

Submitting Patches

@aesophor

Outline

- Coding style
- Tips
- Common problems
- Submitting a patch
- Examples

Coding Style (It does matter)

- Conform to the coding style of the project you're working on.
 - Google Style Guide: [C++](#) | [Java](#)
 - Oracle Java Code Conventions: [pdf](#)
- Naming conventions
 - snake case: `std::string cool_variable;`
 - camel case: `std::string coolVariable;`
 - hungarian notation: `std::string strCoolVariable;`
- Indentation - tab vs space? level?
- Column limit - maximum # of characters per line?

Coding Style

- Put a whitespace before and after `operator =` .
 - `int i=0; // bad`
 - `int i ==0; // good`
- `if / for / while` statements
 - `if(i==0){ // bad`
 - `if(i ==0){ // good`
- Curly braces
 - Differ across projects.
 - Follow the coding style guide of your organization.

Coding Style

- Java

```
7
8 public class Response {
9
10     private final StatusCode statusCode;
11     private Map<String, Object> content;
12
13     public Response(JsonObject json) {
14         statusCode = StatusCode.values()[json.get("statusCode").getAsInt()];
15
16         if (statusCode == StatusCode.SUCCESS) {
17             // Maps the "content" part to a HashMap.
18             JsonObject rawContentJson = json.getAsJsonObject("content");
19             content = NetworkUtils.getGson().fromJson(rawContentJson, Map.class);
20         }
21     }
22
23     public Response(String rawJson) {
24         this(NetworkUtils.getGson().fromJson(rawJson, JsonObject.class));
25     }
26
```

class

- **ClassName** (good)
- className (x)
- Class_Name (x)
- class_name (x)
- class_Name (x)

attributes / methods

- **VarName** (x)
- **varName** (good)
- Var_Name (x)
- var_name (x)
- var_Name (x)

Coding Style

- clang-format ^[1] - automatic code formatter
 - supports: C, C++, Java, JavaScript, JSON, Objective-C, Protobuf, C#
 - --style= LLVM, Google, Chromium, Mozilla, WebKit, File (case-insensitive)
 - example: clang-format -style=file -i main.cc
- Using it in your project
 - clang-format -style=google -dump-config > .clang-format
 - place .clang-format in your project's root directory
 - (optional) edit .clang-format
 - find . -iname "*.h" -o -iname "*.cc" | xargs clang-format -style=file -i

[1] <https://clang.llvm.org/docs/ClangFormat.html>

Tips

- Avoid code duplication

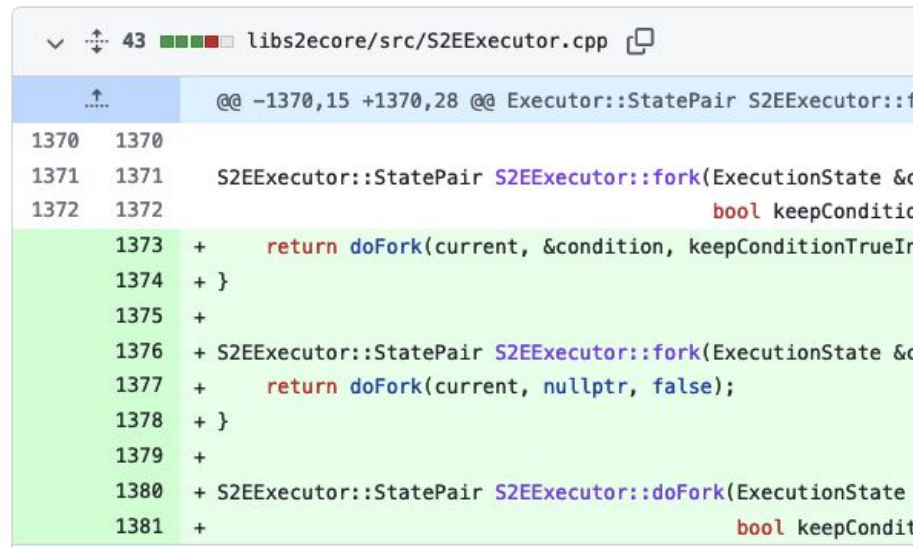
- code reuse
- keep interface clean and intuitive

- Resource management

- C - Single-Exit point rule
- Pre C++11 - BOOST_SCOPE_EXIT
- C++11 - smart pointers (RAII i.e. Scope-bound resource management)

- Time/Space complexity

- avoid excessive copying, e.g., `std::move()`
- avoid unneeded heap allocation, e.g., `std::vector<T> -> llvm::SmallVector<T, N>`





```
43 libs2ecore/src/S2Executor.cpp
@@ -1370,15 +1370,28 @@ Executor::StatePair S2Executor::f
1370 1370
1371 1371 S2Executor::StatePair S2Executor::fork(ExecutionState &
1372 1372 bool keepCondition
1373 + return doFork(current, &condition, keepConditionTrueIn
1374 + }
1375 +
1376 + S2Executor::StatePair S2Executor::fork(ExecutionState &
1377 + return doFork(current, nullptr, false);
1378 + }
1379 +
1380 + S2Executor::StatePair S2Executor::doFork(ExecutionState
1381 + bool keepCondition
```


Tips


- Avoid code duplication
 - code reuse
 - keep interface clean and intuitive
- Resource management
 - C - Single-Exit point rule
 - Pre C++11 - BOOST_SCOPE_EXIT
 - C++11 - smart pointers (RAII i.e. Scope-bound resource management)
- Time/Space complexity
 - avoid excessive copying, e.g., `std::move()`
 - avoid unneeded heap allocation, e.g., `std::vector<T> -> llvm::SmallVector<T, N>`


Tip


 **GitLab** ≡ Menu Search GitLab


 **Samba**


 Project information


 Repository


 Merge requests 149

 CI/CD

 Deployments

 Packages & Registries

 Analytics

 External wiki

```
200     ret = ldb_search(conn->ldb, tmp_ctx, &res, basedn, LDB_SCOPE_BASE, attrs, NULL);
201     if (ret != LDB_SUCCESS) {
202         goto failed;
203     }
204
205     if (res->count != 1) {
206         goto failed;
207     }
208
209     conf_dn = ldb_msg_find_attr_as_dn(conn->ldb, tmp_ctx, res->msgs[0], "configurationNam
210     if (conf_dn == NULL) {
211         goto failed;
212     }
213
214     policy_dn = ldb_dn_copy(tmp_ctx, conf_dn);
215     ldb_dn_add_child_fmt(policy_dn, "CN=Default Query Policy,CN=Query-Policies,CN=Directo
216     if (policy_dn == NULL) {
217         goto failed;
218     }
219
220     ret = ldb_search(conn->ldb, tmp_ctx, &res, policy_dn, LDB_SCOPE_BASE, attrs2, NULL);
221     if (ret != LDB_SUCCESS) {
222         goto failed;
223     }
224
225     if (res->count != 1) {
226         goto failed;
227     }
228
229     failed:
230     DBG_ERR("Failed to load ldap server query policies\n");
231     talloc_free(tmp_ctx);
232     return -1;
233 }
```

free(), release locks here...

Tips

- Avoid code duplication
 - code reuse
 - keep interface clean and intuitive
- Resource management
 - C - Single-Exit point rule
 - Pre C++11 - BOOST_SCOPE_EXIT
 - C++11 - smart pointers (RAII i.e. Scope-bound resource management)
- Time/Space complexity
 - avoid excessive copying, e.g., `std::move()`
 - avoid unneeded heap allocation, e.g., `std::vector<T> -> llvm::SmallVector<T, N>`

Common Problems

- Which project should I contribute to?
 - Do you use any open source library in your **side projects** ?
 - Do you use any open source library in your **thesis** ?
- Project seems difficult to get started with.
 - Trace the code with patience.
 - Make small changes to the source code, rebuild, and see how things break.
 - Start with something small (e.g., fixing typos)
 - Now fix a known bug from the issue tracker, or implement a new feature.

Submitting a Patch

- Write meaningful commit messages
 - subsystem: xxx: summary
 - Signed-off-by: Your Name <email>
- Submit via...
 - Pull request (Github)
 - Merge request (Gitlab)
 - Mailing List
 - git format-patch
 - git send-mail


Staging: comedi: dt9812: fixed a typo in comment.


Fixed a typo in comment.

Signed-off-by: Marco Wang <m.aesophor@gmail.com>

Signed-off-by: Greg Kroah-Hartman <gregkh@linuxfoundation.org>

 master

 v5.17-rc8 ... v5.0-rc1

 **aesophor** authored and **gregkh** committed on Nov 5, 2018

klee,s2e: add support for fork() without symbolic conditions ...

As discussed in [1] and [2], it would be useful if the user can fork a state unconditionally in their plugin code and switch back to it later.

This commit introduces a new way to fork a state without specifying symbolic conditions. Essentially, an overloaded version of ``klee::Executor::fork()`` is added which works similarly to the regular version but without any path constraint code.

Besides, an overloaded version of ``S2EExecutor::fork()`` is added, so that when the user explicitly forks a state using the new API, S2E can write debugging messages just like the regular version of ``S2EExecutor::fork()`` does.

Finally, since the two versions of ``S2EExecutor::fork()`` shares a lot of similarity, the actual forking work has been moved to ``S2EExecutor::doFork()`` in order to avoid code duplication.

Reference:

[1] https://groups.google.com/g/s2e-dev/c/x8tDX8i_v5o/m/JmuWEA4rQTAJ

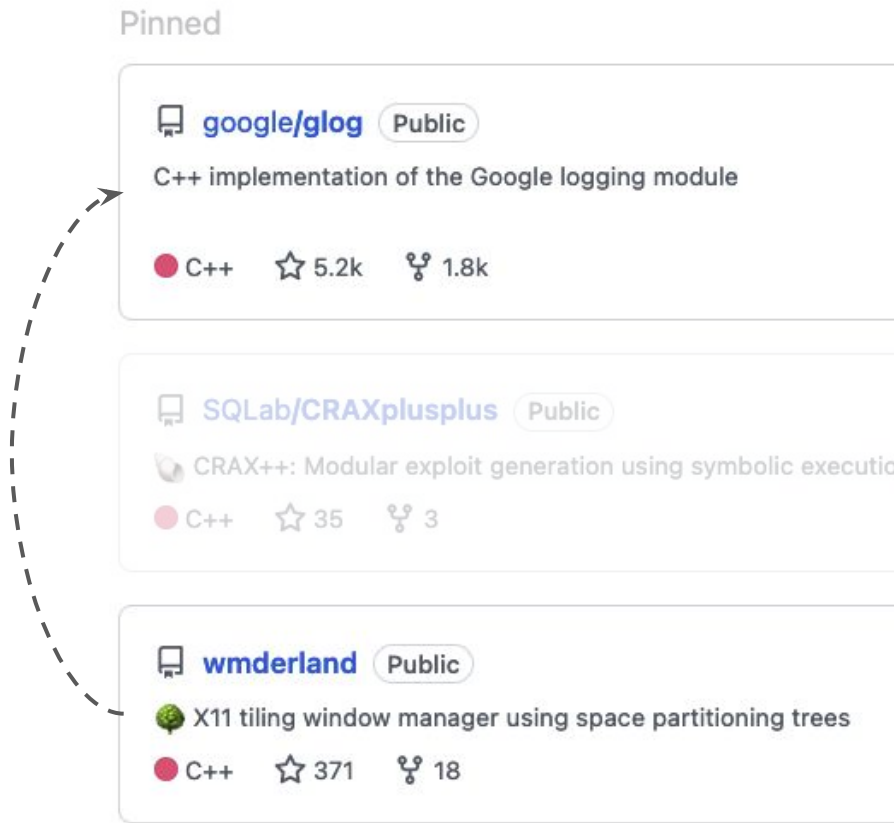
[2] <https://groups.google.com/g/s2e-dev/c/SeBpMlyHx3A/m/Qqe0USguDnIJ>

Signed-off-by: Marco Wang <m.aesophor@gmail.com>

 **aesophor** committed on Nov 4, 2021

Example 1 - google/glog

- Side project - wmderland
 - Tree-based X11 tiling window manager
 - Presented at [SITCON 2019](#)
 - I used **google/glog** to write debugging messages to a log file. I needed this because when an X11 session starts, I don't have access to stdout / stderr. Using gdb to debug a window manager also seems difficult.



Example 1 - google/glog

- google / glog
 - written in C++98
 - browse the issue tab
 - start resolving other people's issues.
 - 11 pull requests merged
 - 8 * non-trivial PRs
 - 3 * trivial PRs

Pinned



google/glog Public

C++ implementation of the Google logging module



C++



5.2k



1.8k



SQLab/CRAXplusplus Public



CRAX++: Modular exploit generation using symbolic execution



C++



35



3



wmderland Public



X11 tiling window manager using space partitioning trees



C++





































371



18

Example 1 - google/glog

<div>  0 Open 11 Closed </div>	Author	Label	Projects	Milestones	Reviews	Assignee	Sort
<div>  LogCleaner: make overdue_days_ unsigned int ✖ enhancement </div>	#760 by aesophor was merged on Dec 14, 2021	 0.6					 2
<div>  LogCleaner: Fix relative paths and add a new test ✖ bug </div>	#754 by aesophor was merged on Dec 9, 2021 · Changes requested	 0.6					 9
<div>  Make LogCleaner support relative paths ✓ cla: yes </div>	#654 by aesophor was merged on Oct 30, 2021 · Approved	 0.6			 1		 11
<div>  src/logging.cc: Fix a typo in comment ✓ cla: yes </div>	#593 by aesophor was merged on Feb 13, 2021	 0.5					
<div>  src/logging.cc: Make LogCleaner aware of filename extension (#589) ✓ cla: yes </div>	#592 by aesophor was merged on Feb 13, 2021	 0.5					 1
<div>  Replace sprintf() with snprintf() (#536) ✓ cla: yes </div>	#551 by aesophor was merged on Sep 30, 2020	 0.5					
<div>  Prepend the year to each glog line (#516) ✓ cla: yes </div>	#530 by aesophor was merged on Mar 25, 2020	 0.5					 1
<div>  src/logging.cc: encapsulate log cleaner, match logfile with <base_filename_><date>-<time>.<pid> ✖ cla: yes </div>	#502 by aesophor was merged on Oct 3, 2020	 0.5					 6
<div>  Fix race conditions in LOG_EVERY_N ✓ cla: yes </div>	#492 by aesophor was merged on Feb 23, 2021 · Changes requested	 0.5					 11
<div>  doc: add information about log cleaner ✓ cla: yes </div>	#491 by aesophor was merged on Nov 5, 2019	 0.5					
<div>  Add support for automatic removal of old logs ✓ cla: yes </div>	#432 by aesophor was merged on Nov 1, 2019	 0.5					 15

Example 2 - S2E/s2e

- S2E / s2e
 - written in C/C++17 (~150,000 LoC)
 - a virtual machine with symbolic execution capabilities
 - <https://github.com/S2E/s2e/pull/34>
 - 要會寫英文