

“

ARQUITETURA DE COMPUTADORES | 21010 | 2024/2025

Período de Realização

Decorre de 19 de dezembro de 2024 a 6 de janeiro de 2025

Data de Limite de Entrega

6 de janeiro de 2025, até às 23:59 de Portugal Continental

Temática

Organização Básica do Computador

Objetivos

- Arquitetura de um Computador
- Conjuntos de Instruções
- Programação em Linguagem Assembly

Trabalho a desenvolver

Implementação de um Programa de Reconhecimento Facial no Simulador P3

Pretende-se o desenvolvimento de um programa de reconhecimento facial simplificado, com base em padrões binários de características faciais, dividido em 4 alíneas, cada uma valendo 1 valor, semelhante ao sistema digital apresentado no EFólioA.

O programa de reconhecimento facial trabalha sobre imagens e devolve como saídas dois códigos de dois *bit* cada.

O código binário a 2 *bits* correspondente ao **traço facial** (TF) encontrado, conforme Tabela 1:

Bit de maior peso TF1	Bit de menor peso TF0	Código do traço facial
0	0	Boca
0	1	Nariz
1	0	Olhos
1	1	Invalido

Tabela 1 - Definição dos códigos binários de 2 bits para traços faciais específicos

O código binário a 2 *bits* do **tipo de traço facial** (TTF) é apresentado na Tabela 2:

Bit de maior peso TTF1	Bit de menor peso TTF0	Tipo de traço Facial	Traço facial
0	0	grande	Boca - 00
0	1	pequena	Boca - 00
1	0	aberta	Boca - 00
1	1	fechada	Boca - 00
0	0	curto	Nariz - 01

0	1	comprido	Nariz - 01
1	0	estreito	Nariz - 01
1	1	largo	Nariz - 01
0	0	castanhos	Olhos - 10
0	1	verdes	Olhos - 10
1	0	azuis	Olhos - 10
1	1	cinzentos	Olhos - 10

Tabela 2 – Detalhe dos códigos binários para diferentes tipos de traços faciais

Alínea A (1 valor)

Escreva uma rotina em Assembly P3 que receba em R1 e R2 (nos 4 bits de menor peso) dois códigos completos de traços faciais (TF e TTF) e devolva em R3:

- 0: se os códigos forem diferentes e o código em R1 é inválido;
- 1: se os códigos forem diferentes e o código em R2 é inválido;
- 2: se os códigos forem diferentes e os códigos em R1 e R2 são ambos inválidos;
- 3: se os códigos forem diferentes e os códigos em R1 e R2 são ambos válidos;
- 4: se os códigos forem iguais e os códigos em R1 e R2 são ambos inválidos;
- 5: se os códigos forem iguais e os códigos em R1 e R2 são ambos válidos.

Apresente, no relatório, um caso de teste para cada resultado de R3, da lista anterior.

Exemplos:

- Entrada:
 - R1 = 0x07 (Nariz largo, código válido)
 - R2 = 0xFF (Código inválido)
- Saída:
 - R3 = **1** (Códigos diferentes, R2 inválido)
- Entrada:
 - R1 = 0x03 (Boca fechada, código válido)
 - R2 = 0x03 (Boca fechada, código válido)
- Saída:
 - R3 = **5** (Códigos iguais e válidos)

Alínea B (1 valor)

Escreva uma rotina em Assembly P3 que recebe, em R1, o código de um traço facial. Recebe em R2, o endereço de uma *string* que deverá ser terminada por 0. Escreva nessa *string* a descrição em ASCII do traço facial recebido:

- R1=0, a *string* deve ficar com "Boca",0
- R1=1, a *string* deve ficar com "Nariz",0
- R1=2, a *string* deve ficar com "Olhos",0
- R1=3, a *string* deve ficar com "Invalido",0

Apresente, no relatório, um caso de teste para cada descrição indicada na lista anterior.

Exemplos:

- Entrada:
 - R1 = 0x00 (Boca)
 - R2 = 0x9000 (Endereço de memória para armazenar a *string*)
- Saída na memória a partir de R2:

- 0x9000: 'B'
 - 0x9001: 'o'
 - 0x9002: 'c'
 - 0x9003: 'a'
 - 0x9004: 0x00 (terminador)
- Entrada:
 - R1 = 0x03 (Invalido)
 - R2 = 0x9100
 - Saída na memória a partir de R2:
 - 0x9100: 'l'
 - 0x9101: 'n'
 - 0x9102: 'v'
 - 0x9103: 'a'
 - 0x9104: 'l'
 - 0x9105: 'i'
 - 0x9106: 'd'
 - 0x9107: 'o'
 - 0x9108: 0x00 (terminador)

Alínea C (1 valor)

Escreva uma rotina em Assembly P3 que recebe, em R1, um código de traço facial completo (TF e TTF) e, em R2, o endereço de uma *string* que será terminada por 0. O conteúdo da *string* deve estar de acordo com a tabela seguinte:

R1	String
0	"Boca grande",0
1	"Boca pequena",0
2	"Boca aberta",0
3	"Boca fechada",0
4	"Nariz curto",0
5	"Nariz comprido",0
6	"Nariz estreito",0
7	"Nariz largo",0
8	"Olhos castanhos",0
9	"Olhos verdes",0
10	"Olhos azuis",0
11	"Olhos cízentos",0
12	"Invalido",0
13	"Invalido",0
14	"Invalido",0
15	"Invalido",0

Apresente, no relatório, um caso de teste para cada descrição indicada na tabela anterior.

Exemplos:

- Entrada:
 - R1 = 0x06 (Nariz estreito)
 - R2 = 0x9200 (Endereço de memória para armazenar a *string*)

- Saída na memória a partir de R2:
 - 0x9200: 'N'
 - 0x9201: 'a'
 - 0x9202: 'r'
 - 0x9203: 'i'
 - 0x9204: 'z'
 - 0x9205: ''
 - 0x9206: 'e'
 - 0x9207: 's'
 - 0x9208: 't'
 - 0x9209: 'r'
 - 0x920A: 'e'
 - 0x920B: 'i'
 - 0x920C: 't'
 - 0x920D: 'o'
 - 0x920E: 0x00
- Entrada:
 - R1 = 0x0F (Inválido)
 - R2 = 0x9300
- Saída na memória a partir de R2:
 - 0x9300: 'l'
 - 0x9301: 'n'
 - 0x9302: 'v'
 - 0x9303: 'a'
 - 0x9304: 'l'
 - 0x9305: 'i'
 - 0x9306: 'd'
 - 0x9307: 'o'
 - 0x9308: 0x00

Alínea D (1 valor)

- a) Usando as rotinas da alínea A, escreva um programa em Assembly P3 para ser implementado a partir do endereço 0000h de memória que tenha como base os vetores [8000h..80FFh] e [8100h..81FFh] contendo códigos completos de traços faciais (TF e TTF) e coloque no vetor [8200h..82FFh] os valores resultados das comparações entre códigos.

Exemplo:

- Vetor de entrada [8000h..80FFh]:
 - 0x8000: **0x03**
 - 0x8001: **0x06**
 - 0x8002: **0xFF**
- Vetor de entrada [8100h..81FFh]:
 - 0x8100: **0x03**
 - 0x8101: **0xFF**
 - 0x8102: **0x03**
- Vetor de saída [8200h..82FFh] após execução:
 - 0x8200: **5** ; Códigos iguais e válidos
 - 0x8201: **1** ; Códigos diferentes, R2 inválido
 - 0x8202: **0** ; Códigos diferentes, R1 inválido

Apresente, no relatório, um caso de teste para cada valor de R3 indicado na lista da alínea A.

b) Usando a rotina da alínea C, faça um programa em Assembly P3 que tenha como base o vetor [8000h..800Fh] e coloque, a partir do endereço 8010h, as *strings* correspondentes aos códigos faciais completos encontrados no vetor. As *strings* devem ficar contíguas com o valor 0 para as separar. Apresente, no relatório, um caso de teste para cada valor de traço facial.

Exemplo:

- Vetor de entrada [8000h..800Fh]:
 - 0x8000: 0x03
 - 0x8001: 0x06
 - 0x8002: 0x0F
- Área de saída [8010h..80FFh] após execução:
 - 0x8010: 'B'
 - 0x8011: 'o'
 - 0x8012: 'c'
 - 0x8013: 'a'
 - 0x8014: ''
 - 0x8015: 'f'
 - 0x8016: 'e'
 - 0x8017: 'c'
 - 0x8018: 'h'
 - 0x8019: 'a'
 - 0x801A: 'd'
 - 0x801B: 'a'
 - 0x801C: 0x00
 - 0x801D: 'N'
 - 0x801E: 'a'
 - 0x801F: 'r'
 - 0x8020: 'i'
 - 0x8021: 'z'
 - 0x8022: ''
 - 0x8023: 'e'
 - 0x8024: 's'
 - 0x8025: 't'
 - 0x8026: 'r'
 - 0x8027: 'e'
 - 0x8028: 'i'
 - 0x8029: 't'
 - 0x802A: 'o'
 - 0x802B: 0x00
 - 0x802C: 'I'
 - 0x802D: 'n'
 - 0x802E: 'v'
 - 0x802F: 'a'
 - 0x8030: 'l'
 - 0x8031: 'i'
 - 0x8032: 'd'
 - 0x8033: 'o'
 - 0x8034: 0x00

Bom Trabalho!

Regras a observar no e-fólio B

Relatório: O relatório deve apresentar todas as decisões relevantes e todas as opções tomadas na construção dos programas.

Forma de entrega: Um único ficheiro com o nome correspondente ao número de aluno (ex: 999999EfolioB.pdf). O ficheiro deverá conter:

- Relatório em formato pdf ou Word até 5 páginas A4 com todas as decisões relevantes e todas as opções tomadas na construção dos programas.
- Em anexo ao relatório (dentro do ficheiro do relatório) deve colocar todo o código e resultados obtidos. O anexo não conta para o limite de páginas definido.

O código deve estar num formato que permita a seleção de modo a ser copiado e colado diretamente para o simulador do P3.

Não são aceites entregas fora da plataforma Moodle.

Avaliação

Cotação: um valor por alínea.

Critérios de Correção:

- Funcionalidade solicitada implementada e testada no relatório: 50%
 - Alínea A (15%)
 - Alínea B (15%)
 - Alínea C (10%)
 - Alínea D (10%)
- Simplicidade e modularidade da solução (utilização correta de funções e código simples): 10%
- Eficiência (contabilizados o número de instruções e ciclos de relógio, relativamente a um valor de referência): 10%
- Apresentação do código (indentação e comentários): 10%
- Relatório: 20%
 - Legibilidade
 - Justificação dos testes e resultados e das opções relevantes

Penalizações:

- Trabalhos não conformes com as regras de entrega do E-fólio B: até 10%
- Código entregue, mas que não permite a seleção de modo a ser copiado e colado diretamente para o simulador do P3: 50%
- Deve ser usado o P3JS Assembler and Simulator (<https://p3js.goncalomb.com/>) descrito na AF8: Simulador P3. Trabalhos entregues que não utilizaram o Simulador P3 e não “correm”: 100%
- Código sem comentários, ou apenas com comentários a refletir o significado da instrução (exemplo MOV R1,R2;mover o conteúdo de R2 para R1) : até 50%
- Deteção de fraude (total ou parcial): 100%
- Trabalhos entregues após a data limite: 10% (o recurso de entrega estará aberto até às 09:00 manhã do dia seguinte à data/hora estabelecida no PUC).