ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN



KIẾN TRÚC MÁY TÍNH – HỢP NGỮ

BÁO CÁO ĐỒ ÁN 02

HỢP NGỮ MIPS – GAME HANGMAN

Nhóm thực hiện: 18120061\_18120062\_18120072\_18120185\_18120419

Lớp: KTMT-HN CQ2018/1

25/05/2020 – THÀNH PHỐ HỒ CHÍ MINH

Table of Contents

[**A.** **Thông tin khái quát** 2](#_Toc42109898)

[**1.** **Thông tin nhóm** 2](#_Toc42109899)

[**2.** **Bảng phân công công việc** 2](#_Toc42109900)

[**B.** **Nội dung** 3](#_Toc42109901)

[**1.** **Môi trường phát triển phần mềm** 3](#_Toc42109902)

[**3.** **Quy trình xử lý trò chơi** 3](#_Toc42109903)

[**4.** **Mô tả các các bước xử lý các hàm quan trọng** 4](#_Toc42109904)

[5. Màn hình các bước xử lý cho testcase cụ thể 13](#_Toc42109905)

[**6.** **Tài liệu tham khảo** 18](#_Toc42109906)

[**7.** **Lời cám ơn** 18](#_Toc42109907)

1. **Thông tin khái quát**
2. **Thông tin nhóm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MSSV | Họ tên | Chức vụ | Mức độ đóng góp | Github Username |
| 18120061 | Lê Nhựt Nam | Thành viên | 100% | lenhutnam298 |
| 18120062 | Nguyễn Hoàng Nam | Nhóm trưởng | 100% | congviec18120062 |
| 18120072 | Phạm Lê Hoài Phương | Thành viên | 100% | PLHPhuong |
| 18120185 | Đoàn Đăng Khoa | Thành viên | 100% | thaGitHubUser |
| 18120419 | Phạm Trường Khoa | Thành viên | 100% | truongkhoa3107 |

1. **Bảng phân công công việc**

|  |  |
| --- | --- |
| MSSV | Tên công việc |
| 18120061 | So sánh string input với string đề bài, một số hàm âm thanh |
| 18120062 | Đọc ghi file, vẽ hình hangman, kiểm thử, xử lý chuỗi |
| 18120072 | Tổng hợp code, hàm random từ, viết hàm bổ trợ, menu game |
| 18120185 | So sánh char input với char trong string đề bài, xử lý số điểm |
| 18120419 | Kiểm tra nhập, sắp xếp top 10 người chơi, kiểm tra đầu vào |

Thời gian thực hiện đồ án: 13/05/2020 – 04/06/2020

Các giai đoạn

Giai đoạn 01 13/05 – 20/05: Tìm hiểu, phân công công việc và khởi tạo Github Repository

Github repository: <https://github.com/khmt-do-an/ktmt-proj2>

Giai đoan 02 21/05-28/05: Code module và ghép vào main project

Giai đoạn 03 29/05 – 03/06: Đọc ghi file và kiểm thử lại projects

1. **Nội dung**
2. **Môi trường phát triển phần mềm**

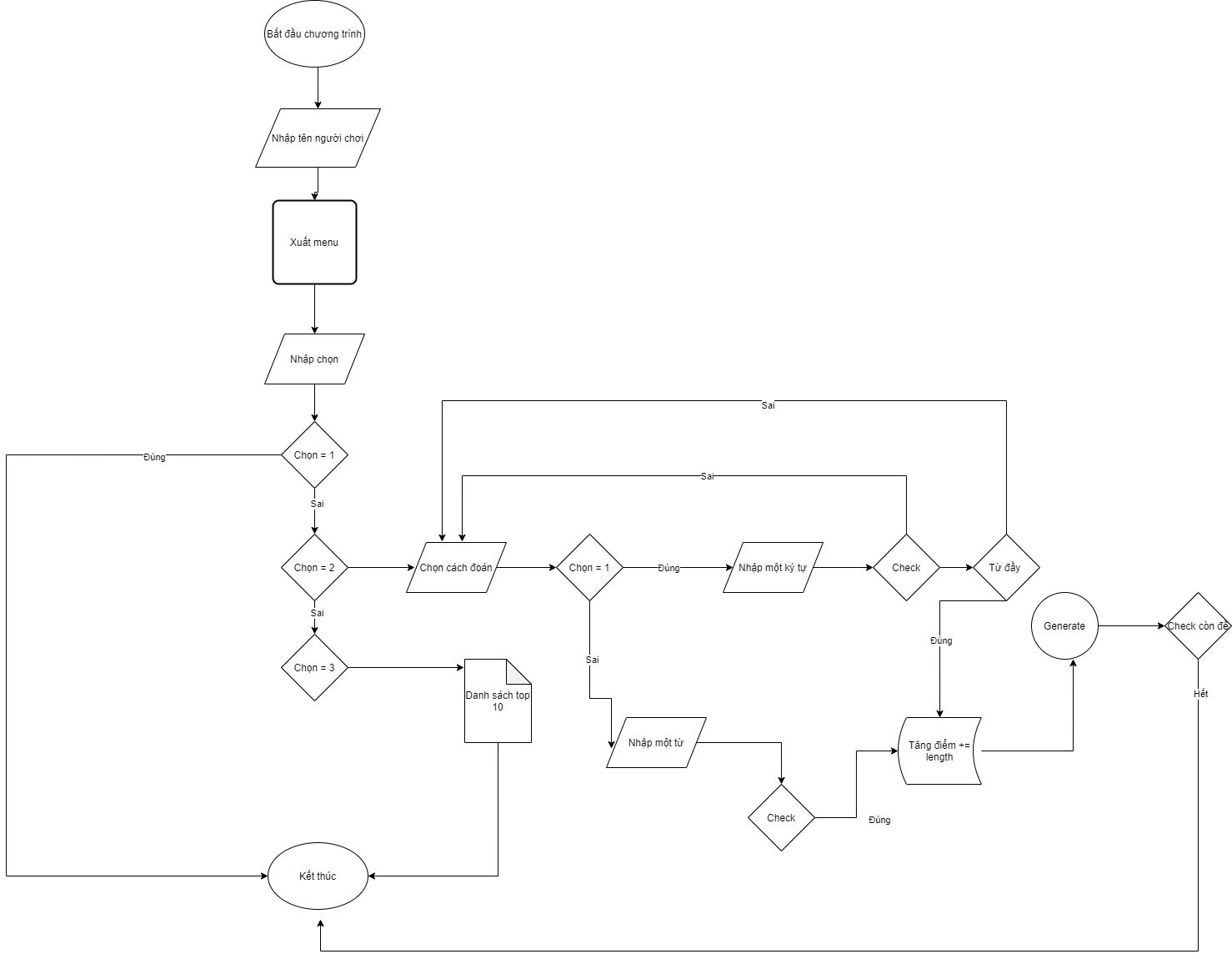
* Mar 4.5 (August, 2014) - MARS MIPS simulator - Missouri State University
* JDK(Java Runtime Environment, Java Virtual Machine) version 8 hoặc mới hơn

1. **Bảng đánh giá các chức năng**

|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Đánh giá |
| 1 | Đoán một ký tự | Tốt |
| 2 | Đoán cả từ | Tốt |
| 3 | Vẽ hangman | Tốt |
| 4 | Console menu | Tốt |
| 5 | Lấy ngẫu nhiên đề bài | Tốt |
| 6 | Xếp hạng | Tốt |
| 7 | Âm thanh | Ổn, một số âm thanh chưa hay |
| 8 | Ghi lại file kết quả | Tốt |

Đánh giá mức độ hoàn thành: 98/100

1. **Quy trình xử lý trò chơi**



1. **Mô tả các các bước xử lý các hàm quan trọng**

Nhóm hàm bổ trợ

* Hàm kiếm tra ký tự đầu vào với chuỗi đề

# == bool \_check\_in\_word(string: word, char: input char)

# - input :

# + @param $a0 = word (string)

# + @param $a1 = $t0 = input char

# - output :

# + @return 0: if $a0 is in $a1

# + @return 1: if $a0 is not in $a1

* Hàm kiểm tra ký tự đầu vào đã có tồn tại trong chuỗi test hay không?

# == bool \_check\_exist(char: char, string: str)

# - input:

# + @param $a0 = char (char)

# + @param $a1 = word (string)

# - output:

# + @return 0: if $a0 is in $a1

# + @return 1: if $a0 is not in $a1

* Hàm kiểm tra ký tự đầu vào có hợp lệ hay không? Tức là bao gồm các ký tự [a-z], [A-Z], [0-9]

# == bool \_check\_input\_valid(char: char)

# - input:

# + @param $a0 = char (char)

# - output:

# + @return 0: $a0 is not valid

# + @return 1: $a0 is valid

* Hàm kiểm tra chuỗi đầu vào có hợp lệ hay không? Tức là bao gồm các ký tự [a-z], [A-Z], [0-9]

# == bool \_check\_valid\_word(string: word)

# - input:

# + @param $a0 = word (string)

# - output:

# + @return 0: $a0 is not valid

# + @return 1: $a0 is valid

* Hàm kiểm tra và đánh dấu những từ đã dùng làm câu hỏi

# == bool \_check\_Marked(int num)

# - input :

# + @param $a0 = num (int)

# - output

# + @return 0: num is in Marked

# + @return 1: num is not in Marked

* Hàm so sánh 2 chuỗi source string và input string. Trả về -1 nếu input string ngắn hơn source string, trả về 0 nếu độ dài 2 chuỗi bằng nhau, và 1 nếu chuỗi input string dài hơn source string

# == int \_strcmp(string: inputStr, string: sourceStr)

# - input:

# + @param $a0 = inputStr (string)

# + @param $a1 = sourceStr (string)

# - output:

# + @return -1: if $a0 shorter than $a1

# + @return 0: if $a0 is the same $a1

# + @return 1: if $a0 longer than $a1

* Hàm tính toán độ dài chuỗi strlen

# == int \_strlen(string: inputword)

# - input:

# + @param $a0 = inputword (string)

# - output:

# + @return : the length of string (int)

* Hàm sao chép chuỗi từ chuỗi nguồn (srcStr) vào chuỗi đích (copyStr)

# == void \_strcopy(string sourceStr, str copyStr)

# - input:

# + @param $a0 = sourceStr (string)

# + @param $a1 = copyStr (string)

# - Use for: Copy sourceStr to copyStr

* Hàm chuyển chuỗi đầu vào về dạng lowcase – uncaplock

# == void \_uncaplock(string: Str)

# - input:

# + @param $a0 = Str (string)

* Hàm đếm số lần mà ký tự đầu vào không xuất hiện trong một chuỗi

# == int \_count\_different\_char\_in\_word (string sourceStr, char: char)

# - input:

# + @param $a0 = sourceStr (string)

# + @param $a1 = char (char)

# - output:

# + @return number of times where $a1 not appear in $a0 (int)

* Hàm sao ghép chuỗi chuỗi (Str1) với chuỗi (Str2) vào chuỗi (result) với kí tự kết nối 2 chuỗi char

# == int \_count\_different\_char\_in\_word (string sourceStr, char: char)

# - input:

# + @param $a0 = sourceStr (string)

# + @param $a1 = char (char)

# - output:

# + @return number of times where $a1 not appear in $a0 (int)

* Hàm chuyển chuỗi số sang số

# == int \_str\_to\_num:(string: str)

# - input :

# + @param $a0 = str (string)

# - output :

# + @return int form of string $a0

* Hàm số sang chuỗi số

# == int \_str\_to\_num:(string: str)

# - input :

# + @param $a0 = str (string)

# - output :

# + @return int form of string $a0

Nhóm hàm IO (Input/Output)

* Hàm nhập tên người chơi

# == void \_enter\_player\_name

# - use for: input or update Name

Chức năng: Dùng để nhập tên người chơi

* Hàm \_input\_Char nhập một ký tự

# == char \_input\_char()

# - output:

# + @return valid char

# - use for: input a character and check whether or not that character is in (0-9)(a-z)(A-Z)

* Hàm \_scanChar nhập một chuỗi đầu vào

# == bool \_scanChar(string: word)

# - input :

# + @param $a0 = word (string)

# - output:

# @return 0: if input char is false(guessed; not in $a0)

# @return 1: if input char is correct(not guessed; in $a0)

# - Use for: input a char and check whether the input char is correct or not

* Hàm \_input\_Word

# == void \_input\_word()

# - use for: input a word and check whether or not that character is in (0-9)(a-z)(A-Z)

# to update inputstr

* Hàm \_scanWord

# == bool \_scanWord(string: word)

# - input :

# + @param $a0 = word (string)

# - output:

# @return 0: if input string is not $a0

# @return 1: if input string is @a0

# - Use for: input a string and check whether the input string is correct or not

* Hàm \_scan

# == int \_scan(int: How\_to\_scan,string: word)

# - input:

# + @param $a0 = How\_to\_scan (int)

# if $a0 = 1 -> scanChar

# if $a0 = 2 -> scanWord

# + @param $a1 = word (string)

# - output:

# @return -1: if How\_to\_scan is not (1 or 2)

# @return 0: if guessing false

# @return 1: if guessing correct

# - Use for: choose how to guess word and check whether player guess is

# correct or not

Nhóm hàm chuẩn bị

* Hàm sort giảm dần điểm của người chơi

# == void \_sort()

# use for: sort form high to low

* Hàm lấy điểm từ phần tử trong chuỗi string gồm điểm-Tên người chơi

# == int \_get\_player\_score(string: str)

# - input :

# + @param $a0 = str (string)

# - output :

# + @return player score in str (int)

* Hàm tạo ngẫu nhiên từ

# == void generate\_guessword(string: word,string: guessword)

# - input:

# + @param $a0 = word (string)

# + @param $a1 = guessword (string)

# - Use for: generate guessword as "-------"

* Hàm printGuessWord

# == void printGuessword(string: guessword)

# - input:

# + @param $a0 = word (string)

# - Use for: print guessword

* Hàm cập nhật từ đoán

# == void \_update\_guessword(string: guessword,char: char,string: word)

# - input:

# + @param $a0 = guessword (string)

# + @param $a1 = char (char)

# + @param $a2 = word (string)

# - Use for: replace '-' in $a0 with $a1 which in $a2

* Hàm trả kết quả khi người chọn đoán hết cả từ

# ==int \_guessing(string word) -------------------------------------------------------

# - input :

# + @param $a0 = word (string)

# - output:

# @return -1: if guessing a competed word is wrong -> lost

# @return 0: if guessing word is correct -> change word

# @return 1: if guessing char is correct and in $a0 but didn’t

# guess the compete word -> continue guessing

# @return 2: if guessing char is wrong and not in $a0

# - Use for: the guessing process of the hangman

* Hàm đếm lại số từ trong bản

# == \_update\_number\_of\_word ()

# - use for update the number of word in buffer N

* Hàm lấy phần tử array[I]

# == int \_get\_array\_I(string: buffer ,int I, string: word)

# - input:

# + @param $a0 = buffer (string)

# + @param $a1 = I (int)

# + @param $a2 = word (string)

# - output:

# + @return 0: input wrong

# + @return 1: successful

# - use for: get value of ArrayStr[I]

* Hàm lấy từ random

# == bool \_generate\_word()

# - use for update word for guessing

# - output:

# + @return 0: out of words

# + @return 1: still has word

* Hàm Hangman xử lý vẽ Hangman

# == void Hangman(string: word)

# - input:

# + @param $a0 = word (string)

# - Use for: the playing process of the hangman

* Hàm xuất top 10 người điểm cao nhất

# == void \_top\_10()

# use for: print top10 player

* Hàm chơi trò chơi tạo game play cho trò chơi

# == void Play()

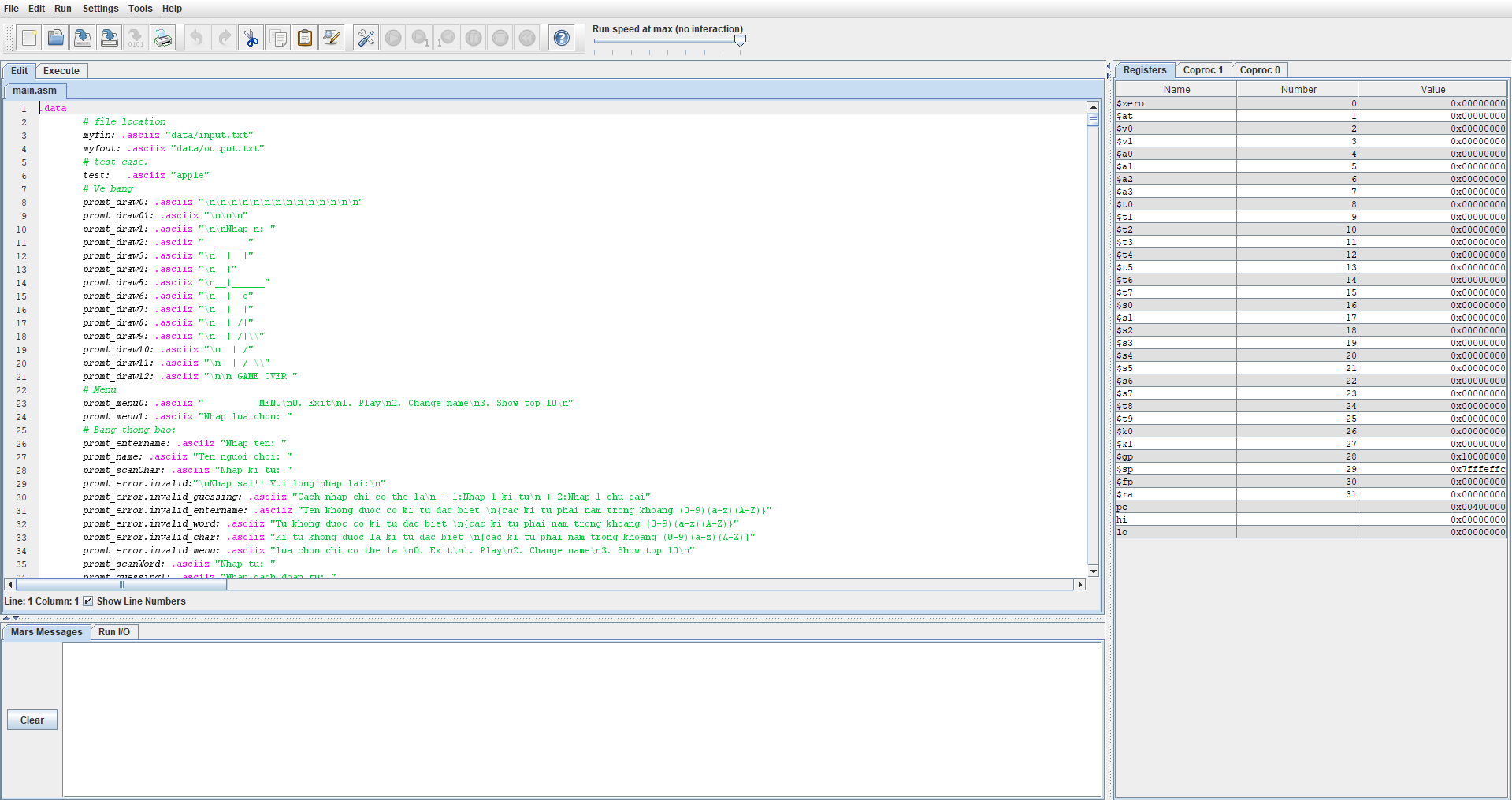
# - Use for: start playing process

* Hàm kết thúc, thoát chương trình

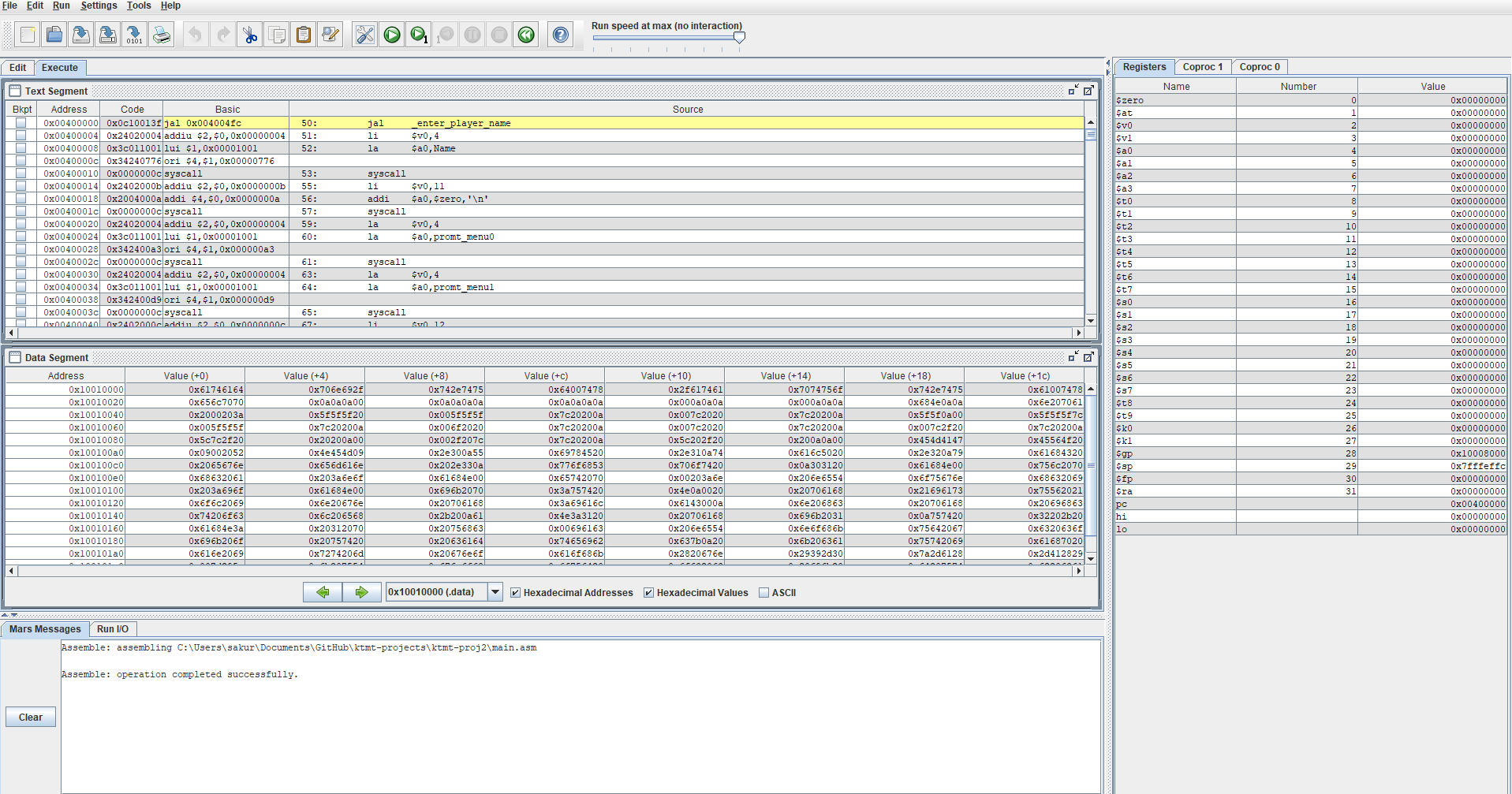
# == \_stop\_runing

# - Use for: Exit program

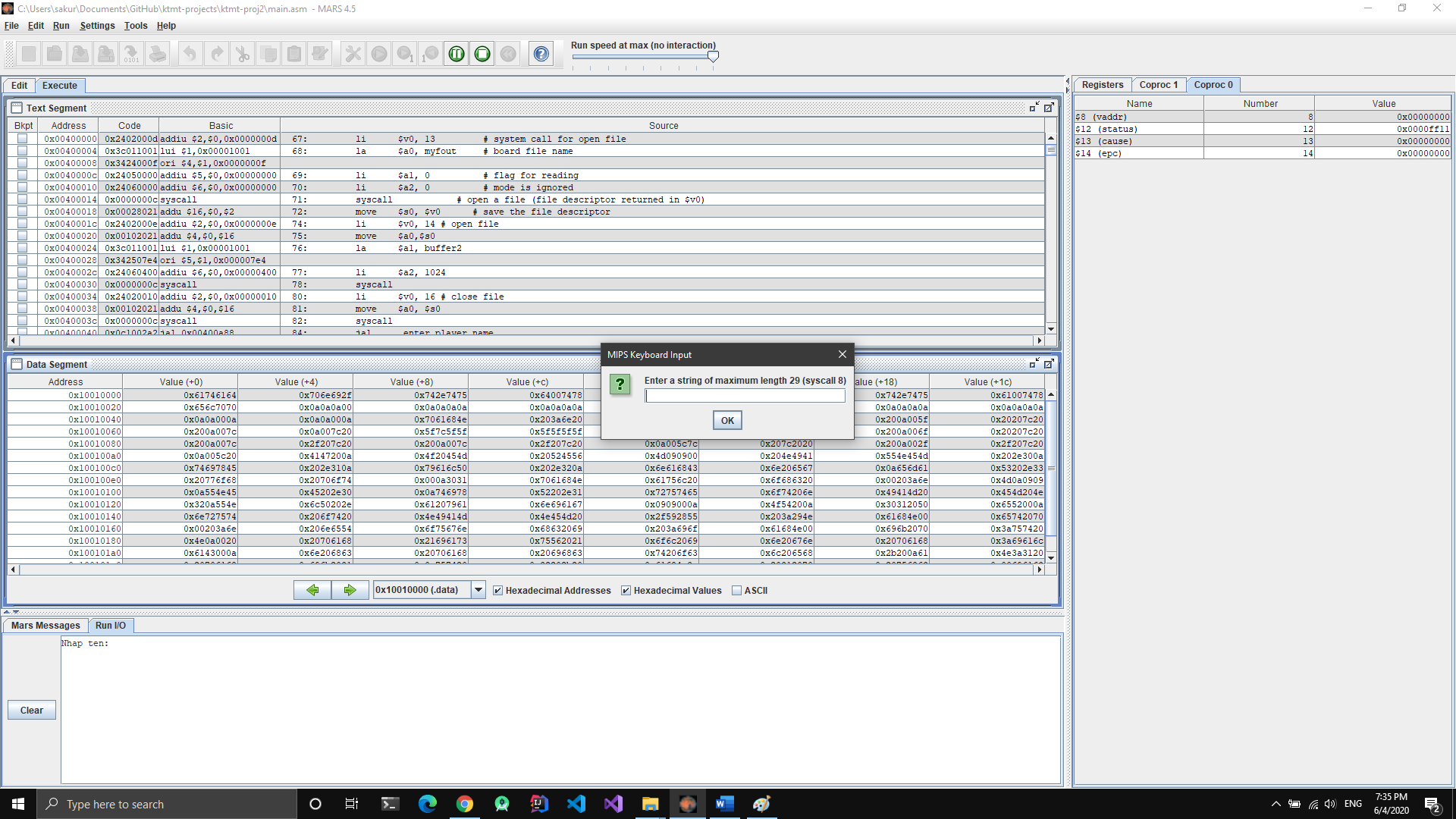
1. **Màn hình các bước xử lý cho testcase cụ thể**



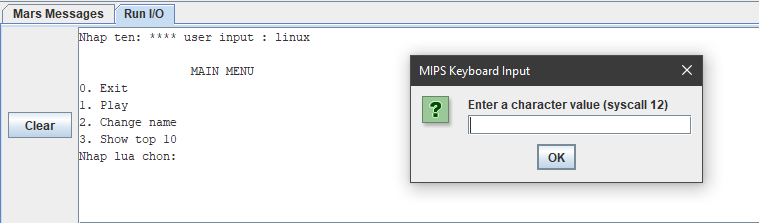
Compile chương trình



Nhập tên người chơi



Menu chính của game khi nhập xong tên người chơi

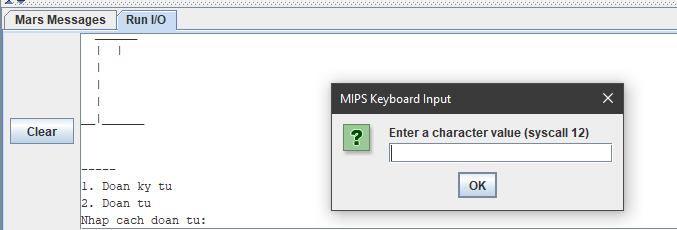


0 – Để thoát khỏi game

1 – Để chơi game

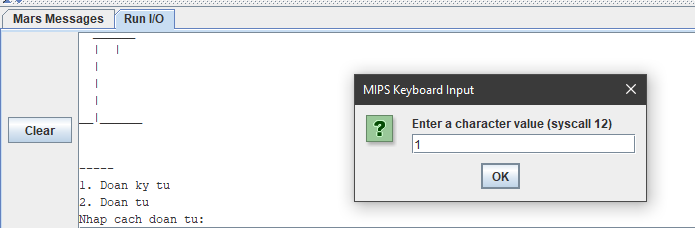
2 – Thay đổi tên người chơi

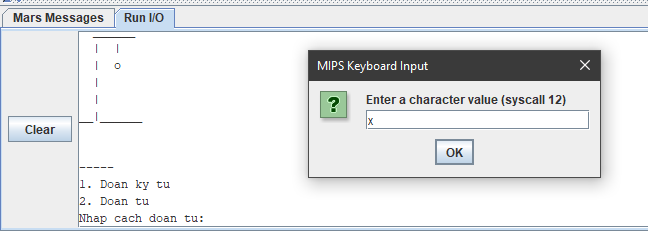
3 – Danh sách top 10 người chơi điểm cao nhất

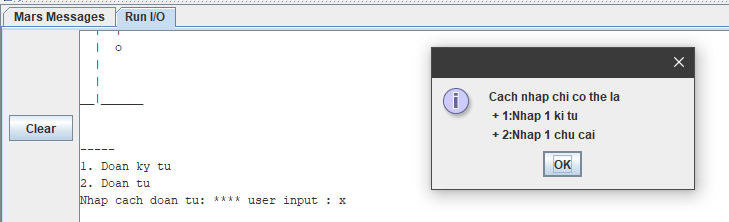


Nhấn 1, và chúng ta sẽ có 1 menu nhỏ, với 1 sẽ đoán 1 ký tự và 2 đoán cả từ

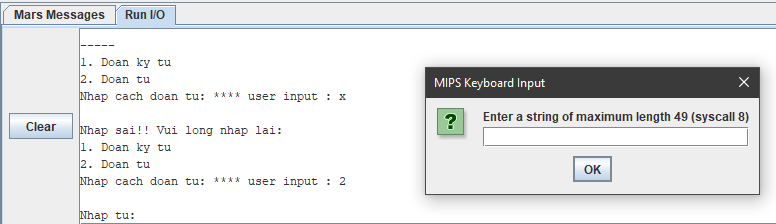
Nếu sai, khung treo cổ sẽ vẽ them một chi tiết

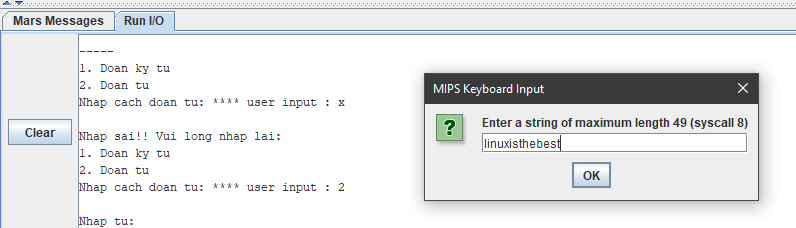




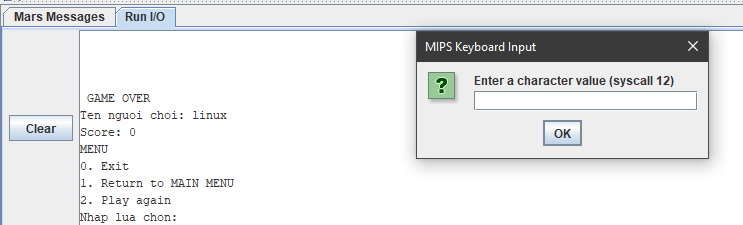


Khi bạn đoán sai cả từ

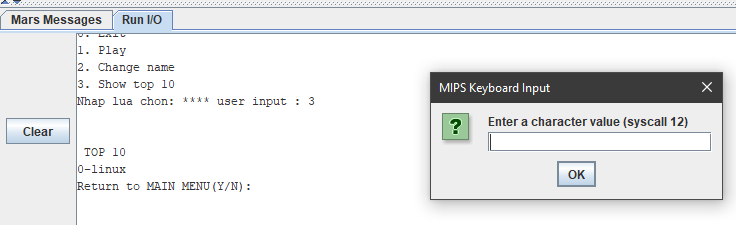




GAME OVER, và xuất hiện menu



Danh sách top 10 người chơi cao nhất



1. **Tài liệu tham khảo**

* MIPS References: <http://www.mrc.uidaho.edu/mrc/people/jff/digital/MIPSir.html>
* Giáo trình Hợp Ngữ - Nguyễn Minh Tuấn, khoa Công nghệ Thông tin, Đại học Khoa học Tự nhiên, ĐH Quốc Gia TPHCM
* Giáo trình Kỹ thuật lập trình, khoa Công nghệ Thông tin, Đại học Khoa học Tự nhiên, ĐH Quốc gia TPHCM
* Slides bài giảng môn Kiến trúc máy tính và Hợp ngữ, phần lập trình hợp ngữ MIPS, khoa Công nghệ Thông tin, Đại học Khoa học Tự nhiên, ĐH Quốc gia TPHCM

1. **Lời cám ơn**

Để hoàn thiện đồ án này, chúng em xin chân thành cám ơn:

* Thầy Lê Viết Long, giảng viên môn Kiến trúc máy tính và Hợp ngữ FIT - HCMUS, đã cung cấp tài nguyên slides bài giảng, kiến thức cơ bản môn học, nhiệt tình hướng dẫn hướng đi, giải quyết những thắc mắc và khó khăn về đồ án của chúng em.
* Các bạn – những thành viên trong team KTMT – HN đã luôn sát cánh, cùng nhau thực hiện và hoàn thiện đồ án này cho tới thời điểm cuối cùng.