

# Pràctiques Estructura de Computadors

2024-25

## Nivell Mig

### 1 Introducció

La pràctica consisteix en implementar un joc de 4 en ratlla, on tenim un tauler en vertical de 6 files i 7 columnes que ens permet inserir fitxes des de dalt de les diferents columnes, de manera que les fitxes cauen per la columna fins a la fila que troben plena. Aquesta estructura es troba emmagatzemada en una matriu anomenada `mBoard` (`int mBoard[6][7]`).

En una partida juguen dos persones que tenen fitxes de colors diferents (en el nostre cas 'O' i 'X'), i van inserint les seves fitxes de forma alternativa fins que una persona aconsegueix fer una fila de 4 fitxes seguides, ja sigui en horitzontal, vertical o diagonal.

La pràctica consta d'un programa en C, que us donem fet i **NO PODEU MODIFICAR** i un esquelet d'un programa en ensamblador en el que heu d'implementar les subrutines que us indiquem. El programa en C, genera un menú en pantalla i permet accedir a les diferents opcions del menú. Les diferents opcions del menú s'han d'implementar amb les subrutines en ensamblador que us indiquem.

El programa en ensamblador conté l'estructura bàsica de les subrutines que heu d'implementar i algunes subrutines ja implementades per a gestionar l'entrada/sortida.

#### 1.1 Objectius de la pràctica:

- Familiaritzar-se amb l'entorn de desenvolupament (Visual Studio).
- Aprendre els conceptes bàsics del llenguatge de baix nivell de l'arquitectura Intel x86.
- Observar els canvis que produeixen les instruccions sobre els registres, flags o registres d'estat i la memòria.
- Treballar en C i llenguatge Assembly Intel x86.

#### 1.2 Material

- Microsoft Visual Studio 2022 Community.
- Documentació sobre l'entorn i configuració del projecte al Campus Virtual.
- Esquelet del Projecte de la pràctica que inclou el programa en C, el fitxer d'encapçalament .h i l'esquelet del programa ensamblador (disponible al Campus Virtual).

#### 1.3 Desenvolupament de la pràctica:

- La pràctica es realitzarà en 6 sessions.
- Hi ha una prova de validació individual un cop hagin acabat les sessions de laboratori.
- La pràctica està dividida en 3 nivells: bàsic, mig i avançat. Quan un grup acabi un nivell, haurà de presentar-lo al membre de l'equip docent responsable del seu grup de pràctiques durant la sessió. Si el grup rep el vist i plau, haurà de lliurar el

seu codi al Campus Virtual i se li obrirà el següent nivell. No es poden saltar nivells (per exemple, presentar directament el nivell mig sense haver presentat el bàsic), ni presentar més d'un nivell per sessió.

- La nota de laboratori depèn del nivell assolit i la qualitat/funcionalitat de la solució proposada. El nivell bàsic dona dret a una nota fins a 6. El mig permet sumar fins a un punt més i l'avançat un altre punt més com a màxim. En el Campus Virtual disposeu d'un document on s' explica amb més detall el mètode d'avaluació de l'assignatura.
- Un cop finalitzades totes les sessions de laboratori caldrà lliurar una memòria final de pràctiques en format pdf, seguint la plantilla que us proporcionarem al Campus Virtual.

## 2 Nivell Mig (2-3 sessions):

En aquest segon nivell s'han d'implementar dues opcions més del menú i afegir noves funcionalitats que obligaran a fer alguna subrutina nova i a modificar algunes de les subrutines desenvolupades al nivell bàsic.

Les noves opcions del menú son:

- **7. Put 2 players:** En aquesta opció, s'ha de permetre que cadascun dels dos jugadors insereixi una fitxa. Primer ho farà el jugador 1 i després el 2. A la posició corresponent de la pantalla s'indicarà a quin jugador li toca introduir fitxa. El jugador 1 introdueix fitxes amb una 'O' i el jugador 2 amb una 'X'. Aquesta opció del menú crida a la subrutina *put2Players* que heu d'implementar en el programa en assemblador.
- **8. Play:** En aquesta opció s'ha de jugar de forma seguida, de manera que els dos jugadors van introduint fitxes de forma alternativa. Aquesta opció del menú crida a la subrutina *Play* que heu d'implementar en el programa en assemblador.

Les noves funcionalitats son:

- Determinar que un jugador ha aconseguit fer una ratlla de 4 fitxes posades en vertical. Aquesta comprovació es farà mitjançant una subrutina nova *checkRow* que haureu de cridar des de la vostra subrutina *putPiece*. Per a facilitar aquest control s'ha introduït:
  - una variable *inaRow* per a fer de comptador i determina si hi ha 4 fitxes en ratlla.
  - una variable *row4Complete* que indica que s'ha assolit la fita de les 4 fitxes en ratlla.

## Subrutines que cal implementar o modificar en assembler per al Nivell Mig:

- **checkRow:**

```
.....  
; A partir de la posició en la que ha quedat la fitxa introduïda comprovar si les fitxes  
; que hi ha per sota en vertical coincideixen amb la que acabem d'introduir.  
; Si coincideix incrementar el comptador i comprovar si hem arribat a 4.  
; Si no hem arribat a 4, seguim baixant.  
; Si hem arribat a 4, posem l'indicador row4Complete a 1.  
;  
; Variables utilitzades:  
; mBoard : matriu del taulell on anem inserint les fitxes  
; inaRow : comptador per a saber el nombre de fitxes en ratlla  
; row4Complete : indicador de si hem arribat a 4 fitxes en ratlla o no  
; pos : posició de la matriu mBoard que estem mirant  
;  
.....
```

- **putPiece:**

```
.....  
; Situa una fitxa en una posició lliure del tauler de joc  
; S'ha de cridar a movCursorContinuous per a triar la columna de la casella desitjada.  
; Un cop som a la columna de casella desitjada premem al tecla ' ' (espai "tirar" la fitxa)  
; Calcular la posició de la matriu corresponent a la posició més baixa de la columna que  
; ocupa el cursor, cridant a la subrutina calcIndex.  
; Anar pujant per al columna fins a trobar una posició buida i posar la fitxa en aquella posició de  
; mBoard.  
; Cridar a la subrutina showBoard per a mostrar com queda el tauler.  
; Si hem pitjat 'q' <Quit>, no hem d'introduir fitxa.  
;  
; Variables utilitzades:  
; row : fila per a accedir a la matriu mBoard  
; col : columna per a accedir a la matriu mBoard  
; colCursor: Columna en la que tenim el cursor  
; pos : índex per a accedir a la matriu mBoard  
; mBoard : matriu 6x7 on tenim el tauler  
; carac : caràcter per a escriure a pantalla  
; tecla: codi ascii de la tecla pitjada  
;  
.....
```

- **put2Players:**

```

; S'han de permetre dues tirades, cridant a putPiece,
; una de cada jugador,
; actualitzant a la casella corresponent de la pantalla
; l'identificador del jugador cridant a showPlayer.
; Si es pitja 'q'<Quit> o s'aconsegueix la ratlla de 4 fitxes
; el procés ha de finalitzar.
;
; Variables utilitzades:
; tecla: codi ascii de la tecla pitjada
; player: Variable que indica quin jugador fa la tirada
; row4Complete: variable que ha actualitzat la subrutina checkRow
; per a indicar si s'ha aconseguit una ratlla de 4 fitxes.

```

- **Play:**

```

; Es va cridant a put2Players mentre no es pitgi 'q'<Quit>
; o un jugador aconsegueixi una ratlla de 4 fitxes.
;
; Variables utilitzades:
; tecla: codi ascii de la tecla pitjada
; row4Complete: variable que ha actualitzat la subrutina checkRow
; per a indicar si s'ha aconseguit una ratlla de 4 fitxes.

```

### Nota:

El codi d'alumne que s'entrega disposa de totes les variables i els esquelets de les funcions d'ensamblador necessàries tant les del nivell bàsic com les nivell mig. Veureu que tots els esquelets, tant les del nivell bàsic com mig, estan buits. És necessari que copieu els codis que ja heu realitzat de les funcions del nivell bàsic dintre de l'esquelet del nivell mig. Això és així perquè l'esquelet del nivell mig conté declaracions i lincatges de funcions que necessitareu en aquest nivell. Si no ho feu així podeu tenir problemes amb les noves declaracions de variables i funcions.

Aquesta nota és igualment vàlida quan realitzeu el nivell avançat.