

構成管理, 版管理, IDE, プラグイン

2023/1/16

海谷 治彦

目次

- 構成管理 (Configuration Management)
 - make, ant, Maven, Gradle
- 版管理 (Version Management)
 - rcs, subversion, git
- 統合開発環境 (IDE Integration Development Environment)
 - Eclipse, IntelliJ/IDEA
- プラグイン
 - Eclipse plugin, astah plugin, VS Code plugin
- まとめ

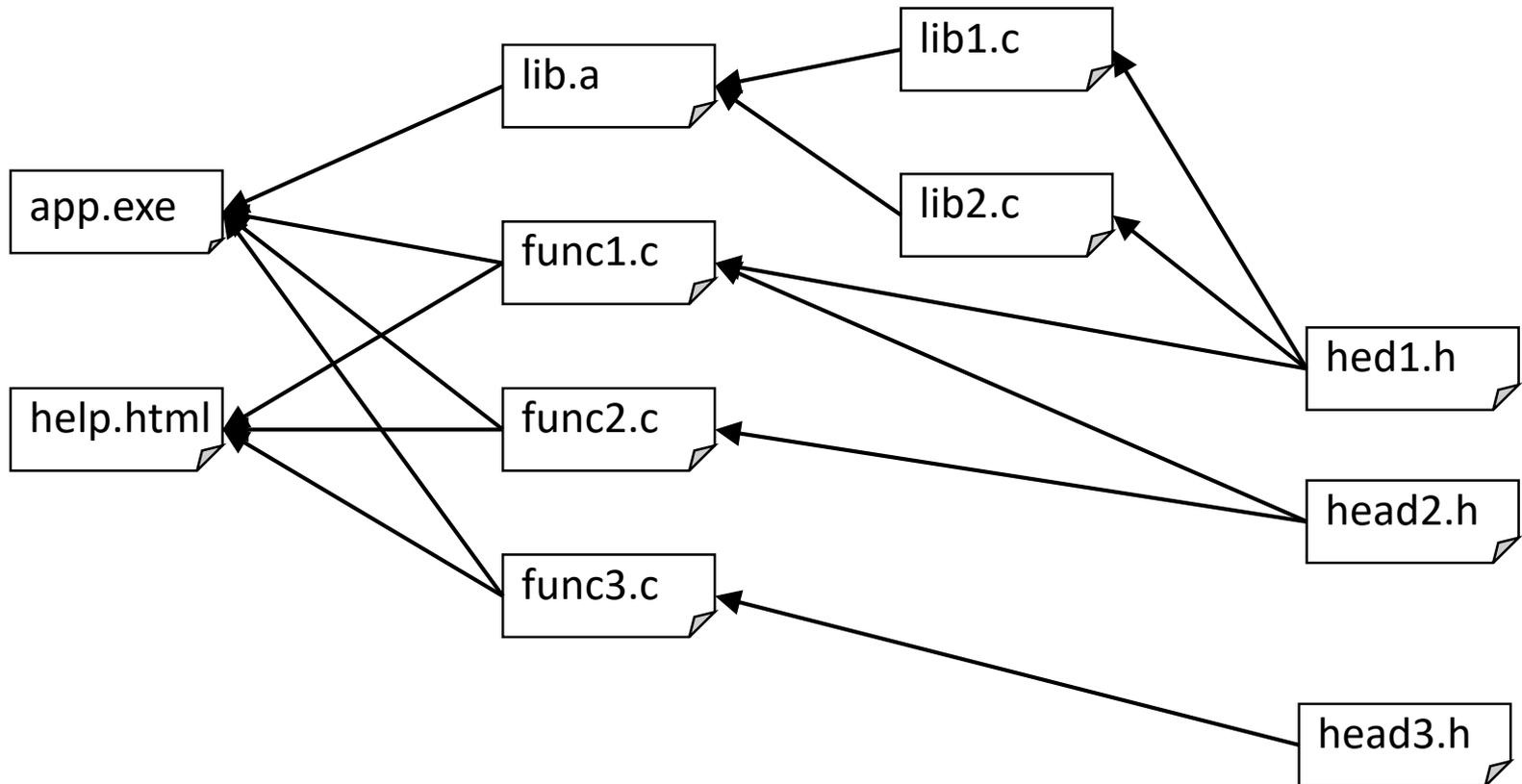
構成管理とは

- 最近のソフトウェアの最終成果物は、多くの他の成果物から生成される。

例

- ロードモジュール: ソースコード ヘッダー ライブラリ
- ヘルプファイル: ソースコードのコメントから...
- このような依存関係を管理し、最終成果物を自動構築するための活動が構成管理。
- 生成だけでなくインストール(配置, Deploy)の管理もする。

例 C のアプリとヘルプ



構成管理のメリット

- 一部直した後に, アウトプットを作り直すのを自動でできる.
- 特に, 影響がある部分のみを作り直す効率化が可能となる.
- 作り直しだけでなく, 再インストールも系統的に行うことができる.

構成管理ツール

- 構成管理を自動化するためのツール
- 歴史も長く種類も多い.

- make 太古からあるツール
- ant XMLベースで管理設定を書くツール, 下火.
- Maven 広く使われている.
- Gradle 新しい目.

make

- UNIX/Linux系で利用される構成管理ツール
- 40年くらい前からある太古のツール
- Makefile というファイルに構成を書き, それが行実行される.
- 実は今でもわりと目にする.
 - C言語系の開発では現役かも.
- 実演ムービー `make.mp4` (9分) 後で見て
 - winアプリのビルド, unzipのビルド

ant

- Apache Ant
- Java のための make みたいな感じだが, Java 専用というわけでもない.
- 設定は XML で書く (下記が例 build.xml)
- 今は, ほぼ Maven に置き換えられてる感じ.

```
<?xml version="1.0" ?>

<project name="bulid" default="jar" basedir=". ">
  <!-- name of jar file -->
  <property name="jar_name" value="ex1build2.jar"/>

  <target name="jar" depends="compile">
    <jar jarfile="{jar_name}" basedir="bin" includes="**/*.class" manifest="manifest.txt"/>
  </target>

  <target name="compile">
    <javac srcdir="src" destdir="bin" includeAntRuntime="yes" verbose="yes"/>
  </target>

</project>
```

Maven

- Apache Maven
- 今は ant にかわり広く使われている.
- 設定ファイルは pom.xml

- ムービー maven.mp4 (20分)
 - astah のプラグイン開発の例の紹介も付属

Gradle

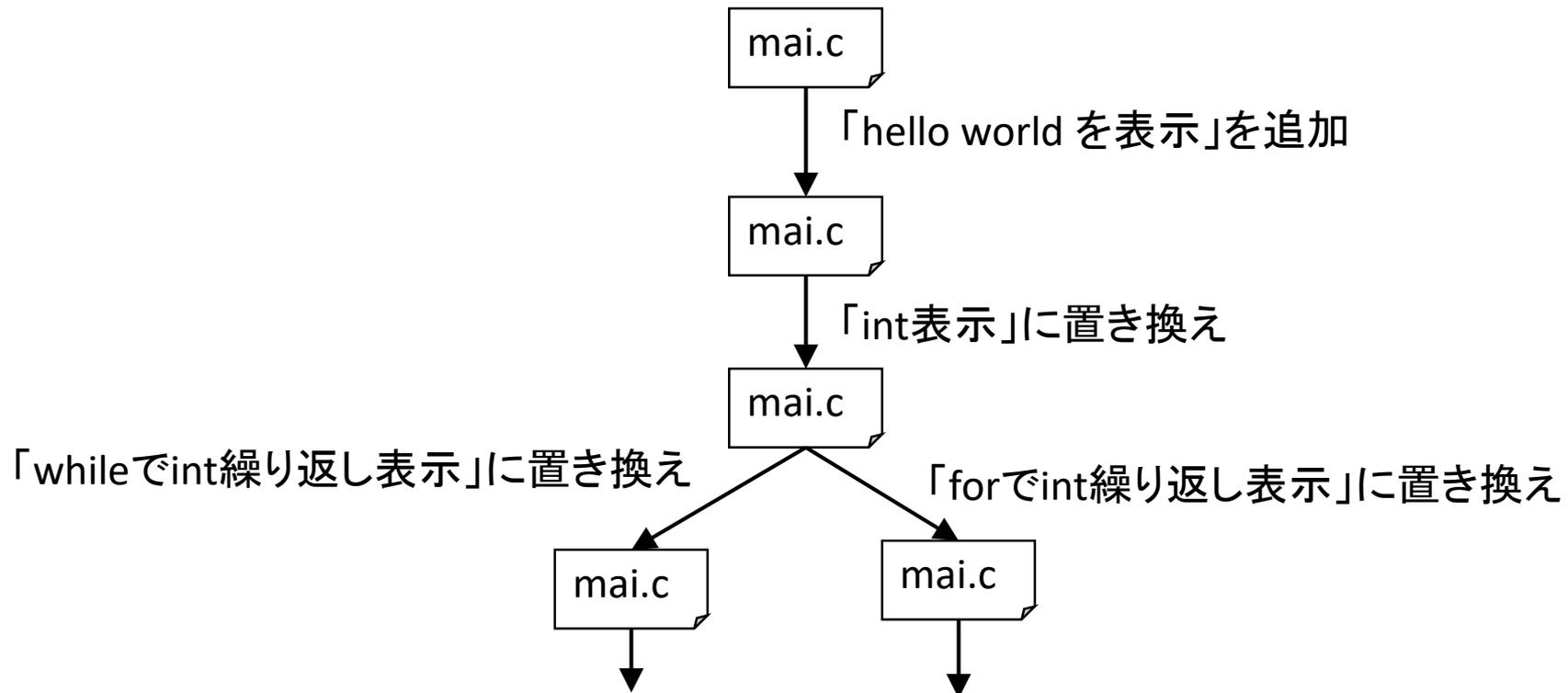
- やはり, 構成管理ツール
- XMLではなく, プログラミング言語 Groovy で設定を書く.
- 比較的, 旬なもの.

版管理とは

- 版 バージョン Version
- ソースコードやマニュアル等は更新を重ねてゆく.
- しかも, リアルには複数人間が, 並行して更新してゆく.
- バリエーションが出てくるかもしれない.
- ソフトウェアにかかわる成果物のバージョン管理を管理するツールが多数, 存在し使われている.

版管理のイメージ例

- ちょっと毎, 直してゆく作業過程を想定する.
- 途中で複数のアイデアが出て, バリエーションが出てくるかもしれない.



RCS

- 古代の版管理システム
- Revision Control System
- いくつかのコマンド(ci, co 等)を使って, ソースコード等の過去のバージョンを保存し, 取り出すシステム.
- 過去バージョンは, 丸コピーではなく, 差分を持っている.
- ネットワーク対応してない.
- CVS (Concurrent Versions System)に引き継がれているが, CVSの利用自体, あまり見かけなくなった.

subversion

- Apache Subversion
- 略称およびコマンド svn

- ネットからの紹介程度
- 今は完全にgitに押されてる

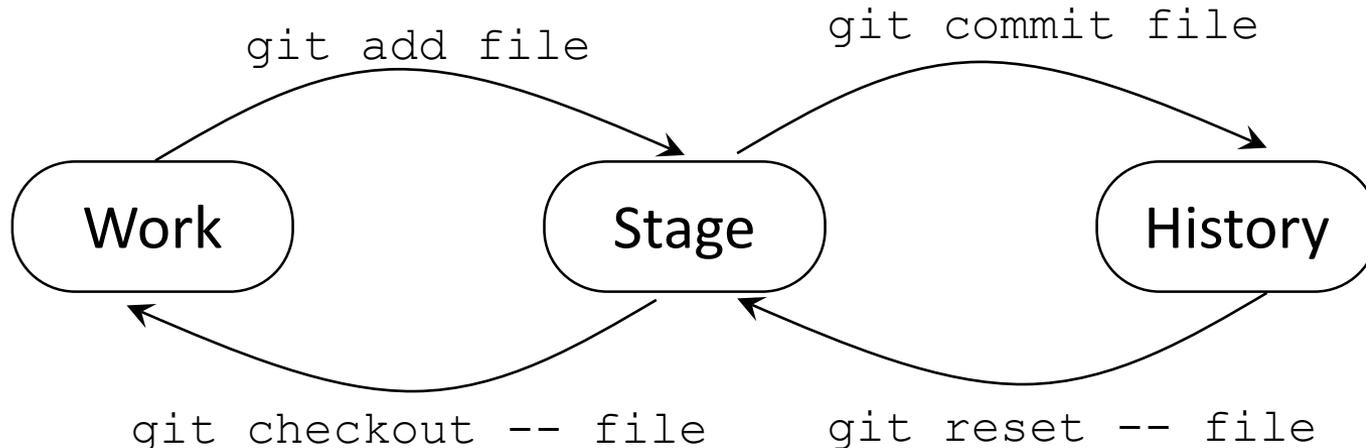
git

- 今, もっとも一般的な版管理ツール
- 基本, ローカルなPC内に repository というバージョンを保存する履歴(History)を持つ.
- 履歴の列を branch (枝)と呼ぶ.
- 枝分かれを許し, バリエーションを保存できる.
- 合流もできる.
- デフォルトの branch は, master という名前が普通.

- ネットワーク上の repository を管理するシステム (github, Bitbucket 等)と連携可能.

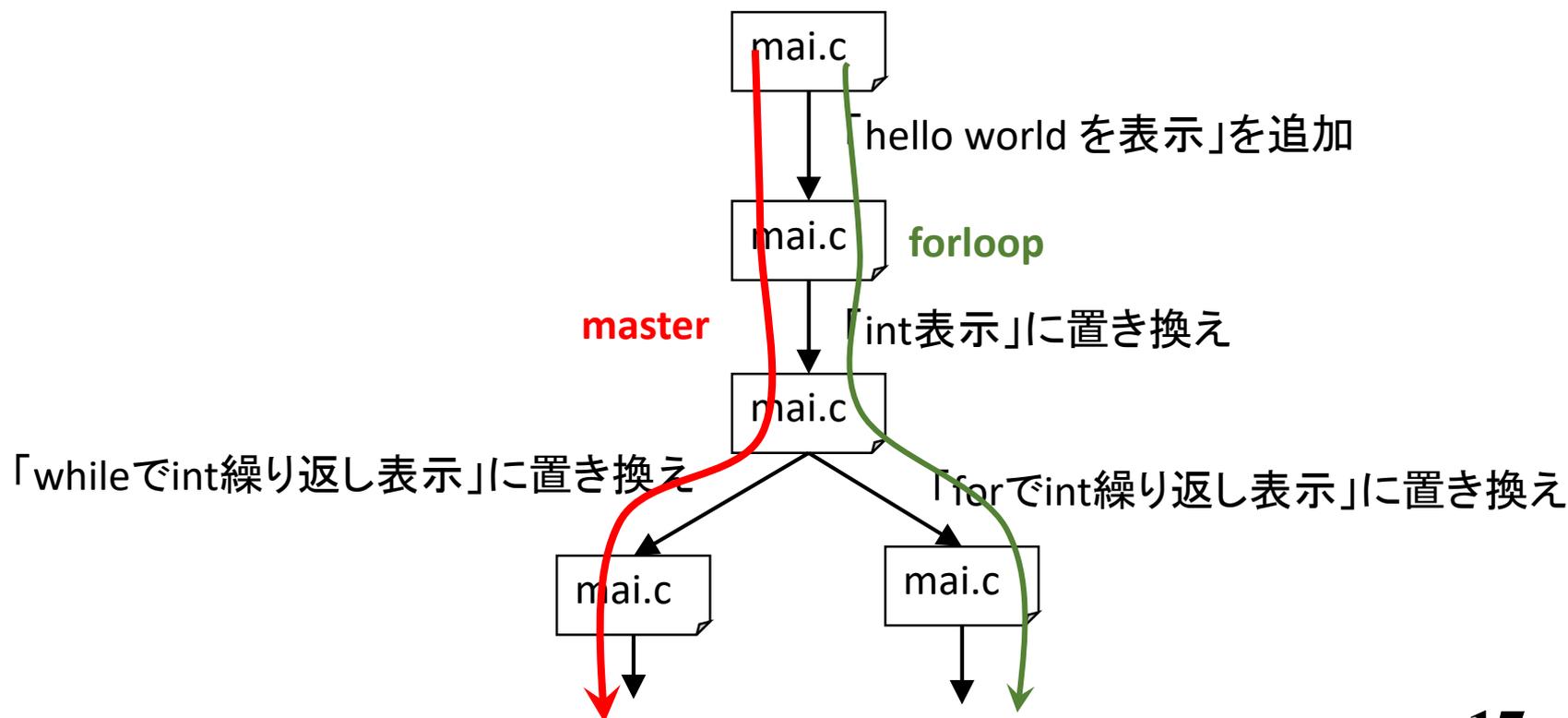
gitでのファイル管理

- git ではファイル等は, Working, Stage, History の3つの状態で管理されている.
 - Working 編集中状態, ココのファイルを編集する.
 - Stage 仮登録のような状態.
 - History 正式に履歴として登録される.
- 以下のようなコマンドで file の状態を変える.



branch

- git branch ブランチ名で作成できる.
- git checkout ブランチ名で切り替えられる.
- git branch で一覧を出せる.



git の インストール

- Windowsの場合
 - 公式サイトからインストーラーをDLしていただけます。
 - git download windows で検索。
- Ubuntu on WSL2の場合
 - 多分, もとから入ってます。
 - なければ, `sudo apt install git` で入れて下さい。

github

- gitで管理されたソース群等をネットで公開, 共有できるシステム.
- 私企業が運営している.
- 全公開するソース群をアップする分には無料.
 - 会社等で非公開で運用する場合は料金がかかる.
- 基本, リポジトリという単位でプロジェクトを管理.

githubのアカウント作成

- 前準備
 - ssh-keygen で公開鍵と秘密鍵を作成.
- jinidaiメールで作成
 - アカウント作成
 - リポジトリの作成
 - 公開鍵を github に登録
- ビデオ 22 分 gitsetup.mp4 webclass にあります.
- ビデオ 5分 gitsetup2.mp4 ちよい忘れてたこと.

gitとgithubの操作 コマンドを利用

- コマンドを利用してgitとgithubを利用.
- ローカルなりポジトリを作成.
- githubにローカルなりポジトリの中身をアップロード
- 23分 gitcmd.mp4 webclass にあります.

gitとgithubの操作 vscodeより

- vscodeからgitとgithubを利用.
 - ローカルなりポジトリを作成.
 - githubにローカルなりポジトリの中身をアップロード
-
- 27分 gitvscode.mp4

古代と現代の違い

- 古代(makeやrcs)では、管理対象がローカルファイルだけだった.
- また、グループ管理もあまり想定してなかった.
- 現代では、管理対象がネットワーク上に分散しており、作業者も分散している.

- 構成管理において、構築および配置(インストール deploy)を対象とするのは変わらない.
- 複数の枝分かれを許容する等の版管理も変わらない.

IDE とは

- Integrated Development Environment
- 統合開発環境
- ソフトの編集, コンパイル, テスト, インストール, 版管理, 構成管理等を一手に引き受けるツール.
- コレを使うと, ちまちま, emacs 編集, gcc コンパイル等やっていたのがバカらしくなる.
- 大抵, 非常に重いソフト.
- Visual Studio も有名は例. (not VScode)

Eclipse

- 代表的なIDE
- もとは Java 用だったが, 今は多様な言語に対応.
- 昨今, 機能つめこみ過ぎで壊れてきたという風評もアリ.

IntelliJ IDEA

- JetBrains という会社が開発している IDE
- 有料版と無料版がある.
- どうも, 今は Eclipse より流行っているっぽい.
- インストールはEclipseより簡単.

プラグイン

- Plugin
- アプリに対して機能追加を再コンパイル等無しで行える仕組み.
- アプリがフレームワーク的にできているため, 可能な場合が多い.

例

- ブラウザの拡張機能
- Eclipse等IDEの拡張機能
- astah UMLエディタのプラグイン (maven.mp4 参照)
- VS Code の拡張機能 extensions
- 実は emacs も emacs lisp で記述された拡張機能が多数ある.

まとめ

- いまどき、コマンドで `emacs gcc/javac` 実行というのは少ない。
- IDEを中心に、各種のツールを使うのが普通。
- ツールは瞬く間に変わっていくが、わりとやってることの根本はかわらない。
 - 根本:
 - 人が業務を見て手作業で仕様書を作る。
 - 人が仕様書を読んで、手作業でコードを作る。
- ソフトウェア工学は単純作業の自動化に注視。
- 現代の実用的なソフト開発は、ネットワーク接続され、多数のブラックボックスを使い、はっきりいって混沌。
- 大学で行うテキストエディタ+コンパイラという作業は、単に言語を学ぶための作業で、本当の開発とはいえない。

演習5

- vscode で簡単なCプログラムを複数バージョン作ってください。
 - hello world と画面表示, 数値を表示, ループを実行程度でOKです.
 - フォルダ名は se2022ex5 をお願いします.
- 上記の複数バージョンをgitでローカルに管理してください.
- それを github のrepositoryにアップしてください。
 - github でのユーザー登録は, なるだけ jindai.jp のメールアドレスを用いてください.
- アップした先のURLを提出してください。
 - git と github の連携ができた証拠となります.
 - URLはテキストファイル url.txt にいれて提出して下さい.
 - **ワードやPDF, Pages は不可です.**
- 提出先 webclass 演習5 ✕切 1月末

複数バージョンの例

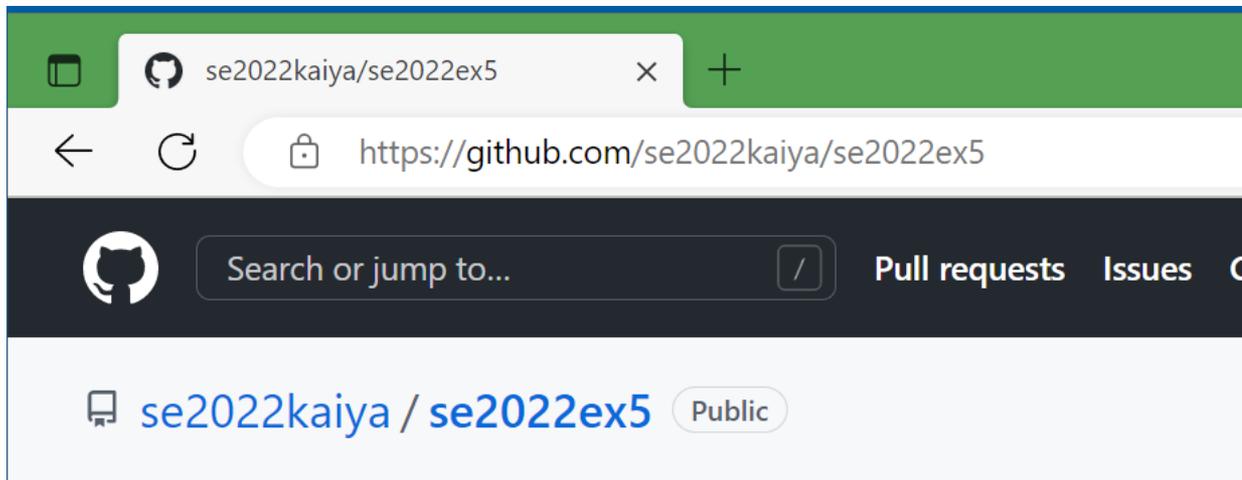
- 考えるのが面倒なら下記を使ってください.
 - ファイル名は main.c でも ex5.c でも好きでOKです.
- 1. printf("hello") 等で直接に文字列を表示.
- 2. char str[]="hello"; 等, 一旦変数に入れてそれを表示.
- 3. 文字列自体と, その長さを両方表示.
 - strlen を使ってもよいし, ループで数えてもよい.

提出するURLの取得

- 右上メニューから Your repositories
- se2022ex5のリポジトリをクリック
- ブラウザに表示されるアドレスをコピーして送ってください, 以下の例だと,

<https://github.com/se2022kaiya/se2022ex5>

ユーザー名が se2022kaiya の場合



演習6

- この講義シリーズ全体の感想文を提出してください。
 - まだ次回があるけど.
- ワード等でA4用紙で1枚以内でお願いします.
- PDFかdocx形式でお願いします.
 - pages (マックのワープロ)は不可です.
- 提出先 webclass 演習6
- ✕切 1月末

以上