

# Homework7 程序优化

## Problem

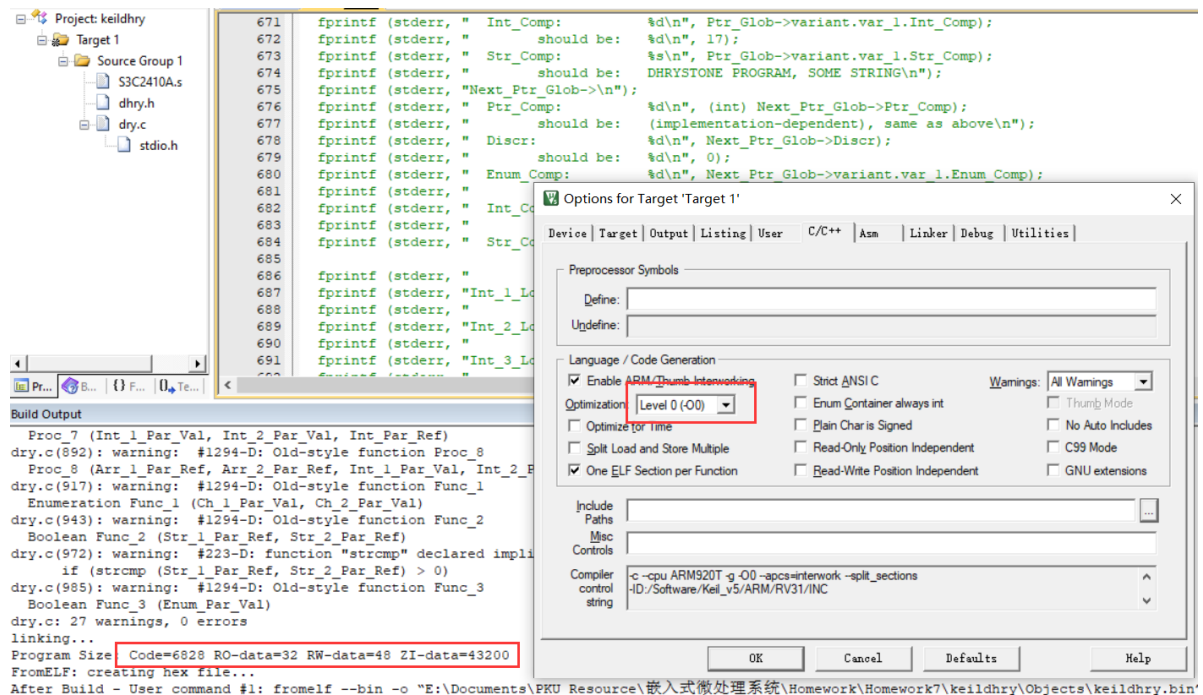
在ARM处理器上, 运行dhrystone测试程序:

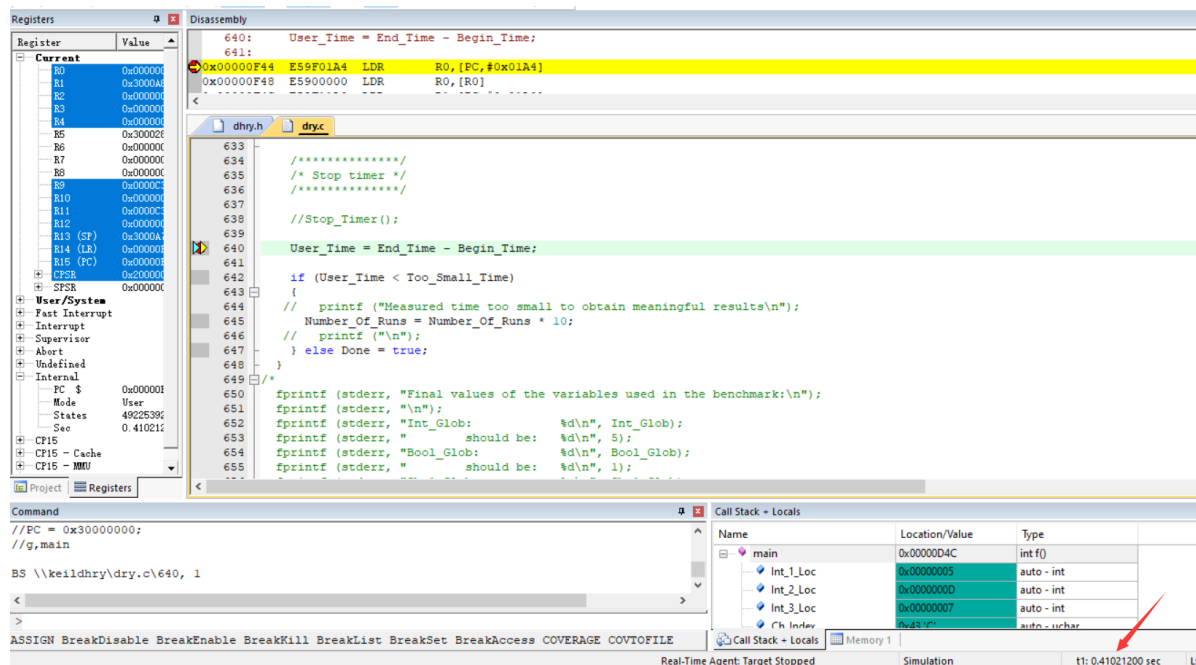
- (1) 分别选择o0, o1, o2, o3 优化选项, 比较所生成程序的大小和执行速度。
- (2) 选择其中一个函数, 尝试分析o0和o3选项所生成的汇编程序的结构和指令数的差别。

## Answer

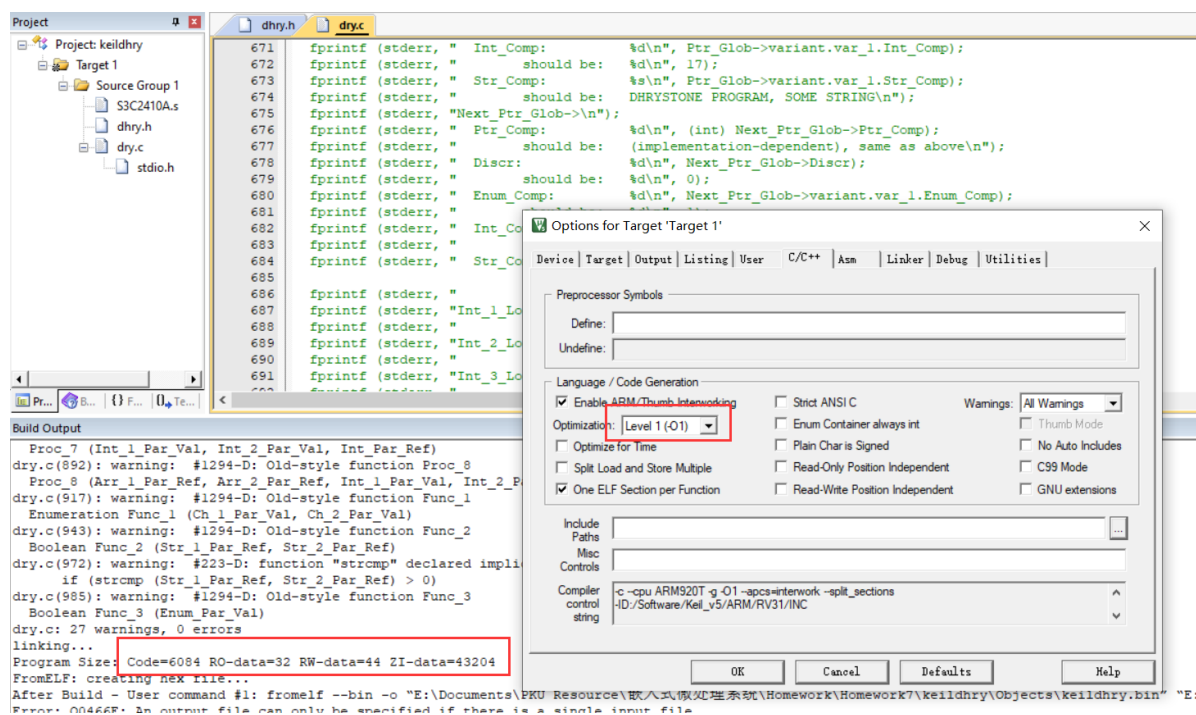
### 实验1

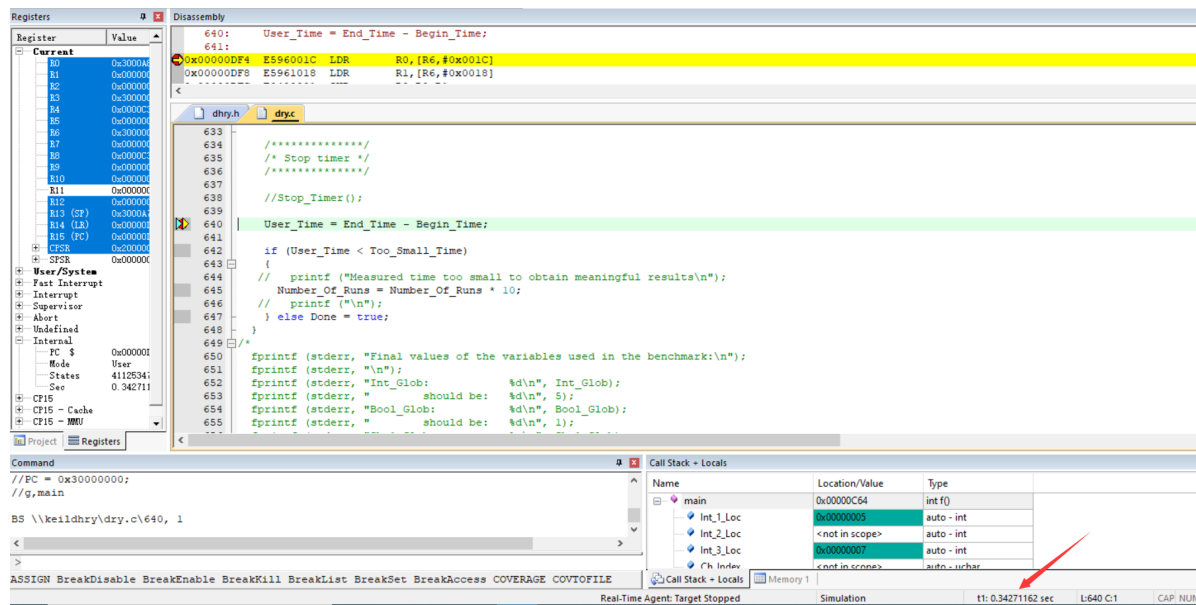
- 由于是模拟器仿真, 注释StartTime与StopTime, 开启o0优化, 编译后生成程序大小与执行时间:
  - Program Size: Code=6828 RO-data=32 RW-data=48 ZI-data=43200
  - Time: 0.41021200s



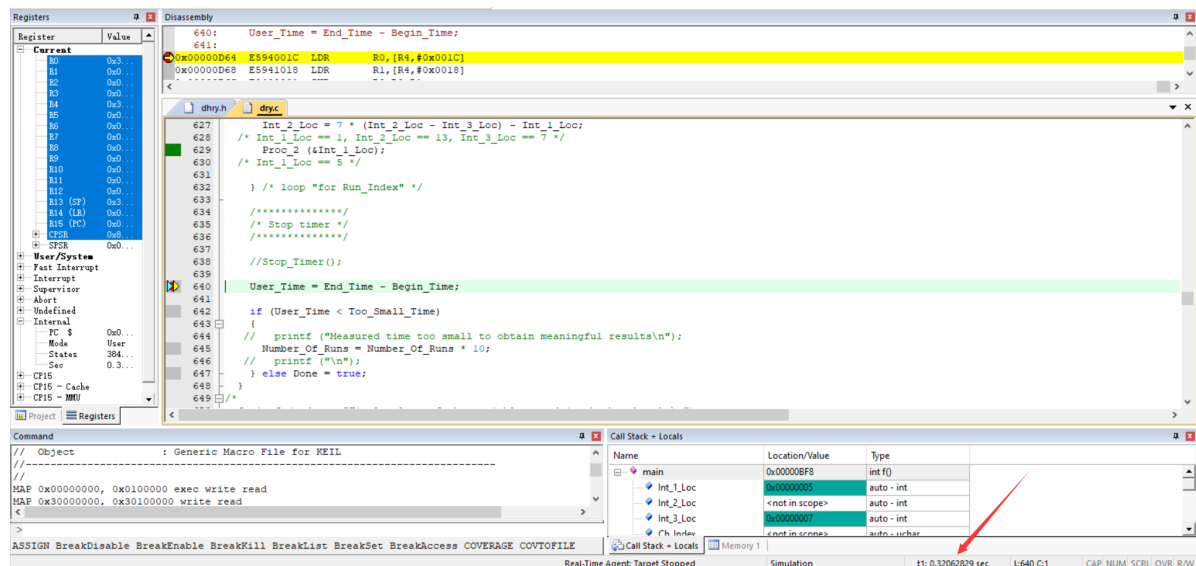
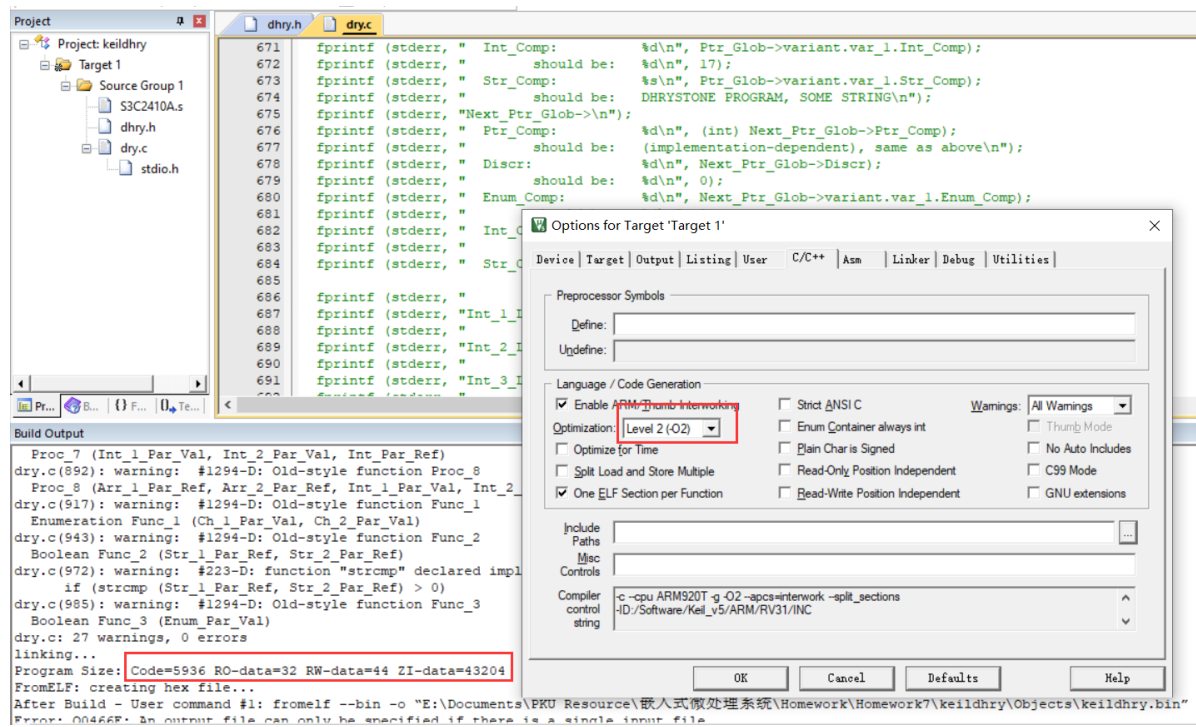


- 开启o1优化，编译后生成程序大小与执行时间：
  - Program Size: Code=6084 RO-data=32 RW-data=44 ZI-data=43204
  - Time: 0.34271162s

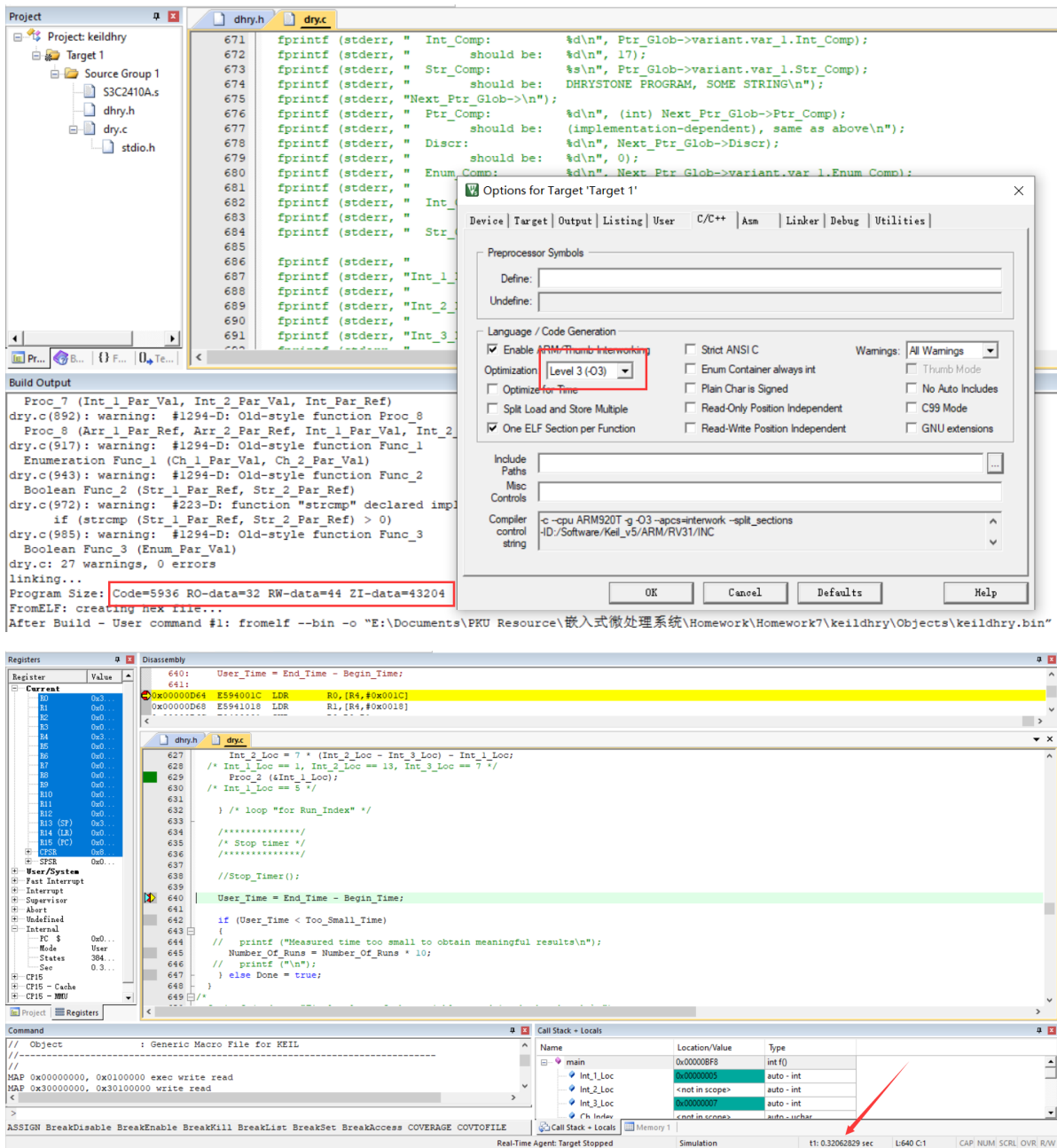




- 开启o2优化, 编译后生成程序大小与执行时间:
  - Program Size: Code=5936 RO-data=32 RW-data=44 ZI-data=43204
  - Time: 0.32062829s



- 开启o3优化，编译后生成程序大小与执行时间：
  - Program Size: Code=5936 RO-data=32 RW-data=44 ZI-data=43204
  - Time: 0.32062829s



## 结论1

- 随着优化级别的提升，生成的程序代码部分的大小越来越小，同时程序执行的时间越来越短（或者不变）
- 最后o2和o3优化的运行时间一样，说明这两次优化的代码对速度提升没有什么差异

## 实验2

- 以函数Proc\_1为例，开o0优化，此时Proc\_1函数的汇编代码有三行，如图：

代码为：

```
0x00000E9C E59F01B0 LDR R0, [PC, #0x01B0]
0x00000EA0 E5900000 LDR R0, [R0]
0x00000EA4 EBFFFEDA BL Proc_1(0x00000A14)
```

- 开o0优化，此时Proc\_1函数的汇编代码有两行，如图：

代码为：

```
0x00000CE0 E5940004 LDR R0, [R4, #0x0004]
0x00000CE4 EBFFF37 BL Proc_1(0x000009C8)
```

## 结论2

- 对于函数Proc\_1，开o0优化与o3优化的指令数相差一条，指令更加精简
- 同时开o3优化时程序不使用PC寄存器内容，而是改用一个R4寄存器。