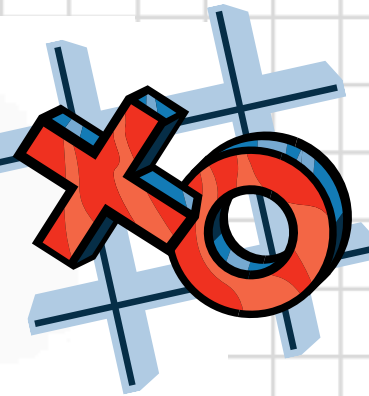
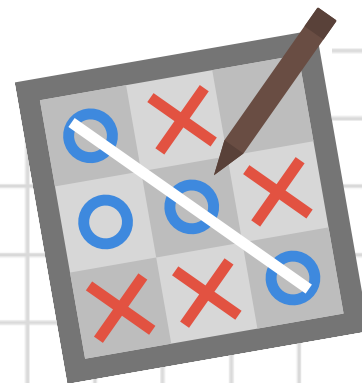
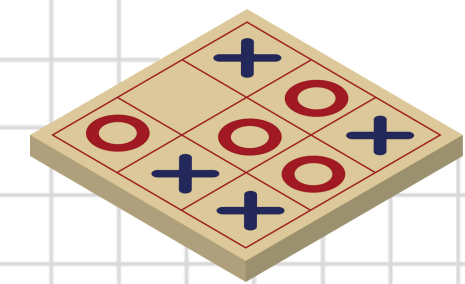
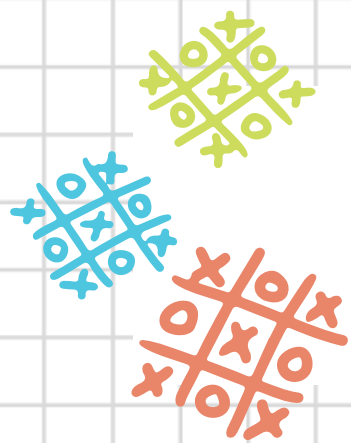


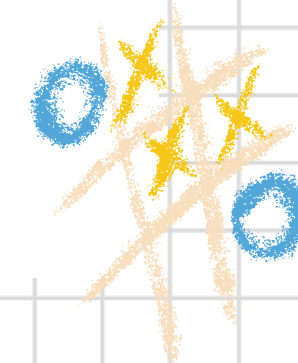
# Đề tài: "Programming Tic Tac Toe game on Emu8086"

Nhóm 5





# Giải thích mã nguồn



Phần khai báo dữ liệu trong chương trình:

```
1  .model small
2  .stack 100h
3  .data
4      ki_tu db '123456789'          ; mảng kí tự trong các ô (ban đầu là 1-9)
5      bang db 13,10, '   |   |   ',13,10, '_____',13,10, '   |   |   ',13,10, '_____',13,10,
        '   |   |   ',13,10,'$' ; giao diện bảng trên màn hình
6      pos dw 3,7,11,29,33,37,55,59,63 ; vị trí các 'ki_tu' trong 'bang' để cập nhật
7      msg_turn_x db 13,10, 'Nhập vị trí (1-9). Lượt X: $'
8      msg_turn_o db 13,10, 'Nhập vị trí (1-9). Lượt O: $'
9      msg_invalid db '      Không hợp lệ! Thu lại.$'
10     msg_x_win db 13,10, 'NGUOI THANG: X$'
11     msg_o_win db 13,10, 'NGUOI THANG: O$'
12     msg_hoa db 13,10, 'HOA VAN$'
13     msg_replay db 13,10, 'Chơi lại? (Y/N): $'
14     turn db 'X'                    ; lưu lượt chơi hiện tại: X hoặc O
15     turn_begin db 'O'              ; lưu lượt chơi đầu mỗi vòng
16     res db ' ', '$'                ; lưu kết quả: X, O hoặc H (hòa)
17     mang_win dw 1,2,3, 4,5,6, 7,8,9, 1,4,7, 2,5,8, 3,6,9, 1,5,9, 3,5,7 ; mảng lưu các bộ 3 vị trí thắng
18
```

# Phần chương trình chính:

```
19 .code
20 ▾ main proc
21     mov ax, @data
22     mov ds, ax
23     call init
24 ▾ game_loop:
25     call ve_bang
26     call nhap_nuoc
27     call check_end
28     cmp al, 1          ; kiểm tra al=1 (có người thắng/hòa)
29     je game_over
30     call doi_luot
31     jmp game_loop
32 ▾ game_over:
33     call ve_bang
34     mov ah, 9          ; dùng hàm 9 để in chuỗi
35     mov al, res        ; lấy kết quả (X, O hoặc H)
36     lea dx, msg_x_win
37     cmp al, 'X'        ; al=X thì sẽ in msg_x_win
38     je hien_kq
39     lea dx, msg_o_win
40     cmp al, 'O'        ; al=O thì sẽ in msg_o_win
41     je hien_kq
42     lea dx, msg_hoa    ; sẽ in msg_hoa
43 ▾ hien_kq:
44     int 21h            ; in kết quả
45     lea dx, msg_replay ; in thông báo hỏi chơi lại
46     int 21h
47     mov ah, 1          ; hàm 1 để nhập ký tự (lưu trong al)
48     int 21h
49     and al, 0dfh       ; chuyển ký tự thành chữ hoa
50     cmp al, 'Y'
51     jne exit           ; al=Y thì exit
52     call init          ; al≠Y thì reset game
53     jmp game_loop
54 ▾ exit:
55     mov ah, 4ch        ; hàm 4Ch thoát chương trình
56     int 21h
57 main endp
```

## Hàm init – Khởi tạo lại trò chơi

```
59  init proc
60      lea si, ki_tu          ; si trỏ đến mảng kí tự
61      mov bl, '1'           ; bl bắt đầu từ ký tự '1'
62      mov cx, 9              ; cx=9 để lặp 9 lần
63      reset:
64          mov [si], bl        ; gán giá trị bl vào ô hiện tại
65          inc bl              ; tăng bl lên 1
66          inc si              ; trỏ đến ô tiếp theo
67          loop reset          ; lặp reset đến khi cx=0
68      mov al, 'X'+'0'        ; đổi lượt chơi đầu bằng 'X'+'0' - turn_begin
69      sub al, turn_begin
70      mov turn_begin, al      ; cập nhật turn_begin và turn hiện tại
71      mov turn, al
72      ret                    ; trở về hàm gọi
73  init endp
```

# Phần chương trình chính:

```
19 .code
20 ▾ main proc
21     mov ax, @data
22     mov ds, ax
23     call init
24 ▾ game_loop:
25     call ve_bang
26     call nhap_nuoc
27     call check_end
28     cmp al, 1          ; kiểm tra al=1 (có người thắng/hòa)
29     je game_over
30     call doi_luot
31     jmp game_loop
32 ▾ game_over:
33     call ve_bang
34     mov ah, 9          ; dùng hàm 9 để in chuỗi
35     mov al, res        ; lấy kết quả (X, O hoặc H)
36     lea dx, msg_x_win
37     cmp al, 'X'        ; al=X thì sẽ in msg_x_win
38     je hien_kq
39     lea dx, msg_o_win
40     cmp al, 'O'        ; al=O thì sẽ in msg_o_win
41     je hien_kq
42     lea dx, msg_hoa    ; sẽ in msg_hoa
43 ▾ hien_kq:
44     int 21h            ; in kết quả
45     lea dx, msg_replay ; in thông báo hỏi chơi lại
46     int 21h
47     mov ah, 1          ; hàm 1 để nhập ký tự (lưu trong al)
48     int 21h
49     and al, 0dfh       ; chuyển ký tự thành chữ hoa
50     cmp al, 'Y'
51     jne exit           ; al≠Y thì exit
52     call init          ; al=Y thì reset game
53     jmp game_loop
54 ▾ exit:
55     mov ah, 4ch        ; hàm 4Ch thoát chương trình
56     int 21h
57 main endp
```

## Hàm ve\_bang – In bàn cờ

```
75  ve_bang proc
76      mov ax, 3          ; hàm 3 của ngắt 10h để xóa màn hình
77      int 10h
78      mov cx, 9          ; cx=9 để lặp 9 lần
79      mov si, 0          ; đặt si=0 (bắt đầu từ ô đầu tiên)
80      update_cell:      ; vòng lặp cập nhật từng ô
81          mov bx, si      ; bx=si*2 (vì pos lưu dạng dw cần 2B)
82          add bx, bx
83          mov di, pos[bx] ; lấy vị trí cần cập nhật trong chuỗi bang và lưu vào di
84          mov al, ki_tu[si] ; lấy ký tự từ mảng bang
85          mov bang[di], al ; cập nhật ký tự vào chuỗi hiển thị
86          inc si          ; tăng si (chuyển đến ô tiếp theo)
87          loop update_cell ; lặp update_cell đến khi cx=0
88      mov ah, 9
89      lea dx, bang        ; in bang
90      int 21h
91      ret
92  ve_bang endp
```



# Hàm nhap\_nuoc proc - Nhập nước đi của người chơi

```
94  ▾ nhap_nuoc proc
95      mov ah, 9                ; hàm 9 để in chuỗi
96      lea dx, msg_turn_x      ; giả sử là lượt X
97      cmp turn, 'X'           ; turn='X' thì in msg_turn_x
98      je hien_nhap
99      lea dx, msg_turn_o      ; không thì in msg_turn_o
100  ▾ hien_nhap:
101      int 21h                 ; in thông báo lượt chơi
102      mov ah, 1               ; hàm 1 để nhập ký tự (lưu trong al)
103      int 21h
104      cmp al, '1'
105      jl nhap_sai              ; nếu nhỏ hơn '1' thì nhập sai
106      cmp al, '9'
107      jg nhap_sai              ; nếu lớn hơn '9' thì nhập sai
108      sub al, '1'              ; chuyển '1'-'9' thành 0-8 (chỉ số mảng)
109      mov bl, al               ; lưu vào bl
110      mov bh, 0                ; bh=0 để bx=bl
111      lea si, ki_tu            ; si trỏ đến mảng kí tự
112      add si, bx                ; si= địa chỉ ô được chọn
113      mov al, [si]              ; lấy giá trị ô và kiểm tra ô đánh chưa (có x hoặc o)
114      cmp al, 'X'
115      je nhap_sai
116      cmp al, 'O'
117      je nhap_sai
118      mov al, turn              ; lấy lượt chơi hiện tại (X/O)
119      mov [si], al              ; đánh dấu X hoặc O vào ô
120      ret
121  ▾ nhap_sai:
122      mov ah, 9                ; hàm 9 để in chuỗi
123      lea dx, msg_invalid      ; in thông báo nước đi không hợp lệ
124      int 21h
125      jmp nhap_nuoc            ; quay lại nhập lại
126  nhap_nuoc endp
```


# Hàm check\_end - Kiểm tra thắng thua

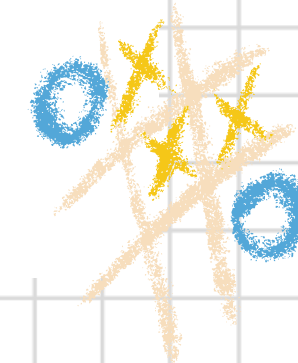
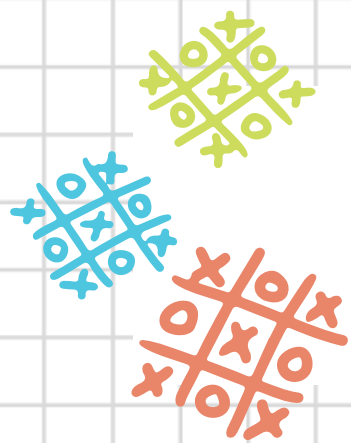
```
135  check_end proc
136      mov si, 0                ; si=0 để bắt đầu từ ô đầu tiên
137      mov cx, 8                ; cx=8 để lặp 8 lần
138      check_win:
139          mov bx, mang_win[si] ; lấy vị trí thứ 1 trong bộ 3 mang_win
140          mov ah, ki_tu[bx-1]  ; lấy ki_tu tại vị trí đó (trừ 1 vì mảng từ 0)
141          mov bx, mang_win[si+2] ; lấy vị trí thứ 2 trong bộ 3 mang_win
142          cmp ah, ki_tu[bx-1]  ; so sánh ki_tu tại vị trí 1 và 2
143          jnz chua_win        ; nếu khác nhau thì chưa thắng
144          mov bx, mang_win[si+4] ; lấy vị trí thứ 3 trong bộ 3 mang_win
145          cmp ah, ki_tu[bx-1]  ; so sánh ki_tu tại vị trí 1 và 3
146          jnz chua_win        ; nếu khác nhau thì chưa thắng
147          mov res, ah          ; nếu giống nhau thì lưu người thắng (X/O)
148          mov al, 1            ; al=1 báo hiệu game kết thúc
149          ret
150      chua_win:                ; xử lý khi chưa thắng ở bộ 3 hiện tại
151          add si, 6             ; tăng si lên 6 (mỗi bộ 3 chiếm 6 byte trong mang_win)
152          loop check_win       ; lặp check_win đến khi cx=0
153      lea si, ki_tu             ; si trỏ đến mảng kí tự
154      mov cx, 9                ; cx=9 để lặp 9 lần
155      check_hoa:
156          mov al, [si]          ; lấy giá trị ô hiện tại
157          cmp al, '9'           ; so sánh với '9'
158          jbe chua_hoa         ; nếu ≤ '9' thì còn ô chưa đánh → chưa hòa
159          inc si                ; tăng si (kiểm tra ô tiếp)
160          loop check_hoa       ; lặp check_hoa đến khi cx=0
161          mov res, 'H'          ; nếu tất cả ô đều đã đánh thì đánh dấu hòa
162          mov al, 1            ; al=1 báo hiệu game kết thúc
163          ret
164      chua_hoa:                ; xử lý khi chưa hòa
165          mov al, 0             ; al=0 (game chưa kết thúc)
166          ret
167  check_end endp
```



## Hàm doi\_luot proc - Đổi lượt chơi

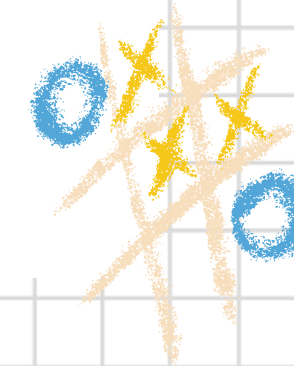
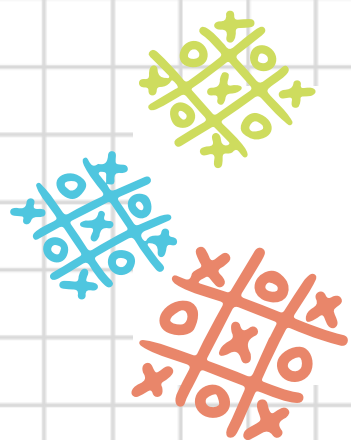
```
128  ✓ doi_luot proc
129      mov al, 'X'+'O'
130      sub al, turn
131      mov turn, al
132      ret
133  doi_luot endp
```





# Đánh giá hiệu quả

- ✓ Hiển thị được bảng trò chơi rõ ràng
- ✓ Nhận đầu vào từ người chơi và không cho phép đánh trùng ô
- ✓ Cập nhật luân phiên chơi và kiểm tra thắng/hòa
- ✓ Hiển thị thông báo thắng hoặc hòa sau khi trò chơi kết thúc



# Tài liệu tham khảo

Tài liệu Emu8086 User Manual:

- <https://fr.slideshare.net/slideshow/kin-thc-c-bn-v-lp-trnh-hp-ng-assembly/266393305>

Emu8086 Tutorial:

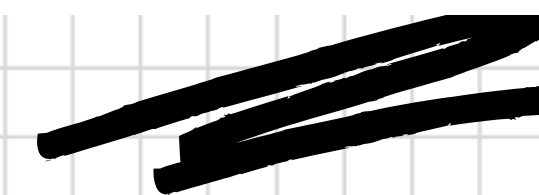
- <https://fr.scribd.com/doc/87705752/Emu8086-Tutorial>

Bài giảng môn Kiến trúc máy tính PTIT :

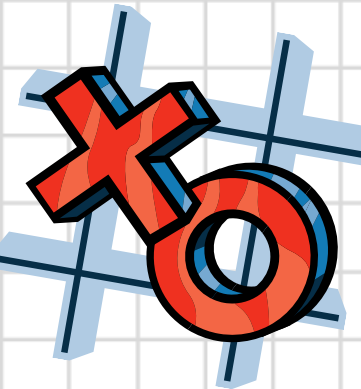
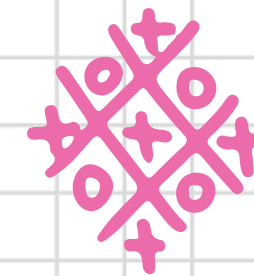
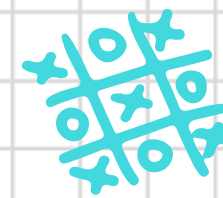
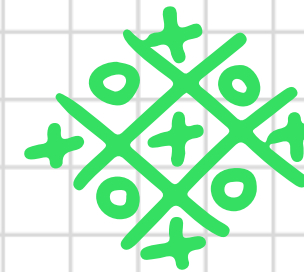
- [docx.com.vn/tai-lieu/giao-trinh-kien-truc-may-tinh-hoc-vien-cong-nghe-buu-chinh-vien-thong-50](http://docx.com.vn/tai-lieu/giao-trinh-kien-truc-may-tinh-hoc-vien-cong-nghe-buu-chinh-vien-thong-50)

Code

- <https://www.ee.nthu.edu.tw/jcliao/mic97/chap08/TICTAC.ASM>
- [https://github.com/trandinhhao/PTIT-KTMT-Assembly/blob/main/Tic\\_Tac\\_Toe.asm](https://github.com/trandinhhao/PTIT-KTMT-Assembly/blob/main/Tic_Tac_Toe.asm)



THANKS



FOR LISTENING

