Automated Patch Porting across Forked Projects

Luyao Ren
Peking University

Porting Patches

- Tedious to **manually track** patches
- **Extra adaptions** on patches may be needed
- **Vulnerabilities** exist due to delayed porting

```
Commit : github.com/vim/vim/commit/d23a823
Message: patch 8.0.1496: clearing a pointer takes two lines
Source : src/edit.c
==========
2936    static void
2937ins_compl_del_pum(void)
2938{
2939    if (compl_match_array != NULL)
2940    {
2941        pum_undisplay();
2942-       vim_clear((void **)&compl_match_array);
2942+       VIM_CLEAR(compl_match_array);
2943    }
2944}
```

Listing 1: A patch in Vim(2018-02-11)

```
Commit : github.com/neovim/neovim/commit/ae846b4
Message: vim-patch:8.0.1496: VIM_CLEAR()
Source : src/nvim/edit.c
==========
2521static void ins_compl_del_pum(void)
2522{
2523  if (compl_match_array != NULL) {
2524    pum_undisplay(false);
2525-   xfree(compl_match_array);
2526-   compl_match_array = NULL;
2525+   XFREE_CLEAR(compl_match_array);
2526  }
2527}
```

Listing 2: Ported Patch in NeoVim(2019-05-21)

Study Results

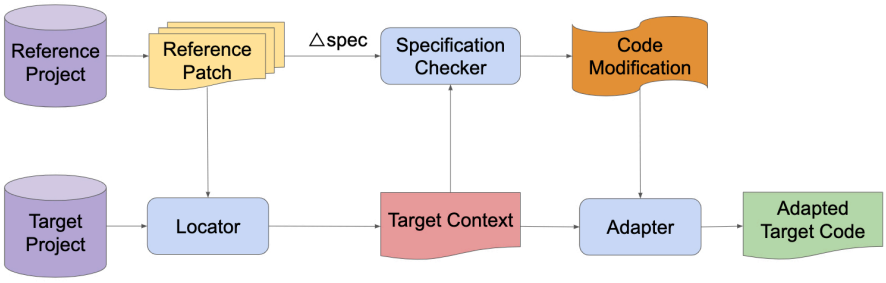
Reference Project	Target Project	#Sample	#Porting Needed		#Syntax Equivalent
			#Resolved	#Unported	
Vim	NeoVim	50	14	2	8
FreeBSD	OpenBSD	50	2	1	2
Marlin	Ultimaker2 Marlin	50	1	2	2
OpenRefine	LODRefine	50	1	18	18
Total		200	18	23	30
			41		

- **20.5%** patches need to be ported, which is a non-negligible portion of all analyzed patches.
- 73.2% patches could be directly ported without additional modifications. For the rest of **26.8%**, **extra adaptations** are required.

Insights

- The **maintenance behaviors** on porting patches among different forked projects are **diverse**.
- To **identify the unported patches**, the patch's current contexts and the evolution history of the contexts are important clues.
- **Comprehension of the semantics of patches** is crucial for adaption on patches if the corresponding contexts are not compatible for directly porting.

Preliminary Approach



- **Locator**: find the corresponding contexts of patches in target projects.
- **Specification Checker**: extract the Δ_{spec} and examine whether the target project satisfies it or not. If not, generate corresponding code modifications based on the Δ_{spec} and the target contexts.
- **Adapter**: adapt the code modifications for the target contexts.

Contact: rly@pku.edu.cn