

OpenFOAM コントリビュート活動

@public_inabower^{1†} 松原 大輔² @tkoyama010¹

¹OpenFOAM-jp ² オープン CAE 勉強会

OpenFOAM Contributing Activities

@public_inabower^{*†} Daisuke MATSUBARA^{**} @tkoyama010^{*}

^{*}OpenFOAM-jp ^{**}OpenCAE Local user group

Abstract

The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog. The quick brown fox jumps over the lazy dog.

Keywords: Keyword1, Keyword2, Keyword3, Keyword4, Keyword5

1. はじめに
2. ESI 版と Foundation 版の違い
3. ESI 版コントリビュート方法
4. Foundation 版コントリビュート方法
5. Git の使い方

本節では <https://github.com/OpenFOAM-jp/OpenFOAM-jp.git> のリポジトリにコントリビュートの練習をする方法について解説します。環境は Ubuntu18.04 の環境を想定します。事前に GitHub のアカウントを作成してください。コントリビュートをする際にはまず、Issue を立て自分が加えたい変更について議論します。

機能の簡単な説明**機能リクエストは問題に関連していますか？ 記述してください。** 問題が何であるかの明確で簡潔な説明。

希望するソリューションを説明してください あなたが何をしたいのかについての明確で簡潔な説明。

検討した代替案を説明してください 検討した代替ソリューションまたは機能の明確で簡潔な説明。

追加のコンテキスト 機能リクエストに関する他のコンテキストまたはスクリーンショットをここに追加します。

OSS のコントリビュートのバージョン管理ソフトには Git が一般的に使用されています。まずは、バージョン管理ソフト git をインストールします。

```
$ sudo apt install git
```

必要なのはソースだけで貢献しないのであれば、次のコマンドでクローンすることで済みます。

```
$ git clone https://github.com/OpenFOAM-jp/OpenFOAM-jp.git
```

貢献をしたい場合は、OpenFOAM-jp のリポジトリを自分のアカウントにフォークします。

TODO: フォークの際の画面キャプチャを挿入する。

フォーク後は自分のアカウントのリポジトリをクローンします。

```
$ git clone https://github.com/your_account_name/OpenFOAM-jp.git
```

クローンをしたら自分の環境でテストを実行してください。

TODO: テスト実行のコマンドを記述する。

テストが全てパスしたらソースを変更します。master ブランチで直接変更することはできないため、ファイルを変更する前に開発ブランチを作成してください。

[†] E-mail address of corresponding author: office@opencae.or.jp

```
$ git branch branch_name  
$ git checkout branch_name
```

branch_name には任意の名前を入れてください。自分のアカウント名と加えたい変更について言及されると分かりやすいです。最初のコマンドでブランチを作成し、2 番目のコマンドでブランチに移動します。これにより、変更を行う準備はほぼ完了です。変更のラベルを付けるために、連絡先の名前と電子メールを以下のコマンドで指定します。

```
$ git config --global user.name "Your Name Comes Here"  
$ git config --global user.email you@yourdomain.example.com
```

もし src/toto.cc というファイルをいくつか変更したり、新しいファイルとして追加したら、ローカルのコミットは次のコマンドで行います。"Your extensive commit message here #1" には変更に関するメッセージを追加します。#1 の部分は自分が追加した 이슈の番号としてください。

```
$ git add src/toto.cc  
$ git commit -m "Your extensive commit message here #1"
```

この段階ではコミットはあなたのローカルリポジトリで行われていますが、GitHub リポジトリでは行われません。十分なテストで変更を検証したら、以下のコマンドで GitHub の自分のアカウントのリポジトリに変更を移すことができます。

```
$ git push origin branch_name  
TODO: コマンドのメッセージを含める
```

このコマンドのメッセージに図のような URL が表示されます。URL にアクセスしプルリクエストを作成します。

TODO: GetFEM のドキュメントから引用を行っているため言及する。

GetFEM++ のマスターブランチにマージすることは許可されていないので、あなたの役割はここで終わります。プルリクエストのページで管理者や他の開発者と議論することができます。管理者が承認した場合には変更がマージされます。

いくつかの便利な git コマンドを示します。

```
$ git status : status of your repository / branch
```

```
$ git log --follow "filepath" : Show all the commits modifying the specified file (and follow the eventual change of name of the file).
```

```
$ gitk --follow filename : same as previous but with a graphical interface
```

6. GitHub および Travis による継続的インテグレーション

7. まとめ