

Disciplina: INF01108 - Arquitetura e Organização de Computadores I
Professores: Marcello Macarthy, Mateus Grellert, Sergio Cechin

Trabalho II - Projeto de Programação para o processador CESAR

1. Descrição Geral

Neste trabalho, você irá desenvolver um jogo da FORCA em linguagem assembly utilizando a arquitetura do processador CESAR.

O jogo da FORCA consiste em adivinhar uma palavra informando uma possível letra a cada rodada. No início do jogo, a palavra está escondida por caracteres "_" (caractere 0x5f). Se a palavra contiver um hífen, este caractere deve também ser apresentado no visor na sua posição. Quando a letra informada consta na palavra, as posições são reveladas. Caso contrário, uma vida é retirada.

O jogo é concluído quando as vidas do jogador acabarem ou quando todos os caracteres da palavra forem informados.

2. Especificação da Implementação

O programa para o jogo da FORCA deverá conter as seguintes etapas (especificações de como isso deve ser apresentado no visor serão descritas a seguir):

1. Antes de o programa iniciar seu laço de jogo, o programa deve verificar as entradas informadas pelo usuário, verificando a presença de caracteres inválidos na palavra (somente letras e hífen podem ser utilizados).
2. Durante o jogo:
 - a. Apresentar a palavra na parte esquerda do visor e a quantidade de vidas restantes na parte direita.
 - b. A cada nova letra informada (utilizando o teclado), o programa atualiza a tela preenchendo a letra no visor (caso o jogador acerte) ou atualizando a quantidade de vidas (caso o jogador erre).
 - c. Ao final do jogo, informar se o jogador ganhou ou perdeu, juntamente com o tempo transcorrido.
 - d. Voltar à etapa (1) após o usuário digitar qualquer tecla.

2.1 Posição dos Dados na Memória

A seguir são apresentadas as posições dos dados na memória para este projeto:

Antes de iniciar o programa:

- **Posição 0x8000:** palavra (word) contendo a quantidade de vidas. Precisa ser um valor entre 0 e 99.
- **Posições 0x8002 a 0x801C:** string (array de bytes) contendo a palavra a ser adivinhada, com no máximo 25 caracteres válidos e terminada com o caractere de terminação '\0'.

Durante o jogo:

- **Posições 0xFFDC a 0xFFFF6:** preencher com os caracteres da palavra em caixa alta ou com "_" para caracteres ocultos. Posições não preenchidas devem ser completadas com espaço.
- **Posições 0xFFFF7 a 0xFFFFF:** preencher com a informação de vidas do jogador. A mensagem segue o padrão "Vidas: [DD]", onde [DD] representa a quantidade de vidas em dois caracteres, com preenchimento '0' no lado esquerdo (equivalente ao padrão de formatação "%02d" da função printf do C).

Após o jogo:

- **Posições 0xFFDC a 0xFFFFF:** preencher com a mensagem de despedida (vitória ou derrota e tempo).

2.2 Estados do Visor

As figuras a seguir apresentam alguns exemplos de estado do visor, considerando que o número de vidas foi configurado para 5 e que a palavra configurada é "paralelepipedo" (sem acento):

- Tela de boas-vindas:



- Jogador informou a letra "a":



- Jogador informou a letra "u":



- Jogo concluído com vitória do jogador:



3. Desafio

O desafio adicional proposto para este projeto envolve implementar um modo para 2 jogadores com repetição. Neste modo, o primeiro jogador deve digitar a palavra a ser adivinhada. Após o registro dessa palavra, o segundo jogador irá tentar adivinhá-la. Ao concluir o jogo, o programa exibe a opção de jogar novamente ou concluir a execução.

Os(as) alunos(as) que optarem por implementar este desafio terão 1 ponto adicional em qualquer um dos 3 trabalhos (RAMSES, CESAR ou INTEL).

4. Avaliação

As seguintes condições devem ser respeitadas para obter nota integral no projeto:

1. O programa deve funcionar sem bugs e **checar situações de entradas inválidas. Exemplos:**
 - a. número de vidas diferente do limite de 1 a 99.
 - b. Caracteres inválidos na palavra.
2. A checagem de caracteres deve ser insensível ao caso, isto é, "A" e "a" são considerados iguais. **O visor irá sempre imprimir a versão em caixa alta.**
3. A entrada de dados pelo teclado deve ser feita utilizando o mecanismo de interrupções.
4. O tempo deve ser contabilizado utilizando interrupções de timer.
5. Uso de subrotinas ao menos para os procedimentos de checagem de dados e de busca do caractere na palavra.
6. O trabalho deve ser realizado de forma individual. Cópias e plágio terão nota final igual a zero.

5. Instruções para Submissão

Os alunos devem submeter no Moodle **o arquivo .CED** do programa dentro do prazo estipulado.

~~**Prazo de submissão: 23/12/2023**~~

6. Observações

Recomenda-se a troca de ideias entre os alunos. Entretanto, a identificação de cópias de trabalhos acarretará na aplicação do Código Disciplinar Discente e a tomada das medidas cabíveis para essa situação. Inicialmente, nesses casos, ambos os trabalhos: original e cópias, receberão nota zero. O professor da disciplina reserva-se o direito, caso necessário, de solicitar uma demonstração do programa, onde o aluno será arguido sobre o trabalho como um todo. Nesse caso, a nota final do trabalho levará em consideração o resultado da demonstração.