被験者用資料

概要

目的

本実験の目的は，普段 git を利用しない人に対して，バージョン管理の利便さを感じて貰う事と git コマンド操作の補助です．

この実験では，後の検証のためにコマンドの操作ログを記録します．

やること

こちらで用意したツールを利用し，C 言語による簡単なプログラムの実装，そして，git 操作を行います．

補足

なるべく初心者でもソースコードを読めるように，ポインタなどはなるべく使わない書き方をしています．

テンプレートのプログラムの書き方に従う必要はありません．より良い方法があれば，自由に書き換えてください．

なお，操作ログは対象プロジェクトのディレクトリに入っている時のみ記録されます．記録する操作ログは，シェルのコマンド履歴とその実行時間です．

準備

実験に取り組む前に準備として，(1)実験で利用するサポートツール kani のインストール，(2)対象プロジェクトのクローンの 2 つの作業を行ってください．それぞれ，以下に詳細を述べます．順番に実行してください．

(1) 実験で利用するサポートツール kani のインストール

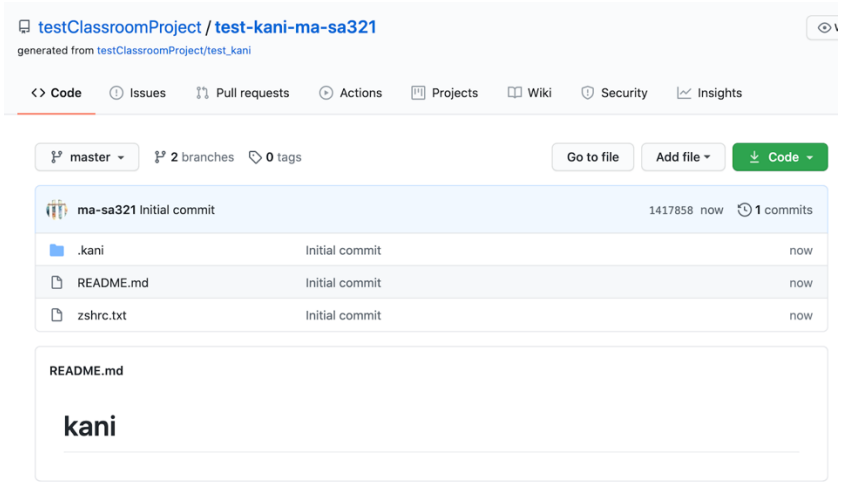
kani は Homebrew でインストールします．Homebrew がインストールされていない場合は，https://brew.sh/index_ja を参照し，インストールしてください．インストール後，次のコマンドをターミナル上で入力し，kani をインストールしてください．

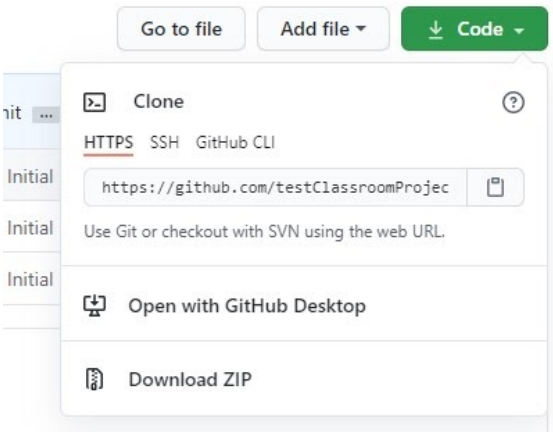
```
brew tap tamadalab/brew
brew install kani
echo 'eval "$(git kani init -)"'>> ~/.zshrc
```

(2)対象プロジェクトのクローン

1. GitHub アカウントなどの準備(今回は割愛します).

2. 招待リンクを踏む，利用してもらうリポジトリが生成されます.
(今回はこのリンク．押すと即座にリポジトリが生成されます.
https://classroom.github.com/a/oOLCBvtY)
リポジトリ名が tmdlabtest- (自身の GitHub アカウント名)になっていることを確認してください.
3-5 は clone して main ブランチにいるかどうかチェックするだけなので，分かる方は読み飛ばしてください.





3. 2.の Code(緑色のボタン)を押すと https..とリンクが表示されるので，コピーしてください.

4. ターミナルを開き，作業したいディレクトリに移動してください．
   git clone https...(コピーしたもの)を入力し，実行すると自身のローカル環境で作業できるようになります(クローンした後，クローンディレクトリに移動してください)．

5. clone したディレクトリで git branch を打つと，main ブランチに「*」がついていて，現在いる場所が確認できます．今回は main ブランチに居ることを確認してください．

```
git branch
*main
```

6. kani は zsh で動作します. bash を利用されている方はターミナルで「zsh」と入力し，zsh を起動してください．
   すでに zsh で実行中の場合, 一度ウィンドウを立ち上げ直してください，上書きした~/.zshrc に更新がかかります．
   （「source ~/.zshrc」と打つだけだと，更新されません．）

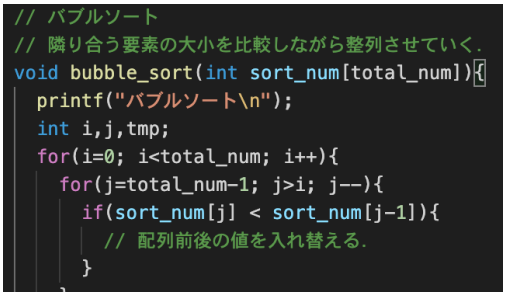
7. clone したリポジトリのあるディレクトリに移動し，「git kani init」を実行してください．kani が作動する対象に設定されます．
   「kani が導入されました！」の表示が出ない場合，作動していないので担当者に連絡してください．
   また，.kani/test.log が生成されていない場合も連絡してください．

テスト内容

概要

いくつかのソートプログラムを実装してください．ソースコードは 2 種類ありますが，どちらも与えられた int 型の配列をソートするプログラムになっています．

今回時間の関係上少しの箇所を穴埋めする形式ですが，全体を始めから書いている感覚で commit してください．



```c
// バブルソート
// 隣り合う要素の大小を比較しながら整列させていく．
void bubble_sort(int sort_num[total_num]){
  printf("バブルソート\n");
  int i,j,tmp;
  for(i=0; i<total_num; i++){
    for(j=total_num-1; j>i; j--){
      if(sort_num[j] < sort_num[j-1]){
        // 配列前後の値を入れ替える．
      }
```

図．参考例

手順

このツールは一定数変更を行うと commit を促します．それに沿って commit を行っていただきます．

もちろん自信が commit すべきタイミングであると判断した場合は，commit してください．

1. 1 つ目の課題は，ツールなしで行います．プロジェクトのディレクトリで，「git kani disable」と入力してツールを無効化してください．
   commit のタイミングはお任せしますが，目安としてデータの取捨選択が出来たら，ソートが出来たら，といったタイミングでしてください．

2. 2 つ目の課題は，ツールを作動させて行います．
   「git kani enable」と打ち，ツールを有効化してください．

3. 3 つ目の課題は，再びツールなしで行います．プロジェクトのディレクトリで，「git kani disable」と入力してツールを無効化してください．

4. これまでのシェルでのコマンド履歴が「.kani/test.log」に記録されています．
   実験の分析に必要になるため，次の手順でコミットしてください．

```
git add ../.kani/test.log
git commit –m "add command histories"
```

5. 「git push origin main」でリモートリポジトリにあげてください．
   master から main に仕様が変更されています，注意してください．

最後にアンケート回答をお願いします！

https://forms.gle/bwwfnxGBhycjTd8a6

ご協力ありがとうございました！

報奨金について

今回学生アルバイトの扱いになっています．
玉田先生に出勤簿を提出後，指定の口座に振り込まれます．
玉田先生から個別に連絡がいきますので，指示に従ってください．

今回利用する kani(仮)について

kani に出来ること

・git 初心者対象の git 操作補助をします．
　add,commit,push の操作を補助する表示が出ます．
・commit のタイミングを提案します．
　commit のタイミングがわからない人だとまとめて commit する事が発生します．その場合 git(バージョン管理)の利点を有効に使えません．特定タイミングで commit を促すことで，頻度をあげることを目指しています．

kani をアンインストールする

次のコマンドでインストールした kani をアンインストールできます．

```
brew uninstall kani
brew untap tamadalab/brew
```

また，~/.config/kani, ~/.zshrc の「eval “$(git kani init -)”」の行を削除してください.

クローンしたプロジェクトを削除する
プロジェクトのディレクトリ自体を削除(ゴミ箱に入れる or rm -rf)してください.