PROJECT REPORT ON MIPS
COAL LAB - 3B1 2020

Instructor's Name: Sir Shiraz Malkani

Name: Muhammad Danish
Roll No: CS182019
Section: 3B1

Project Name: Connect-4 Game

Introduction:

The project include a 6 by 7 grid of connect-4 game. The game runs with an array of 42 bytes. There are two players player1 and player2 both of the player have their own tokens. When a player chooses a column number for token entry than there are different cases are checked for a player when. But before that every entry is check on two cases:

If the index where the token is entered, available?
And if the entered column number is valid or not that is ranging between 0-6?
After this the requested token is processed an entered into the array.
Than the combination of 4's are check each try in diagonal, vertical and horizontal. The player which makes the 4's combination first will win.

Array is first filled with whitespaces than it every entry is checked with a whitespace that's how it shows if Index is empty for entry or not.

Description:

The functions that run this program are described below.

1) Function createBoard:

First of all I took the initial address of the array. I have used a character array that's why the size of the array is 42 bytes because every character in MIPS requires 1 byte of storage. And there are 42 entries (6*7 = 42).
Then I loaded a whitespace (" ") character into a register and then this whitespace character is filled in every index of array. The exit condition for this loop is that, when index is greater than 42.

Filling every index with whitespace is helping in further checking for the valid token entries and available index space.

2) Function displayBoard:

Here is using again a loop which prints the board now here I have hardcoded a top row and then with a loop the columns and the middle rows are printed. And then at last the last row which is hardcoded is printed.

The way it works is that, first I loaded the initial address of array (filled with whitespaces) in a register.

And then a bar ("|") is loaded. First the bar is printed and after that a whitespace (" ") from the first array index is loaded and printed. And then again this repeats 6 times. That's how it shows columns to the user. And last a whole printed grid is in front of the user. After that the middle row is printed and again the cycle repeats for the column printing.

The column printing is not hardcoded expect the bar ("|"). Whitespaces are coming from the array.

3) Function requestPlayerMove and processPlayerMove:

And now when the moves are requested for player1 and player2.
When the column number is entered than the entry is checked for:
Valid column entry, available space, player win and player game tie.

when the player move is processed the above condition are checked and valid column entry and available space is mandatory to be true otherwise the program tell the player for error by printing a string of "invalid column number" and "no space available column is full"

The column number is entered and then the value is checked and store in the valid index. After that the valid index if filled and taken.

4) Function getArrayIndex:

This functions return the array index of the current array. And it is used for checking available space and checking for wins for player.
The way is works is that, the current column and current row index are entered as arguments. Then these values are used to fetch the data and column and row number from the array. These values are returned and are used in processing.

5) Function spaceAvailable:

This functions returns a flag for available space, the way it works is that it checks the whitespace in array index. If it's available than in means that token can be entered there.
Otherwise if a value is present it return a -1 flag and shows that space is not available.

5) Function checkWin:

Here the win for each player is checked either he made a combination of 4's (with his valid token) diagonally, vertically or horizontally.

The diagonal win is checked both side left and right. Vertical win is checked form down to up. And horizontal win is checked form left to right.
And after that game tie is checked.

The way every check work is like that I loaded the current array index. And then the value is checked with the adjacent value if the value is same for any player token than a counter is incremented. If the counter is greater than or equals to 4. Than check is exited and win is awarded to the player. And who won the match we know is by the player value. Player value have 1 as a value and player-2 has a 2 value.

If the entry is entered into the array and processed and checked and none of that satisfy the match.
Than tie game is checked is works as follow, every index (42) of the array is checked if it's they are without the whitespaces than it means it is filled and none of the winning condition is satisfied. Hence it's a ties without any combination of 4's.

If a player satisfies a win then a string is printed for the player and program is exited. And if none is satisfied than the player are switched by decrementing and incrementing for player 2 and player 1 respectively and then jumping to their playerRequest functions and processing.

Every time the player entered the column number an entry of -1 is also checked, as it works as program exit condition and it is checked in the validIndex function everytime. If the entered column value is -1 than the program is directly exited with a message of the player that quit the game

X ----------X

Name: Muhammad Danish

Roll No: CS182019

Section: 3B1