

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



KIẾN TRÚC MÁY TÍNH (CO2008)

Bài tập lớn

BATTLESHIP

GIẢNG VIÊN: PGS. TS PHẠM QUỐC CƯỜNG
SINH VIÊN: NGUYỄN TUẤN HUY - 2211253

TP. HỒ CHÍ MINH, THÁNG 11/2023



Mục lục

1	Giới thiệu	2
2	Flowchart	4
3	Khởi tạo	4
4	Hiện thực chương trình	5
4.1	Giai đoạn thiết lập	5
4.1.1	Xử lý dữ liệu nhập vào	6
4.1.1.a	Kích thước tàu	6
4.1.1.b	Tọa độ đầu và đuôi tàu	7
4.1.2	Cập nhật bản đồ	8
4.2	Giai đoạn chiến đấu	8
4.2.1	Xử lý dữ liệu nhập vào	9
4.2.1.a	Ô muốn tấn công	9
4.2.2	Cập nhật bản đồ	9
4.3	File ghi dữ liệu	10

1 Giới thiệu

Trong bài tập lớn này, ta sẽ tái hiện lại trò chơi Battleship bằng ngôn ngữ lập trình MIPS assembly. Vì tài nguyên hạn chế và để đơn giản hóa mọi thứ, ta sẽ làm việc trên bản đồ nhỏ có kích thước **7x7**. Chương trình thực hiện các công việc sau:

- Một lưu ý trước khi vào chương trình: Vị trí của một con tàu được chỉ định bằng tọa độ của đầu và đuôi của tàu (row_{bow} , $column_{bow}$, row_{stern} , $column_{stern}$). Ở [Hình 1](#), tọa độ của tàu vàng 4x1 là “0 0 0 3”, những tàu xanh dương 2x1 là “1 4 2 4”, “2 1 3 1”, và “4 4 4 5”; những tàu xanh lá 3x1 là “5 3 5 5” và “6 0 6 2”.
- Ta sẽ bắt đầu với giai đoạn chuẩn bị, nơi mà người chơi sẽ thiết lập vị trí các con tàu của họ. Giá trị của các ô trên bản đồ sẽ là **1** hoặc **0**. 1 có nghĩa là ô đó đã bị chiếm bởi một chiếc tàu, còn 0 có nghĩa là ô đó không bị chiếm. Người chơi sẽ cần thiết lập chính xác số lượng tàu cùng với kích thước để hoàn thành giai đoạn thiết lập. Cụ thể là, mỗi người chơi sẽ có 3 tàu **2x1**, 2 tàu **3x1**, và 1 tàu **4x1**. Lưu ý rằng các tàu không được chồng lên nhau. [Hình 1](#) là một ví dụ về cách sắp xếp chấp nhận được.

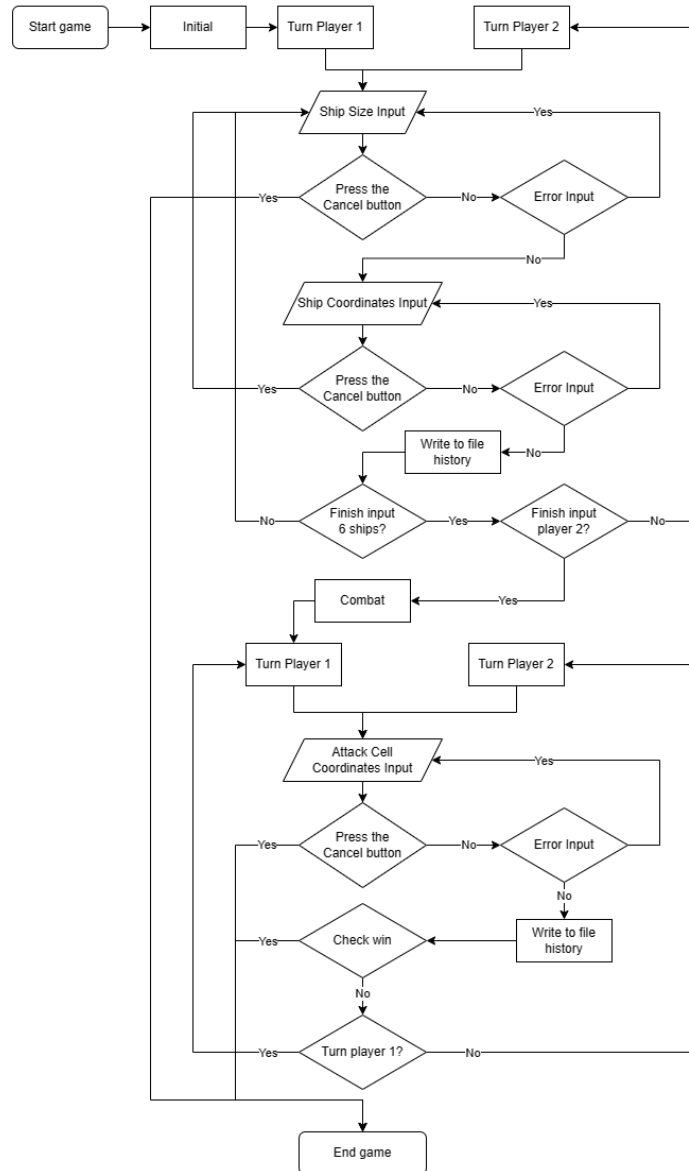
1	1	1	1	0	0	0
0	0	0	0	1	0	0
0	1	0	0	1	0	0
0	1	0	0	0	0	0
0	0	0	0	1	1	0
0	0	0	1	1	1	0
1	1	1	0	0	0	0

Hình 1: Ví dụ về những chiếc tàu mà người chơi có thể đặt trong trò chơi này.



- Sau giai đoạn thiết lập, cả hai người chơi sẽ lần lượt tấn công một ô trên bản đồ của đối phương mà họ không thấy trước. Nếu tấn công trúng một ô đã bị chiếm bởi tàu, một thông báo sẽ xuất hiện trên dialog là “HIT!”. Điều này sẽ cho phép người tấn công biết liệu họ có đánh trúng tàu đối thủ hay không. Người chơi sẽ cần nhớ các ô mà họ đã tấn công. Nếu một ô bị tấn công, giá trị của nó sẽ thay đổi thành **0**, vì nó không còn được chiếm bởi tàu nữa. Một chiếc tàu bị phá hủy hoàn toàn nếu tất cả các ô mà nó đang chiếm bị tấn công. Một người chơi sẽ thua nếu tất cả các tàu của họ bị phá hủy trước. Nói cách khác, trò chơi sẽ kết thúc khi bảng của một người chơi chỉ chứa các giá trị **0**.
- Sau khi trò chơi kết thúc, người chơi sẽ được xem lại vị trí đã đặt tàu, lịch sử các ô tấn công của bản thân và đối thủ.

2 Flowchart



3 Khởi tạo

Mỗi người chơi sẽ được khởi tạo 2 bản đồ 7×7 gồm bản đồ đặt tàu và bản đồ để lưu vị trí các ô tấn công trúng bằng mảng 1 chiều có độ dài 49 chứa các kí tự có kiểu byte. Tọa độ $A_{i,j}$ trong bản đồ 7×7 sẽ nằm ở vị trí $7 \times i + j$ trong mảng 1 chiều.

Dùng syscall với $\$v0 = 13$ và $\$a1 = 1$ (ghi file) mở 2 file `history_player_1.txt` và `history_player_2.txt` để lưu vị trí đặt tàu và lịch sử các ô tấn công của mỗi người chơi. **Hình 2**

open file	13	$\$a0$ = address of null-terminated string containing filename $\$a1$ = flags $\$a2$ = mode	$\$v0$ contains file descriptor (negative if error). <i>See note below table</i>
-----------	----	---	---

Hình 2: Syscall với $\$v0 = 13$.

4 Hiện thực chương trình

4.1 Giai đoạn thiết lập

Trong giai đoạn này, sẽ xuất hiện thông báo cho người chơi biết rằng còn bao nhiêu tàu cần đặt, bản đồ đánh dấu những vị trí đã đặt tàu và yêu cầu người chơi sẽ nhập kích thước của tàu, tọa độ đầu, đuôi của tàu. Người chơi sẽ phải nhập đúng định dạng đầu vào, nếu không sẽ có thông báo lỗi xuất hiện và bắt buộc phải nhập lại.

Thông báo số tàu còn lại mà người chơi cần phải nhập và bản đồ với những tàu đã đặt

```
You have 3 2x1 ships, 2 3x1 ships, and 1 4x1 ship
0 1 2 3 4 5 6
```

```
-----
0 | 0 0 0 0 0 0 0 |
1 | 0 0 0 0 0 0 0 |
2 | 0 0 0 0 0 0 0 |
3 | 0 0 0 0 0 0 0 |
4 | 0 0 0 0 0 0 0 |
5 | 0 0 0 0 0 0 0 |
6 | 0 0 0 0 0 0 0 |
-----
```

Thông báo nhập kích thước tàu

Choose a type ship (2,3,4)

Dữ liệu nhập của kích thước tàu có định dạng như sau:

size

Trong đó: **size** là các số 2, 3, 4 tương ứng với các kích thước tàu 2x1, 3x1, và 4x1.

Thông báo nhập tọa độ đầu và đuôi tàu

Enter the location of the bow and the stern of the ship (row col row1 col1)

Dữ liệu nhập tọa độ của đầu, đuôi tàu có định dạng như sau:

row_col_row1_col1

Trong đó:

- Đầu tàu nằm ở hàng **row** và cột **col**.
- Đuôi tàu nằm ở hàng **row1** và cột **col1**.
- **row**, **col**, **row1**, **col1**: Là một số nguyên có giá trị từ 0 đến 6.

4.1.1 Xử lý dữ liệu nhập vào

4.1.1.a Kích thước tàu

Sử dụng syscall với $\$v0 = 51$ để yêu cầu người chơi nhập số nguyên **size**. **Hình 3**.

InputDialogInt	51	$\$a0$ = address of null-terminated string that is the message to user	$\$a0$ contains int read $\$a1$ contains status value 0: OK status -1: input data cannot be correctly parsed -2: Cancel was chosen -3: OK was chosen but no data had been input into field
----------------	----	--	---

Hình 3: Syscall với $\$v0 = 51$.

1. Kiểm tra kiểu dữ liệu nhập, nếu dữ liệu nhập không phải là số nguyên ($\$a1 = -1$) hoặc dữ liệu rỗng ($\$a1 = -3$) thì sẽ có thông báo (1) nhập lại.
2. Kiểm tra kích thước tàu nhập, nếu **size** < 2 hoặc **size** > 4 đó thì sẽ có thông báo (1) nhập lại.

(1) Thông báo nhập kích thước tàu không đúng định dạng

Type ship range from 2 to 4. Enter again

3. Kiểm tra loại tàu cần nhập, nếu không còn loại tàu đó thì sẽ có thông báo (2) nhập lại.

(2) Thông báo không còn loại tàu cần nhập

You no longer have this type of boat. Enter again

4. Nếu người chơi bấm nút Cancel ($\$a1 = -2$) thì trò chơi sẽ kết thúc.

4.1.1.b Tọa độ đầu và đuôi tàu

Sử dụng syscall với $\$v0 = 54$ và độ dài chuỗi lớn nhất để đọc là 8 ($\$a2 = 8$, độ dài chuỗi nhập là 7 và kí tự `'\n'` khi nhấn nút OK) để yêu cầu người chơi nhập chuỗi `row_col_row1_col1`.
Hình 4.

InputDialogString	54	<p>$\\$a0$ = address of null-terminated string that is the message to user</p> <p>$\\$a1$ = address of input buffer</p> <p>$\\$a2$ = maximum number of characters to read</p>	<p>See Service 8 note below table</p> <p>$\\$a1$ contains status value</p> <p>0: OK status. Buffer contains the input string.</p> <p>-2: Cancel was chosen. No change to buffer.</p> <p>-3: OK was chosen but no data had been input into field. No change to buffer.</p> <p>-4: length of the input string exceeded the specified maximum. Buffer contains the maximum allowable input string plus a terminating null.</p>
-------------------	----	--	--

Hình 4: Syscall với $\$v0 = 54$.

- Kiểm tra độ dài dữ liệu nhập (str), nếu độ dài chuỗi nhập lớn hơn 7 ($\$a1 = -4$) hoặc chuỗi rỗng ($\$a1 = -3$) thì sẽ có thông báo (3) nhập lại.
- Kiểm tra các kí tự nhập (gồm 4 số nằm trong khoảng từ 0 đến 6 và cách nhau bởi dấu `_`), nếu $str[0], str[2], str[4], str[6] < '0'$ hoặc $> '6'$ và $str[1], str[3], str[5] \neq '_'$ thì sẽ có thông báo (3) nhập lại.
- Kiểm tra tọa độ đầu và đuôi có trùng nhau hoặc không thẳng hàng, nếu $str[0] = str[2]$ và $str[1] = str[3]$ hoặc $str[0] \neq str[2]$ và $str[1] \neq str[3]$ thì sẽ có thông báo (3) nhập lại.

(3) Thông báo nhập tọa độ đầu và đuôi tàu không đúng định dạng

Location range from 0 to 6. Enter again (row col row1 col1)

- Kiểm tra khoảng cách giữa 2 tọa độ đầu và đuôi tàu có bằng kích thước tàu đã chọn ở trên hay không, nếu $|str[0] - str[2]| \neq size$ (tàu nằm dọc) hoặc $|str[1] - str[3]| \neq size$ (tàu nằm ngang) thì sẽ có thông báo (4) nhập lại.

(4) Thông báo nhập tọa độ đầu và đuôi tàu không đúng loại tàu đã chọn ở trên

Please enter correct length of 4x1 ship

- Kiểm tra tàu vừa nhập có bị chồng lấn, tàu nằm ngang thì duyệt các ô nằm ở hàng $str[0]$ và cột từ $str[1]$ đến $str[3]$, tàu nằm dọc thì duyệt các ô nằm ở hàng từ $str[0]$ đến $str[2]$ và cột $str[1]$ nếu đã có giá trị bằng `'1'` thì sẽ có thông báo (5) nhập lại.

(5) Thông báo tàu vừa nhập bị chồng lấn

The ships are overlapping with each other. Enter again

- Nếu người chơi bấm nút Cancel ($\$a1 = -2$) thì người chơi sẽ quay lại nhập kích thước tàu.

4.1.2 Cập nhật bản đồ

Nếu qua được các bước kiểm tra trên, ghi kích thước và tọa độ đầu, đuôi của tàu vừa đặt vào file. Các ô trong khoảng tọa độ vừa nhập của người chơi sẽ được cập nhật thành giá trị '1'.
Sử dụng syscall với $\$v0 = 15$ để ghi dữ liệu vào file

write to file	15	$\$a0$ = file descriptor $\$a1$ = address of output buffer $\$a2$ = number of characters to write	$\$v0$ contains number of characters written (negative if error). See note below table
---------------	----	---	--

Hình 5: Syscall với $\$v0 = 15$.

4.2 Giai đoạn chiến đấu

Trong giai đoạn này, sẽ xuất hiện thông báo cho người chơi biết rằng đã tấn công trúng những ô nào bị chiếm bởi thuyền trên bản đồ và yêu cầu người chơi sẽ nhập tọa độ ô muốn tấn công. Người chơi sẽ phải nhập đúng định dạng đầu vào, nếu không sẽ có thông báo nhập lại.

Thông báo những ô bị chiếm bởi thuyền mà người chơi tấn công trúng

Cells attacked by player 1

0 1 2 3 4 5 6

```
-----  
0 | 1 0 0 0 0 0 0 |  
1 | 1 1 0 0 0 0 0 |  
2 | 0 1 0 0 0 0 0 |  
3 | 0 1 0 0 0 0 0 |  
4 | 0 1 0 0 0 1 0 |  
5 | 0 0 0 0 0 1 0 |  
6 | 0 0 0 0 0 0 0 |  
-----
```

Thông báo nhập tọa độ ô người chơi muốn tấn công

Turn player 1. Enter target cell (row col)

Dữ liệu nhập tọa độ ô người chơi muốn tấn công có định dạng như sau:

row_col

Trong đó:

- ô muốn tấn công nằm ở hàng **row** và cột **col**.
- **row**, **col**: Là một số nguyên có giá trị từ 0 đến 6.

4.2.1 Xử lý dữ liệu nhập vào

4.2.1.a Ô muốn tấn công

Sử dụng syscall với $\$v0 = 54$ và độ dài chuỗi lớn nhất để đọc là 4 ($\$a2 = 8$, độ dài chuỗi nhập là 3 và kí tự `'\n'` khi nhấn nút OK) để yêu cầu người chơi nhập chuỗi `row_col`. Hình 4.

1. Kiểm tra độ dài dữ liệu nhập (str), nếu độ dài chuỗi nhập lớn hơn 3 ($\$a1 = -4$) hoặc chuỗi rỗng ($\$a1 = -3$) thì sẽ có thông báo (6) nhập lại.
2. Kiểm tra các kí tự nhập (gồm 2 số nằm trong khoảng từ 0 đến 6 và cách nhau bởi dấu `_`), nếu $str[0], str[2] < '0'$ hoặc $> '6'$ và $str[1] \neq '_'$ thì sẽ có thông báo (6) nhập lại.

(6) Thông báo nhập tọa độ ô muốn tấn công không đúng định dạng

Error location of the target cell. Enter again

3. Nếu người chơi bấm nút Cancel ($\$a1 = -2$) thì trò chơi sẽ kết thúc.

4.2.2 Cập nhật bản đồ

Nếu qua được các bước kiểm tra trên, ghi tọa độ của ô vừa tấn công vào file (Hình 5). Nếu tấn công đúng ô bị chiếm bởi thuyền, giá trị của ô đó trên bản đồ đối phương sẽ trở thành `'0'` đồng thời trên bản đồ lưu vị trí của người chơi sẽ cập nhật thành `'1'` và hiện thông báo (7) trên dialog.

(7) Thông báo tấn công trúng ô bị chiếm bởi thuyền

HIT!

Sau đó, duyệt qua hết các ô trong bản đồ đối phương, nếu có ô có giá trị `'1'` thì trò chơi sẽ tiếp tục với lượt của đối phương. Nếu không còn ô nào có giá trị `'1'` thì trò chơi sẽ kết thúc, hiện thông báo (8) chiến thắng thuộc về người chơi vừa nhập.

(8) Thông báo chiến thắng

YOU WIN!

Sử dụng syscall với $\$v0 = 16$ để đóng 2 file `history_player_1.txt` và `history_player_2.txt` khi trò chơi kết thúc. Hình 6.

close file	16	$\$a0 = \text{file descriptor}$	
------------	----	---------------------------------	--

Hình 6: Syscall với $\$v0 = 16$.



4.3 File ghi dữ liệu

Kết quả file history ghi được

```
Type ship: 4 at location 1 0 1 3
Type ship: 3 at location 2 3 4 3
Type ship: 3 at location 6 4 6 6
Type ship: 2 at location 5 1 5 2
Type ship: 2 at location 2 5 3 5
Type ship: 2 at location 0 6 1 6
0 0
1 0
1 1
2 1
3 1
4 1
3 3
3 4
4 5
5 5
6 2
6 3
6 4
1 6
2 6
3 6
YOU WIN!!
```