

Trabalho Final

Microprocessadores

Turno P1

12/12/2024

Alunos:

Sidimohamed Brahim Nº 64532

Guilherme Silva Nº 64937

Tiago Leal Nº 66181

Índice

1. Introdução	3
2. Descrição dos requisitos	4
3. Apresentação da arquitectura implementada	5
4. Detalhe dos módulos do Sistema	9
5. Testes e Resultados	11
6. Conclusões	12
7. Referências bibliográficas	13

. INTRODUÇÃO

Este trabalho foi realizado no âmbito da unidade curricular de Microprocessadores, onde o objetivo consiste no desenvolvimento do software de uma estação de voto.

O sistema é dividido em duas vertentes principais: Front-office e Back-office.

A parte de Front-office é responsável pela recolha de votos, onde se procede à identificação do votante através do respetivo número de eleitor, regista-se o voto do eleitor e garante também que cada eleitor tem direito a apenas um voto.

Por sua vez, o Back-office é mais voltado para a parte estatística e análise de resultados. Para obtermos acesso a este modo será necessário introduzir a password correta, para assim desbloquear um novo menu que dá acesso, à lista de eleitores bem como ao gráfico de resultados.

Este relatório encontra-se dividido em 7 capítulos:

- 1) Introdução, apresentação geral do nosso relatório.
- 2) Descrição dos requisitos, explicação da estrutura do nosso trabalho para responder aos objetivos pedidos no enunciado.
- 3) Apresentação da arquitetura implementada, vista geral dos
- 4) Detalhes dos módulos do sistema
- 5) Testes e resultados
- 6) Conclusões obtidas
- 7) Referências bibliográficas

2. DESCRIÇÃO DOS REQUISITOS

Neste segundo capítulo falamos dos objetivos pedidos para este trabalho. Aqui iremos descrever mais especificamente cada funcionalidade do programa.

O programa inicia-se com a apresentação do menu principal, onde nos deparamos com quatro opções disponíveis: votar, gerir, créditos e sair.

Ao seleccionar a opção “Votar”, o programa irá pedir ao utilizador que coloque o número de eleitor. De seguida o programa valida o número introduzido com a lista de eleitores e verifica se o respetivo eleitor ainda não votou, tendo em conta que cada eleitor apenas pode votar uma vez. Só após estas validações, é que o programa nos vai apresentar o boletim de voto com todos os candidatos e um respetivo quadrado à frente de cada nome para o registo do voto. Finda uma votação e o programa retorna ao menu principal.

Ao seleccionar a opção “Gerir”, é pedido que o utilizador coloque a palavra chave de administração, guardada como constante no programa. No caso de erro, o programa volta a pedir a palavra chave, e para que não fiquemos presos neste menu, a tecla ESC voltará para o menu principal. No caso de estar correta é apresentado um segundo menu com três opções:

- **Carregar eleitores**– apresenta a lista de eleitores, apenas disponível se o programa arrancou sem dados, ou seja, sem ficheiro.bin existente.
- **Gráfico de Resultados** – apresenta o número de votos obtidos até ao momento, e ilustra com gráficos de barras.
- **Sair** – o programa retorna ao menu principal.

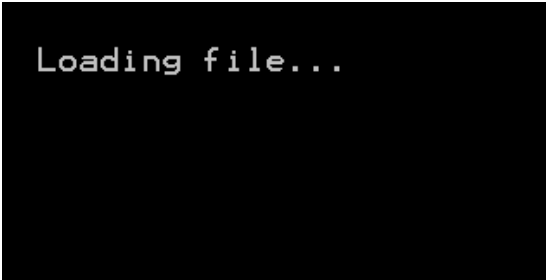
Selecionando a opção “Créditos”, serão apresentados os nomes e respetivos números dos alunos que participaram na realização do trabalho.

Clicando na opção “Sair”, saímos do programa, guardando a lista de eleitores, inclusive se votaram ou não, para o ficheiro de binário de dados.

3. APRESENTAÇÃO DA ARQUITECTURA IMPLEMENTADA

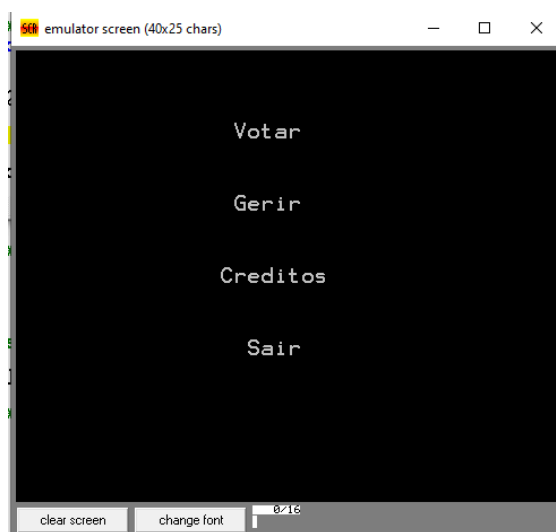
Neste capítulo apresentamos o funcionamento do programa a partir do menu principal, para demonstrar a nossa implementação que respondesse aos objetivos do enunciado.

Quando corremos o programa, é verificada a existência do ficheiro binário “dados.bin”, e prossegue para o menu principal, com a estrutura preenchida no caso do ficheiro existir.

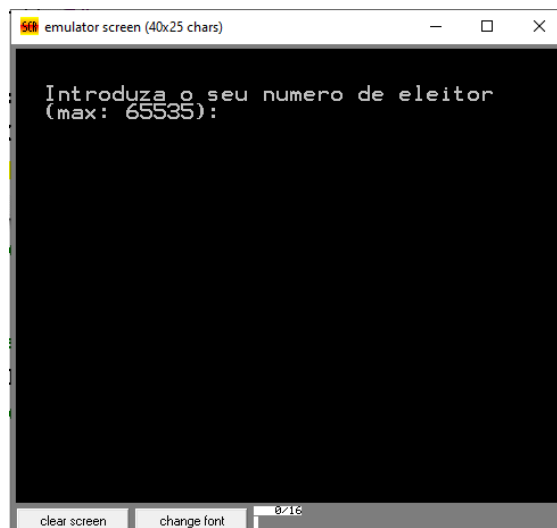


```
Loading file...
```

Vemos então as quatro opções disponíveis, bastando clicar em cada uma para seleccionar a opção pretendida pelo utilizador.



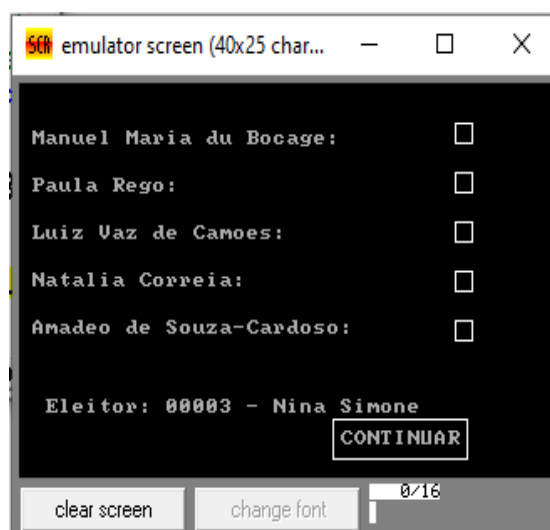
Ao seleccionar a opção “Votar” deparamo-nos com o seguinte:



O utilizador tem de introduzir de 1 até 12 dígitos (limite apenas simbólico, pode ser alterado), se o número final for maior que 65530, um erro será assinalado. Pode também clicar no ESC para voltar para o ecrã inicial.

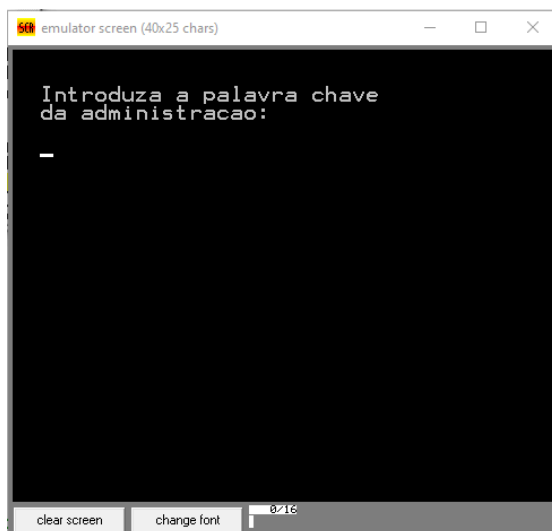
Aqui o voto será comparado com os eleitores existentes na estrutura, e seguirá caso o número seja encontrado.

Após indicar-mos o número de eleitor, é apresentado o boletim de voto:

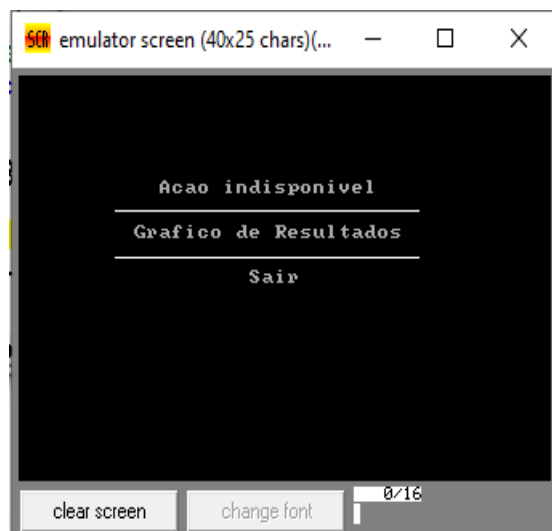


De seguida basta clicar num dos quadrados, o voto é guardado no ficheiro de logs, e é verificado o número de votos para saber se será nulo ou válido ou branco.

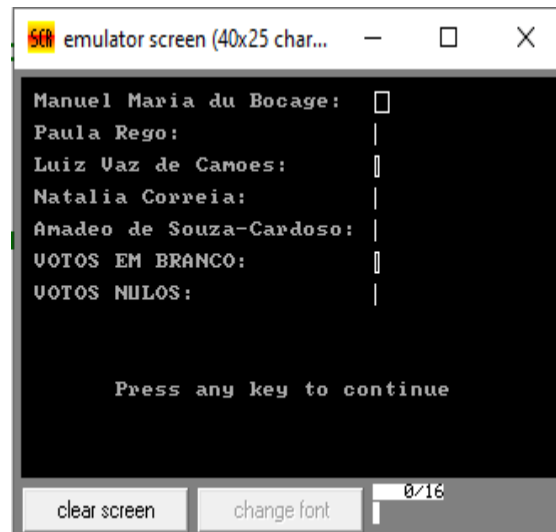
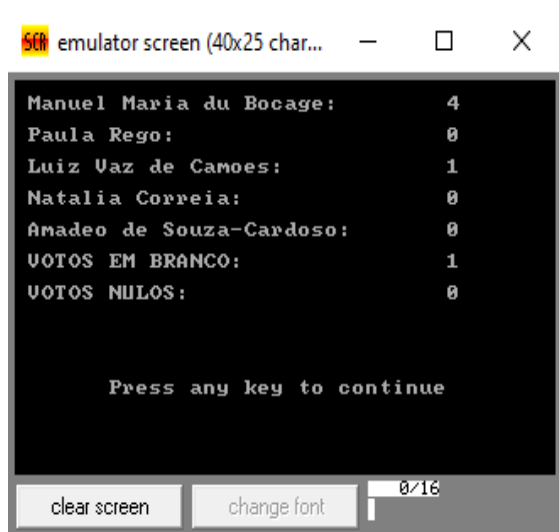
Se no menu principal, o utilizador optar por escolher a opção “Gerir”, terá de introduzir a password correta:



De seguida, aparece um segundo menu com as opções de carregar eleitores (caso não tenham sido carregados no arranque do programa) e gráfico de resultados:



Quando seleccionamos o gráfico de resultados, podemos ver quantos votos cada candidato recebeu, e um gráfico de barras:



Ao sair, o programa salvaguarda a estrutura dos eleitores, atualizando os dados com os votos de cada eleitor (se votaram ou não). A estrutura por sua vez é definida deste modo:

lista eleitores db (02 + 30 + 01) * num eleitores dup(00)

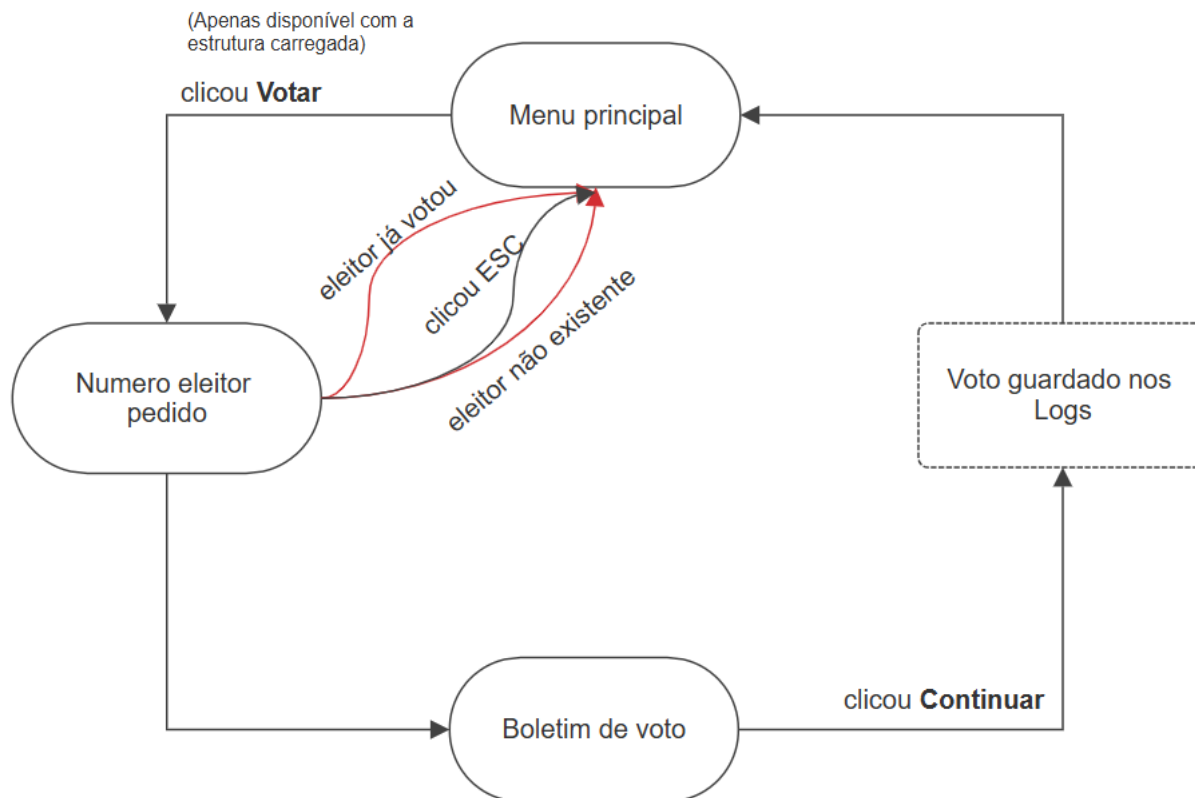
Saving...

Assim, guardamos o mínimo espaço possível por cada eleitor, 2 bytes para o seu número, 30 para o nome e um último booleano para saber se votou ou não. Apontamos também que ao escrever para o ficheiro binário, os bytes a nulo que não são ocupados pelo nome (ex: “Desiderio Murcho” ocupará 16 bytes, 14 deles continuaram nulos) NÃO serão copiados, pois apenas ia encher o ficheiro com dados não essenciais.

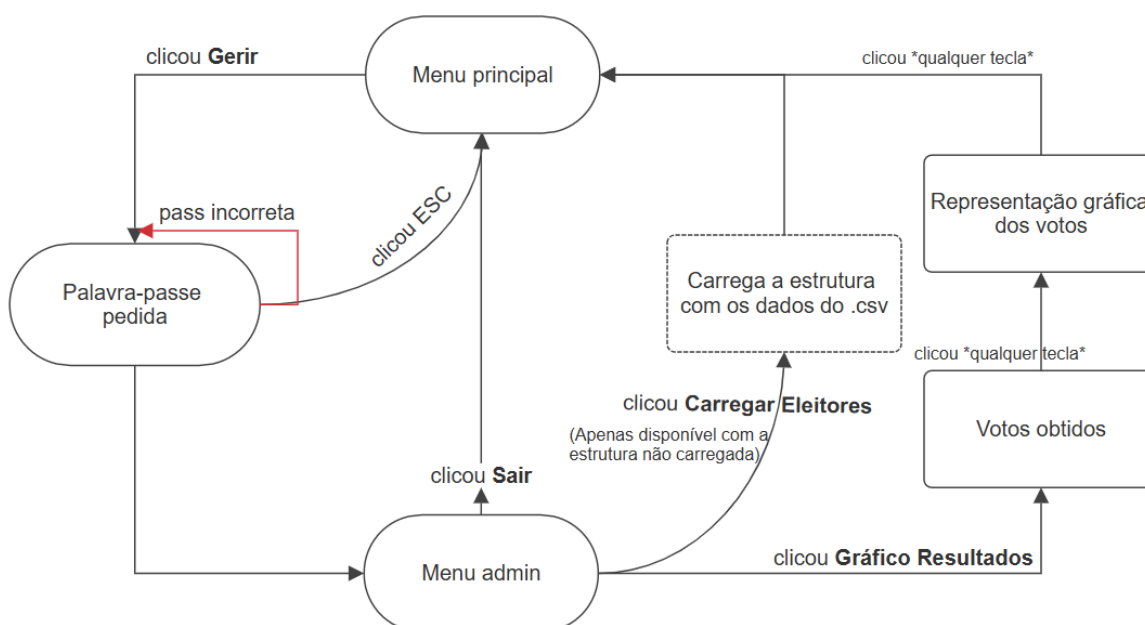
4. DETALHE DOS MÓDULOS DO SISTEMA

Neste capítulo fazemos a apresentação dos algoritmos recorrendo a fluxogramas.

Votar:



Gerir:



No que toca à opção “Créditos”, apenas são exibidos os nomes e respetivos números dos alunos que realizaram o trabalho e quando o utilizador pressiona uma tecla, é reencaminhado para o menu principal.

```
credits:
    call    setTextMode

    mov     Dx, 070Bh
    call    setCursorPosition
    lea     Dx, authors
    call    printStr$

    call    keyToContinue
    jmp     main_loop
```

```
Tiago leal 66181

Sidi brahim 63452

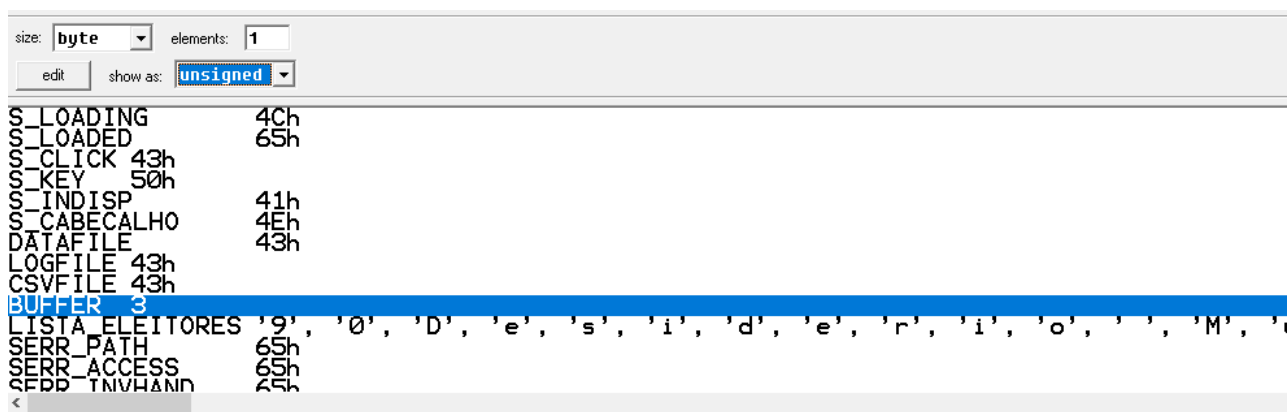
Guilherme Silva 64937

Press any key to continue_
```

5. TESTES E RESULTADOS

O programa foi testado usando as ferramentas dadas pelo emu8086, destas, falamos então das mais utilizadas, para a conclusão deste trabalho.

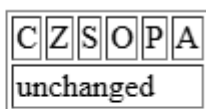
A janela que nos permite de ver os valores que cada byte de cada variável tem, possivelmente interpretando a informação como signed, unsigned, ascii, etc... , foi essencial para o debug da estrutura de dados dos eleitores, desta forma conseguimos ter a certeza do que estava a ser guardado.



Halt the System.

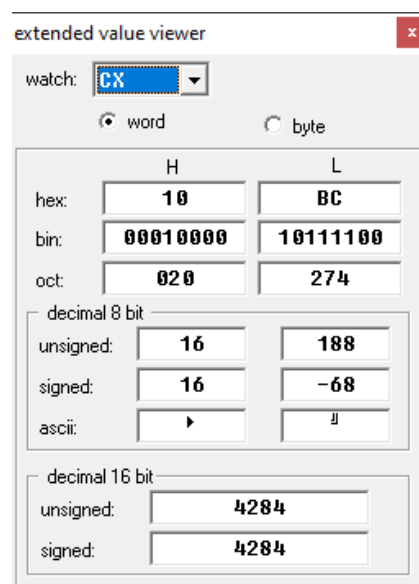
Example:

```
MOV AX, 5
HLT
```

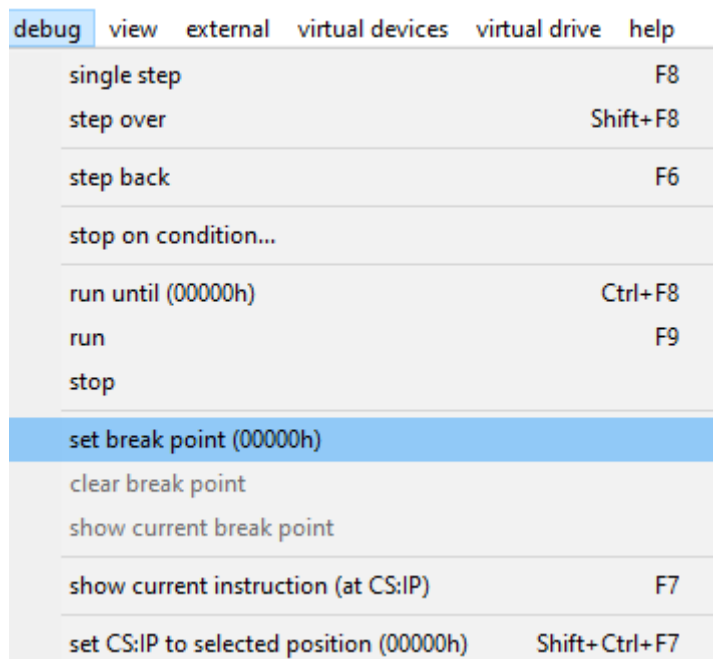


Juntamente, a instrução HLT foi indispensável. Serve de breaking point para pararmos o código numa linha exata, para depois podermos ver passo a passo o que o código fez sem precisarmos de confusões a parar a execução a tempo. Infelizmente, não é perfeito pois para a execução por completo e para prosseguirmos com o código é preciso recompilar o programa.

Também temos de realçar a utilidade do visualizador para cada registo, que nos dá todas as interpretações necessárias aos bytes que se encontram lá, realçando a mais importante do valor total dos 16 bits que seria absurdamente fastidioso debugar diretamente pelo código.



Como nota, queremos também explicar que o emulador tem ferramentas melhores que só foram “encontradas” a seguir à entrega. Possui um hotkeys para andar de passo em passo a possibilidade de meter breakpoints, e até regras para que o programa pare até um dos registo chegar a certo valor.



A utilização deste menu, facilitaria consideravelmente o debug do código, e convidamos o leitor a usá-lo se precisar.

6. CONCLUSÕES

Após a realização do trabalho concluímos que o programa cumpre perfeitamente todos os objetivos essenciais requisitados, com um código devidamente estruturado e comentado. Foi tomada em conta a utilização de diversas formas de executar certas funcionalidades como o uso do base pointer, destination/source index, de modo a mostrar versatilidade, e para saber exatamente do que nos serve cada ferramenta disponível ao programar em Assembly. Ainda assim reconhecemos que não está a 100%, e sabemos onde falha e/ou o que está em falta:

Os ficheiros funcionam como pedido, em casos extremos que não iriam ser pedidos nem testados, como erro de criação (no caso de a abertura do ficheiro falhar por permissões) escrita ou leitura (caso o ficheiro seja apagado a meio da leitura), não foi implementado o código para lidar com estas situações, embora estejam devidamente comentados os sítios onde iriam ser precisas as exceções.

O máximo que podemos obter ao pedir o input do número do eleitor é teoricamente 65535, o máximo que podemos guardar com uma word, mas no caso de ser acima de 65530, assumimos como erro. Neste caso apenas nos faltou tempo para completar caso estivesse entre 65530 e 65535, e talvez fosse mais simples usar a flag AF no caso de overflow do registo.

Ao imprimir na tela o número de votos de cada candidato, apenas é feita a impressão de um dígito. Esta falha seria muito facilmente resolvida tendo em conta que já utilizamos um rotina que nos imprime números com 2 dígitos na tela, facilmente adaptável para mais dígitos, e a mudança não foi feita por termos de resolver outros erros de maior importância no tempo que nos restava.

Por último, o caminho para o ficheiro .csv não é pedido como input do utilizador e é apenas guardado como string. Por não nos apercebemos a tempo que isto fazia parte do enunciado, não foi possível completar esta parte, ainda assim formulando hipóteses sobre como resolver, por ex: guardando uma string vazia para ser completada depois, ou talvez usando o INT 21h / AH= 39h - make directory...

Por fim reconhecemos também a importância dada ao comprimento do código, com 2300 linhas não por faltarem implementações mas pela eficiência, as funções foram feitas de modo a serem reutilizadas e utilizadas em qualquer parte do código, embora ainda assim haja muito espaço para melhorias neste aspeto, que só faria sentido se o código tivesse completo a 100%.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- Sousa, 2024 Pedro Sousa e João Pimentão, Slides das aulas teóricas e práticas, Universidade Nova de Lisboa, 2024-2025
- Grysztar, 1999 Tomasz Grysztar, Documentação emu8086, (?)