



Tokyo Tech

情報収集

2020年10月2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

- 課題を含めたりポジトリとレポートを作成し, 提出
- またあとで詳しく説明します



Tokyo Tech

情報収集

文字`#'のASCIIコードを調べよ

- どうにかして調べる。
 - 一番低コストな方法は何だろう？
- 制限時間2分

- 調べ方の例 : man ascii
 - Linuxのコマンド
 - 一瞬で調べられる。ネットワーク通信も発生しない。

```
$ man ascii
```

040	32	20	SPACE
041	33	21	!
042	34	22	"
043	35	23	#
044	36	24	\$
045	37	25	%

- 情報収集はシステム開発の超基本
- 自分なりの情報収集源を揃えていこう

- ヘルプコマンド (man, info, tldrなど)
 - デモ man expr, man strtol, man write
- 公式ドキュメント (仕様書, Getting Startedなど)
- 書籍
- 質問サイト (Stack Overflowなど)
 - デモ <https://stackoverflow.com/questions/3221812/how-to-sum-up-elements-of-a-c-vector>
- ブログ (Qiita, はてなブログなど)
- Gitのコミットコメント, Issue
 - デモ <https://github.com/torvalds/linux/commit/99f62a746066fa436aa15d4606a538569540db08>

tarでディレクトリabcを圧縮し,
abc.tar.gzというファイルを作る
コマンドラインを調べよ

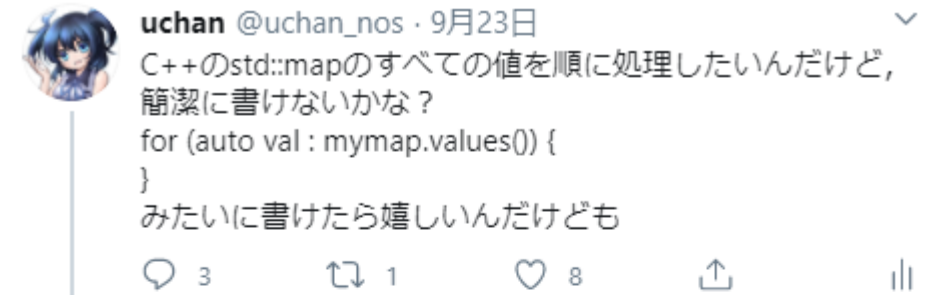
- ヒント : tar X
 - Xは空白を含み18文字
- 制限時間10分

- 答え : `tar czf abc.tar.gz abc`
- manは強力なツールだが、役に立たないこともある
- また、特殊な事例は非常に調べにくい
 - 「良くある事例」の記事が相対的に多い
- 優秀な情報収集ツール「経験者に聞く」

皆さんが良く使う情報収集ツール
や手法を互いに紹介しよう

- 数人のグループに分かれて話します
- 自己紹介タイムも兼ねています
- 最後にグループの1人がまとめて発表します
- 議論20分+発表10分

- Googleで検索する
 - その結果Stack Overflowやブログがヒットすることが多い
- その筋のサイトで探す
 - C++の規格ならcpprefjp
- 仕様書をあたる
 - x86-64の仕様ならIntel SDM
- Twitterで質問する



返信先: @uchan_nosさん

できますよ



午前3:17 · 2020年9月24日 · TweetDeck

- 逆説的だが、情報を発信するところに情報が集まる
- その心は：関連情報が集まる、ユーザーが集まる、間違いを正してくれる、など
- 一例：私が開講している「fuchan C++講習会」
- 私のC++スキルが低いのに耐えかねた猛者たちからC++の最新機能を教えてもらえる

- 特殊な事例，ニッチな分野，最先端の調査では情報収集ツールは役立たないかも
- 自分の手を動かして調べよう！
- システムに負荷をかけ，ボトルネックを把握する
- 実際にAPIを叩いて結果を見る
- コンパイル後のバイナリを調べる
- などなど...

- コマンドを実行したり，調べものをする過程を記録する
- 記録を取らないと...
 - 何をしたか忘れ同じことを繰り返す
 - 環境を変えたことを忘れる
 - 調査結果を忘れ，再度実験する羽目になる
 - 編み出した便利なコマンドを忘れる
- 講義では，皆さんのGitHubリポジトリのIssuesに作業を記録してください（成績評価対象です）
 - サンプル <https://github.com/oss-gate/workshop/issues/880>

次のプログラムについて
最適化しない場合とした場合を比べ、
最適化の影響を調べよ

- コラッツの予想：
任意の初期値に対し、
有限回の操作で1になる
- 制限時間10分
- 作業記録を取る

```
int collatz(int n) {  
    if (n == 1) return 1;  
    if (n%2) return collatz(3*n + 1);  
    else      return collatz(n/2);  
}
```

- 最適化により大きく変わる
- -O3最適化の結果, 単に1を返す関数に

```
collatz(int):  
    mov     eax, 1  
    ret
```

- Compiler Explorer
 - <https://godbolt.org/z/9PPqcz>
 - 手軽にコンパイラの出力を確認できる

作業記録を取りながら，自分のOSSプロジェクトに関する情報収集をしよう

- 自分のOSSが無ければ何か作ってください
 - 次回の授業のことを考えるとC/C++で作るのがおススメ
- 情報収集はなんでもいいです。例えば...
- 必要なAPIの使い方を調べる
- 性能のボトルネックを探す

- 課題を含めたご自身のリポジトリとレポートを提出
- 提出先は内田のGitHubリポジトリ
 - <https://github.com/uchan-nos/titech-sysdev-2020>
 - プライベートリポジトリのためアカウント登録必須
皆さんのGitHubアカウントを教えてください
- このリポジトリに対し、レポートを送る
 - レポートには、トピックに対する回答を含める
- 提出期限は講義の1週間後の10:00 (JST)
 - 「情報収集」の課題提出期限は10/09 (金) 10:00

- GitHubアカウントを内田に教える作業（初回のみ）
- 情報を uchida.k.af あつと m.titech.ac.jp に送る
- メールに含める内容
 - GitHubアカウント名
 - 本名
 - 学籍番号
- 今作業してください
- ここで、皆さんからアカウント情報が来るのを待つ

1. 独自のブランチを作る

1. titech-sysdev-2020:master

↓ branch

titech-sysdev-2020:report-YOUR_NAME

2. 回答の概要をまとめたファイルを加える

1. titech-sysdev-2020/reports/TOPIC/YOUR_NAME.md

2. Commit & Push

3. プルリクを送る (リポジトリ内プルリク)

1. titech-sysdev-2020:report-YOUR_NAME

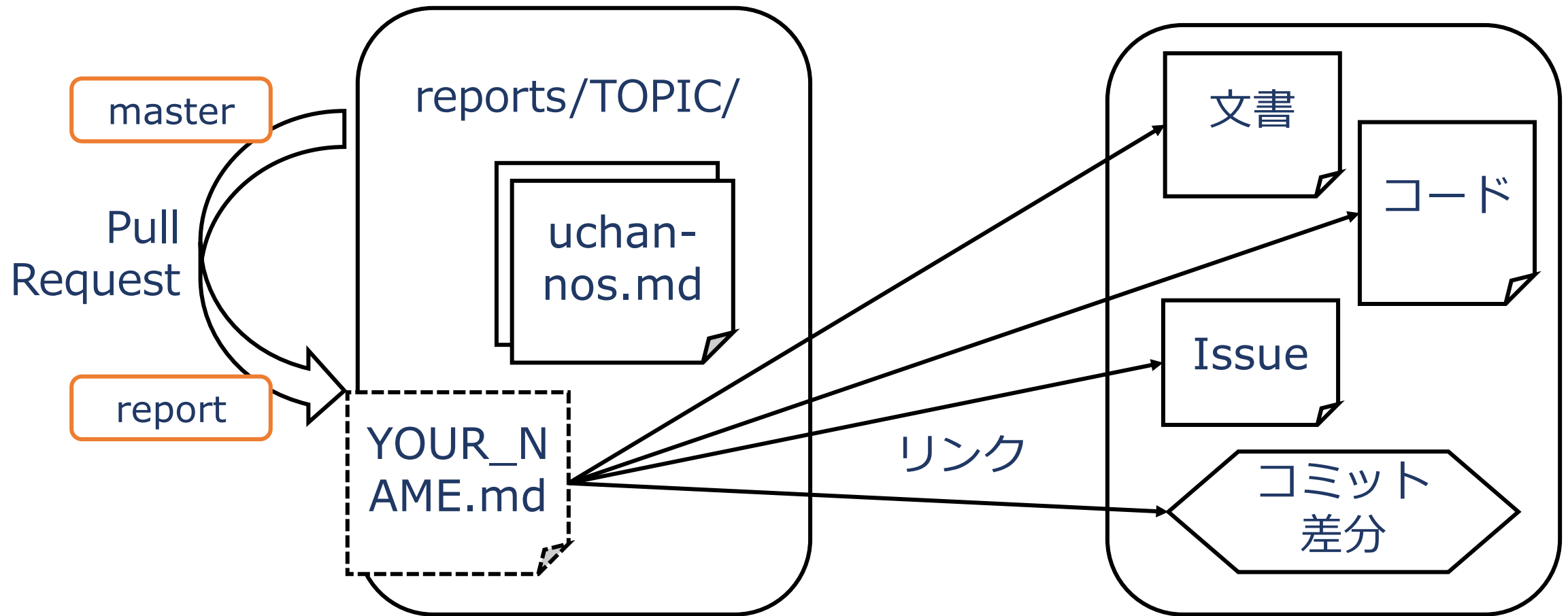
↓ pull request

titech-sysdev-2020:master

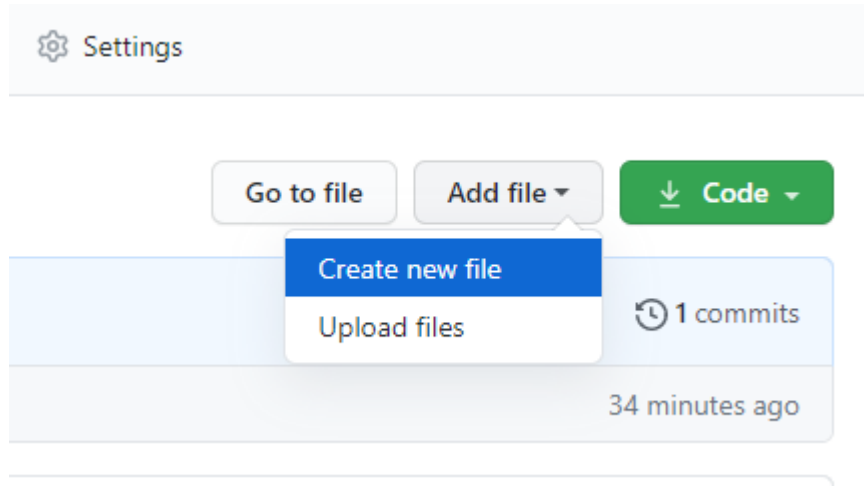
- reports/TOPIC/YOUR_NAME.md
- このファイルに課題への回答を記載する
- 必要なら以下のものを含める
 - Issueへのリンク
 - コミット差分へのリンク
[https://github.com/HOGE/REPO/
compare/COMMIT1...COMMIT2](https://github.com/HOGE/REPO/compare/COMMIT1...COMMIT2)
 - その他
- 要するに、成績評価に必要な情報をYOUR_NAME.md自体に記載するか、そこから辿れるようにする

uchan-nos/titech-sysdev-2020

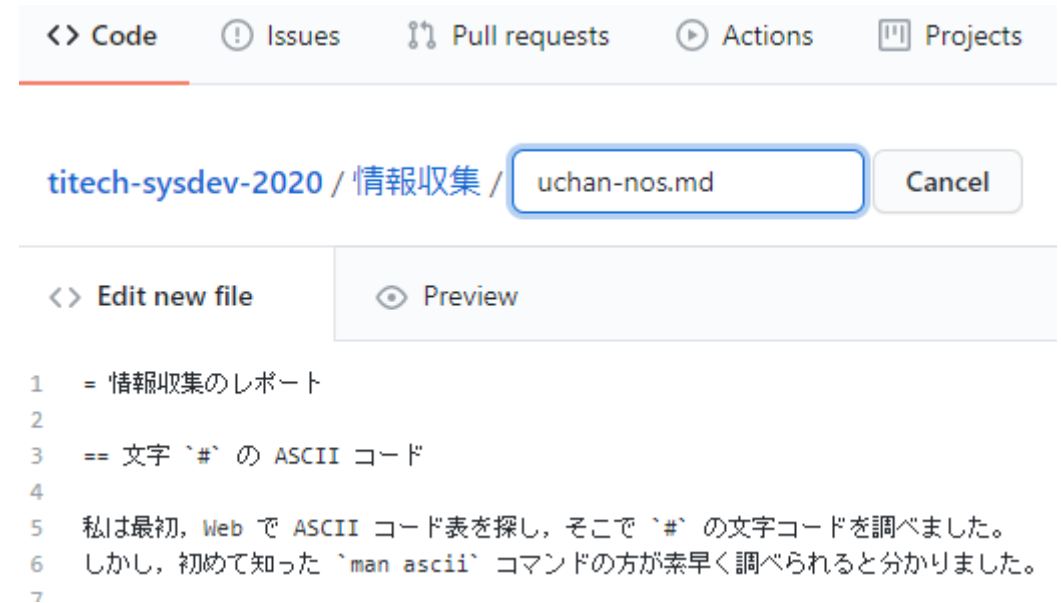
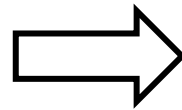
受講者のリポジトリ



レポートの送り方 1/2



ファイルを新規作成



YOUR_NAME.mdの内容を記述

レポートの送り方 2/2

- ☐ Commit directly to the `master` branch.
 - ☒ Create a new branch for this commit and start a pull request. [Learn more](#)
-

Propose new file

Cancel





新規ブランチにコミット

プルリクを作成

Open a pull request

The change you just made was written to a new branch named `report-uchan-nos`. Create a pull request to








 base: master < compare: report-uchan-nos > ✓ Able to merge. These branches can be merged.




レポート提出 uchan-nos

Write

Preview

H B I  <>     @  

「情報収集」に関するレポートを提出します

Attach files by dragging & dropping, selecting or pasting them. 

Create pull request

レポートの送り方 2/2

- ☐ Commit directly to the `master` branch.
- ☒ Create a new branch for this commit and start a pull request. [Learn more](#)

report-uchan-nos

Propose new file

Cancel

新規ブランチにコミット

プルリクを作成

Open a pull request

The change you just made was written to a new branch named `report-uchan-nos`. Create a pull request to

base: master

compare: report-uchan-nos

✓ Able to merge. These branches can be merged.

作成したブランチから
masterへのプルリク
となっている！


「情報収集」に関するレポートを提出します


Attach files by dragging & dropping, selecting or pasting them.

Create pull request

レポートの受理

レポート提出 uchan-nos #1


 Open uchan-nos wants to merge 1 commit into master from report-u

 Conversation 0  Commits 1  Checks 0



uchan-nos commented 4 minutes ago

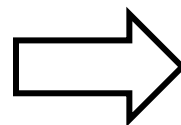
「情報収集」に関するレポートを提出します

 Create uchan-nos.md



uchan-nos commented now

内容が薄いですよ。もっと付け足しませんか？



レポート提出 uchan-nos #1

 Merged uchan-nos merged 1 commit into master from report-u

 Conversation 0  Commits 1  Checks 0



uchan-nos commented 5 minutes ago


「情報収集」に関するレポートを提出します

 Create uchan-nos.md



uchan-nos commented 1 minute ago

内容が薄いですよ。もっと付け足しませんか？

 uchan-nos merged commit 032af9a into master now

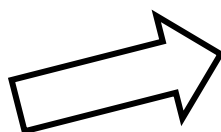
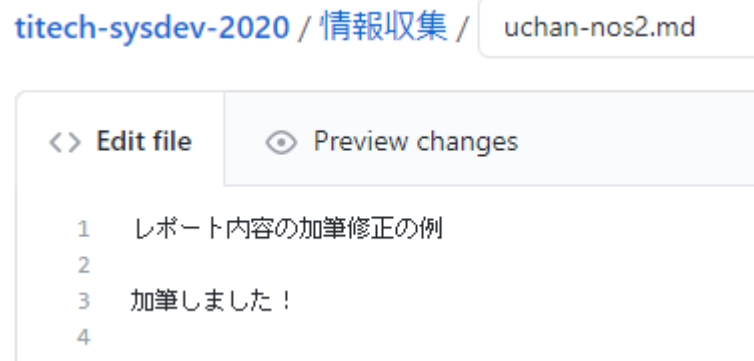
プルリクにコメントが付くことも

Merged : レポート受理済み

レポートの更新 1/2

レポートに不十分な個所があった！

まだマージされていないときの
更新方法を紹介



Commit changes

加筆

Add an optional extended description...

uchan0@gmail.com

Choose which email address to associate with this commit






☒ Commit directly to the `report-uchan-nos` branch.

☐ Create a new branch for this commit and start a pull request. [Learn more](#)

納得いくまで加筆修正

report-Xブランチにコミット

不十分な内容のレポートを作成 #4

 Openuchan-nos wants to merge 2 commits into `master` from `report-uchan-nos`  Conversation 0 Commits 2 Checks 0 Files changed 1

uchan-nos commented 2 minutes ago

これは不十分なレポートなので、まだマージしないでください！



不十分な内容のレポートを作成



加筆



uchan-nos commented now

レポート完成しました。マージして大丈夫です。

一発目のコミット



Pull Requestに
コミットが追加されていく



- GitHubのWebインターフェースを使う必然性はない
- コマンドラインで作業してもよい
 - 具体的なコマンドラインは示しません
 - この講義は「情報収集」でしたね？
 - コマンドラインについて情報収集すれば、レポートをさらに充実させるネタになりますよ

- 次回は10/16（金） 14:20から
 - 10/09（金）は全学で授業は休み
- 「GDBによるデバッグ」を行います
- C/C++で作ったOSSがあると学習効果が高まります