

# OneLinePrinter

JANOG46 ハッカソン

Team MF（後藤 / 佐藤 / 萩原）

開通作業

よくある  
トラブルシュー  
ティング

みなさん、  
運用自動化進んでいますか？

監視設定追加/  
削除

廃止作業

開通作業

予期せぬ  
トラブル

よくある  
トラブルシュー  
ティング

ツール/ライ  
ブラリ非対応

手動オペレーション、  
まだまだ必要な場面ありますよね？

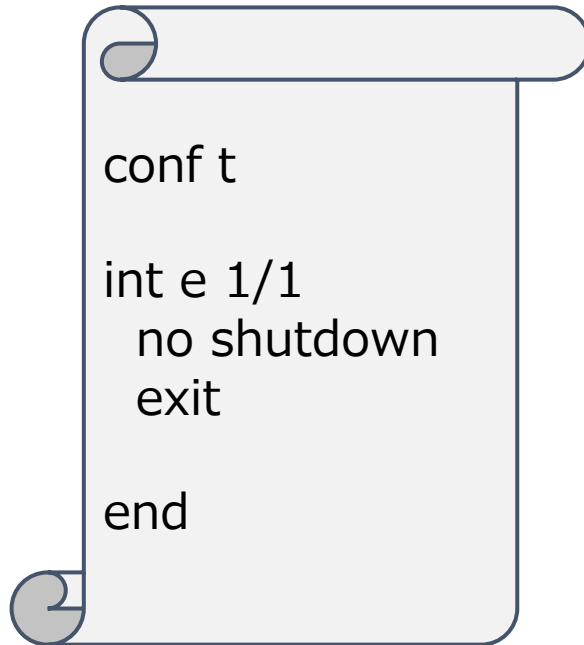
非定型作業

監視設定追加/  
削除

自動化ができ  
ない作業

廃止作業

## 手順書



一行ずつ手でコピーし、  
対象にペーストする



## ターミナル

```
Router1#
Router1#
Router1#
Router1#
Router1#
Router1#
Router1#
Router1# conf t
Router1(config)# int e 1/1
Router1(config-if)# no shutdown
Router1(config-if)# exit
Router1(config)# end
Router1#
```

- 気を付けていてもやってしまう “コピペ” の課題
  - → 全然違う行貼り付けてしまった
  - → 一行だけ貼り忘れてしまった
  - → 間違えて次の行も一緒に貼り付けてしまった etc
- Enter 前の確認や複数の作業者による目視である程度防ぐことはできる
  - コピペが正しく行えているかのチェックはストレスがかかる
- 違う Window を行ったり来たりするのは非常に面倒

# OneLinePrinter

これは何?

- 複数行のコマンドをバッファに入れて、ターミナルなどに**一行ずつ貼り付ける**のをサポートしてくれる GUI ツール

誰が作ったの?

- MF の大先輩 (OB) が数年前に個人で開発したツール
- Web で公開されている: <http://hogetan.net/code/>

誰が使っているの?

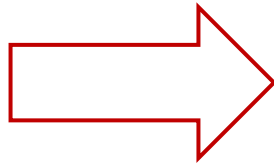
- MF の人はみんな OneLinePrinter が大好き



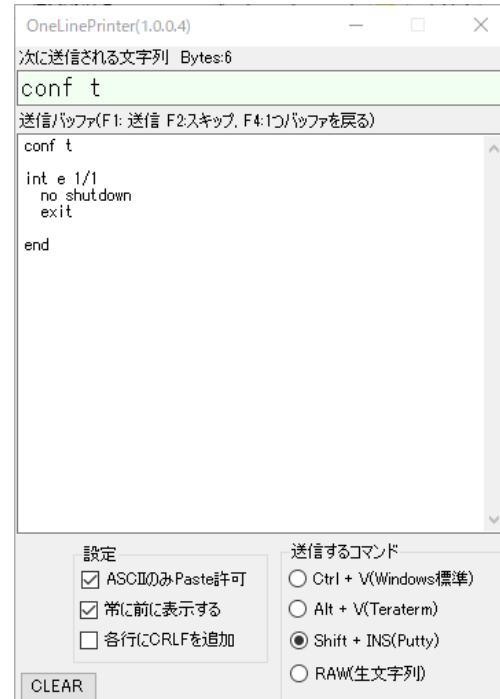
## 手順書

```
conf t
int e 1/1
no shutdown
exit
end
```

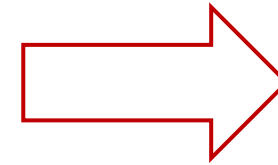
手順全体をまとめて  
ツールにコピペ



## OneLinePrinter



キーボード操作で  
一行ずつ  
対象にペースト



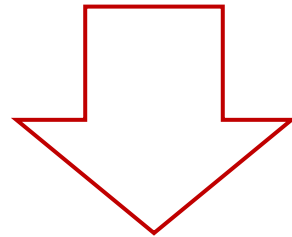
## ターミナル

```
Router1#
Router1#
Router1#
Router1#
Router1#
Router1#
Router1#
Router1#
Router1# conf t
Router1(config)# int e 1/1
Router1(config-if)# no shutdown
Router1(config-if)# exit
Router1(config)# end
Router1#
```

F1: 次の1行をペースト  
F2: 次の1行を Skip  
F4: History から1行呼び戻し



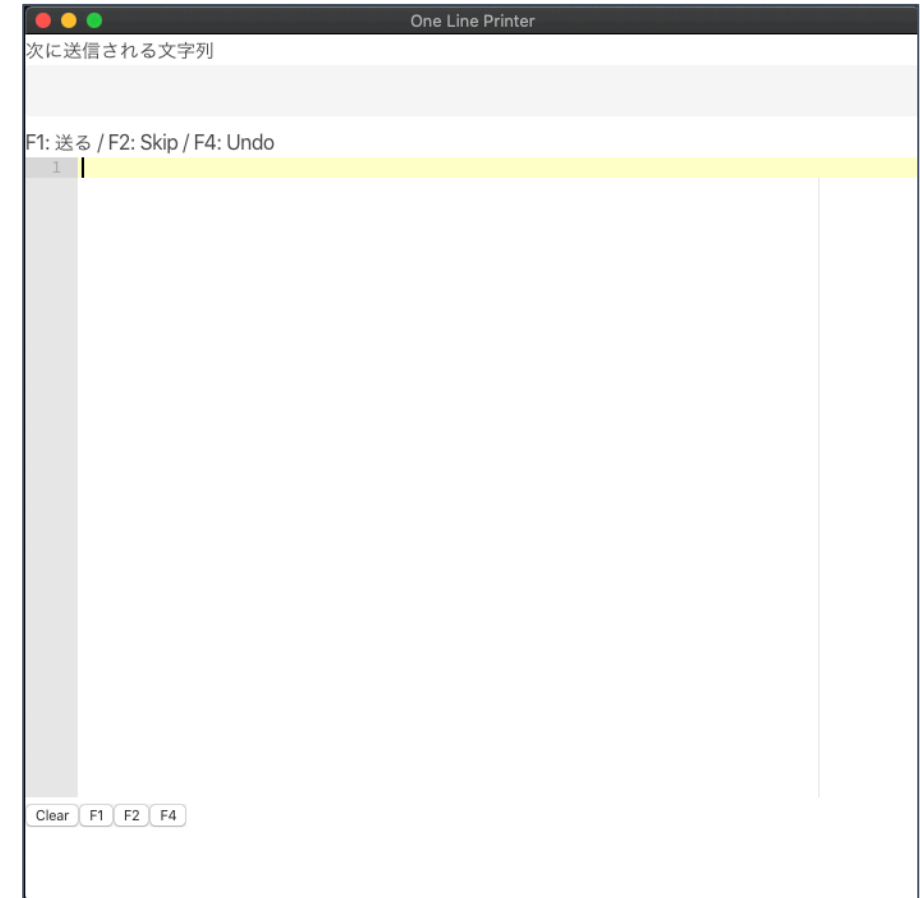
- 社内にソースコードがない
  - あくまで先輩が趣味/個人で作成されたものなので
- ソースコードがあったとしても、メンテ/機能開発できる人が社内にはいない
  - コードは C#
- Windows のみ対応
  - Mac でも使いたい



新しく作り直そう！！

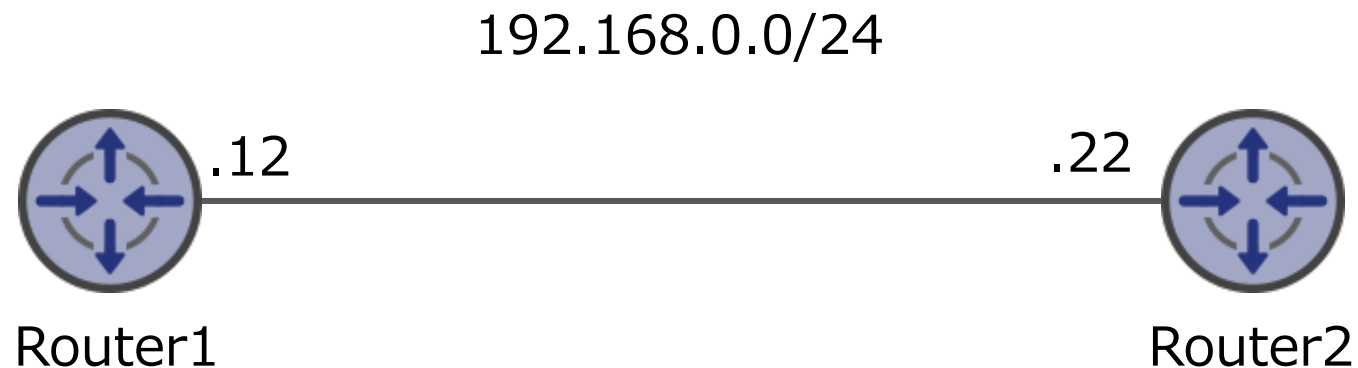
What's new? (= Challenge!)

- モダンな言語/フレームワークによる実装
  - Framework : Electron (go-astilelectron)
  - Frontend : TypeScript + React
  - Backend : Golang
- マルチプラットフォーム対応
  - Windows / MacOS / Linux ...



デモ

あらかじめ用意していた config に対し、  
2台のルータ間で Ping が通るように設定を追加する



- 既存機能の全移植
  - ハッカソンでは最低限の機能のみを実装したため
- CI (Continuous Integration) の導入
- 機能の拡充
- (OSS として公開)

- 良かった点
  - 事前に役割分担を決め、(最初のうちは) 時間を意識して開発できた
  - 事前に使用するライブラリの目星を (ある程度は) つけていた
- 苦労した点
  - バグに悩まされ、動くようになったのは発表の30分前
  - Go で書きたい派 vs Electron で書きたい派で揉めた
    - 結局、Go だけでも Electron だけでもライブラリが不足することがわかり、Go + Electron の形になった