

# プログラミング応用第2回 バージョン管理システムの基礎(1)

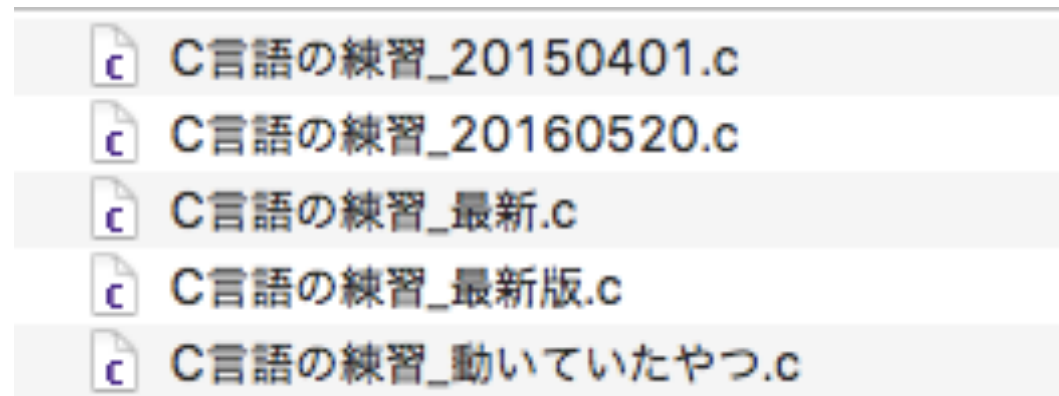
4月14日

# バージョン管理の概要

- バージョン管理
  - ファイルの変更履歴(バージョン)を記録すること
- バージョン管理の利点
  - プログラムが動かなくなっても動いていた状態に戻すことが出来る
  - ソースコードの配布が容易になる
  - 共同開発者との共同作業が容易になる
    - 現在の商業開発ではバージョン管理システムの導入はmust

# 補足：従来のバージョン管理

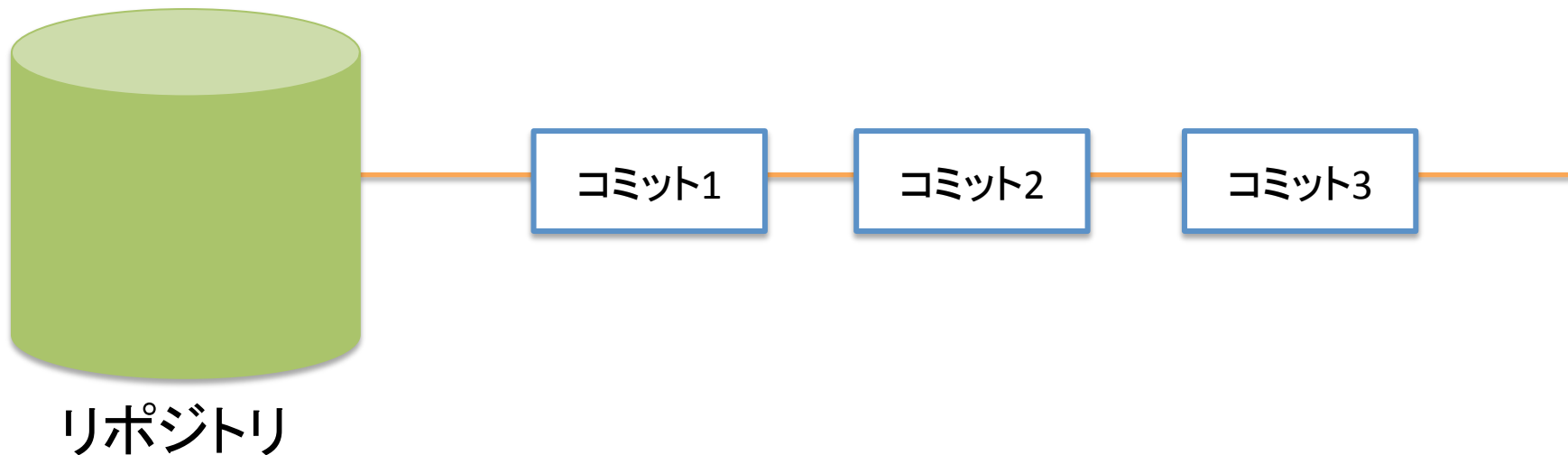
- バージョン管理を用いない変更履歴管理
  - たとえばファイル名に日付を入れたりする



- この場合変更履歴が増えたと管理が困難

# バージョン管理システムによる管理

- “リポジトリ”に変更履歴を記録していく
  - ステージング = バージョン管理するファイルを指定
  - コミット = ステージングしたファイルの変更を記録



- コミットすると変更内容、変更時間、コメント等が自動記録される

# Gitによるバージョン管理

- ステージング
  - 基本のコマンド  
`$ git add ファイル名`
  - すべてのファイルを管理する場合  
`$ git add -A`
- コミット
  - 基本のコマンド  
`$ git commit -m “なにかコメント”`
  - \* コメントにはあとから見ても分かりやすくなるようなコメントを書く  
例) 課題2完了  
例) printf()の記述ミスを修正

# 演習0

- Gitリポジトリのセットアップ
  - (終わっていない場合)先週の演習1, 演習2を行う

# 演習1

- バージョン管理の利点をまとめる  
(100字程度)
  1. c-programming-practice-(学籍番号)以下にweek2というディレクトリを作成
  2. week2以下に「gitの利点.txt」というファイルを作成
  3. テキストエディタで利点を100字程度で記述
    - バージョン管理システムを使わない場合と比較して書くと良い
  4. GitHubに提出

```
$ git add -A
$ git commit -m “第2週課題提出”
$ git push origin master
```

- 早く終わった人
  - 先週の課題のやり残しがあればやっておく
  - 周りの人に教えてあげる
- 次週はGitを複数人での共同開発に使う方法について学習します