

大規模ソフトウェアを手探る 提出物・レポートについて

田浦 健次郎

2026/10/27

この課題の提出物

- いじったコード → github
- 最終回発表資料 → github
- レポート → github からリンク
- UTOL に (レポート出しましたとだけ言う形式的な) 提出

- スライドでプレゼン
- 改めて何をしたかを始めから説明
- 動いているもののデモ
- アプローチ
 - ▶ 関係箇所の特定、
 - ▶ いじった後の顛末など色々...
- 演習の感想、得たもの
- 喋り時間 : 11 分 (10 分のつもりがよい)
- 質問時間 : 4 分 (他の班の発表に質問して下さい!!)
- あるある 何を話せば決まってないほどその場で思いついたことで時間が伸びがち

レポートについて

- 田浦とTAにしか読めないPDFファイル → **世の中に公開されるブログ記事**
- **世の中の**読者に役に立つものを残そうという気持ちで
- 世の中にあふれる「〇〇をやってみた」という系のブログの大部分は「インストールしてみた」「使ってみた」の類で、この課題の方が深いが、それでも同じソフトをインストールしてハマった人とかには有用な情報 (それも AI で要らなくなっていくのだろうか...)
- あまり方に肩を入れず
- ♥がつくように頑張ってください

役に立つ記事の書き方

- 要点・目的をはっきりと
 - ▶ LLVM を〇〇してみた
 - ▶ Firefox を...
 - ▶ OpenCode を...
 - ▶ lamma2.c を...
 - ▶ ...
- 他の人が真似できるだけの情報を, 「ありのまま」書く
 - ▶ download 先, バージョン, ビルド方法
- (長くなり過ぎない限り) 「コマンドやログをそのまま」載せるのがよい
- ググって得た情報などは, その情報源へのリンクをつける (出典明記)

- 日本中の「通りすがりの人」を読者だと思って書く
 - ▶ 何度も聞いてる人にしか通じない話をしない
- 結果的にこういじればできた, おしまい, ではなく, どうやってその情報へ至ったか, の過程も書く
- 課題の過程で得た色々なノウハウを, できるなら (そのソフト固有の断片知識ではなく) 一般的に有用なノウハウ, スキルとして書く
- 課題を始める前の自分へ, なに (注: 答よりもやり方) を教えたらいいかを想像しながら書くといい

より形式的なこと

- どこに書くか? ⇒ お気に入りがあある人は自由
- 公開に足る内容がないんですが... ⇒ github の wiki 機能でもよいことにする
- だが、課題で当初やろうとしたことに限らず、自分が今回経験したことで、「同じ問題に遭遇した人に役立つ内容」は何らか切り出せるのではないかと思う。その内容を公開すればよい。
- サンプルブログ、前のタームのレポートも参考に