Trabalho de Programação 3

Processador Intel (80x86)

1. Descrição Geral

Neste trabalho você deverá desenvolver um programa capaz de ler um arquivo em formato texto (arquivo de entrada), processar os dados lidos e, ao final, escrever um arquivo com os resultados obtidos (arquivo de saída).

No arquivo de entrada estará disponível uma lista de caracteres ASCII visíveis (20_{16} até $7E_{16}$) e de controle (00_{16} até $1F_{16}$). Quaisquer outros valores devem ser ignorados. Seu programa deverá ler esses bytes e processá-los.

Depois de lidos os bytes, seu programa deverá organizá-los em grupos de 4 bytes e realizar a soma, byte a byte, dos grupos obtidos.

Depois de realizada a leitura e os cálculos, o resultado deverá ser escrito em um arquivo de saída.

Seu programa deverá ser desenvolvido em linguagem simbólica de montagem do processador 8086 da Intel e executado no ambiente DosBox. Para montagem de seu programa fonte será usado o montador MASM 6.11.

2. Arquivo de Entrada

O nome do arquivo de entrada deverá ser fornecido, para seu programa, via digitação no teclado pelo usuário. O nome desse arquivo deverá ser um nome válido do DOS. Esse nome de arquivo pode ser formado por um NOME (até 8 caracteres) seguido de um TIPO (até 3 caracteres), separados por um "." (ponto). Exemplo:

```
arquivo.txt
```

O nome do arquivo fornecido também poderá ser formado sem o TIPO. Exemplo:

```
arquivo
```

No arquivo de entrada estará disponível uma lista de caracteres, codificados em ASCII, cada um deles ocupando um byte. O arquivo pode estar vazio (com 0 bytes) até o máximo de 128kBytes.

Observar que o final dos bytes que formam o arquivo de entrada é identificado através da sinalização de que se chegou ao final do arquivo. (ver exemplos de programas de leitura de arquivos, vistos em aula)

Exemplo de arquivo de entrada:

```
1AB0

```

Observar que, no exemplo, a segunda linha está vazia. Observar, também, que o arquivo contém um CR e LF no final da primeira linha. Portanto, ao analisar os bytes que estão contidos no arquivo, encontra-se o seguinte conteúdo (os caracteres estão representados por byte em formato hexa-decimal):

```
314142300D0A
```

3. Processamento

Os bytes que estão no arquivo devem ser lidos e agrupados em grupos de 4 bytes. Os bytes de cada grupo devem ser somados com os bytes que estão na posição (coluna) correspondente nos outros grupos. Eventuais valores de overflow das somas devem ser descartados.

Considere o seguinte arquivo de entrada, que está representado em ASCII e em hexa-decimal:

ASCII	1	A	B	0	C	D	2	E	F	\r	\n
Hexa-decimal	31	41	42	30	43	44	32	45	46	0D	0A

Agrupe os bytes em grupos de 4:

31	41	42	30
43	44	32	45
46	0D	0A	

Efetuar a soma dos bytes que estão na mesma coluna. Ou seja, soma 31+43+46, depois 41+44+0D, e assim por diante. Quando faltar um valor, como ocorre com a coluna mais a direita, deve-se considera-la como se fosse zero. O resultado das somas será:

BA	92	7E	75

Após realizar a soma, seu programa deverá colocar na tela um resumo das informações geradas. Essas informações são o número de bytes lidos do arquivo de entrada e o resultado da soma realizada (ver Figura 1).

```
C:>p1
Bytes: 11
Soma:  BA 92 7E 75

C:>
```

Figura 1 – Exemplo de resultado na tela do primeiro programa

Seu programa também deverá enviar os resultados obtidos para um arquivo de saída. O arquivo de saída deve ser formatado conforme descrito na próxima sessão.

4. Arquivo de Saída

O nome do arquivo de saída deverá ser formado pela concatenação do nome do arquivo de entrada e o TIPO “RES”. Exemplo: Se o nome do arquivo de entrada for “arquivo.txt”, o nome do arquivo de saída deverá ser:

```
arquivo.res
```

Outro exemplo: Se o nome do arquivo de entrada for “arquivo” (sem extensão de tipo), o nome do arquivo de saída deverá ser:

```
arquivo.res
```

Seu programa deverá colocar cada grupo de 4 bytes, em formato ASCII (portanto, 8 símbolos) em uma linha do arquivo de saída. Após cada conjunto de 8 símbolos, deve haver um CR e LF para a próxima linha.

Logo depois dos bytes lidos, seu programa deve escrever no arquivo de saída o resultado do somatório, organizado também em grupos de 8 símbolos.

Para o caso do exemplo de arquivo de entrada que tinha 11 bytes, o resultado no arquivo de saída (em ASCII) seria o seguinte:

```
31414230
43443245
460D0A0B
0A090207
0E0705
```

Observar que os bytes do arquivo de entrada estavam escritos em ASCII. Mas, ao serem colocados no arquivo de saída, foram apresentados em hexa-decimal. Por exemplo, o primeiro caractere do arquivo de entrada era "1". Assim, no arquivo de saída, aparecem os caracteres "31" (valor ASCII do símbolo "1").

Observar, também, que o resultado foi escrito no arquivo de saída em ASCII, mas cada byte representando um único "nibble" (conjunto de 4 bit). Portanto, todos são antecedidos de "zero". Por exemplo, o resultado do exemplo é "BA927E75". Então, no arquivo texto de saída, aparecerá "0B0A0902070E0705".

5. Entregáveis: o que deve ser entregue?

Deverá ser entregue, via Moodle da disciplina, **APENAS** 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, o programa será montado usando o montador **MASM 6.11** no ambiente **DosBox 0.74** e executado 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**.

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 (**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.