



UFFS – Universidade Federal da Fronteira Sul
Curso de Ciencia da Computação
Organização de Computadores

Instruções:

Trabalho individual ou em dupla
Cronograma com as datas limite:

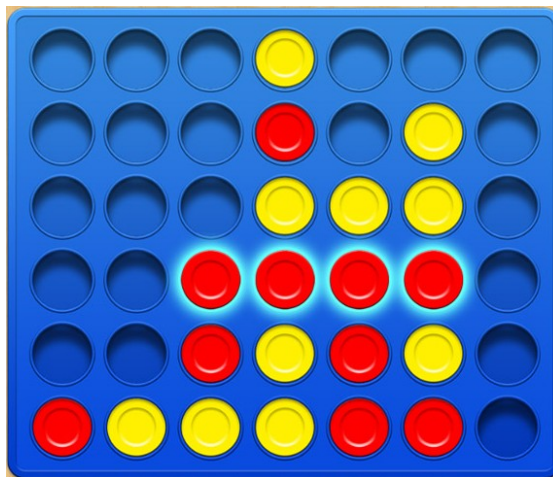
	Em dia	Atrasada
Entrega intermediária:	16/09	20/09
Entrega final:	30/09	02/10
Apresentação:	Cronograma até 02/04	

O que entregar:

Um arquivo com extensão .s ou .asm
O arquivo deve conter obrigatoriamente um cabeçalho com a matrícula e o nome dos integrantes do trabalho

Descrição:

Faça um programa utilizando o conjunto de instruções RV32I que implementa a gerencia do jogo "4 em linha".



O programa deve ser implementado utilizando o conjunto de instruções do processador RISC-V e deve ser executado no simulador RARS.

O programa principal deve apresentar um menu para o usuário com as seguintes opções:

- 1) Configuração
- 2) Jogar
- 3) Sair

A opção 'Configuração' deve apresentar um sub-menu onde o usuário pode configurar as seguintes funcionalidades:

- 1) Quantidade de jogadores: 1 ou 2
- 2) Tamanho da tabuleiro: 7x6 ou 9x6
- 3) Modo de dificuldade: ao menos 2 (fácil: aleatório; médio: tenta impedir vitória)
- 4) Zerar contadores
- 5) mostra as configurações atuais e o valor dos contadores;

A opção 'Jogar' deve permitir o(s) usuário(s) realizar as suas jogadas, isto é, selecionar a coluna que deseja colocar uma 'ficha' e também atualizar a situação do tabuleiro. Ao iniciar a opção 'Jogar' o controle deve levar em consideração as configurações atuais definidas para a quantidade de jogadores e tamanho do tabuleiro.

A cada jogada o controle do jogo deve: (a) verificar se algum dos jogadores conseguiu colocar '4 em linha'; (b) em caso de vitória, informar qual jogador foi vencedor; (c) em caso de vitória, incrementar o contador com a quantidade de vitórias do jogador

```
J1 venceu: 4
J2 venceu: 1
```

```
0 1 2 3 4 5 6
- - - - - - -
- - - - - - -
- - - - - - -
- - - - - - -
- - & - - - -
- # & # # - -
```

No caso da configuração com 2 jogadores o controle do jogo deve solicitar a jogada a cada um jogadores de forma alternada. Caso o jogador informe uma coluna inexistente o controle deve evitar e pedir para o jogador realizar sua jogada novamente.

No caso da configuração com apenas 1 jogador, o controle do jogo deve realizar a jogada de um dos jogadores (perguntar no início quem começa, jogador ou controlador). Na configuração de modo de dificuldade fácil a escolha da coluna por parte do controlador deve ser aleatória. Na configuração de modo de dificuldade média o controlador deve prioritariamente tentar evitar que o jogador humano ganhe; para isso deve verificar se há alguma sequencia de 3 em linha e, se possível, tentar bloquear a posição de vitória.

As informações de configuração e do tabuleiro devem ser armazenadas em memória. O "tabuleiro" também deve estar armazenado em memória, em um rótulo chamado `matriz_jogo`.

A implementação deve tornar o código mais modular possível, implementando funções para as diferentes funcionalidades, passando e retornando os valores necessários como parâmetro das funções.

Funções obrigatórias:

```
void inicia_tabuleiro(int *matriz_jogo, int colunas);
```

Parâmetros recebidos:

a0: a posição de memória do rótulo `matriz_jogo`;

a1: quantidade de colunas presentes no tabuleiro;

Retorno da função:

A função não precisa retornar nada;

Funcionalidade: a função deve limpar o tabuleiro;;

```
void imprime_tabuleiro(int *matriz_jogo, int colunas);
```

Parâmetros recebidos:

a0: a posição de memória do rótulo `matriz_jogo`;

a1: quantidade de colunas presentes no tabuleiro;

Retorno da função:

A função não precisa retornar nada;

Funcionalidade: a função deve imprimir o estado do tabuleiro a cada jogada;

```
int verifica_vencedor(int *matriz_jogo, int colunas, int jogador);
```

Parâmetros recebidos:

a0: a posição de memória do rótulo `matriz_jogo`;

a1: quantidade de colunas presentes no tabuleiro;

a2: o jogador que realizou a ultima jogada

Retorno da função:

caso encontre vencedor: retorna 1 em a0 e o jogador vencedor em a1

caso não encontre vencedor: retorna -1 em a0

Funcionalidade: a função deve verificar no tabuleiro se o jogador informado possui uma sequencia de '4 em linha';

Detalhamento do Cronograma:

Entrega intermediária (16/09) - lista de funcionalidades que devem estar prontas:

- (a) a função *main* com o menu principal e os sub-menus, lendo as opções do usuário e gravando as configurações na memória;
- (b) a função *inicializa_tabuleiro*;
- (c) a função *imprime_tabuleiro*;

Entrega final (30/09) – Todas as funcionalidades solicitadas.

Nota do trabalho:

A nota será composta da seguinte forma:

- 50% implementação das funcionalidades e modularidade do código
- 50% apresentação e explicações sobre a implementação

Penalidades das entregas inicial e intermediária do cronograma em relação a nota final do trabalho:

- não fez uma entrega: 10% por entrega não realizada;
- entrega atrasada: 5% por entrega atrasada;

Em caso de plágio a nota será zero.