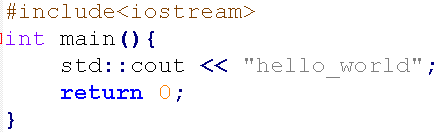
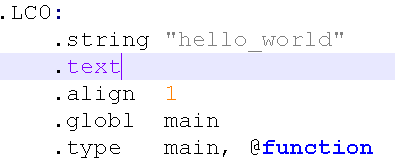
B10815057 廖聖郝 Project 5

1. Write a report to elaborate the compiled RISC-V assembly code in correspondence of your C/C++ source code

Program 1:



從組合語言中可以發現hello\_world與main function的蹤跡:



Program 2:

一張含有 文字 的圖片

自動產生的描述

一張含有 桌 的圖片

自動產生的描述

int的size為4，所以第二行的.size欄位為4

有初始化的變數，初始值會放在.word欄位

無初始化的全域變數，就會有.zero欄位，自動初始化為0

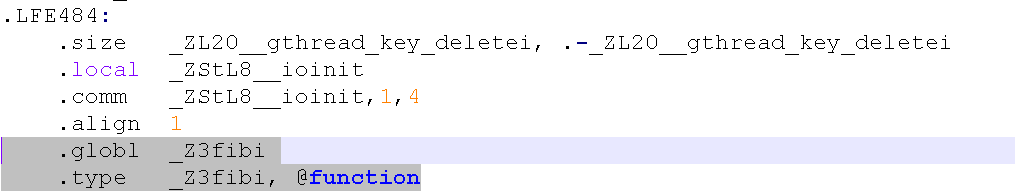
Program 3

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

可以看到提示的string都放在.string欄位

定義fib function的地方

一張含有 文字 的圖片

自動產生的描述

Fib function的內部實作

1. Compare both the elf-gcc and linux-gnu-gcc compilation results with and without using the -static compilation option by using objdump

依據觀察結果得到以下dumpfile指令數比較:

linux(static)(約幾十萬) > elf(not staitc) = elf(static)(約幾萬) > linux(not static)(約幾百行)

elf不管有無static編譯，結果似乎都沒差多少

而linux有static的dump file指令數會遠大於沒有static的數量。

有include iostream的程式碼dump file指令數(約14萬，linux(not static)例外)，會遠大於沒有include iostream的程式碼指令數(約2萬)

1. In your program, declare a variety of C/C++ variable types with and without non-zero initialization and identify the actual physical locations in either final binary program or run-time memory. Summarize your observations.

觀察後的結論:

全域變數:若無初始值，則會自動初始為0

Static變數:若無初始值，則會自動初始為0

以上2種變數因為在編譯時期就已決定記憶體位址(只會有一份，並且生命週期與程式本身一樣長)，所以初始化值不會對執行時期造成影響(無運行成本)。

區域變數:若無初始值，則可能會是任何值(根據該記憶體位址之前寫入之值)

動態配置記憶體(存於heap) :若無初始值，則可能會是任何值(根據該記憶體位址之前寫入之值)

以上2種變數會因為function call或是記憶體配置無法預期，所以在每次的調用都會有不同的記憶體位址，所以初始化該變數的時機只能在執行時期，因而增加運行成本。