**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO ĐỒ ÁN 2**

**6 Bạch Tăng Thắng - 18120237**

**Trần Thanh Tùng - 18120258**

**Trần Hữu Chí Bảo - 18120288**

**Vòng Cảnh Chi - 18120293**

**Cao Tất Cường - 18120296**

**Môn : Kiến trúc máy tính và hợp ngữ**

**Thành phố Hồ Chí Minh - 2020**

# Bảng đánh giá chức năng

|  |  |  |
| --- | --- | --- |
| STT | Chức năng | Đánh giá |
| 1 | Random đề | Chuẩn |
| 2 | Đoán kí tự + Đoán đáp án | Đủ 2 cách đoán + trường hợp đoán từ đã đoán |
| 3 | Hiển thị thông báo kết quả | Đầy đủ |
| 4 | Xuất top 10 điểm | Load từ file sau đó sắp xếp nên giảm chi phí sửa file |
| 5 | Lưu kết quả chơi vào file | Lưu vào cuối file, tiếp nối chuỗi trong file cũ |
| 6 | Nhập tên | Có cho nhập lại nếu tên không thỏa mãn |

# Quy trình xử lý trò chơi

* Vào game
* Gọi hàm nhập tên, tên bị nhập nếu chứa kí tự đặc biệt sẽ bị bắt nhập lại
* Gọi hàm random cho 1 số, lấy từ có số thứ tự bằng số đó
* Đọc từng kí tự của word rồi lưu tất cả vào 1 thanh ghi
* Bắt đầu game, in ra màn hình 2 lựa chọn đoán kí tự và đoán từ:
  + Nếu chọn đoán kí tự, cho nhập vào 1 chữ và gọi hàm kiểm tra kí tự đoán có trong word đã lưu hay ko, việc kiểm tra kí tự có bao gồm kiểm tra kí tự đó đã được đoán chưa, nếu đã được đoán xuất thông báo riêng
  + Kí tự đoán sai sẽ làm tăng giá trị “Lần sai”, giá trị lần sai này sẽ quyết định thông báo xuất trên màn hình (hình người treo cổ). Tùy vào giá trị của “Lần sai”, gọi hàm tương ứng để xuất thông báo đó
  + Mỗi lần đoán sai chữ, những biến lưu thông báo (hình người treo cổ) sẽ được thay đổi.
  + Kí tự đoán đúng sẽ không làm tăng giá trị “Lần sai”, sau đó vẫn xuất ra thông báo dựa trên giá trị ấy
* Sau khi word được đoán đúng hoặc biến đếm số “Lần sai” chạm giới hạn, xuất thông báo game over và kết thúc 1 game.
* Đưa các thanh ghi lưu quay về giá trị ban đầu trước khi bắt đầu game
* Khi người chơi thua cuộc hoặc lựa chọn kết thúc trò chơi, xuất ra 10 người điểm cao nhất từ file Input
* Lượt chơi tiếp theo

# Một số hàm quan trọng

1. Hàm Doankitu (kiểm tra kí tự nhập):

* Xuất menu, nhập kí tự:
* kí tự nhập vào lưu tại $s2

li $v0,12

syscall

move $s2,$v0

* nhảy tới tag kiemtratudoan (so sánh $s2 và $t2)

lb $t2,word($t1)

beq $s2,$t2,KiemTraDuDoan\_Equal

j KiemTraDuDoan\_NotEqual

* Nếu kí tự đoán sai dẫn đến các tag số lần đoán sai ($s7 lưu lại số lần sai)

beq $s7,0,SaiLan1

beq $s7,1,SaiLan2

beq $s7,2,SaiLan3

beq $s7,3,SaiLan4

beq $s7,4,SaiLan5

beq $s7,5,SaiLan6

beq $s7,6,SaiLan7

* Các tag SaiLanX (X từ 1 đến 7) dung để set giá trị cho ảnh (pic) xuất ra màn hình (thông báo sai lần sau được dựa trên thông báo sai lần trước nên chỉ cần gọi 1 hàm mỗi lần sai, mang hơi hướng quy đệ quy)
* Trong tag KiemTraDuDoan\_Equal nếu đoán từ đã có sẽ dẫn đến tag KiemTraDuDoan\_TrungLap dùng để xuất thông báo
* Quay lại xuất menu ở B1, lặp lại đến khi đoán đúng hoặc biến đếm số lần sai chạy quá số quy định

1. Mở File và thao tác:
   * Mở file:

li $v0,13

la $a0,tbfile1

li $a1,0

li $a2,0

syscall

* + Đọc file xử lý chuỗi, mỗi từ được cách nhau bở dấu \*
  + Load các giá trị

#xu li chuoi

la $t3, '\*'

la $t4, '-'

li $t0,0 # giá trị src[i]

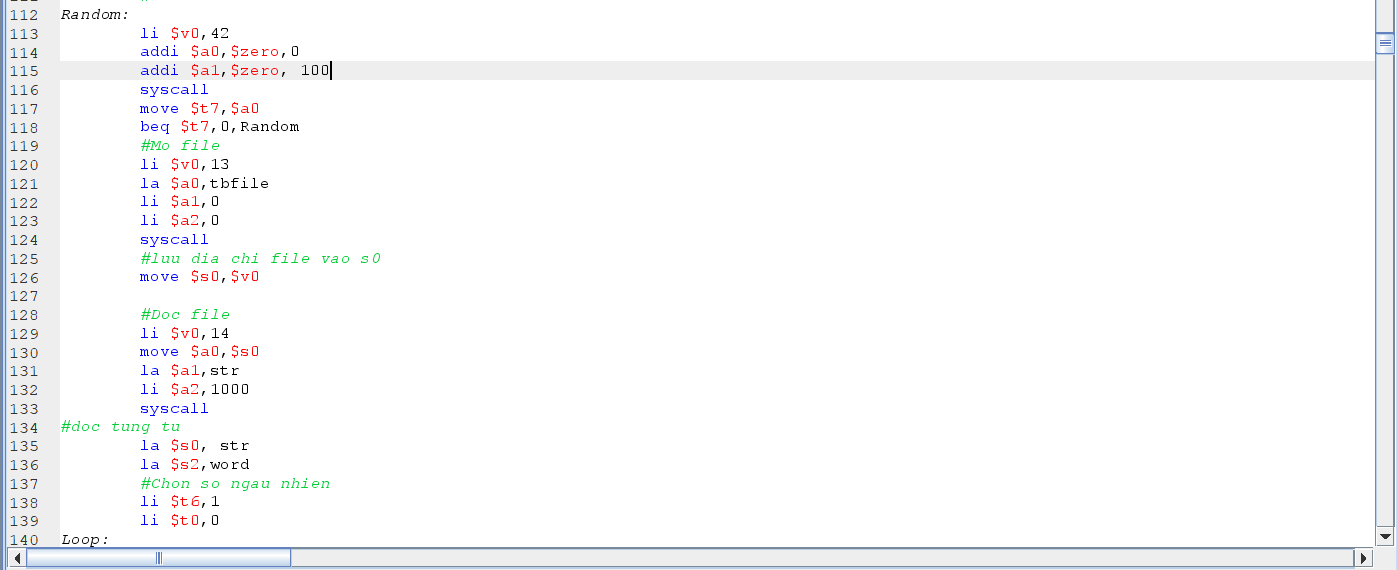
li $t5,0 #số lượnng mảng trong int

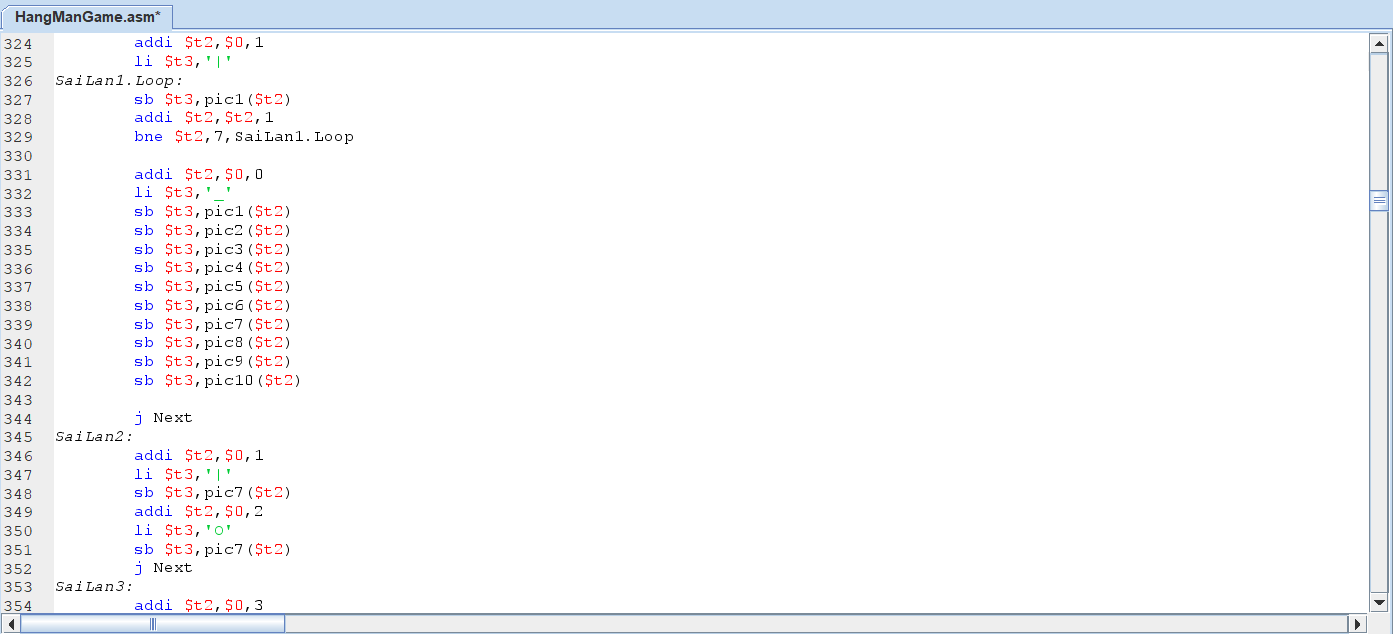
li $t7,0 # int t = 0

la $t2,'\0'

li $t9,0

* + Xếp điểm vào 1 mảng, sắp xếp mảng chọn 10 người cao điểm nhất rồi xuất ra màn hình

1. Hàm random ra số ngẫu nhiên và xuất hiện từ cần đoán:
   * Syscall $v0 =42, gọi hàm random trong khoảng với $a1 là cận trên ($a1=100), load file gồm 100 phần tử nên random từ 1🡪100
   * Số được ngẫu nhiên chọn là số thứ tự của word cần đoán, trích xuất word này từ file
   * Nếu random được số 0 cho gọi lại hàm random
2. Hàm đoán kí tự và xuất từng mảnh hình người khi đoán sai:



* Đầu tiên là tạo cột treo với câu lệnh ($t3 vai trò như biến tạm sao chép):

addi $t2,$0,1

li $t3,'|'

* Sau đó cho lặp 7 lần tạo ra cột treo (lưu vào $t2)

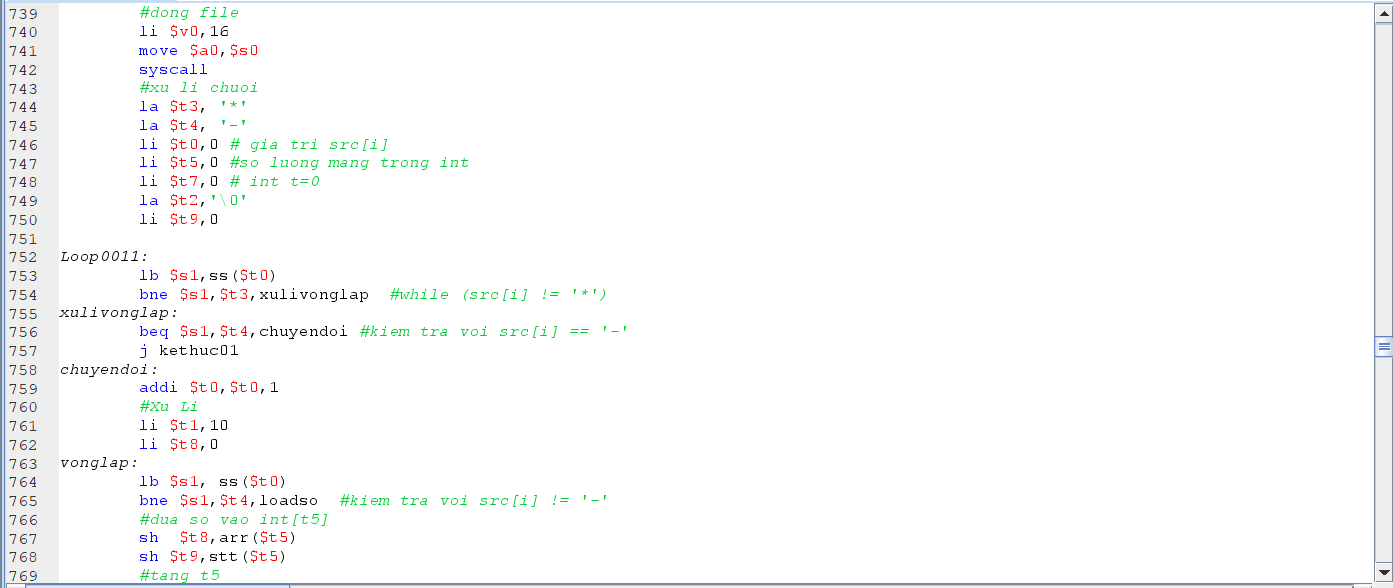
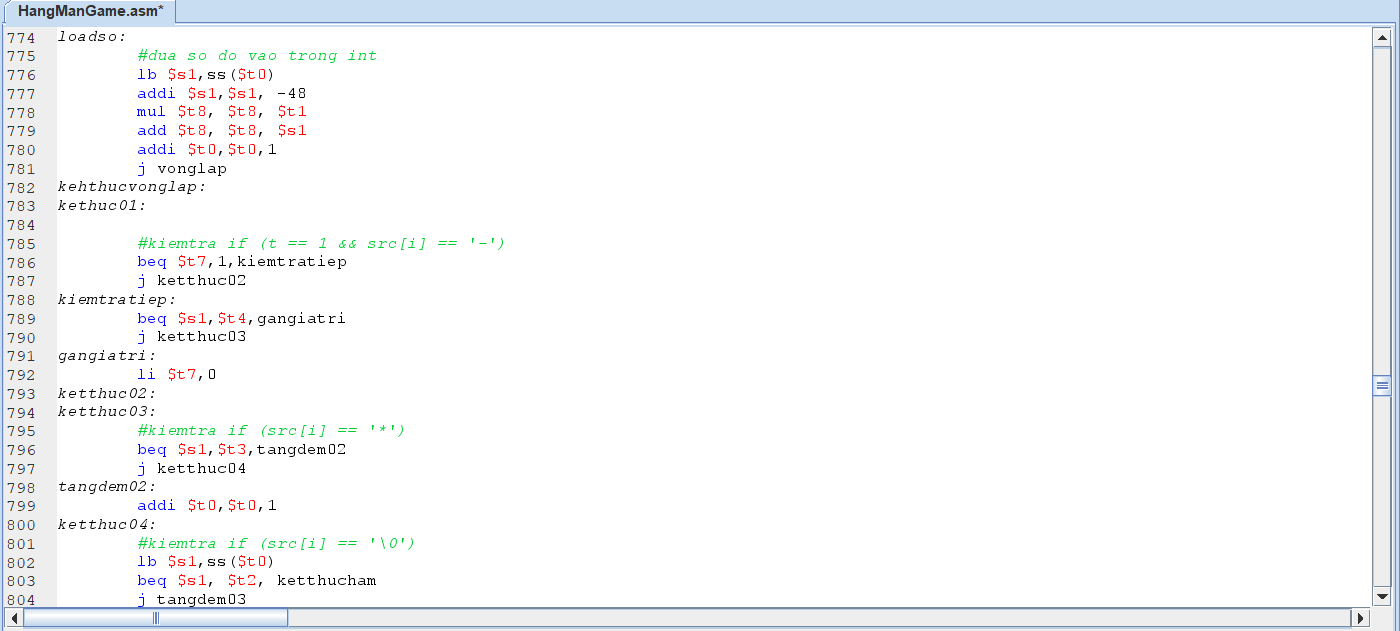
SaiLan1.Loop:

sb $t3,pic1($t2)

addi $t2,$t2,1

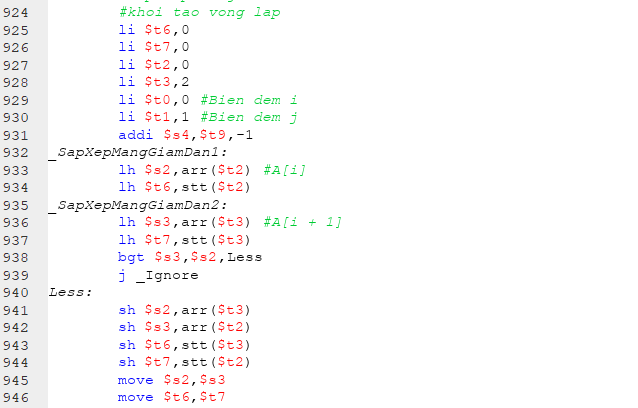
bne $t2,7,SaiLan1.Loop

* Đưa $t2 lại về 0 (quay về đỉnh của cột treo), gán $t3 = “-”, rồi dung $t3 để sao chép vào $t2, hoàn thành thanh treo cổ (trường hợp sai lần 1)
* Mỗi lần sai sau dung cách tương tự lần sai 1, $t3 vẫn là biến tạm sao chép, $t2 vẫn là thông báo xuất ra, lưu ý mỗi lần sai sau đều kế thừa kết quả lần sai trước và vẽ them

1. Hàm lấy ra từng số trong file ouput và các dấu \* :

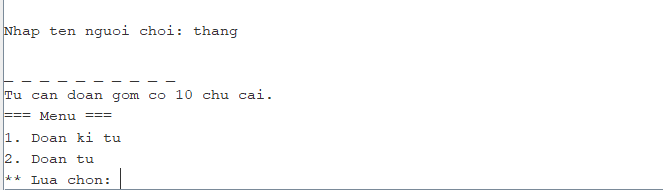
* Hàm kiểm tra giá trị đọc vào khác “ \* ” , sau đó kiểm tra nếu bằng “ - ” thì cộng giá trị lưu tại $t0 lên 1
* Khi $t0 đạt giá trị 2, tức là số đã được đọc qua ( điểm nằm giữa 2 dấu -), load số lưu vào mảng int, reset $t0 về lại 0
* Nếu kí tự đọc vào là “\*” thì $t0 cộng 1
* Nếu kí tự đọc vào là “\0” thì kết thúc hàm, xóa kí tự đó và ghi vào \*

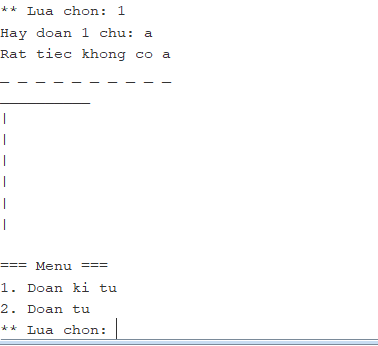
1. Hàm sắp xếp giảm dần:

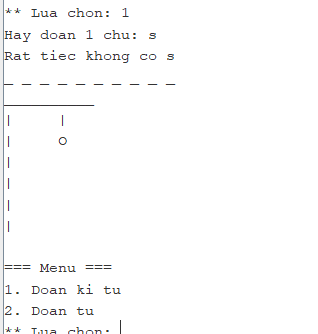


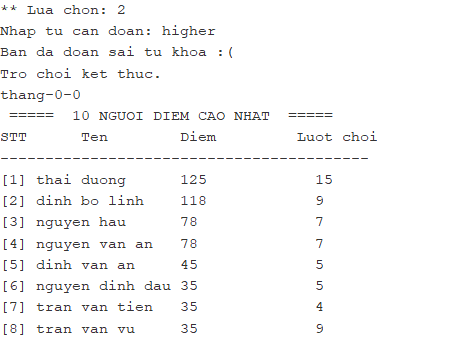
* Sắp xếp mảng giảm dần theo phương pháp Bubble sort, so sánh giữa 2 giá trị tại $t2 (lưu A[i]) và $t3(lưu A[i+1])
* Nếu $t2 <$t3, tiến hành gọi hàm Less, đổi chỗ $t2 và $t3, đồng thời đổi 2 giá trị mảng index ở vị trí tương đương (lưu tại $t6 và $t7)
* Lặp đến khi chạm đến cuối mảng
* Xuất ra màn hình top 10 người dựa trên mảng index đã sắp xếp

# Màn hình các bước xử lý

1. Nhập tên, chọn xử lý:
2. Lựa chọn 1, nhập chữ sai, xuất thông báo:



1. Lựa chọn 1, nhập chữ sai, xuất thông báo:
2. Lựa chọn 2, nhập chữ, xuất thông báo, đoán sai xuất ra top 10 điểm cao nhất:



1. Hiện menu tiếp tục, chọn thoát, xuất ra màn hình tên, điểm và số lần đoán đúng:

