

# Especificação do Mini-Projeto 2019.2

## O que fazer

Modifique o visitor em

```
compilers-cin/mini-projeto/compiler/visitor.c
```

para gerar código llvm-ir (usando printf's mesmo).

## Como fazer

(Tudo dentro da pasta compilers-cin/mini-projeto/compiler)

Comando para compilar o compilador:

```
make
```

Comando para rodar o compilador com test.c como entrada:

```
./compiler test.c
```

Se você não criar no código um FILE \* para entregar o output em llvm-ir, mas usar o stdout, redirecione o stdout para um arquivo:

```
./compiler test.c > test.ll
```

## Referências

- Use esta instrução para gerar código llvm-ir dum código fonte em C

```
clang -S -emit-llvm -O2 test.c
```

e ver como se dá a conversão. O arquivo test.ll é gerado a partir dela.

Altere a flag de otimização -O2 para o nível de otimização que quiser ter como parâmetro: -O0, -O1, -O2 ou -O3.

Há no test.ll partes do código geradas que não são necessárias para o código estar correto, portanto podem ser omitidas no output do seu visitor. Ex.: tudo que começa com ';' é comentário em llvm-ir.

- A especificação de 2018.2:

compilers-cin/mini-projeto/Mini Projeto.pdf

- E tem a internet.

## De llvm-ir para binário executável

Para gerar o executável:

```
clang test.ll -o test
```

Para rodar o executável:

```
./test
```

## Equipe

Em dupla ou individual

## Dúvidas

- Google Classroom
- e-mail: fcgr@cin.ufpe.br
- Telegram: @felipe\_guerra
- CIn