

Let's make your compiler

言語処理系実験

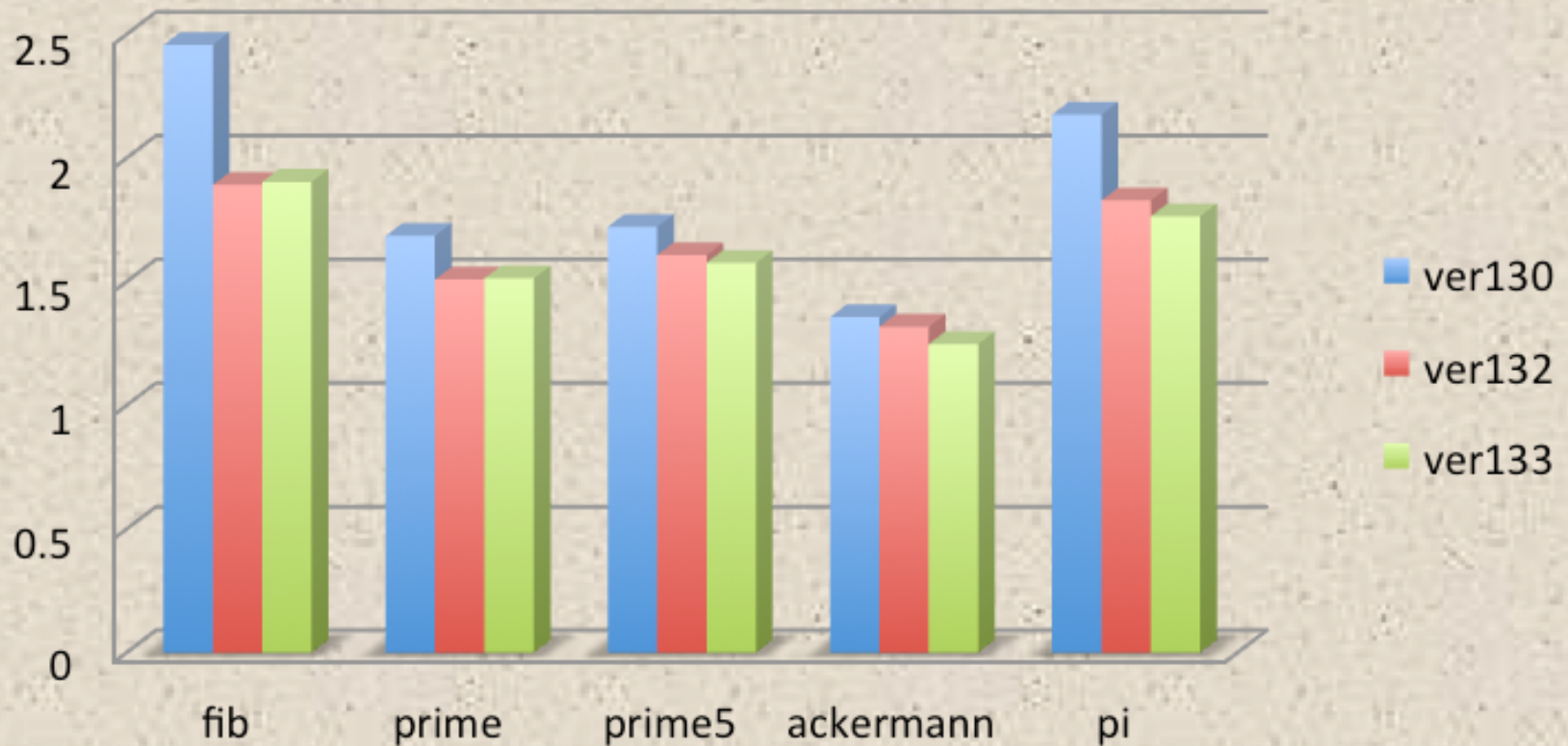
io3班 石見 徳永

達成度

- `.run_test.sh`でokは出るようになった。
 - 出来たのが昨日なので追加機能は・・・
- とりあえず実行してみましょう
 - Ver130:冗長性を許してとりあえず実行できるようになった段階
 - Ver132:アセンブリ言語生成関数をメモリを想定した冗長なものを全域に使っていたのを`addl,subl,imull`に対して簡略化した。
 - Ver133:`if.while`における条件文の判定結果を特定のレジスタに入れていることを考え、それを利用するときにそのレジスタから参照するようにした

速度比較

gccとの速度(cycle数)比較



工夫

- Mac上でも開発が出来るよう工夫した。
- MacのgccはLinuxのものとは異なり、内部でClangが動いているため、吐き出すアセンブラが少し特殊なものであった。
- そこで仮想debianのgccを簡易に使えるようなシェルスクリプトを書いてみた。

- 具体的には、scpによるファイルの送受信とsshでログインし、gccを行う動作をするシェルを組んだ。

感想

- システムを字句解析、構文解析、コード生成に分けて作成するという方針は他のプログラムを作る上でも有用だと感じたので応用していきたい。
- 部品化されたsyntreeなどの部品は完全にブラックボックス化されていて使いやすかったなので今後自分で何か作るときも汎用性の高い部分をブラックボックス化して部品化していこうと思いました。