

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



## BÁO CÁO

### ĐỀ TÀI PHẦN MỞ RỘNG KSTN - KIẾN TRÚC MÁY TÍNH

---

## SỬ DỤNG HỢP NGỮ MIPS

### HIỆN THỰC TRÒ CHƠI Ô ĂN QUAN

---

GV hướng dẫn: PGS.TS Phạm Quốc Cường  
SV thực hiện: Nguyễn Đức An – 2010102  
Nguyễn Quang Khánh – 2010330  
Huỳnh Tấn Lộc – 2010391  
Lưu Quốc Hưng Thịnh – 2010651

Tp. Hồ Chí Minh, Tháng 12/2021



## Mục lục

<b>1</b>	<b>Mở đầu:</b>	<b>2</b>
<b>2</b>	<b>Luật chơi Ô ăn quan:</b>	<b>2</b>
2.1	Định nghĩa: . . . . .	2
2.2	Luật chơi . . . . .	3
<b>3</b>	<b>Giải thuật hiện thực:</b>	<b>3</b>
<b>4</b>	<b>Hướng dẫn chơi và Kiểm thử chương trình:</b>	<b>5</b>
4.1	Hướng dẫn chơi: . . . . .	5
4.2	Kiểm thử chương trình: . . . . .	7
<b>5</b>	<b>Kết luận:</b>	<b>9</b>
	<b>Tài liệu tham khảo</b>	<b>10</b>

## 1 Mở đầu:

**Kiến trúc máy tính** là môn học rất quan trọng, giới thiệu cho sinh viên khoa Khoa học và Kỹ thuật máy tính các khái niệm quan trọng trong lĩnh vực máy tính, và đồng thời giúp cho sinh viên có thể đánh giá hiệu năng của một chương trình, hiểu được về hoạt động của phần cứng máy tính, đặc biệt là giúp sinh viên có khả năng vận dụng hợp ngữ để lập trình một số bài toán. Môn học giúp sinh viên có thể đọc hiểu tài liệu, chương trình hợp ngữ và lập trình được bằng hợp ngữ MIPS thông qua công cụ MARS, giúp sinh viên hiểu được bản chất hoạt động của máy tính, giúp ích cho việc giải thích nhiều vấn đề quan trọng trong quá trình học tập và làm việc sau này.

Việc vận dụng hợp ngữ MIPS để giải quyết một số vấn đề thực tế sẽ giúp cho sinh viên hệ tài năng hiểu sâu sắc hơn về hợp ngữ cũng như có hứng thú hơn trong việc nghiên cứu các vấn đề nâng cao, ứng dụng vào thực tế. Vì vậy, trong bài tập lớn mở rộng này, nhóm sẽ thực hiện đề tài "**Sử dụng hợp ngữ MIPS hiện thực trò chơi Ô ăn quan**".

## 2 Luật chơi Ô ăn quan:

### 2.1 Định nghĩa:

- **Bàn chơi:**

Ô ăn quan hay gọi tắt là ô quan, ô làng, là trò chơi dân gian quen thuộc, hấp dẫn, gắn liền với tuổi thơ của hầu hết các trẻ em Việt Nam.

Bàn chơi được kẻ thành một hình chữ nhật rồi chia thành 10 ô vuông, mỗi bên có 5 ô đối xứng nhau. Ở 2 cạnh chiều rộng của hình chữ nhật, vẽ 2 ô hình bán nguyệt hướng ra phía ngoài. Các ô hình vuông gọi là ô dân còn 2 ô hình bán nguyệt gọi là ô quan.



- **Quân chơi:**

Bao gồm 2 loại quân: Dân và Quan và có thể được làm bằng từ nhiều vật liệu như sỏi, gạch, đá, hạt ... có kích thước vừa phải để người chơi dễ cầm nắm cùng lúc nhiều quân. Quan có kích thước to hơn hẳn các quân Dân để dễ dàng phân biệt. Số lượng quân chơi quan luôn là 2 còn dân có số lượng tùy theo luật chơi, phổ biến nhất là 50.



- **Bố trí quân chơi:**

Quan được đặt trong 2 ô hình bán nguyệt, mỗi ô 1 quân. Dân được bố trí vào các ô vuông với số quân đều nhau. Trong đề tài mở rộng này, nhóm sẽ dùng **số nguyên** để biểu thị số lượng quan và dân.

- **Người chơi:**

Thường gồm hai người chơi, mỗi người ngồi ở phía ngoài cạnh dài hơn của hình chữ nhật và những ô vuông bên nào thuộc quyền kiểm soát của người chơi ngồi bên đó.

## 2.2 Luật chơi

- **Tiến hành trò chơi:**

Người chơi đầu tiên sẽ chọn 1 ô của mình, gồm 5 quân, chọn chiều đi quân rồi rải lần lượt từng quân vào các ô bên cạnh.

Khi rải hết quân, có các tình huống như sau:

1. Nếu liền sau đó là một **ô vuông có chứa quân** thì tiếp tục dùng tất cả số quân đó để rải tiếp theo chiều đã chọn. Tuy nhiên nếu ô đó là một ô Quan thì chỉ được phép lấy 1 quân để rải.
2. Nếu liền sau đó là một **ô trống** rồi đến một ô có chứa quân thì người chơi sẽ được ăn tất cả số quân trong ô đó. Số quân bị ăn sẽ được loại ra khỏi bàn chơi để người chơi tính điểm khi kết thúc.
3. Nếu liền sau ô có quân đã bị ăn lại là một ô trống rồi đến một ô có quân nữa thì người chơi có quyền ăn tiếp cả quân ở ô này... Do đó trong cuộc chơi có thể có phương án rải quân làm cho người chơi ăn hết toàn bộ số quân trên bàn chơi chỉ trong một lượt đi của mình.
4. Nếu liền sau đó là **ô quan không có dân** hoặc 2 ô trống trở lên thì người chơi bị mất lượt và quyền đi tiếp thuộc về đối phương.

- **Các lượt chơi Ô quan quan:**

Từng người chơi sẽ chơi lần lượt nối tiếp nhau như vậy theo các tính toán đã đặt ra. Trường hợp đến lượt đi nhưng cả năm ô vuông thuộc quyền kiểm soát của người chơi đều không có dân thì người đó sẽ phải dùng 5 dân đã ăn được của mình để đặt vào mỗi ô 1 dân để có thể thực hiện việc di chuyển quân. Nếu người chơi không đủ 5 dân thì phải vay của đối phương và trả lại khi tính điểm.

- **Chiến thắng trò chơi Ô ăn quan:**

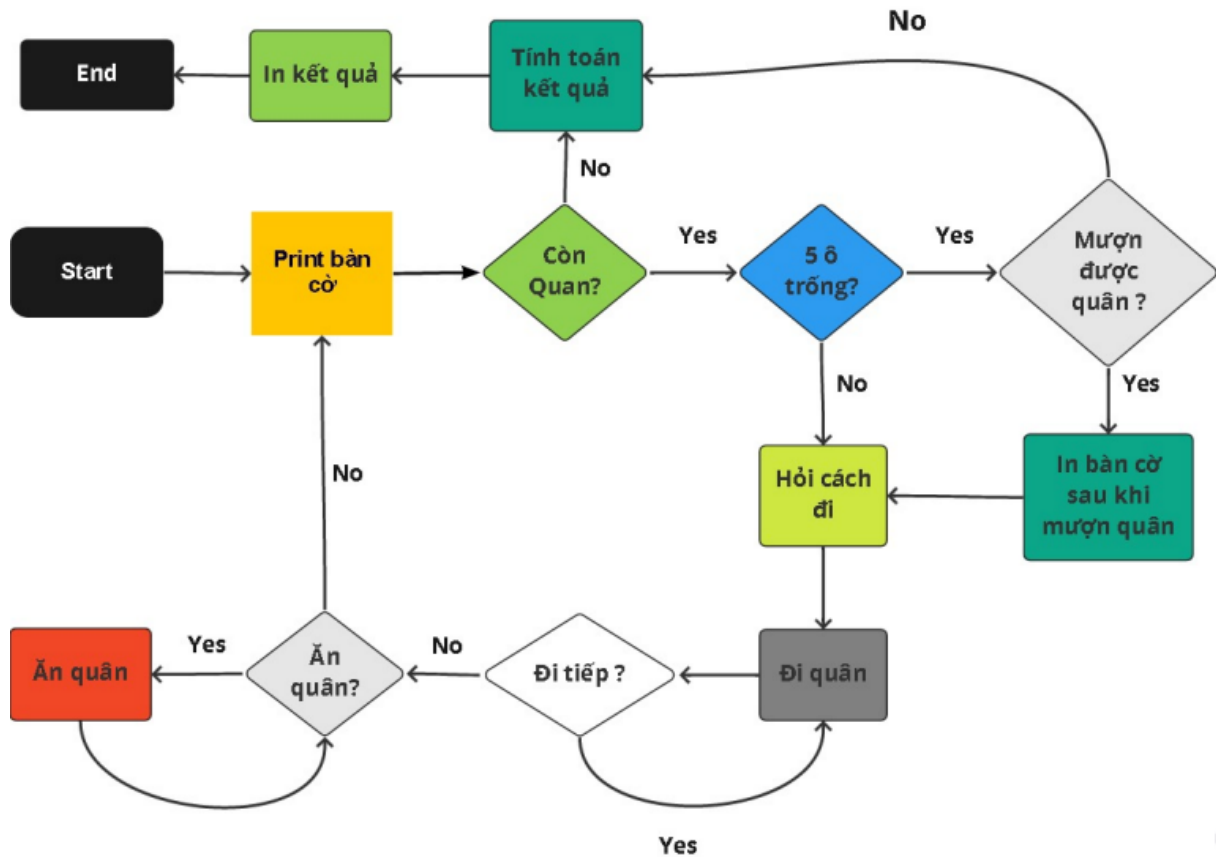
Trò chơi sẽ kết thúc khi Quan của hai bên đều bị ăn mất. Trường hợp hai ô quan đã bị ăn hết nhưng vẫn còn dân thì quân trong những hình vuông phía bên nào coi như thuộc về người chơi bên ấy. Người **thắng cuộc** trong trò chơi này là người mà khi cuộc chơi kết thúc có **tổng số dân quy đổi nhiều hơn**.

- **Một số quy ước trong hiện thực:**

1. Quy đổi điểm của quân: Quan tương ứng với 5 điểm, dân tương ứng với 1 điểm.
2. Trường hợp 5 ô của người chơi đều trống thì mượn quân của đối thủ, trường hợp đối thủ chưa có đủ 5 quân thì coi như trò chơi dừng lại.

## 3 Giải thuật hiện thực:

Dựa vào Luật chơi Ô ăn quan ở trên, nhóm xây dựng lưu đồ giải thuật cho bài toán như hình bên dưới.



### Hiện thực trên MIPS:

- **Biểu diễn:**

- numOfPawn: Mảng **int** có 12 phần tử với dạng: [0, 1, 2, ..., 6, 7,...,10, 11] sẽ lưu **số dân** có trong từng ô.
- sA, sB: Biến **int** lưu số điểm hiện tại của A và B.
- sBorrowA, sBorrowB: Biến **int** lưu số quân hiện tại A mượn B và B mượn A.
- Ban đầu khởi tạo mỗi nhà có 5 dân, kí hiệu bằng số 5. Còn quân được thể hiện bằng kí hiệu "X"

- **Pseudo code:**

```

While (true){
    print
    tmp := checkEnd
    if tmp = true
        break
    tmp := checkBlock
    if tmp = true
        if borrowPawn = false
            break
        print("Ban co sau khi phan phoi")
        print
        interact
        processing
        calPoint
    }
    printResult
}
  
```

- **Giải thích 1 số hàm:**

1. **print**: không nhận vào tham số và thực hiện in ra bàn cờ hiện tại.
2. **checkEnd**: không nhận vào tham số và thực hiện kiểm tra xem có kết thúc chưa và trả về giá trị 1 vào thanh ghi \$v0 nếu dừng .

**Ý tưởng:** Trò chơi kết thúc khi hai ô quan (ô 0 và ô 6) không còn quân.

3. **checkBlock**: nhận 1 tham số là \$a0 hoặc bằng 0 nếu là người chơi A hoặc bằng 1 nếu là người chơi B, kiểm tra xem hiện tại người chơi có thể thực hiện đi quân hay không, nếu có trả về giá trị 0 trong thanh ghi \$v0.

**Ý tưởng:**

- Người chơi A có thể thực hiện đi quân khi ở trong 5 nhà (ô 1-5 của board) có tồn tại ít nhất một ô chứa quân.
- Người chơi B có thể thực hiện đi quân khi ở trong 5 nhà (ô 7-11 của board) có tồn tại ít nhất một ô chứa quân.

4. **borrowPawn**: nhận 1 tham số là \$a0 hoặc bằng 0 nếu là người chơi A hoặc bằng 1 nếu là người chơi B, thực hiện kiểm tra có thể phân phối lại hoặc có thể mượn hay không, nếu có thì thực hiện phân phối lại và trả về giá trị 1 trong thanh ghi \$v0.

**Ý tưởng:**

- Nếu trường hợp đến lượt đi của player này nhưng lúc này không đủ quân để rải ( số quân ăn được < 5) thì phải mượn quân của player kia.
- Nếu số quân của player kia cũng < 5 thì xem như mượn không được.

5. **interact**: nhận 1 tham số là \$a0 hoặc bằng 0 nếu là người chơi A hoặc bằng 1 nếu là người chơi B, thực hiện tương tác với người dùng và trả về các giá trị: \$v0 thể hiện ô người dùng chọn từ 1 đến 5 và \$v1 trả về hướng đi hoặc 1 nếu là trái hoặc 0 nếu là phải. Hàm trả về địa chỉ của ô player chọn để di chuyển.

6. **processing**: nhận vào 3 tham số: \$a0 hoặc bằng 0 nếu là người chơi A hoặc bằng 1 nếu là người chơi B, \$a1 thể hiện ô người dùng chọn từ 1 đến 5, \$a2 là hoặc 1 nếu là trái hoặc 0 nếu là phải, hàm trả về giá trị idx của ô mà nó dừng vào \$v0. Hàm trả về địa chỉ ô cuối cùng trước khi dừng sau khi loop process.

7. **calPoint**: nhận 2 tham số là \$a0 hoặc bằng 0 nếu là người chơi A hoặc bằng 1 nếu là người chơi B, \$a1 là vị trí ô dừng trước đó, chức năng là thực hiện tính toán lại số điểm trong biến toàn cục sA và sB.

## 4 Hướng dẫn chơi và Kiểm thử chương trình:

### 4.1 Hướng dẫn chơi:

Khi chạy chương trình, giao diện tương tác sẽ như sau:

```

Left <- Player A -> Right
  1  2  3  4  5
/  |  |  |  |  |  |  |  \
|  | 05 | 05 | 05 | 05 | 05 |  | |
| X |  |  |  |  |  |  | X |
| 00 |-----| 00 |
|  |  |  |  |  |  |  |  |
|  | 05 | 05 | 05 | 05 | 05 |  |
\  |  |  |  |  |  |  |  /
  5  4  3  2  1
Left <- Player B -> Right
Current score A: 0          Current score B: 0
Current borrow score A: 0    Current borrow score B: 0
__PLAYER A__
Input your block (from 1 to 5): |

```

*Giải thích giá trị:*

- **Current score A, Current score B:** Giá trị điểm số của **PLAYER A** và **PLAYER B** trong quá trình trò chơi diễn ra, sẽ được cập nhật sau khi mỗi người chơi đi xong lượt của mình.
- **Current borrow score A, Current borrow score B:** Số quân mà **PLAYER A** và **PLAYER B** mượn đối thủ trong quá trình diễn ra trò chơi, sẽ được cập nhật khi người chơi có xảy ra mượn quân.
- **Input your block:** yêu cầu nhập block mà người chơi muốn di chuyển trong quá trình chơi cờ, giá trị nhập vào là một số từ 1 đến 5. Chương trình sẽ sử dụng quân cờ này để di quân theo luật của Ô ăn quan.
- **Input your way:** yêu cầu nhập hướng di chuyển mà người dùng muốn di quân, giá trị nhập vào chỉ nhận giá trị là 0 hoặc 1. Chương trình sẽ sử dụng quân cờ trong block đã chọn và đi theo hướng như ý muốn của người dùng.

Khi bắt đầu, chương trình mặc định **PLAYER A** sẽ là người đi trước, chương trình sẽ yêu cầu người nhập block cần để di chuyển quân, mặc định đánh số từ 1 đến 5. Sau khi chọn block, chương trình sẽ yêu cầu người dùng chọn hướng đi, **nhập 0** tương ứng với đi quân về **bên phải**, **nhập 1** tương ứng với đi quân về **bên trái**.

Giả sử người dùng nhập vào `block = 4`, và nhập chiều di chuyển là 0, tức là lấy quân ở ô có số thứ tự 4 của người chơi A và di chuyển về phía bên phải. Chương trình sẽ tiếp tục in ra bàn cờ sau khi đi xong, và thông báo tới lượt chơi của **PLAYER B**, yêu cầu người dùng nhập block cần di quân:

```

__PLAYER A__
Input your block (from 1 to 5): 4
Input your way (Right(0) and Left(1)): 0
    Left <- Player A -> Right
    1    2    3    4    5
/      |      |      |      |      |      |      \
|      | 06 | 06 | 06 | 00 | 00 |      |
| X   |   |   |   |   |   |   X   |
| 01 |-----| 01 |
|      |      |      |      |      |      |
|      | 06 | 00 | 06 | 06 | 06 |      |
\_____|_____|_____|_____|_____|_____|_____/
        5     4     3     2     1
    Left <- Player B -> Right
Current score A: 6           Current score B: 0
Current borrow score A: 0    Current borrow score B: 0
__PLAYER B__
Input your block (from 1 to 5): |

```

Giả sử người dùng nhập vào `block = 3`, và nhập chiều di chuyển là 1, tức là lấy quân ở ô có số thứ tự 3 của người chơi B và di chuyển về phía bên trái. Chương trình sẽ in ra màn hình bàn cờ sau khi người chơi **PLAYER B**, thông báo đến lượt chơi của người chơi **PLAYER A** và yêu cầu người dùng nhập `block` cần di quân:

```

__PLAYER B__
Input your block (from 1 to 5): 3
Input your way (Right(0) and Left(1)): 1
    Left <- Player A -> Right
    _____
    /      |      |      |      |      |      |      \
    |      | 07 | 07 | 07 | 00 | 00 |      |      |
    | X   |   |   |   |   |   |   | X   |
    | 02 |-----| 01 |
    |      |   |   |   |   |   |   |      |
    |      | 07 | 01 | 00 | 06 | 06 |      |
    \_____|_____|_____|_____|_____|_____|_____/
          5     4     3     2     1
    Left <- Player B -> Right
Current score A: 6          Current score B: 0
Current borrow score A: 0   Current borrow score B: 0
__PLAYER A__
Input your block (from 1 to 5): |

```

Người chơi sẽ tiếp tục nhập các số liệu yêu cầu chương trình đi quân theo mong muốn cho đến khi chương trình dừng lại và tổng kết điểm cho người chơi.

## 4.2 Kiểm thử chương trình:

Trong phần này, nhóm sẽ kiểm thử chương trình bằng một số bước đi.

```

    Left <- Player A -> Right
    _____
    /      |      |      |      |      |      |      \
    |      | 05 | 05 | 05 | 05 | 05 |      |      |
    | X   |   |   |   |   |   |   | X   |
    | 00 |-----| 00 |
    |      | 05 | 05 | 05 | 05 | 05 |      |
    \_____|_____|_____|_____|_____|_____|_____/
          5     4     3     2     1
    Left <- Player B -> Right
Current score A: 0          Current score B: 0
Current borrow score A: 0   Current borrow score B: 0

```

**PLAYER A:** Chọn block 4 và đi quân về phía bên phải.

```

__PLAYER A__
Input your block (from 1 to 5): 4
Input your way (Right(0) and Left(1)): 0
    Left <- Player A -> Right
    _____
    /      |      |      |      |      |      |      \
    |      | 06 | 06 | 06 | 00 | 00 |      |      |
    | X   |   |   |   |   |   |   | X   |
    | 01 |-----| 01 |
    |      | 06 | 00 | 06 | 06 | 06 |      |
    \_____|_____|_____|_____|_____|_____|_____/
          5     4     3     2     1
    Left <- Player B -> Right
Current score A: 6          Current score B: 0
Current borrow score A: 0   Current borrow score B: 0

```



**PLAYER B:** Chọn block 2 và đi quân về phía bên phải.

```

__PLAYER B__
Input your block (from 1 to 5): 2
Input your way (Right(0) and Left(1)): 0
Left <- Player A -> Right
  1     2     3     4     5
/-----\
|  X  | 00 | 07 | 07 | 01 | 01 |  X  |
| 02 |-----| 02 |
| 07 | 01 | 07 | 01 | 08 | 02 |
|-----\
  5     4     3     2     1
Left <- Player B -> Right
Current score A: 6          Current score B: 0
Current borrow score A: 0   Current borrow score B: 0

```

**PLAYER A:** Chọn block 5 và đi quân về phía bên trái.

```

__PLAYER A__
Input your block (from 1 to 5): 5
Input your way (Right(0) and Left(1)): 1
Left <- Player A -> Right
  1     2     3     4     5
/-----\
|  X  | 02 | 09 | 01 | 03 | 01 |  X  |
| 04 |-----| 00 |
| 09 | 00 | 09 | 03 | 00 | 00 |
|-----\
  5     4     3     2     1
Left <- Player B -> Right
Current score A: 14         Current score B: 0
Current borrow score A: 0   Current borrow score B: 0

```

**PLAYER B:** Chọn block 2 và đi quân về phía bên phải.

```

__PLAYER B__
Input your block (from 1 to 5): 2
Input your way (Right(0) and Left(1)): 0
Left <- Player A -> Right
  1     2     3     4     5
/-----\
|  X  | 03 | 10 | 02 | 00 | 02 |  X  |
| 04 |-----| 01 |
| 09 | 00 | 09 | 00 | 01 | 00 |
|-----\
  5     4     3     2     1
Left <- Player B -> Right
Current score A: 14         Current score B: 0
Current borrow score A: 0   Current borrow score B: 0

```

Sau một số bước chơi, chương trình sẽ có thể xảy ra tình huống như sau:

```

      Left <- Player A -> Right
      1 2 3 4 5
      /-----\
      | 11 | 00 | 00 | 00 | 00 |
      | X  | 13 |-----| 06 |
      |    |    |    |    |    |
      | 00 | 00 | 00 | 00 | 00 |
      \-----/
      5 4 3 2 1
      Left <- Player B -> Right
      Current score A: 14      Current score B: 11
      Current borrow score A: 0    Current borrow score B: 0
      __PLAYER A__
      Input your block (from 1 to 5): 1
      Input your way (Right(0) and Left(1)): 0
      Left <- Player A -> Right
      1 2 3 4 5
      /-----\
      | 00 | 00 | 01 | 01 | 01 |
      | X  | 14 |-----| 07 |
      |    |    |    |    |    |
      | 01 | 01 | 01 | 01 | 01 |
      \-----/
      5 4 3 2 1
      Left <- Player B -> Right
      Current score A: 15      Current score B: 11
      Current borrow score A: 0    Current borrow score B: 0
  
```

Giả sử **PLAYER B** chọn block tiếp theo để di chuyển là 5, và di chuyển về bên phải thì cuộc chơi sẽ kết thúc, chương trình sẽ in ra dòng thông báo điểm của cả 2 người chơi.

```

      __PLAYER B__
      Input your block (from 1 to 5): 5
      Input your way (Right(0) and Left(1)): 0
      Left <- Player A -> Right
      1 2 3 4 5
      /-----\
      | 00 | 00 | 00 | 00 | 00 |
      | 00 |-----| 00 |
      |    |    |    |    |    |
      | 00 | 00 | 00 | 00 | 00 |
      \-----/
      5 4 3 2 1
      Left <- Player B -> Right
      Current score A: 15      Current score B: 45
      Current borrow score A: 0    Current borrow score B: 0
      Total score A: 15
      Total score B: 45
      -- program is finished running (dropped off bottom) --
  
```

## 5 Kết luận:

**Ô ăn quan** là trò chơi dân gian phổ biến tại Việt Nam. Việc vận dụng hợp ngữ MIPS để hiện thực trò chơi này sẽ giúp sinh viên hệ tài năng nâng cao kỹ năng lập trình bằng hợp ngữ, kỹ năng làm việc nhóm cũng như ứng dụng những kiến thức đã học cho các vấn đề thực tế. Từ đó, giúp cho sinh viên có khả năng mở rộng thêm các kiến thức đã học cho các vấn đề yêu cầu kiến thức của ngành máy tính cần giải quyết trong cuộc sống.

## Tài liệu

- [1] Phạm Quốc Cường, *Giáo trình Kiến trúc máy tính*, Nhà xuất bản Đại học Quốc gia TP. Hồ Chí Minh, 2019.
- [2] William Stallings, *Computer Organization and Architecture: Design for Performance*, Prentice Hall, 4th edition, 2010.
- [3] David A. Patterson and John L.Hennessy, *Computer Organization and Design: The Hardware/software Interface*, Morgan Kaufmann, 6th edition, 2020.
- [4] Trích nguồn: <https://thuthuatchoi.com/cach-choi-o-an-quan-danh-cho-2-nguoi-choi.html>