Coding Challenge- Coccinelle

My Experience with Coccicheck in embedded system

I read through the coccinelle documentation and out of the 2 tasks required for the challenge I tried one more exercise to learn about coccicheck. I am from embedded background, so I tried out coccicheck in a basic Embedded system application- bit masking. One of the issues there is once we define any macro definition we need to go into each line to substitute the macro. So, I thought of using cocci for the bitwise operation macro. I wrote a simple cocci demo to read and write the bit mask. I then wrote a simple sample program in C which sets and reads the register value in the sixth bit. I then ran spatch as below:

```
# spatch -sp_file demos/bitmask.cocci demos/bitoperation.c -o /tmp/bitoperation.c
init_defs_builtins: /usr/local/bin/../lib/coccinelle/standard.h
HANDLING: demos/bitoperation.c
diff =
--- demos/bitoperation.c
+++ /tmp/cocci-output-6156-ac5030-bitoperation.c
@@ -12,17 +12,17 @@ int main(int i) {
  const int ZOUT8 = 0x08;       // = 0b0001000  // 8-bit z register address

  printf("MCTL     = %04x \n", MCTL);          // Display binary MCTL address
-  printf("writeMask = %04x (OR)\n", (1u<<6));    // Display binary writeMask
+  printf("writeMask = %04x (OR)\n", WRITEMASK(6));    // Display binary writeMask

-  cmd = MCTL | (1u<<6);                    // MCTL OR writeMask to cmd
+  cmd = MCTL | WRITEMASK(6);                     // MCTL OR writeMask to cmd

  printf("-------------------------\n");       // Draw line
  printf("cmd      = %04x \n\n", cmd);         // Display cmd result
  printf("ZOUT8    = %04x \n", ZOUT8);         // Display binary ZOUT address
  printf("readMask  = %04x (AND)\n",           // Display binary readMask
-       ~(1u<<6) & 0b1111111);                // Bits 31...7 -> 0 for display
+       READMASK(6) & 0b1111111);                // Bits 31...7 -> 0 for display

-  cmd = ZOUT8 & ~(1u<<6);                    // ZOUT8 AND readMask to cmd
+  cmd = ZOUT8 & READMASK(6);                     // ZOUT8 AND readMask to cmd

  printf("-------------------------\n");       // Draw line
  printf("cmd      = %04x \n", cmd);           // Display cmd result
```

- Please refer the code details here: https://github.com/ckadithya/coccinelle/pull/1/files

My Experience with running coccicheck in Linux kernel

- I already cloned the Linux mainline and Linux stable kernel as a part of the course “A Beginner’s Guide to Linux Kernel Development" I did lately. In that course I performed an exercise while learning to create a patch, for which I made a small change to the uvc driver at **drivers/media/usb/uvc/uvc_driver.c** then, recompiled the kernel and installed it.
- So, I was set with the kernel installation part.
- Next, I cloned and installed coccinelle.
- While installing I encountered the below error when trying to install python.

 # Error

```
# make install
mkdir -p /usr/local/bin /usr/local/lib/coccinelle
mkdir -p /usr/local/lib/coccinelle/ocaml
if test -f bundles/pyml/dllpyml_stubs.so; then \
   /usr/bin/install -c -m 755 bundles/pyml/dllpyml_stubs.so \
      /usr/local/lib/coccinelle; \
fi
if test -f bundles/pcre/dllpcre_stubs.so; then \
   /usr/bin/install -c -m 755 bundles/pcre/dllpcre_stubs.so \
      /usr/local/lib/coccinelle; \
fi
/usr/bin/install -c -m 755 spatch.opt /usr/local/bin/spatch
/usr/bin/install -c -m 644 standard.h /usr/local/lib/coccinelle
/usr/bin/install -c -m 644 standard.iso /usr/local/lib/coccinelle
/usr/bin/install -c -m 644 ocaml/*.cmi /usr/local/lib/coccinelle/ocaml/
if test -f ocaml/coccilib.cmx; then \
   /usr/bin/install -c -m 644 ocaml/*.cmx /usr/local/lib/coccinelle/ocaml/; \
fi
/usr/bin/install -c -m 755 tools/spgen/source/spgen.opt \
    /usr/local/bin/spgen
/usr/bin/install -c -m 644 python/coccilib/*.py \
   /usr/local/lib/coccinelle/python/coccilib
/usr/bin/install: target '/usr/local/lib/coccinelle/python/coccilib' is not a directory
Makefile:332: recipe for target 'install-python' failed
make: *** [install-python] Error 1
```

- Then I created the directory in the present folder and proceeded to get it installed.

 # Solution

```
# mkdir -p /usr/local/lib/coccinelle/python/coccilib
# make install
mkdir -p /usr/local/bin /usr/local/lib/coccinelle
mkdir -p /usr/local/lib/coccinelle/ocaml
if test -f bundles/pyml/dllpyml_stubs.so; then \
   /usr/bin/install -c -m 755 bundles/pyml/dllpyml_stubs.so \
      /usr/local/lib/coccinelle; \
fi
if test -f bundles/pcre/dllpcre_stubs.so; then \
   /usr/bin/install -c -m 755 bundles/pcre/dllpcre_stubs.so \
      /usr/local/lib/coccinelle; \
fi
/usr/bin/install -c -m 755 spatch.opt /usr/local/bin/spatch
/usr/bin/install -c -m 644 standard.h /usr/local/lib/coccinelle
/usr/bin/install -c -m 644 standard.iso /usr/local/lib/coccinelle
/usr/bin/install -c -m 644 ocaml/*.cmi /usr/local/lib/coccinelle/ocaml/
if test -f ocaml/coccilib.cmx; then \
   /usr/bin/install -c -m 644 ocaml/*.cmx /usr/local/lib/coccinelle/ocaml/; \
fi
/usr/bin/install -c -m 755 tools/spgen/source/spgen.opt \
    /usr/local/bin/spgen
/usr/bin/install -c -m 644 python/coccilib/*.py \
   /usr/local/lib/coccinelle/python/coccilib
if test "x/usr/share/bash-completion/completions" != "xno"; then \
   mkdir -p /usr/share/bash-completion/completions; \
   /usr/bin/install -c -m 644 scripts/spatch.bash_completion \
      /usr/share/bash-completion/completions/spatch; \
fi
mkdir -p /usr/local/share/man/man1
```

```
mkdir -p /usr/local/share/man/man3
/usr/bin/install -c -m 644 docs/spatch.1 /usr/local/share/man/man1/
/usr/bin/install -c -m 644 docs/pycocci.1 /usr/local/share/man/man1/
/usr/bin/install -c -m 644 docs/spgen.1 /usr/local/share/man/man1/
/usr/bin/install -c -m 644 docs/Coccilib.3cocci /usr/local/share/man/man3/
```

- I went through the .cocci files in misc folder. I took **scripts/coccinelle/misc/array_size.cocci** as example.
- In that file (sizeof(E)/sizeof(*E)) was set to be replaced with ARRAY_SIZE(E). I wanted to try it out so, in the same module in which I performed change for my earlier course [**uvc driver at drivers/media/usb/uvc/uvc_driver.c**] I changed ARRAY_SIZE() with sizeof() in two place and saved it.
- Then I built coccicheck with verbose flag set, in patch mode as and for the drivers directory as **make V=99 coccicheck MODE=patch COCCI=scripts/coccinelle/misc/array_size.cocci M=drivers/media/usb/uvc**
- I got the following output

Output

```
# make V=99 coccicheck MODE=patch COCCI=scripts/coccinelle/misc/array_size.cocci M=drivers/media/usb/uvc

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

Processing array_size.cocci
with option(s) " --no-includes --include-headers"

Message example to submit a patch:
Use ARRAY_SIZE instead of dividing sizeof array with sizeof an element

The semantic patch that makes this change is available
in scripts/coccinelle/misc/array_size.cocci.

More information about semantic patching is available at
http://coccinelle.lip6.fr/

Semantic patch information:
This makes an effort to find cases where ARRAY_SIZE can be used such as
where there is a division of sizeof the array by the sizeof its first
element or by any indexed element or the element type. It replaces the
division of the two sizeofs by ARRAY_SIZE.

Running (4 in parallel): /usr/local/bin/spatch -D patch --very-quiet --cocci-file scripts/coccinelle/misc/array_size.cocci --no-
includes --include-headers --patch . --dir drivers/media/usb/uvc -I ./arch/x86/include -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 --jobs 4 --chunksize 1
diff -u -p a/drivers/media/usb/uvc/uvc_driver.c b/drivers/media/usb/uvc/uvc_driver.c
--- a/drivers/media/usb/uvc/uvc_driver.c
+++ b/drivers/media/usb/uvc/uvc_driver.c
@@ -237,7 +237,7 @@ struct usb_host_endpoint *uvc_find_endpo

static struct uvc_format_desc *uvc_format_by_guid(const u8 guid[16])
{
-    unsigned int len = sizeof(uvc_fmts)/sizeof(uvc_fmts[0]);
+    unsigned int len = ARRAY_SIZE(uvc_fmts);
     unsigned int i;

     for (i = 0; i < len; ++i) {
```

When I tried git diff I got the output as

```
root@ckadithya:/home/adithya/Workspace/linux_work/linux_mainline# git diff
diff --git a/drivers/media/usb/uvc/uvc_driver.c b/drivers/media/usb/uvc/uvc_driver.c
index 431d86e1c94b..b28c99a2395f 100644
--- a/drivers/media/usb/uvc/uvc_driver.c
+++ b/drivers/media/usb/uvc/uvc_driver.c
@@ -237,7 +237,7 @@ struct usb_host_endpoint *uvc_find_endpoint(struct usb_host_interface *alts,

static struct uvc_format_desc *uvc_format_by_guid(const u8 guid[16])
{
-       unsigned int len = ARRAY_SIZE(uvc_fmts);
+       unsigned int len = sizeof(uvc_fmts)/sizeof(uvc_fmts[0]);
        unsigned int i;

        for (i = 0; i < len; ++i) {
@@ -259,7 +259,7 @@ static u32 uvc_colorspace(const u8 primaries)
                V4L2_COLORSPACE_SMPTE240M,
        };

-       if (primaries < ARRAY_SIZE(colorprimaries))
+       if (primaries < sizeof(colorprimaries)/sizeof(u8))
                return colorprimaries[primaries];

        return 0;
@@ -2126,6 +2126,8 @@ static int uvc_probe(struct usb_interface *intf,
        int function;
        int ret;

+       pr_info("I changed uvcvideo driver in the Linux Kernel\n");
+
        if (id->idVendor && id->idProduct)
                uvc_trace(UVC_TRACE_PROBE, "Probing known UVC device %s "
                                "(%04x:%04x)\n", udev->devpath, id->idVendor,

root@ckadithya:/home/adithya/Workspace/linux_work/linux_mainline#
```

- I had fun learning throughout this exercise.

Scripts/coccicheck - Explanation

- Firstly, it checks and prints if SPATCH is installed and which version of it is there. If no SPATCH found it echos 'spatch is part of the Coccinelle project and is available at http://coccinelle.lip6.fr/' prompting user to get it.
- Then we go on to check if the verbose is set or is it 0. The flag is to show quiet.
- The way spatch calls include directories and the gcc calls is different. spatch only allows include directories with the syntax "-I include" while gcc also allows "-Iinclude" and "-include include". It is according given to cocciinclude.
- In line 61 we check flag c set 1 or 2 and if online=1, take only the last argument and that c file is checked against all files in the directory. And also parallelisation is not needed nproc=1. If online=0, we enable parallelisation and coccinelle try to see no.of processors given by user as in the J variable, even though by default coccicheck tries to run as parallel as possible.If nproc is not 1 then we set the jobs for cpu.
- In the line 99 the mode is checked. If no mode is set and online=0 then we echo that the default report mode is used and proceed with report mode.

- If mode is set to chain by user, we echo it and will proceed to try all available mode-patch, report, context, org in that order. when online =0 we echo to check for false positives in the output before submitting a patch and also echo When using "patch" mode, carefully review the patch before submitting it.
- At line 256, we check for all the files in directory to find the .cocci file and once sorted we take file one byb one and give as input to the coccinelle function. If the user set coccinille test case, we do it else we test all.
- Then in line 173 we go in to coccinelle function and get grep of all cocci parameters. We try to run with current spatch version and if it is not 1.0 or above a print statement is put out to update. If not proceed to run test cases.
- Then when verbose is set and online is 0 we go in to see what mode is set and accordingly the print statement is put out to make user aware of mode details.
- In line 227, once mode is known we check if it is a combination mode- chain or rep+ctxt. chain tries all modes in the order above until one succeeds. rep+ctxt runs the report mode and the context mode. If the C option is used checks the code on a file basis.If coccinelle is run with debug option stderr is redirected to /dev/null.
- If mode is not chain or rep+ctxt we go into the run_cmd() check and see if use_job is set to yes we go to line 125 run_cmd_parmap() where it gets all job and runs parallelly. If debug option is set it is redirected accordingly.
- If use_job is no then we go to run_cmd_old() and get the job sequentially and run it. In my case when trying it 4 jobs in 4 CPU's were run.