

Trabalho de Programação 3

Processador Intel (80x86)

1. Descrição Geral

Neste trabalho você deverá desenvolver um programa para o processador Intel 8086 que lê arquivos de texto e exibe o texto lido na tela, um caractere de cada vez.

A novidade é que esta exibição será feita com velocidade variável. A velocidade de exibição é alterada por “tags” embutidas no próprio arquivo. As “tags” serão sequências de 3 caracteres no meio do texto, iniciadas pelo caractere '#' seguido de dois dígitos decimais. Por exemplo: **#50**, **#99**, etc. Os dois dígitos definem quantos “ticks” do relógio do sistema o programa deve esperar antes de exibir o próximo caractere lido do arquivo. As “tags” não devem ser exibidas na tela.

O programa deve ter capacidade de processar vários arquivos e encerrar sua execução somente quando o usuário especificar isto digitando um caractere **ENTER** em vez do nome do arquivo a ser lido ou quando o usuário interromper a exibição de um arquivo usando a tecla **ESC** (caractere “escape”).

Notas:

1. A qualquer momento, o usuário poderá interromper a exibição de um arquivo pressionando qualquer tecla. Dependendo da tecla pressionada, o programa deverá tomar a ação correspondente, conforme especificado na etapa 8, a seguir.
2. Podem existir duas “tags” consecutivas. Por exemplo: ...**#36#87**...

2. Especificação do Programa

O programa deverá realizar as seguintes tarefas

1. Ao ser chamado, o programa deve “limpar” a tela (usando um atributo diferente do padrão usado pelo DOS), exibir a identificação do aluno (**nome e número do cartão**) e posicionar o cursor no início da próxima linha da tela.
2. Pedir o **nome do arquivo de entrada**, mantendo o cursor na mesma linha da mensagem de pedido e ecoando na tela os caracteres lidos (processando corretamente o caractere **BACKSPACE**), até receber um **ENTER**. Se o usuário digitar apenas o **ENTER**, encerrar o programa com uma mensagem indicando que **o programa foi encerrado normalmente** por decisão do usuário. Após ler o nome do arquivo, o programa deve abrir o arquivo. Se não ocorrer erro na abertura do arquivo, continuar na etapa 3. Caso contrário, exibir na linha seguinte da tela uma mensagem **identificando qual foi o erro** e repetir esta etapa 2.
3. Limpar a tela e posicionar o cursor na primeira coluna (0) da linha 0. Definir o tempo de espera (número de “ticks” antes de exibir o próximo caractere lido) como sendo 1. Obter o tempo do sistema (usando a função 0 da INT 1AH) e armazenar como tempo inicial.
4. Ler um caractere do arquivo. Se o arquivo terminou, ir para a etapa 7. Se o caractere lido não for um '#', passar para a etapa 5. Ler os próximos dois caracteres do arquivo e converter para um valor binário, que deverá ser o novo valor do tempo de espera. Caso os dois caracteres após um '#' não sejam dígitos decimais ('0' a '9'), exibir uma mensagem indicando que havia um erro no conteúdo do arquivo e ir para a etapa 7. Ler mais um caractere do arquivo. Se não for um '#', passar para a etapa 5. Se for, repetir o processo de atualizar o tempo de espera até encontrar um caractere diferente de '#'.
5. Esperar até que tenha decorrido o tempo de espera definido.
6. Exibir na tela o último caractere lido. Atualizar o tempo inicial. Verificar se o usuário pressionou uma tecla qualquer (sem ler o caractere digitado). Se pressionou, ir para a etapa 8. Se não, voltar à etapa 4.
7. Fechar o arquivo, limpar a tela, posicionar o cursor na primeira posição da linha 0 e voltar à etapa 2.
8. Ler um caractere do teclado (sem eco). Se for um 'r' ou 'R', reiniciar a exibição do arquivo atual a partir do primeiro caractere. Se for um 'n' ou 'N', ir para a etapa 2 para processar um novo arquivo. Se for um **ESC**, terminar a execução do programa com uma mensagem indicando que **a execução do programa foi interrompida** a pedido do usuário. Se não for nenhum destes caracteres, ignorar o caractere lido e voltar para a etapa 4.

3. Formato do Arquivo de Entrada

O arquivo de entrada será um arquivo de texto no formato padrão do MS-DOS, ou seja, as informações estarão dispostas no arquivo de entrada em linhas terminadas por um par de caracteres **CR** (0DH) e **LF** (0AH). O programa deverá simplesmente ler o arquivo, um caractere por vez, e exibir o caractere lido na tela, exceto quando encontrar uma “tag” que especifica um novo valor (de 1 a 99) para o tempo de espera. A figura 1 mostra um exemplo de arquivo de entrada.

```
Arquivo de entrada exibido em velocidade maxima. Arquivo de entrada
exibido em velocidade maxima. Arquivo de entrada exibido em velocidade
maxima. Arquivo de entrada exibido em velocidade maxima.
Arquivo de entrada exibido em velocidade maxima. Arquivo de entrada
exibido em velocidade maxima.#99 E agora na velocidade minima possivel.
E agora na velocidade minima possivel. E agora na velocidade minima
possivel. E agora na velocidade minima possivel. E agora na velocidade
minima possivel. E agora na velocidade minima possivel.
Em seguida, mudando para velocidade intermediaria com o valor 50#50 na
"tag".
E continua nesta velocidade ate o fim do arquivo. Ate o fim do arquivo.
Ate o fim do arquivo. Ate o fim do arquivo. Ate o fim do arquivo. Ate o
fim do arquivo. Ate o fim do arquivo. Ate o fim do arquivo. Ate o fim do
arquivo. Ate o fim do arquivo. Ate o fim do arquivo. Ate o fim do
arquivo. Ate o fim do arquivo. FIM!
```

Figura 1 – Exemplo de arquivo de entrada

4. Entregáveis: o que deve ser entregue?

Deverá ser entregue, via Moodle da disciplina, o arquivo fonte com a solução do problema apresentado, escrito *na linguagem simbólica de montagem* dos processadores 80X86 da Intel (arquivo .ASM). Além disso, esse programa fonte deverá conter comentários descritivos da implementação.

Para a correção, os programas serão montados usando o montador **MASM** no ambiente **DosBox 0.74** e executados com diferentes arquivos de dados de entrada. A nota final do trabalho será proporcional às funcionalidades que forem atendidas pelo programa.

O trabalho deverá ser entregue até a data prevista, conforme programado no MOODLE. **Não será aceita a entrega de trabalhos após a data estabelecida.**

5. 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 (**tanto o trabalho original quanto os copiados 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.