



Tokyo Tech

バグトラッキング

2020年10月30日（金）
システム開発プロジェクト応用第一

東京工業大学
特任助教 内田公太

- 実際のシステム開発プロジェクトの現場で使われている現代的な開発ツールや手法を学ぶ
 - 正しいツールや手法の選択はソフトウェア開発を効率的に、そして楽しいものにする

到達目標：

- 現代的な開発ツールの基本的な使い方と適する用途が分かる

- 情報収集
- GDB
- Git
- バグトラッキング
- GitHub & Pull Request
- ユニットテスト
- 継続的インテグレーション
- デプロイと冪等性
- コミュニケーション

自己紹介

- 内田公太
- Twitter @uchan_nos
- 週3日：サイボウズ・ラボ株式会社
週2日：東工大の特任助教
- osdev-jpコアメンバー
- 『30日でできる! OS自作入門』の校正担当
- 『自作エミュレータで学ぶ
x86アーキテクチャ』の著者



スタイル：

- 少し講義して演習，の繰り返し

成績評価：

- 現代の開発技術・手法の理解度を評価する
- 各トピックを受講者自身のソフトウェア開発プロジェクトに適用し，レポートおよびリポジトリを提出する
- レポートおよびリポジトリの充実度で成績を決定する

- 色々な要素がある
 - トピックに対する回答
 - ドキュメント
 - コミットメッセージ
 - プルリクのやり取り
 - Etc.
-
- 総合的に判断して評価します

- 課題を含めたリポジトリとレポートを作成し, 提出
- 初回 (10/2) 説明したので詳しい話はしないつもり
 - 改めて聞きたい方がいたらお知らせください



Tokyo Tech

バグトラッキング

- バグが修正済かどうか, どのバージョンに修正が取り込まれたかを追跡して管理
- 専用システム (BTS: Bug Tracking System) がある

▲						
46792	clang	C	efriedma@quicinc.com	NEW	---	<u>ARM long double NaN cannot be negated</u> 2020-07-21
45157	clang	C	ndesaulniers@google.com	NEW	---	<u>Clang and gcc disagree on whether a `const struct` is a compile-time constant</u> 2020-03-12
42034	clang	C	unassignedclangbugs@nondot.org	NEW	---	<u>Incorrect expansion of '@' in inline asm macros</u> 2019-05-27
42098	clang	C	unassignedclangbugs@nondot.org	NEW	---	<u>__CHAR16_TYPE__ and __CHAR32_TYPE__ are defined on MacOS but they don't actually work</u> 2019-06-01
42120	clang	C	unassignedclangbugs@nondot.org	NEW	---	<u>Inconsistent (missing) incompatible function pointer type warnings</u> 2019-06-04
42142	clang	C	unassignedclangbugs@nondot.org	NEW	---	<u>Missing switch case checking in Clang diagnostics</u> 2019-11-12
42554	clang	C	unassignedclangbugs@nondot.org	NEW	---	<u>__BIGGEST_ALIGNMENT__ not the same as gcc for AVX and later</u> 2019-07-09

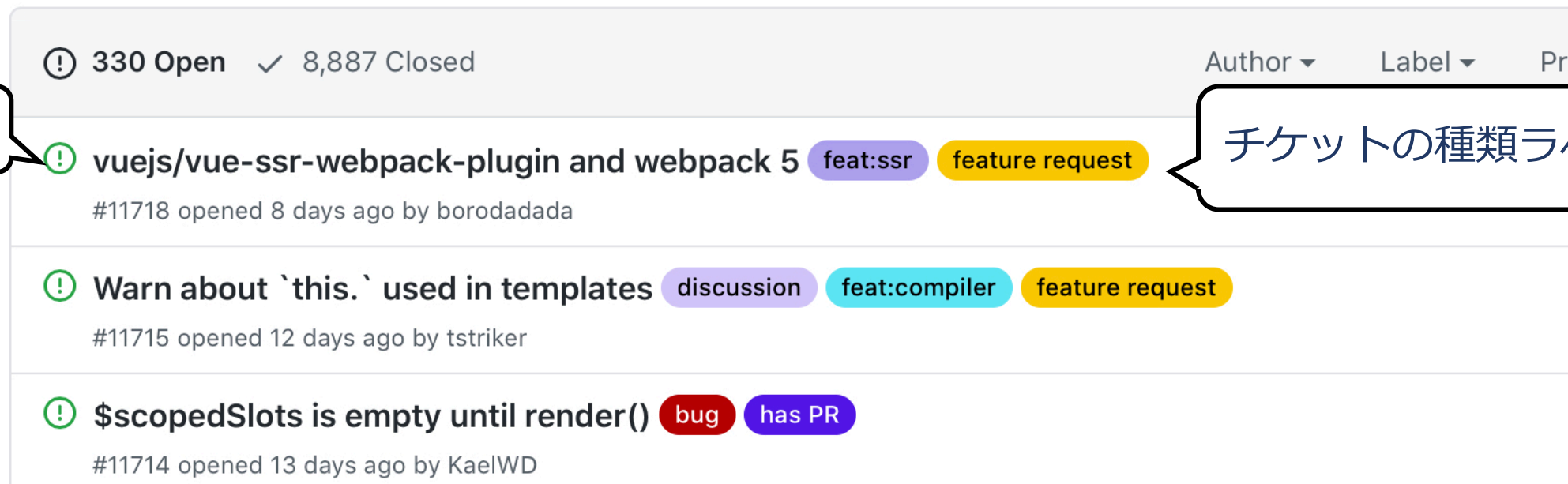
LLVM(Clang)のBTS <https://bugs.llvm.org/>

- バグを追跡管理
 - チケット=1つのバグと，関連する投稿をまとめる
- バグ以外を扱う事もある
 - 機能追加要望（feature request），質問
- 基本機能：類似バグの検索
 - 重複するバグ報告は開発者の負担となる
- あると便利：バージョン管理システムとの連携
 - バグが修正されたらチケットを自動で閉じたい
 - 修正が取り込まれたバージョンを自動的に追跡したい

- Bugzilla: MozillaによるOSS
- Redmine: プロジェクト管理のためのOSS
- Launchpad: Ubuntuの開発管理システム
- GitHub Issues: GitHubの標準機能
- JIRA: Atlassian社の製品
- FogBugz: Fog Creek Software社の製品
 - 『ジョエル・オン・ソフトウェア』は超おすすめ！

- GitHubが提供するBTS
- コミットメッセージやPull Requestなどと連携

チケット



The screenshot shows a GitHub Issues page with the following details:

- Summary: 330 Open, 8,887 Closed
- Filters: Author, Label, Pro
- Issue 1: vuejs/vue-ssr-webpack-plugin and webpack 5 (feat:ssr, feature request)
- Issue 2: Warn about `this.` used in templates (discussion, feat:compiler, feature request)
- Issue 3: \$scopedSlots is empty until render() (bug, has PR)

チケットの種類ラベル

Vue.jsのIssues <https://github.com/vuejs/vue/issues>

普段使っているOSSに関する
バグ報告をいくつか探せ

- OSSにより使っているBTSが異なる
 - お気に入りのOSSはどのBTSを使っているだろうか
- 制限時間9分

- 良いバグ報告と悪いバグ報告とは何だろうか

良いバグ報告

- バグの内容が簡潔・正確に書いてある
- 再現性が高いシンプルな再現手順がある
 - バグを直すにはバグを再現できることが必須
- バグの影響度が書いてある
 - 広い範囲に甚大な影響があると修正対象になりやすい

バグの概要

発生する問題と その原因

再現するバージョン

再現手順（リンク）

Bug Description

mdadm 3.3-2ubuntu7.6 (for xenial) has a bug which may break an internal write-intent bitmap.

Due to this bug, a bitmap overlaps with a bbl (bad block log), then the bbl gets to be inconsistent.

The cause of the bug is that a bbl is placed at a fixed location (80 sectors from the beginning of a disk).

Because an internal write-intent bitmap is placed before the bbl (16 sectors from the beginning of a disk), it overlaps with the bbl when it grows too large (> 64 sectors).

See this figure. https://cdn-ak.f.st-hatena.com/images/fotolife/u/uchan_nos/20171126/20171126111715.png

This bug has been fixed in upstream of mdadm. The fixing commit is <https://github.com/neilbrown/mdadm/commit/1b7eb672f7792313cc1517feaae8267575fc496b>

I tested the bug on the following versions:

- mdadm 3.3-2ubuntu7.6 (for xenial) : The bug is reproduced
- mdadm 3.4-4ubuntu0.1 (for zesty) : The bug is reproduced
- mdadm 4.0-2 (for artful) : The bug has been fixed

I want you to backport the patch to Ubuntu 16.04 LTS because the bug is serious.

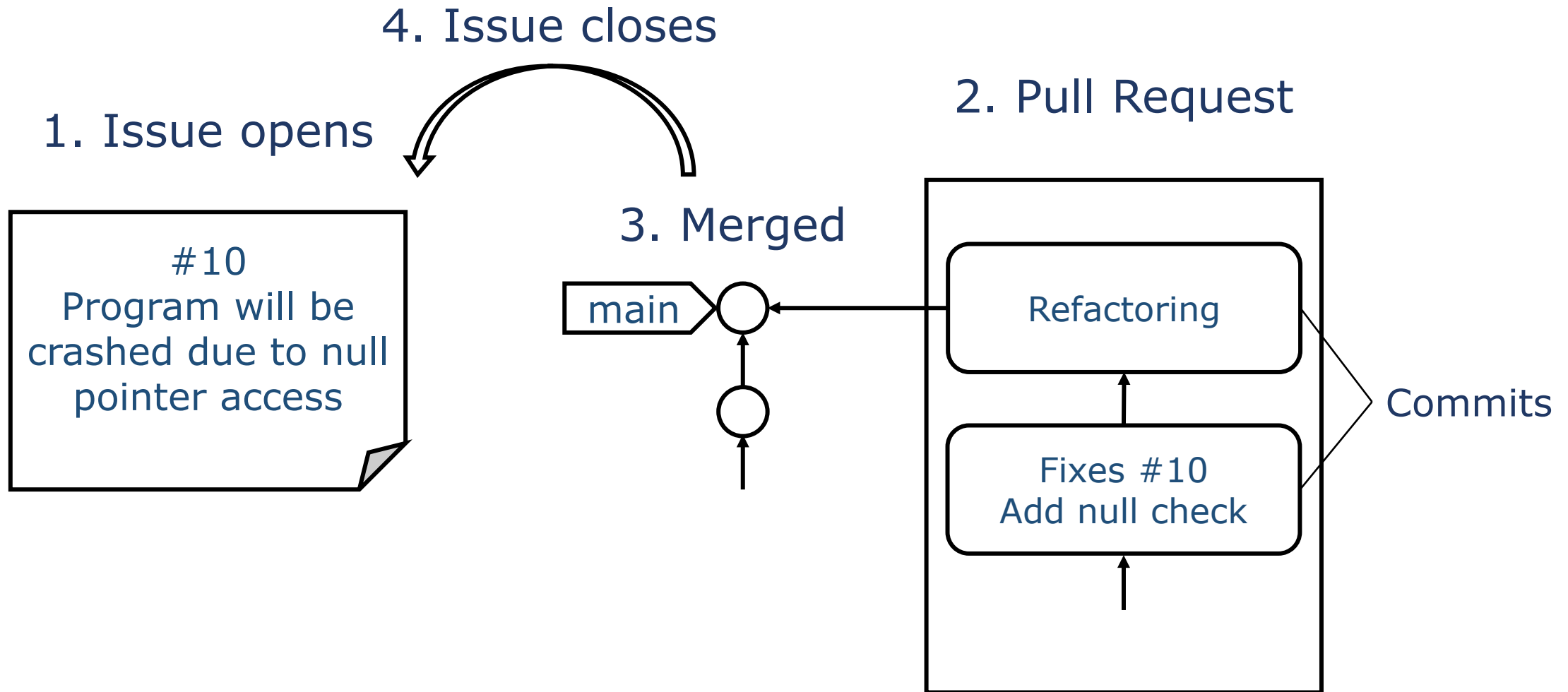
Detailed description (containing scripts to reproduce the bug) is here: https://gist.github.com/uchan-nos/4a4894f3c8e4a59ed0b870b5a516dcad#file-mdadm_bug_bbl_disruption-md

良いバグ報告と悪いバグ報告を探せ

- お気に入りのOSSに関する物でも良いし，そうでなくても良い
- バグ報告の良し悪しに影響する要素は何だろう
- 制限時間15分

- GitHub Issuesのチケットとコミットを連携できる
 - 「プルリクエストをIssueにリンクする」
 - <https://docs.github.com/ja/free-pro-team@latest/github/managing-your-work-on-github/linking-a-pull-request-to-an-issue>
- コミットにキーワードを入れてプルリクを出す
 - Closes #10
 - Fixes uchan-nos/titech-sysdev-2020#3
- mainリポジトリにマージされるとチケットが閉じる

GitHub Issuesのチケットとコミットの連携 2/2



- Ticket-driven development
- タスクをチケットに割り当てて開発を進める方式
- BTSの機能を活用してタスク管理できる
 - 個人へのタスク割り当て
 - タスク完了・未完了のリスト化
 - タスクに関する相談
 - GitHub Issuesなら、タスクを実現したPRを自動追跡可能
- スクラム開発の各種バックログと相性が良いかも
 - PBL (Product Backlog)やSBL (Sprint backlog)の各アイテムをチケットとして表現

自分のOSSに対しIssueを投稿し, 修正コミットを投げてIssueを自動で閉じよ

- リアルなバグに関するIssueでも良いし, 機能追加チケットでも良い
- 使えるキーワード: Fixes, closes, resolves, 解決, etc.
- 制限時間45分



Tokyo Tech

後半は GitHub & Pull Request