic patch foo.cocci to the C file foo.c.

     ./spatch --sp-file foo.cocci foo.c -o /tmp/newfoo.c

       The same as the above, except that a modified version of foo.c is stored in
/tmp/newfoo.c.

     It is also possible to apply spatch to all of the C files in a directory:

         ./spatch --cocci-file foo.cocci --dir foodir

     If the semantic patch is not working as expected, the option --debug shows
selection of information about the application of a semantic patch to a file or
directory.
```

A convenient script – ***coccichk*** - to invoke spatch on either a single file, several files (with

wildcard), is available in my *usefulsnips* repo:

```
$ coccichk
Usage: coccichk <source-pathname(s)> file1 [file2] [file3] ...
  -OR-
coccichk source-folder
$
<< try out our script on the Apache2 httpd v2.4.27 codebase >>
$ coccichk httpd-2.4.27/srclib/apr/network_io/unix/sendrecv.c  << single C file >>
@@@ coccichk: WARNING! Don't blindly act on coccinelle's output @@@
@@@ False positives can and do occur. Verify your code.         @@@
@@@ (Also, it works only on C source/header files, nothing else)@@@

#FILE:<...>/httpd-2.4.27/srclib/apr/network_io/unix/sendrecv.c
$ coccichk httpd-2.4.27/srclib/apr/network_io/unix/*.[ch]     << several C files >>
@@@ coccichk: WARNING! Don't blindly act on coccinelle's output @@@
@@@ False positives can and do occur. Verify your code.         @@@
@@@ (Also, it works only on C source/header files, nothing else)@@@

#FILE:httpd-2.4.27/srclib/apr/network_io/unix/inet_ntop.c
#FILE:httpd-2.4.27/srclib/apr/network_io/unix/inet_pton.c
#FILE:httpd-2.4.27/srclib/apr/network_io/unix/multicast.c
#FILE:httpd-2.4.27/srclib/apr/network_io/unix/sendrecv.c
#FILE:httpd-2.4.27/srclib/apr/network_io/unix/sockaddr.c
httpd-2.4.27/srclib/apr/network_io/unix/sockaddr.c:173:59-60: Unneeded semicolon
#FILE:httpd-2.4.27/srclib/apr/network_io/unix/sockets.c
httpd-2.4.27/srclib/apr/network_io/unix/sockets.c:560:17-24: ERROR: ( * sock ) is NULL but
dereferenced.
httpd-2.4.27/srclib/apr/network_io/unix/sockets.c:330:4-76: code aligned with following code on
line 331
#FILE:httpd-2.4.27/srclib/apr/network_io/unix/socket_util.c
#FILE:httpd-2.4.27/srclib/apr/network_io/unix/sockopt.c
$ coccichk httpd-2.4.27/srclib/apr/network_io/unix/ << on a folder >>
@@@ coccichk: WARNING! Don't blindly act on coccinelle's output @@@
@@@ False positives can and do occur. Verify your code.         @@@
@@@ (Also, it works only on C source/header files, nothing else)@@@

#Folder:httpd-2.4.27/srclib/apr/network_io/unix/
httpd-2.4.27/srclib/apr/network_io/unix/sockets.c:560:17-24: ERROR: ( * sock ) is NULL but
dereferenced.
httpd-2.4.27/srclib/apr/network_io/unix/sockets.c:330:4-76: code aligned with following code on
line 331
httpd-2.4.27/srclib/apr/network_io/unix/sockaddr.c:173:59-60: Unneeded semicolon
$
```

Tip- *within the script code, set the var VERBOSE=1 to see the Coccinelle scripts being tried one by one...*

Additional Resources

As an introduction to learning how to write a Coccinelle semantic patch, this article is useful:
Coccinelle for the newbie.
JuliaLawall Wiki page: coccinelle developer

Related:

The Intel OpenSource 0Day project:
"This 0-Day service is an automated Linux kernel test service that provides comprehensive test coverage of the Linux kernel. It monitors various kernel trees, spanning the mainline tree, the next tree, maintainers' trees, and key developers' trees for changes. 0-Day also monitors the *Linux Kernel Mail List* (LKML) itself. 0-Day performs build, boot, functional, performance, and

power tests whenever it detects changes. Our goal is to assist developers to find problems as early as possible so they can be fixed at the first opportunity. ...

0-Day does build tests using more than 100 different kernel configurations. For Intel x86 architecture, static analysis is also performed for selected widely-used configurations. The turnaround time of build results is within hours after a code change is detected. If there is any failure during the build stage, 0-Day will bisect the failure to the first code patch that introduces the failure. That patch author is then notified with the failure information and the steps to reproduce the problem. This allows developers to reproduce the problem in their local environments and to verify their fixes.”
Tools To Cleanup Linux Kernel.

Sparse

Sparse is a semantic checker for C programs; it can be used to find a number of potential problems with kernel code. See https://lwn.net/Articles/689907/ for an overview of sparse.

<<
Source: https://www.kernel.org/doc/html/v4.11/dev-tools/sparse.html

Using sparse

Do a kernel make with “make C=1” to run sparse on all the C files that get recompiled, or use “make C=2” to run sparse on the files whether they need to be recompiled or not. The latter is a fast way to check the whole tree if you have already built it.

The optional make variable CF can be used to pass arguments to sparse. The build system passes -*Wbitwise* to sparse automatically.
>>

```
$ make C=2
  CHK     include/config/kernel.release
  CHK     include/generated/uapi/linux/version.h
  CHK     include/generated/utsrelease.h

...

  CHECK   init/main.c           << the 'CHECK' directives are sparse >>
init/main.c:162:12: warning: symbol 'envp_init' was not declared. Should it be static?
  CHK     include/generated/compile.h
  CHECK   init/version.c
  CHECK   init/do_mounts.c
init/do_mounts.c:8:2: warning: "Sparse checking disabled for this file"
include/linux/slab.h:307:43: error: attribute '__assume_aligned__': unknown attribute
include/linux/slab.h:308:58: error: attribute '__assume_aligned__': unknown attribute
include/linux/slab.h:322:58: error: attribute '__assume_aligned__': unknown attribute

...

  CHECK   init/calibrate.c
init/calibrate.c:260:37: warning: symbol 'calibrate_delay_is_known' was not declared.
Should it be static?
init/calibrate.c:270:28: warning: symbol 'calibration_delay_done' was not declared.
Should it be static?
  CHECK   init/init_task.c
  CHECK   arch/x86/entry/syscall_64.c
arch/x86/entry/syscall_64.c:30:10: warning: Initializer entry defined twice
arch/x86/include/generated/asm/syscalls_64.h:1:1:   also defined here
  CHECK   arch/x86/entry/common.c
  CHECK   arch/x86/entry/vdso/vma.c
  CHECK   arch/x86/entry/vdso/vclock_gettime.c
  CHECK   arch/x86/entry/vdso/vgetcpu.c
arch/x86/entry/vdso/vgetcpu.c:14:1: warning: symbol '__vdso_getcpu' was not declared.
Should it be static?
```

```
  CHECK   arch/x86/entry/vdso/vdso-image-64.c
  CHECK   arch/x86/entry/vsyscall/vsyscall_gtod.c
  CHECK   arch/x86/entry/vsyscall/vsyscall_64.c
  CHECK   arch/x86/kernel/process_64.c
arch/x86/kernel/process_64.c:55:11: warning: symbol 'rsp_scratch' was not declared.
Should it be static?
  CHECK   arch/x86/kernel/signal.c
arch/x86/kernel/signal.c:170:17: warning: cast removes address space of expression

...
```

Source below: http://elinux.org/Sparse

Use sparse on a C normal project

Running sparse on a C project is quite easy if you use the cgcc wrapper script installed with the sparse package. Just call cgcc instead of your normal gcc.

For make use

```
make CC=cgcc
```

With configure use

```
./configure CC=cgcc
make
```

Example: we try cgcc on the netcat-0.7.1 codebase:

```
$ cd <...>/netcat-0.7.1/
$ ls
ABOUT-NLS   AUTHORS    config.guess*  config.rpath*  configure*    COPYING  INSTALL
lib/  Makefile.am  missing*        NEWS  README  TODO
aclocal.m4  ChangeLog  config.h.in    config.sub*    configure.ac  doc/     install-sh*
m4/   Makefile.in  mkinstalldirs*  po/   src/
$ ./configure CC=cgcc
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
...
config.status: creating po/Makefile
$ make
make  all-recursive
make[1]: Entering directory '.../netcat-0.7.1'
Making all in m4
...
Making all in src
make[2]: Entering directory '.../netcat-0.7.1/src'
cgcc -DLOCALEDIR=\"\/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I..     -g -O2 -
Wall -c `test -f 'core.c' || echo './'`core.c
cgcc -DLOCALEDIR=\"\/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I..     -g -O2 -
Wall -c `test -f 'flagset.c' || echo './'`flagset.c
cgcc -DLOCALEDIR=\"\/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I..     -g -O2 -
Wall -c `test -f 'misc.c' || echo './'`misc.c
cgcc -DLOCALEDIR=\"\/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I..     -g -O2 -
```

```
Wall -c `test -f 'netcat.c' || echo './'`netcat.c

   << output below is from the sparse checker >>

netcat.c:52:6: warning: symbol 'opt_udpmode' was not declared. Should it be static?
netcat.c:60:6: warning: symbol 'opt_exec' was not declared. Should it be static?
netcat.c:216:11: warning: Using plain integer as NULL pointer
netcat.c:216:17: warning: Using plain integer as NULL pointer
netcat.c:374:16: warning: Using plain integer as NULL pointer

...

cgcc -DLOCALEDIR=\"\/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I..     -g -O2 -
Wall -c `test -f 'telnet.c' || echo './'`telnet.c
   << output below is actually a regular warning from the compiler (gcc) >>
telnet.c: In function ‘netcat_telnet_parse’:
telnet.c:134:7: warning: ignoring return value of ‘write’, declared with attribute
warn_unused_result [-Wunused-result]
       write(ncsock->fd, putrq, 3);  /* FIXME: handle failures */
       ^~~~~~~~~~~~~~~~~~~~~~~~~~
...
$
```

In order to actually see the sparse checker tool being invoked:

1. Edit the ‘cgcc’ perl script (/usr/bin/cgcc), changing the line
   `my $verbose = 0;`
   to 1
2. do a ‘make clean’
3. make with verbose mode on:

```
$ make V=2 CC=cgcc

...

sparse -DLOCALEDIR=\"/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I.. -g -O2 -Wall -c
netcat.c -Dx86_64=1 -D__x86_64=1 -D__x86_64__=1 -D__LP64__=1 -D__CHAR_BIT__=8 -
D__SCHAR_MAX__=127 -D__SHRT_MAX__=32767 -D__INT_MAX__=2147483647 -
D__LONG_MAX__=9223372036854775807L -D__LONG_LONG_MAX__=9223372036854775807LL -
D__LONG_LONG_LONG_MAX__=170141183460469231731687303715884105727LLL -D__FLT_RADIX__=2 -
D__FINITE_MATH_ONLY__=0 -D__DECIMAL_DIG__=33 -D__FLT_MANT_DIG__=24 -D__FLT_DIG__=6 -
D__FLT_MIN_EXP__='(-125)' -D__FLT_MAX_EXP__=128 -D__FLT_MIN_10_EXP__='(-37)' -
D__FLT_MAX_10_EXP__=38 -D__FLT_HAS_INFINITY__=1 -D__FLT_HAS_QUIET_NAN__=1 -
D__FLT_DENORM_MIN__=1.40129846e-45F -D__FLT_EPSILON__=1.19209290e-7F -
D__FLT_MAX__=3.40282347e+38F -D__FLT_MIN__=1.17549435e-38F -D__DBL_MANT_DIG__=53 -
D__DBL_DIG__=15 -D__DBL_MIN_EXP__='(-1021)' -D__DBL_MAX_EXP__=1024 -D__DBL_MIN_10_EXP__='(-307)'
-D__DBL_MAX_10_EXP__=308 -D__DBL_HAS_INFINITY__=1 -D__DBL_HAS_QUIET_NAN__=1 -
D__DBL_DENORM_MIN__=4.9406564584124654e-324 -D__DBL_EPSILON__=2.2204460492503131e-16 -
D__DBL_MAX__=1.7976931348623157e+308 -D__DBL_MIN__=2.2250738585072014e-308 -
D__LDBL_MANT_DIG__=113 -D__LDBL_DIG__=33 -D__LDBL_MIN_EXP__='(-16381)' -D__LDBL_MAX_EXP__=16384
-D__LDBL_MIN_10_EXP__='(-4931)' -D__LDBL_MAX_10_EXP__=4932 -D__LDBL_HAS_INFINITY__=1 -
D__LDBL_HAS_QUIET_NAN__=1 -D__LDBL_DENORM_MIN__=6.475175119438025110924438958227646 55e-4966L -
D__LDBL_EPSILON__=1.9259299443872358530559779425849 2732e-34L -
D__LDBL_MAX__=1.1897314953572317650857593266280 0702e+4932L -
D__LDBL_MIN__=3.3621031431120935062626778173217 5260e-4932L -U__SIZE_TYPE__ -D__SIZE_TYPE__=long\
unsigned\ int -D__SIZEOF_POINTER__=8 -Dunix=1 -D__unix=1 -D__unix__=1 -D__linux__=1 -D__linux=1
-Dlinux=linux -gcc-base-dir /usr/lib/gcc/x86_64-linux-gnu/6/ -multiarch-dir x86_64-linux-gnu
netcat.c:52:6: warning: symbol 'opt_udpmode' was not declared. Should it be static?
netcat.c:60:6: warning: symbol 'opt_exec' was not declared. Should it be static?
netcat.c:216:11: warning: Using plain integer as NULL pointer
netcat.c:216:17: warning: Using plain integer as NULL pointer
netcat.c:374:16: warning: Using plain integer as NULL pointer
```

```
cc -DLOCALEDIR=\"/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I.. -g -O2 -Wall -c netcat.c
cgcc -DLOCALEDIR=\"\/usr/local/share/locale\" -DHAVE_CONFIG_H -I. -I. -I..     -g -O2 -Wall -c
`test -f 'network.c' || echo './'`network.c

...

$
```
