TICKLAB

PROJECT: ASSEMBLY RUNNER

# DOCUMENTARY

*Người thực hiện:*
Nguyễn Tấn Đạt
Châu Thành Đạt

*Team:*
Web apps

*Người hướng dẫn:* Huỳnh Hoàng Kha

Ngày 3 tháng 4 năm 2019

# Mục lục

**Tóm tắt nội dung**

Báo cáo hoàn thành.

# 1 Ý tưởng của project

## 1.1 Ý tưởng ban đầu

- Viết một chương trình C++ để dịch một file code assembly sau đó thực thi file code này từng lệnh một để cho phép người dùng có thể nhìn thấy sự thay đổi sau từng dòng code.

## 1.2 Ý tưởng chính

- Assembly là ngôn ngữ lập trình bậc thấp tác động trực tiếp đến các thanh ghi. Việc thực thi một chương trình assembly sẽ ảnh hưởng đến trạng thái hoạt động của máy tính. Việc đó dẫn đến nhu cầu cần máy ảo để giả lập các hành vi của chương trình assembly. Máy ảo phổ biến là máy ảo Java.

- Ý tưởng chính của project là sẽ viết một chương trình bằng ngôn ngữ C++, giả lập lại hành vi của một chương trình assembly và cho phép người sử dụng quan sát trạng thái thanh ghi sau từng dòng lệnh khi thực thi chương trình.

- Ý tưởng của chương trình:
  - Về thanh ghi:
    - Như đã nêu ở trên, việc thay đổi thanh ghi cần được tránh. Chính vì thế chúng ta sẽ gỉa lập lại các thanh ghi trong chương trình bằng cách ô nhớ.
    - Các thanh ghi được giả lập sẽ có độ lớn 16 bit. Mỗi ô nhớ đóng vai trò của một thanh ghi, được đặt tên theo tên các thanh ghi ấy và có hành vi giống như vậy (thanh ghi PC giữ địa chỉ của lệnh hiện tại).
  - Về mạch chương trình:

1. Chương trình sẽ compile code trước khi chạy cho phép người dùng chạy từng lệnh, chương trình sẽ chỉ kiểm tra lỗi cú pháp lúc này.
2. Khi compile chương trình sẽ thực hiện các bước sau: Kiểm tra cú pháp của một câu, kiểm tra thanh ghi, lưu vào danh sách câu lệnh nếu là câu lệnh. Kiểm tra label đã tồn tại hay chưa, đúng cú pháp hay không và lưu vào danh sách các label nếu thỏa mãn.
3. Sau đó, người dùng mới được chạy chương trình và quan sát thay đổi qua từng lệnh. Lúc này sẽ dừng chương trình sẽ kiểm tra lỗi logic, dừng ngay khi gặp lỗi logic.

- Về việc sử dụng OOP:
  - Trong ngôn ngữ, các lệnh từ được phát triển từ 4 loại chính là: R, I, pseudo, J. Hành vi của các lệnh trong cùng một nhóm có phần giống nhau, tuy nhiên lại khác nhau về cú pháp, biến, tham số... Nên việc áp dụng OOP để giải quyết vấn đề nan giải này là hợp lý.
  - Áp dụng tính bao đóng để quản lý các biến của từng loại lệnh cũng như các method của lệnh đó.
  - Áp dụng tính trừu tượng để viết các class cha (loại lệnh).
  - Áp dụng tính kế thừa để viết chi tiết cho từng lệnh từ các abstract của các loại lệnh.
  - Áp dụng tính đa hình để lưu tất cả các loại lệnh vào một mảng, gọi các method...

# 2 Quá trình nghiên cứu và hoàn thành đề tài

## 2.1 Kiến thức cần nghiên cứu

### 2.1.1 Kiến trúc máy tính

1. Tài liệu tìm hiểu: Computer organization and design 5th edition 2014.
2. Người hướng dẫn: anh Huỳnh Hoàng Kha.

3. Quá trình tìm hiểu:

    - Nghe sơ bộ về cách thức một chương trình hoạt động như thế nào qua anh Kha. Sau đó được gợi ý đọc cuốn sách nêu trên cùng với việc tìm hiểu ngôn ngữ Assembly.
    - Đọc và tìm hiểu cách hoạt động thì nhận được các vấn đề cần quan tâm cho việc hoạt động của máy tính như sau:
        (a) Hệ điều hành
        (b) Thanh ghi
        (c) Cách lưu một số: nguyên có dấu, không dấu, số thực chuẩn IEEE...
        (d) Cách máy tính thực hiện các lệnh cộng, trừ, nhân, chia, dịch bit, phép toán logic...
        (e) Thứ tự thực hiện lệnh máy tính: Các lệnh được nạp lên bộ nhớ, biến được nạp lên thanh ghi sau đó thực hiện. Các lệnh được load lên bộ nhớ theo thứ tự nào thì được thực hiện theo thứ tự đó.
    - Tổng kết, tạm ngưng tìm hiểu qua tài liệu, tập code assembly.

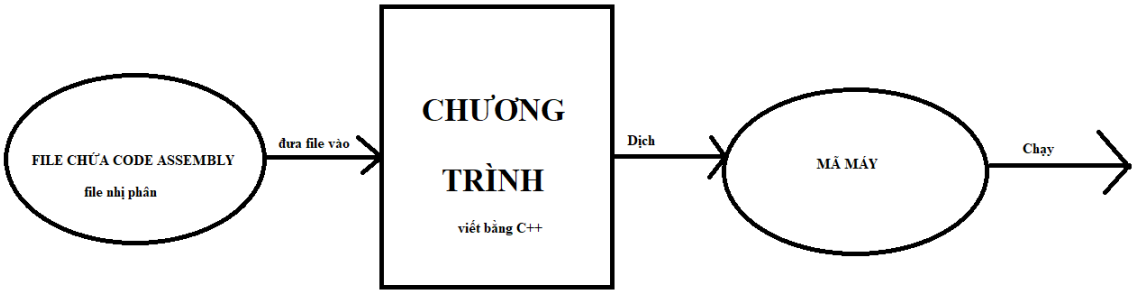
#### 2.1.2 Viết assembly

1. Tài liệu tìm hiểu: Computer organization and desgin 5th edition, Computer organization and design 4th edition.
2. Người hướng dẫn: anh Châu Thành Đạt.
3. Môi trường code: MIPS MARS
4. Quá trình tìm hiểu:

    - MIPS MARS thực hiện lệnh assembly trên một máy ảo Java.
    - Cách chạy một chương trình assembly bằng MIPS MARS và quan sát sự thay đổi của các biến sau một lệnh.
    - Cú pháp viết một chương trình assembly.
    - Các biến trên assembly.
    - Các lệnh trong assembly.
    - Ngừng việc tìm hiểu, lên ý tưởng cho dự án.

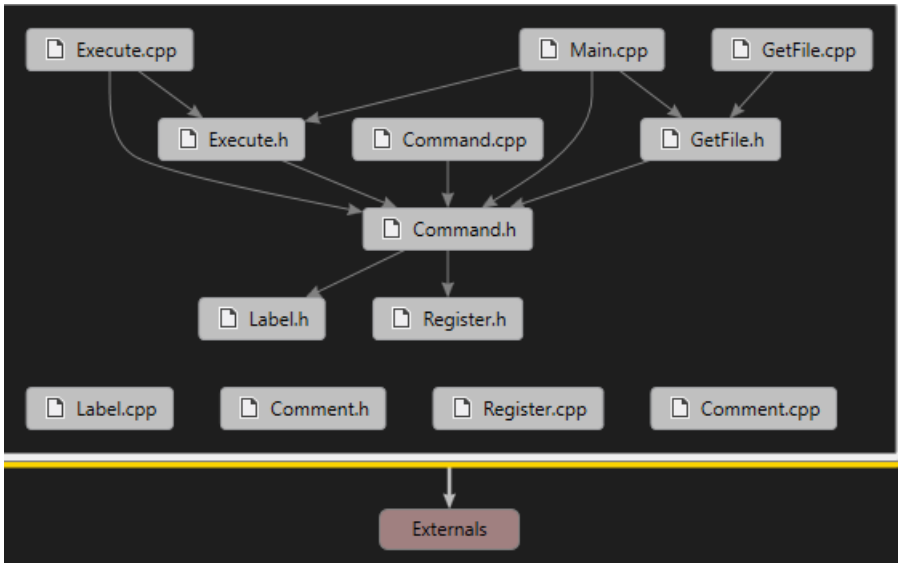
## 2.2 Lên ý tưởng cho dự án

1. Người hướng dẫn: anh Huỳnh Hoàng Kha.
2. Cộng sự: anh Châu Thành Đạt.
3. Môi trường làm việc:
   - IDE: Visual studio 2017
   - Source control: Git, GitLab.
4. Quá trình lên ý tưởng:
   (a) Lên ý tưởng cho mạch chính chương trình:




Hình 1: Mạch chương trình

   (b) Viết các file đầu tiên để quy định các phần chung, riêng của chương trình. Tổ chức các class, struct, tổ chức file, viết mạch chính chương trình...




Hình 2: Liên hệ giữa các file

5. Tạo ra một class chứa các thanh ghi để quản lý.

## 2.3 Quá trình thực hiện

1. Người hướng dẫn: anh Huỳnh Hoàng Kha.
2. Cộng sự: anh Châu Thành Đạt.
3. Quá trình thực hiện:
   (a) Xử lý file input:
      i. Nhận một file code assembly bất kì dưới dạng text.
      ii. Xử lý text để lấy từng dòng một sau đó nhận dạng là command hay là label.
      iii. Lưu các label vào một linklisted, các command vào một linklisted.
   (b) Xử lý code đọc được:
      i. Linklisted lưu command được lưu lại vào một mảng nhằm có thể sử dụng số thứ tự trong mảng để thực hiện chương trình.

ii. Từ các struct cha là RTypeCommand, ITypeCommand, JTypeCommand, PseudoTypeCommand, viết các class lệnh. Mỗi class lệnh bao gồm các biến của lệnh, method check để kiểm tra lệnh và tìm các thanh ghi để lưu vào biến, method execute thực thi lệnh đó.

iii. Trong main, sử dụng method execute để thực hiện từng method execute của các lệnh đọc được từ bước trước đó.

(c) Kết thúc chương trình: Các vùng nhớ khai báo ở heap nếu chưa được xóa ngay vì lý do còn dùng lại sẽ được giải phóng ngay trước khi kết thúc chương trình.

# 3 Mô tả sản phẩm

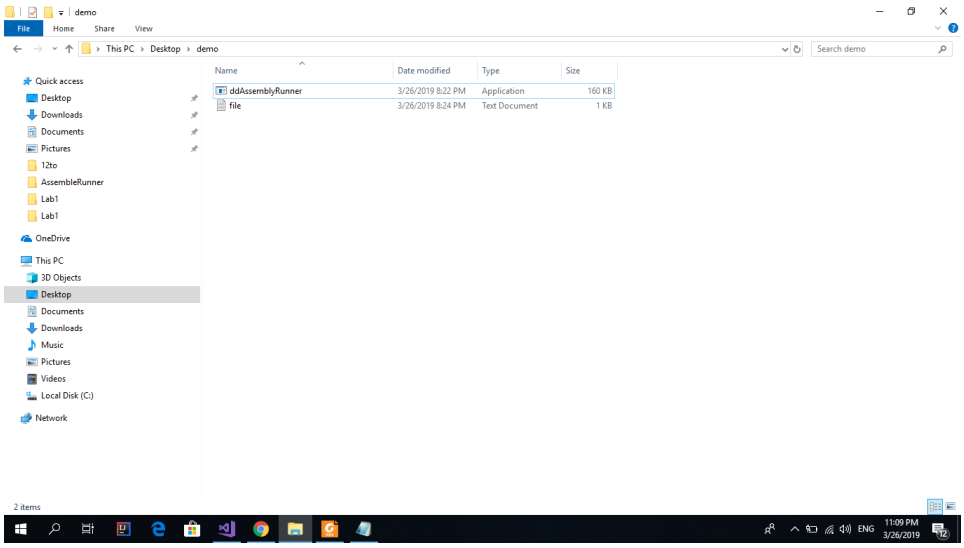
## 3.1 Nguyên lý hoạt động

- Các chương trình assembly làm việc với các thanh ghi trên phần cứng, mỗi lệnh assembly thực hiện sẽ ảnh hưởng trực tiếp đến trạng thái hoạt động máy tính.
- Vì vấn đề trên, ta sẽ sử dụng C++ để giả lập các thanh ghi này bởi các ô nhớ.
- Từng lệnh assembly sẽ được thực thi với đối tượng thực hiện là các thanh ghi được giả lập hay chính xác là các ô nhớ được cung cấp bởi chương trình.
- Sau mỗi lệnh, các thanh ghi (ô nhớ) này sẽ được in ra màn hình để người dùng nhìn thấy.

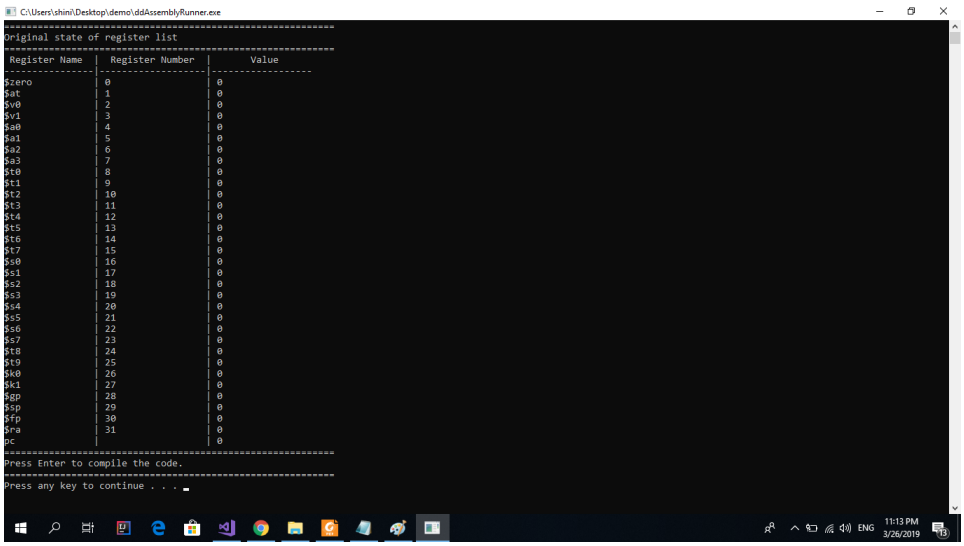
## 3.2 Cách sử dụng sản phẩm

1. Chạy chương trình.

2. Chạy chương trình và quan sát các các register ở trạng thái original.

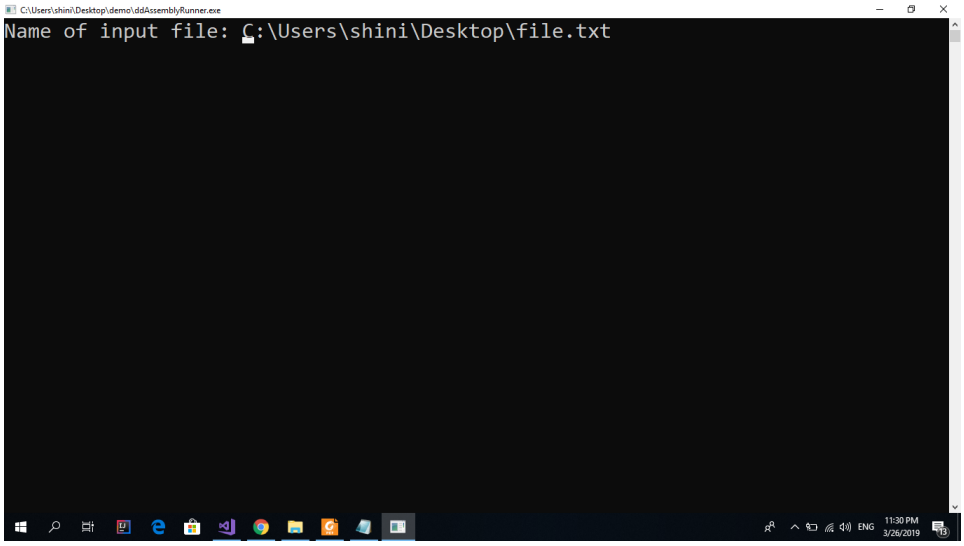



Hình 3: Bước 1: Chạy chương trình



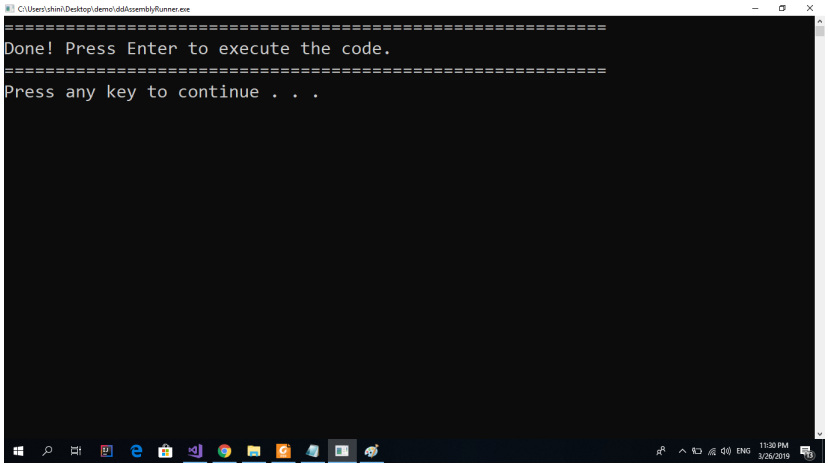

Hình 4: Bước 2: Trạng thái các thanh ghi ban đầu

3. Enter để chuyển sang màn hình nhập tên file. hoàn toàn có thể nhập đường dẫn của file đó.

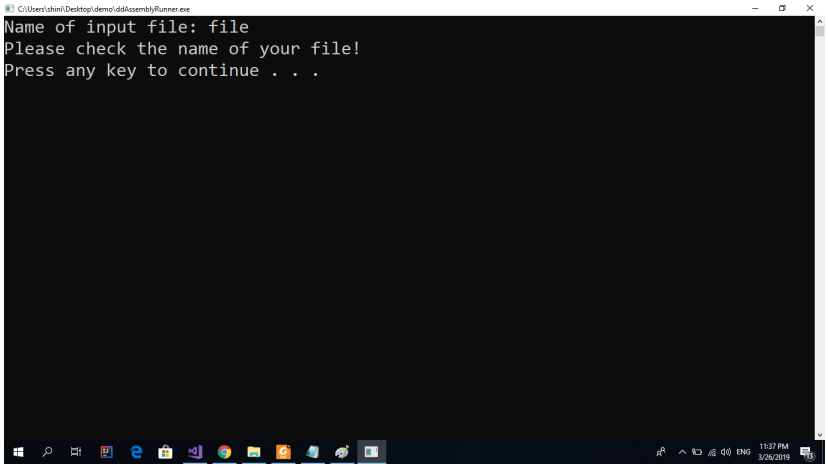



Hình 5: Bước 3

4. Nhập phím bất kì để yêu cầu chương trình chạy file vừa nhập, nếu đúng sẽ hiện ra màn hình sẵn sàng thực thi còn không thì sẽ yêu cầu kiểm tra tên file đó. Khi đọc file code thấy sai syntax, chương trình sẽ xuất ra thông báo lỗi.

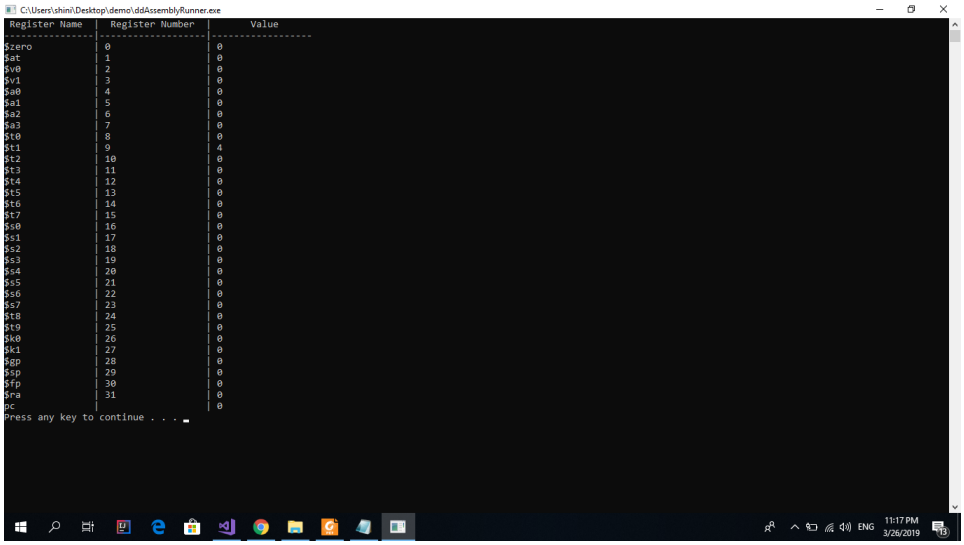



Hình 6: Bước 4: Tìm thấy file




Hình 7: Bước 4: Không tìm thấy file

5. Chương trình sẽ thực thi các lệnh, sau mỗi lệnh sẽ xuất ra trạng thái của tất cả các thanh ghi. Trong quá trình chạy, lệnh nào chạy sai logic sẽ được xuất ra thông báo và kết thúc chương trình.




Hình 8: Bước 5: Thực thi

# 4 Tổng kết đánh giá

## 4.1 Tổng kết

1. Sản phẩm đã hoàn thành các lệnh tính toán cơ bản: cộng, trừ, logic, shift, jump, jump có điều kiện, so sánh lớn hơn, và nhỏ hơn...

2. Sản phẩm đã có thể demo được file code như sau

```
li $t1, 4
li $t2, 4
li $t3, 3
li $t4, 2

addi $t3, $t3, 2
jal test_cal
bne $t1, $t3, doubleT3

Continue:
li $t1, 100
j enda

doubleT3:
add $t3, $t3, $t3
j Continue

test_cal:
sub $t1, $t2, $t3
sll $t3, $t3, 3
and $t4, $t2, $t3
slt $t1, $t2, $t3
jr $ra

end:
```

3. Tìm hiểu được về cách viết một chương trình assembly.

4. Tìm hiểu cơ bản về cách tổ chức, thiết kế máy tính, cách hoạt động của một chương trình máy tính...

## 4.2 Đánh giá quá trình làm việc

1. Trong quá trình làm việc có trễ deadline mà không xin gia hạn. Lý do chủ quan: không quan tâm đến project đúng mực. Lý do khách quan: do ảnh hưởng của nhiều task khác nhau dẫn đến không đủ thời gian để thực hiện nhanh chóng project này.

2. Trong quá trình làm việc có sử dụng Git để quản lý dự án tuy nhiên vẫn chưa tận dụng hết, hay sử dụng không được các chức năng của Git. Đến thời điểm hiện tại, đã tự tin có thể sự dụng các chức năng thông thuộc của Git.

3. Trong quá trình làm việc, nhờ có việc lên ý tưởng chung từ ban đầu và viết các phần interface mà project có thể làm hai người mà không gặp vấn đề quá lớn vì xung đột ý tưởng.

## 4.3 Rút kinh nghiệm

1. Lưu ý về quản lý file, tổ chức các class, struct vì nó ảnh hưởng trực tiếp đếnkhông chỉ ý tưởng mà còn cả quá trình code. Lần này do thực hiện chưa tốt vấn đề trên mà phải đối mặt với file code rất lớn hay có thể rút gọn code nhưng do tổ chức chương trình mà không thể.

2. Theo project tới cùng để không tạo thói quen từ bỏ giữa chừng.

3. Khi thực hiện một project, team phải thường xuyên thảo luận tìm phương án để không xung đột ý tưởng.

4. Sử dụng Git hiệu quả để không gặp các vấn đề về làm việc team, hay thậm chí là giúp quản lý dự án của bản thân tốt hơn.

Trên đây là báo cáo hoàn thành đề tài làm máy tính đơn giản.