

Trabalho de Programação 3
Processador Intel (80x86)

1. Descrição Geral

Tomando por base o seu trabalho com a compactação de imagens implementada no processador RAMSES, você está sendo solicitado a implementar um algoritmo de compactação no processador 8086 da Intel, que deverá rodar no ambiente DosBox.

Agora, entretanto, você receberá a imagem original em um arquivo em disco, que deverá ser lido para a memória. Após, essa imagem na memória deverá ser compactada. Finalmente, as duas imagens: a original e a compactada deverão ser apresentadas na tela.

2. Especificação do Trabalho

O algoritmo a ser implementado no trabalho está descrito na figura 1, e deverá ser rigorosamente seguido.

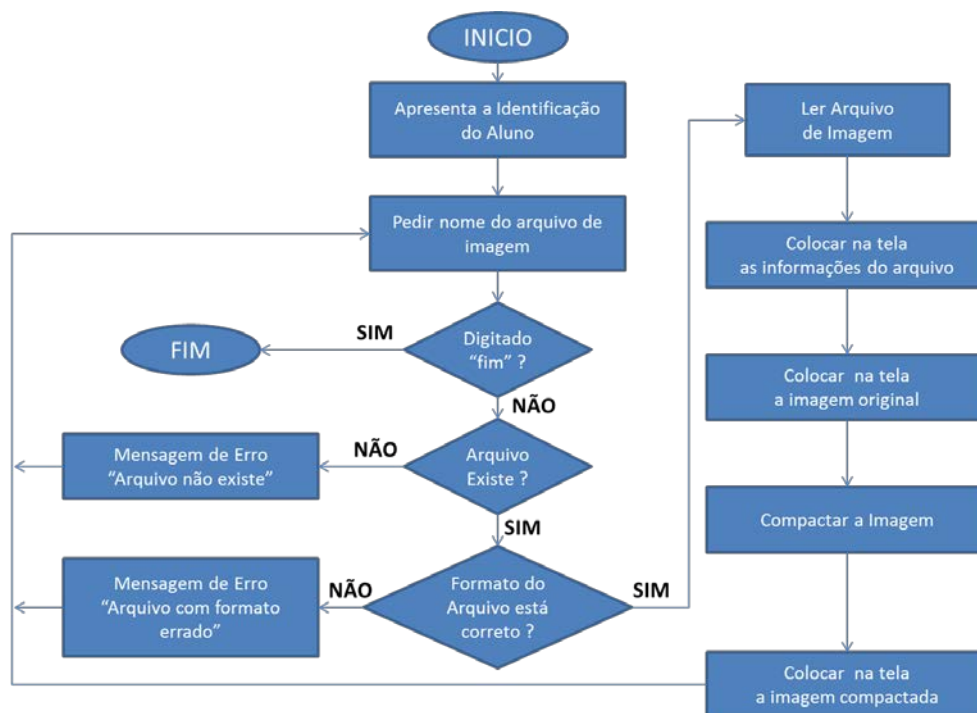


Figura 1 – Algoritmo do Trabalho

Ao iniciar, o programa deve colocar na tela a identificação do aluno: nome e número do cartão.

Em seguida, deve ser solicitado ao usuário que informe o nome do arquivo “.CW”, que contém a imagem original. A forma como os bytes estão organizados no arquivo está descrita no item “Formato do Arquivo .CW”. O nome do arquivo deve incluir o sufixo “.CW”. Se for digitado um nome vazio (somente a tecla ENTER), o programa deverá ser encerrado; qualquer outra palavra será tratada como o nome de um arquivo.

O programa deve tentar abrir o arquivo da imagem para verificar se ele existe. Caso ocorra um erro, deve-se considerar que o arquivo não existe.

Se o arquivo existir, o programa deve verificar se o formato de seus bytes está correto. Para isso devem ser verificados os campos ID, FTYP, HCSIZE, CID e BPP do arquivo (vide “Formato do Arquivo .CW”).

Se o arquivo existir e se estiver no formato correto, então o programa deve ler os dados de controle e verificar se o tamanho do arquivo, em *pixels*, pode ser apresentado na tela. Se o tamanho do arquivo for maior do que a área disponível na tela (vide “Leitura da Tela”), isto deve ser informado e nenhuma compactação deve ser realizada. Caso contrário, ler o

restante dos dados do arquivo, ou seja, os *pixels* da imagem. O programa deve colocar na tela uma mensagem de texto apresentando as dimensões da imagem: largura e altura.

Depois de lidos os *pixels*, o programa deverá colocá-los na tela, calcular a imagem compactada (vide “Compactação da Imagem”), colocar a imagem compactada na tela e voltar a solicitar um novo nome de arquivo.

3. Compactação da Imagem

A imagem original, lida do arquivo, é uma imagem monocromática que utiliza um bit para representar cada *pixel*. Esses *pixels* devem ser usados para calcular o *pixel* da imagem compactada usando o mesmo cálculo empregado no trabalho do RAMSES. Entretanto, no caso das imagens usadas neste trabalho, o cálculo da compactação será realizado usando *pixels* que podem assumir apenas os valores “0” ou “1”. Portanto, o cálculo de um *pixel* da imagem compactada deverá ser o seguinte:

- Se, dentre os quatro *pixels* analisados forem encontrados 0, 1 ou 2 *pixels* em “1”, o *pixel* compactado receberá o valor “0”;
- Se, forem encontrados 3 ou 4 *pixels* em “1”, o *pixel* compactado receberá o valor “1”.

4. Leiaute da Tela, Modo Gráfico e Modo Texto

O programa deve colocar na tela várias informações. Algumas informações são em formato “texto” e outras em formato “gráfico”. Mesmo nessas condições, quando o programa iniciar sua operação, deverá colocar a tela no modo gráfico, que tem 640 colunas por 480 linhas de *pixels*, e onde serão apresentadas as informações gráficas e texto. Ao encerrar sua operação, o programa deve retornar a tela para o formato texto padrão.

Para colocar a tela em formato gráfico o programa deverá utilizar uma chamada de sistema para selecionar esse modo, que tem as seguintes características:

- Modo gráfico com dimensões 640 colunas por 480 linhas de *pixels*, identificadas pelas coordenadas nos intervalos [0, 639] e [0, 479], respectivamente, sendo a coordenada (0,0) o canto superior esquerdo da tela e (639,479) o canto inferior direito;
- Suporta escrita de caracteres ASCII utilizando um sistema de coordenadas de caracteres, formando uma área de 80 colunas por 30 linhas de caracteres, identificados pelas coordenadas nos intervalos [0;79] e [0;29], respectivamente, sendo o caractere da coordenada (0,0) o do canto superior esquerdo e (79,29) o do canto inferior direito;
- Caracteres com 8 colunas por 16 linhas de *pixels*;
- Modo monocromático (P&B – Preto e Branco)

Para ativar o modo gráfico deve-se usar a INT 10H com AH=0 e AL=11H (modo gráfico, com resolução de 640 colunas por 480 linhas de *pixels*); para retornar ao modo texto padrão do DOS deve-se usar a INT 10H com AH=0 e AL= 7 (modo texto, com 80 colunas por 25 linhas de caracteres). Note que no modo gráfico utilizado pode-se superpor gráfico (utilizando as coordenadas de *pixel*) e texto (utilizando as coordenadas de caractere)

As informações a serem colocadas na tela devem obedecer a um leiaute formado por três áreas, conforme representado na figura 2, e cujas características são as seguintes:

- **Área para imagem original:** área com 430 colunas por 400 linhas onde deve ser apresentada a imagem original, sendo que o *pixel* superior-esquerdo da imagem original deve ser colocado nas coordenadas gráficas (0,0) da tela;
- **Área para imagem compactada:** área com 210 colunas por 400 linhas onde deve ser apresentada a imagem compactada, sendo que o *pixel* superior-esquerdo da imagem compactada deve ser colocado nas coordenadas gráficas (430,0) da tela;
- **Área para texto:** área com 640 colunas por 80 linhas de *pixels* onde serão apresentadas todas as mensagens de texto. A primeira linha de *pixels* dessa área corresponde à linha 25 de texto. Portanto, as coordenadas de texto (0,25) correspondem ao caractere da posição superior-esquerda dessa área.

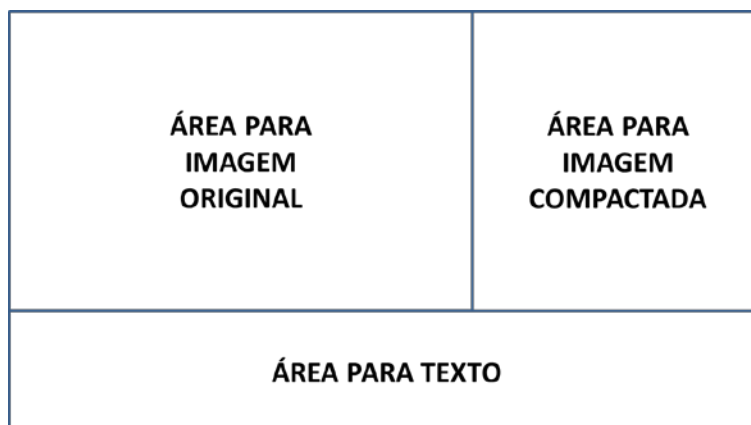


Figura 2 – Leiaute da Tela

5. Escrevendo na Tela

O programa deverá apresentar na tela caracteres e *pixels*. Para isso, devem ser utilizadas diferentes chamadas de sistema (interrupções de software).

Para ligar (ponto branco) ou desligar (ponto preto) um *pixel* na tela **deve-se** utilizar a interrupção INT 10H com AH=0CH. Essa interrupção requer a passagem dos seguintes valores nos registradores:

- BH = 0
- AL = 00H (preto) ou 01H (branco)
- CX = coordenada gráfica “X” do *pixel* (coluna)
- DX = coordenada gráfica “Y” do *pixel* (linha)

Para escrever um *string* na tela deve-se chamar duas interrupções: uma para posicionar o cursor e, em seguida, outra para escrever o *string*. Para posicionar o cursor **deve-se** utilizar a interrupção INT 10H com AH=02H. Essa interrupção requer a passagem dos seguintes valores nos registros:

- BH = 0
- DL = coordenada de texto “X” do caractere (coluna)
- DH = coordenada de texto “Y” do caractere (linha)

Para escrever o *string* na tela pode-se utilizar, por exemplo, a INT 21H com AH=2, que coloca um caractere na tela, ou então a INT 21H, com AH=9, que escreve um *string* terminado por ‘\$’ na tela.

6. Formato do Arquivo .CW

Os bytes de um arquivo no formato .CW estão organizados em três áreas:

- Cabeçalho (*header*) do arquivo, que ocupa 6 bytes;
- Cabeçalho de controle, que ocupa 6 bytes;
- Área da imagem, que ocupa o restante do arquivo e seu tamanho depende do tamanho da imagem armazenada.

O **Cabeçalho do Arquivo** tem três campos:

- ID: dois bytes usados para identificar o tipo do arquivo. Sempre deverá conter os valores ASCII da letra “C” seguido pela da letra “W”;
- FTYP: dois bytes, em formato *little-endian*, usados para identificar o conteúdo do arquivo. No caso deste trabalho, deverá conter, sempre, o valor 1 (0001H);
- HCSIZE: dois bytes, em formato *little-endian*, usados para informar o tamanho do Cabeçalho de Controle. No caso deste trabalho deverá conter, sempre, o valor 6 (0006H).

O **Cabeçalho de Controle** tem quatro campos:

- CID: um byte que informa se a imagem é P&B ou colorida. No caso deste trabalho deverá conter, sempre, o valor 0 (00H), pois as imagens usadas no trabalho serão sempre monocromáticas;

- BPP: um byte que informa o número de bits usados para representar um *pixel*. No caso deste trabalho, deverá conter, sempre, o valor 1 (01H), pois as imagens usadas no trabalho serão monocromáticas com 1 bit para cada *pixel*;
- WIDTH: dois bytes, em formato *little-endian*, usados para identificar a largura da imagem, em *pixels*. Caso a imagem armazenada no arquivo tenha uma largura maior do que 430 *pixels* (largura da área de tela para a imagem original), o programa deverá informar que a imagem excede a área da tela e solicitar novo arquivo a ser processado.
- HEIGHT: dois bytes, em formato *little-endian*, usados para identificar a altura da imagem, em *pixels*. Caso a imagem armazenada no arquivo tenha uma altura maior do que 400 *pixels* (altura da área de tela reservada para a imagem original), o programa deverá informar que a imagem excede a área da tela e solicitar novo arquivo a ser processado.

A **Área da Imagem** contém os bytes onde cada bit representa um *pixel*. Os *pixels* estão armazenados como bits da seguinte forma:

- A imagem está armazenada nessa área “por linha”. Isso significa que os primeiros bytes (consecutivos) correspondem à primeira linha de *pixels* da imagem.
- A imagem está armazenada “de cima para baixo”. Isso significa que a primeira linha armazenada nessa área corresponde à linha superior da imagem; a última linha de *pixels* dessa área corresponde à linha inferior da imagem;
- O bit 7 do byte representa o *pixel* mais à esquerda. Portanto, por exemplo, o bit 7 do primeiro byte de uma linha corresponde ao *pixel* mais à esquerda da imagem. Notar que, com isso, o bit 7 do primeiro byte dessa área corresponde ao *pixel* superior esquerdo da imagem;
- Último byte de imagens cuja largura não seja múltipla de 8. Nesse caso, alguns bits do último byte das linhas não pertencem à imagem. Por exemplo, se a imagem tiver 50 *pixels* de largura, serão necessários 7 bytes para representá-los. Destes bytes, os primeiros 6 bytes (que representam 48 bits) são totalmente utilizados enquanto que são utilizados apenas 2 bits do sétimo byte (bits 7 e 6 do byte).

7. 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* do 8086 da Intel (arquivo .ASM). Além disso, esse programa fonte deverá conter comentários descritivos da implementação. Sugere-se utilizar os comandos da linguagem “C” para essa finalidade.

Para a correção, os programas serão montados e executados no ambiente DosBox, com diferentes arquivos de dados. 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 serão aceitos trabalhos entregues além do prazo estabelecido.**

8. 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.