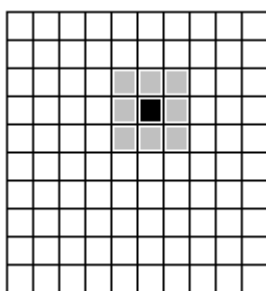


Trabalho Prático 3 - INTEL

Escrever para a família de processadores x86 da INTEL um simulador do "Jogo da Vida", um autômato celular desenvolvido por John Conway (detalhes podem ser vistos em http://en.wikipedia.org/wiki/Conway's_Game_of_Life). A definição do jogo é a mesma utilizada para o trabalho do RAMSES.

Definições

O autômato é formado por uma matriz de células que podem estar em um de dois estados: morta ou viva. Essas células morrem, nascem ou permanecem vivas, dependendo do número de células vivas em sua vizinhança (as oito células que a cercam).



Uma célula viva (em preto) e sua vizinhança (em cinza)

Modelo de Nascimento/Morte

As células nascem ou morrem em turnos, chamados de "geração". A cada geração, verifica-se a vizinhança de cada célula e decide-se seu destino de acordo com um conjunto de regras que determinam se ela deverá nascer, morrer ou permanecer como está. As regras a serem observadas são as seguintes:

1. Uma célula viva, que tenha menos de dois vizinhos vivos, morre (de solidão);
2. Uma célula viva, com mais de três vizinhos vivos, morre (de superlotação);
3. Uma célula viva com dois ou três vizinhos vivos permanece viva;
4. Uma célula morta, com exatamente três vizinhos vivos, nasce.

Os nascimentos/mortes ocorrem simultaneamente em todas as células, ou seja, todas as células de uma matriz passam para a próxima geração ao mesmo tempo. Dada uma matriz com um padrão inicial de células vivas e mortas, aplica-se o conjunto de regras acima e calcula-se o novo estado de cada célula.

Conforme os novos valores de estado forem determinados, deverão ser armazenados em uma matriz auxiliar, de maneira que todas as células sejam calculadas a partir de um mesmo conjunto de valores. Ao término da determinação do estado de todas as células, a matriz auxiliar estará completa e, então, poderá ser copiada para a matriz principal e iniciada uma nova geração.

A implementação

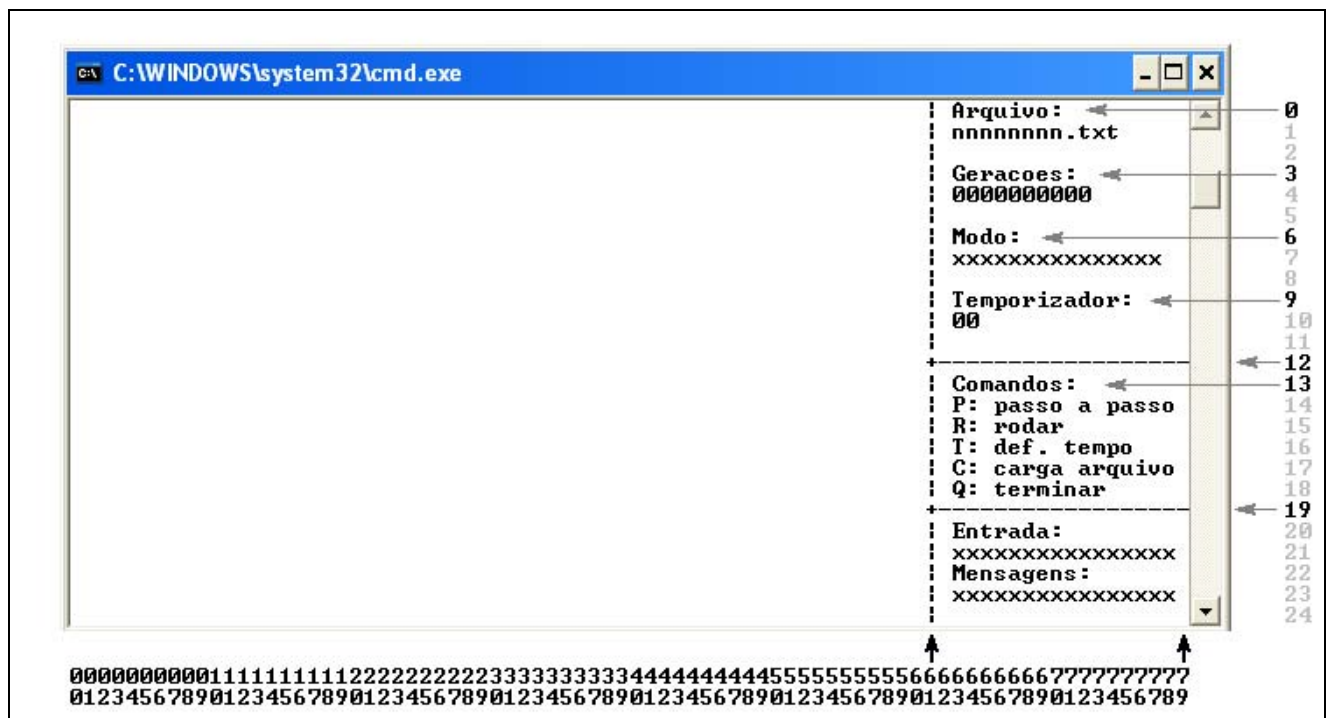
Escrever um programa para um processador x86 e ambiente DOS que efetue a simulação de um autômato celular em uma matriz com 25 linhas e 60 colunas. Diferentemente do trabalho do RAMSES, não existe uma borda externa de células mortas. Entretanto, considera-se que as células fora da matriz definida estão sempre mortas.

O estado inicial das células deverá ser lido de um arquivo texto armazenado em disco. Este arquivo será sempre constituído de 25 linhas com 60 caracteres cada, ou seja, o arquivo contém uma

“imagem” da matriz de células. No arquivo, o caractere ASCII “.” (ponto) representa um célula morta, e o caractere ASCII “#” representa uma célula viva. O arquivo está no formato texto DOS, ou seja, cada linha termina com os caracteres ASCII *carriage return* (código 13) e *line feed* (código 10). Garante-se que a última linha do arquivo também tem esta terminação.

Notar que, ao colocar um padrão de células na tela, as células mortas não devem aparecer enquanto que as células vivas deve ser assinaladas, de tal forma que estarão visíveis apenas as células vivas.

O programa deverá executar em modo texto, de 80 colunas por 25 linhas. Observe-se que, pelo DOS, a linha superior é a de índice zero, e a linha mais inferior da tela é a de índice 24. De forma semelhante, a coluna mais à esquerda é a de índice zero enquanto que a mais à direita é a de índice 79. A matriz de células deverá ser apresentada nas linhas 0 até 24 e nas coluna 0 até 59. As colunas 60 a 79 serão utilizadas para apresentar várias informações relativas à operação do programa e para a entrada de dados, conforme definido a seguir:



Os comandos (P, R, T, C e Q) devem ser lidos do teclado, sem ecoar para a tela, e sem ser necessário digitar *Enter*. Eles devem ser entendidos tanto em caixa baixa (minúsculas) como em caixa alta (maiúsculas). Os comandos a serem implementados são:

Carga de arquivo (C): carrega o estado inicial da matriz de células a partir de um arquivo em disco, e exibe este estado na tela. O usuário deverá fornecer o nome do arquivo na área de entrada (campo “Entrada”). O arquivo deverá estar no mesmo diretório onde está o programa em execução, e terá no máximo 8 caracteres no nome e 3 caracteres de sufixo (afinal, estamos no tempo do DOS!). Na entrada do nome, deve ser processado *Backspace*, e deve ser previsto o caso do arquivo não existir. Após uma carga com sucesso, o nome do arquivo deve ser escrito na área de informações (campo “Arquivo:”), a matriz deve ser exibida na tela (colunas 0 a 59) e o contador de gerações deve ser zerado. Caso ocorra algum erro na leitura do arquivo, uma mensagem adequada deve ser apresentada no campo de “Mensagens”.

Modo passo-a-passo (P): realiza o cálculo de uma geração, exibindo o resultado na tela (e incrementando o contador de gerações). Enquanto estiver neste modo, a área de informações (campo “Modo”) deve indicar que o modo é “Passo a passo”.

Def. Tempo (T): define o temporizador do modo de execução contínua, em ticks (1 tick corresponde a 1/18,2 segundos). Um valor de zero significa que a execução deve ser a mais rápida possível, e valores iguais ou superiores a 1 indicam a quantidade de ticks que deve transcorrer entre uma geração e a seguinte. O valor do temporizador deve ser lido da área de entrada (campo “Entrada”). Deve ser tratado BACKSPACE, e deve ser feita a consistência do número. Valores fora do intervalo [0;99] devem ser descartados e a entrada deve ser repetida. Um valor válido para o temporizador deve ser escrito na área de informações (Campo “Temporizador”).

Rodar continuamente (R): rodar continuamente, atualizando a tela, tanto na parte da matriz de células como no contador de gerações. A velocidade de execução deve ser determinada pelo temporizador. Para o contador de gerações deve ser utilizada uma variável sem sinal, em 32 bits. Este número deve ser mostrado na área de informações, com 10 dígitos (incluir zeros à esquerda, caso necessário). Quando em modo de execução contínua, a área de informações deve indicar que o modo é “Contínuo”. Qualquer tecla digitada deve interromper a execução (após completar o cálculo da próxima geração e a sua exibição), mas esta tecla não deve ser interpretada como o próximo comando.

Quit (Q): Sair, ou seja, o programa deve ser encerrado e deve-se retornar ao DOS.

Entrega do Trabalho

O trabalho deve ser desenvolvido **individualmente** e usando um montador assembler para os processadores da INTEL e em ambiente DOS (terminal do Windows).

Garante-se que os dados de entrada fornecidos nos arquivos de descrição estão corretos, segundo a definição acima.

Na avaliação do trabalho será levada em consideração a “interface com o usuário“. Quanto mais amigável e intuitiva for, e quando melhor o aspecto visual, melhor poderá ser a nota obtida.

O trabalho deverá ser entregue via Moodle, contendo um arquivo fonte comentado (.asm) e um executável (.exe). Para nomear os arquivos, utilize os seus dois primeiros nomes, sem espaço em branco e sem acentos. Por exemplo, o aluno Um Três Dois de Oliveira Quatro deverá nomear seus arquivos UmTres.asm e UmTres.exe. Os arquivos do trabalho deverão ser compactados em um arquivo UmTres.rar (ou UmTres.zip) e entregues através do Moodle.

Arquivos de teste estarão disponíveis via Moodle.

Data final de entrega: dia 25 de novembro, conforme especificado no Moodle.