

TRABALHO DE PROGRAMAÇÃO ASSEMBLY

Componentes	1 ou 2 alunos por trabalho.
Apresentação	O trabalho deve ser apresentado no dia 2 de julho na sala 109 do bloco 71. Após a apresentação será requisitado que o aluno envie o trabalho (código fonte e toda e qualquer documentação relacionada) via webfólio.
Avaliação	Os trabalhos serão avaliados através de diversos fatores, tais como, atendimento aos requisitos do trabalho, organização do código, documentação do código, utilização da linguagem Assembly, e nível algorítmico da solução. Verifique a ficha de avaliação no final deste documento.
Relatório	<p>O relatório deve seguir o formato para publicação de artigos da Sociedade Brasileira de Computação (disponível no acervo da disciplina). Não é necessário incluir o Abstract. Cada grupo deve produzir um relatório com pelo quatro seções (não serão aceitos mais de um arquivo para o relatório):</p> <ol style="list-style-type: none">1. Algoritmo: esta seção deve descrever o algoritmo utilizado para resolver o problema. Deve descrever os principais elementos do programa, apontando as principais funções. Esta seção pode ter o suporte de diversas figuras (devidamente identificadas e referenciadas no texto) incluindo um fluxograma da solução que facilite a compreensão tanto do algoritmo como do código.2. Estruturas de Dados: esta seção deve descrever como foram armazenadas em memória as informações dos arquivos (exceto o arquivo de ativação, o qual não deve ser armazenado em memória) e outras auxiliares.3. Conclusões: o grupo deve discutir as dificuldades encontradas no desenvolvimento do programa e o que aprendeu com o trabalho. Que melhorias o programa ainda pode sofrer.
Arquivo Fonte	Deve incluir comentários das instruções ou blocos de código, um cabeçalho que identifique o nome do arquivo e o grupo.

CENÁRIO

Uma casa possui um determinado número de sensores sem fio de movimentos e de abertura/fechamento de portas. Como estes sensores não são confiáveis, o protocolo

de envio de sinais (o sinal contém a identificação do sensor) foi definido de tal forma que cada sensor envia 5 vezes o mesmo sinal a cada ativação.

Dado os diversos problemas que podem ocorrer durante a transmissão (sinal fraco, perda do sinal, corrompimento do sinal, interferência, reflexo do sinal, entre outros) uma antena receptora poderá receber 0, 1, 2, ..., 5 vezes o sinal enviado. Ao ser recebido, o sinal é prontamente gravado (data/hora de recebimento e a identificação do sinal) em um arquivo no computador.

PROBLEMA

Uma das medidas de declínio cognitivo mental é a velocidade do caminhar. Por isso, alguns sensores de movimentos foram colocados em uma sequência em um corredor da casa para realizar a medição da velocidade do caminhar (medido através dos tempos de ativações dos 3 sensores). Entretanto, o arquivo de ativações coletado na casa contém sensores que não pertençam a casa, códigos corrompidos, replicação de sinais (como o mesmo é enviado 5 vezes). Por isso, é necessário realizar a limpeza deste arquivo e a detecção destas sequências de 3 sensores, previamente definidos.

OBJETIVO

O programa deve ler um arquivo de ativações coletado pelo computador e a lista de sensores instalados na casa e produzir as sequências de sensores detectados para uma determinada casa.

A fim de poder avaliar a eficácia do sistema, outros dois arquivos também são gerados:

- 1) arquivo com somente os sensores que pertençam à lista de sensores válidos e o número de ativações por sensor e
- 2) outro arquivo com a mesma informação, mas com somente os sensores instalados na casa. Esta lista deve seguir a ordem cronológica de ativações.

Maiores informações sobre o formato dos arquivos de entrada e de saída serão apresentadas adiante.

SINAL DO SENSOR

Cada vez que o sensor é ativado (ou desativado), ele envia cinco vezes o mesmo sinal (protocolo X10). O sinal é composto por dois bytes que serão recebidos isoladamente. O primeiro byte corresponde à identificação do sensor, o qual é gravado pela sua representação hexadecimal. O segundo byte representa o estado do sensor: ligado e desligado (como o código do sensor, esta informação também é representada em hexadecimal). Dois sensores podem enviar sinais simultaneamente. Entretanto o computador somente poderá recebê-los sequencialmente. Após o sensor de movimento ser ativado ele pode ser ativado novamente somente depois de 5 segundos.

A fim de garantir a qualidade dos dados, cada byte é codificado utilizando 2 bytes. Assim, cada byte possui uma representação com 4 caracteres hexadecimal. Para ser considerado válido, o primeiro e terceiro caracteres devem somar 15 e o segundo e quarto também devem somar 15. Por exemplo, o sinal 609f é válido pois 6+9 = 15 e 0+f = 15. Se o sensor não for válido, não é necessário buscar o código.

REGRAS

O programa deve seguir as seguintes regras:

1. Para que a ativação seja válida, toda ativação deve possuir uma entrada no arquivo com o código do sensor (primeiro byte) seguida de uma linha com o código do estado do sensor (segundo byte) com uma diferença de tempo não maior que 20 ms;
2. Ativações emitidas pelo mesmo sensor com menos de 4 segundos de diferença serão consideradas pertencentes ao mesmo grupo de repetição do sinal.

FORMATO DOS ARQUIVOS

ENTRADA

Arquivo de Sensores Válidos

Este arquivo, chamado X10Map.txt, armazena os dados dos sensores de movimentos e de portas válidos. Cada linha representa um sensor com as seguintes informações:

<identificação> **<código>** **<tipo>**

onde:

<identificação>: cadeia de 4 caracteres (representando um número em hexadecimal) que identifica o sensor.

<código>: representa o código numérico do sensor (inteiro). Este número somado com o número do estado do sensor define a identificação decimal do sensor.

<tipo>: representa o tipo do sensor (caractere): **M** para sensor de movimento e **P** para sensor de portas.

A seguir, um exemplo do arquivo de sensores é apresentado:

```
609f 0 M
708f 16 M
40bf 32 M
f3fc 760 D
767c 770 D
5b54 780 D
```

O exemplo acima mostra 6 sensores. Os 3 primeiros sensores são sensores de movimento (terceira coluna é igual a M) e os 3 últimos são sensores de porta (terceira coluna é igual a D). Note que o primeiro sensor aparece no exemplo do arquivo de ativação. Este arquivo é chamado X10Map.txt.

Arquivo de Estado de Sensores

Este arquivo, chamado X10OnOff.txt, armazena os códigos para os estados dos sensores. Cada linha representa um tipo de estado com as seguintes informações:

<identificação> <código> <tipo>

onde:

<identificação>: cadeia de 4 caracteres (representando um número em hexadecimal) que identifica o estado do sensor. Este valor sempre segue a identificação do sensor (conforme exemplo do arquivo de ativações).

<código>: representa o código numérico do estado (inteiro). Este número somado com o número do sensor (segunda coluna do arquivo de sensores) define a identificação decimal do sensor.

<tipo>: representa o estado do sensor (caractere). No caso do sensor de movimento, 1 para ativado e 0 para desativado. No caso do sensor de portas, 1 para fechada e 0 para aberta.

A seguir, um exemplo do arquivo de estados é apresentado:

```
00ff 1 1
10ef 2 1
20df 1 0
30cf 2 0
```

Arquivo de Ativações

Este arquivo armazena todas as ativações de sensores de movimento e de portas de uma residência. Cada linha representa um sinal de um sensor (note que cada sensor pode enviar até 5 ativações por evento) e a mesma possui a seguinte informação:

<número sequencial> <data> <hora> <sinal do sensor>

onde:

<número sequencial>: este número é utilizado para garantir que as ativações estão sendo recebida na sequência correta. Este número será ignorado neste trabalho.

<data>: representa a data da ativação no formato aaaammdd (por exemplo: 20061030 representa 30 de outubro de 2006).

<hora>: representa a hora da ativação no formato hhmmssiii, onde i representa milissegundos (por exemplo: 143040333 representa 14 horas, 30 minutos, 40 segundos e 333 milissegundos)

<sinal do sensor>: uma cadeia de 4 caracteres que identifica o sensor que gerou a ativação ou o tipo de sinal (isto é, ligado/desligado). Estes sinais estarão armazenados em 2 arquivos diferentes: um para as identidades dos sensores e outros para o estado dos sensores.

A seguir, um exemplo de ativações é apresentado:

18D	20060107	000037656	609f
18E	20060107	000037656	20df
18F	20060107	000037765	609f
190	20060107	000037765	20df
191	20060107	000037875	609f
192	20060107	000037875	20df

O exemplo mostra 3 ativações do sensor 609f. Existem várias informações neste exemplo:

1. O sinal 20df representa o desligamento do sensor;
2. O sinal de desligamento possui a mesma hora que a identificação do sensor;
3. Os três sinais chegaram com um intervalo de 110ms.

Arquivo de Sensores Instalados

Este arquivo armazena os dados dos sensores de movimentos e de portas válidos. Cada linha representa um sensor com as seguintes informações:

<código> **<sequência>**

onde:

<código>: representa o código numérico do sensor (inteiro). Este número somado com o número do estado do sensor define a identificação decimal do sensor.

<sequência>: define se o sensor está na sequência de sensores de interesse através da sua posição na sequência (valores de 1 a 3). Caso o sensor não pertencer a sequência, este campo terá o valor 0.

A seguir, um exemplo do arquivo de sensores é apresentado:

18	0
86	3
103	1
171	2

O exemplo acima mostra 4 sensores: 18, 86, 103, e 171. Os sensores 103, 171, e 86 formam uma sequência nesta ordem.

SAÍDA

Arquivo de Ativações Detectadas

O arquivo resultante deve possuir o seguinte formato:

<data> <hora> <ident-decimal> <estado> <nro de ativações>

onde:

<data>: representa a data da ativação (ou da primeira ativação de um grupo de ativações) no formato aaaammdd.

<hora>: representa a hora da ativação (ou da primeira ativação de um grupo de ativações) no formato hhmmssiii.

<ident-decimal>: representa o código decimal resultante da soma do código do sensor e do seu estado.

<estado>: representa o estado do sensor, conforme o arquivo de estado dos sensores.

<nro de ativações>: representa o número total de ativações que foram recebidas (lembre-se que ativações com horas de chegada separadas por menos do que 4 segundos correspondem ao mesmo evento que ativou o sensor).

Por exemplo, a seguinte sequência de ativações:

9DC	20060107	000028703	708f
9DD	20060107	000028703	10ef
9DE	20060107	000028812	708f
9DF	20060107	000028812	10ef
9E0	20060107	000028921	708f
9E1	20060107	000028921	10ef
9E2	20060107	000029031	708f
9E3	20060107	000029031	10ef
9E4	20060107	000029140	708f
9E5	20060107	000029140	10ef
9E6	20060107	000037000	708f
9E7	20060107	000037015	10ef
9E8	20060107	000037109	708f
9E9	20060107	000037125	10ef
9EA	20060107	000037218	708f
9EB	20060107	000037234	10ef
9EC	20060107	000037328	708f
9ED	20060107	000037343	10ef
9EE	20060107	000037656	609f
9EF	20060107	000037656	20df
9F0	20060107	000037765	609f

9F1	20060107	000037765	20df
9F2	20060107	000037875	609f
9F3	20060107	000037875	20df

produzirá a seguinte saída:

20060107	000028703	18	1	5
20060107	000037000	18	1	4
20060107	000037656	1	0	3

As identificações dos sensores 18 e 1 são obtidas da seguinte maneira:

- Sensor 18: o código de 708f possui valor 16 no arquivo de sensores (X10Map.txt) e o estado correspondente, 10ef, possui valor decimal 2 no arquivo de estado dos sensores (X10OnOff.txt). A soma dos dois valores é 18, a identificação decimal do sensor.
- Sensor 1: o código de 609f possui valor 0 no arquivo de sensores (X10Map.txt) e o estado correspondente, 20df, possui valor decimal 1 no arquivo de estado dos sensores (X10OnOff.txt). A soma dos dois valores é 1, a identificação decimal do sensor.

Os estados dos sensores correspondem ao código que segue o código do sensor: 1 para o código 10ef e 0 para o código 20df.

Note que foram produzidas 2 linhas para o sensor 18. Isto se deve ao fato que as ativações possuem um intervalo de aproximadamente 8 segundos entre as horas 00:00:29.140 e 00:00:37.000.

Arquivo de Ativações Válidas

Este arquivo possui o mesmo leiaute do arquivo de ativações detectadas, mas com somente as ativações dos sensores instalados.

Assumindo que os arquivos de sensores instalados e de ativações sejam iguais aos dos exemplos acima, este arquivo terá a seguinte saída:

20060107	000028703	18	1	5
20060107	000037000	18	1	4

Note que as ativações do sensor 1 foram eliminadas do arquivo.

Arquivo de Sequências Detectadas

O arquivo resultante deve possuir o seguinte formato:

<data 1> <hora 1> <data 2> <hora 2> <data 3> <hora3>

onde:

<data>: representa a data da ativação no formato aaaammdd de cada sensor da sequência.

<hora>: representa a hora da no formato hhmmssiii de cada sensor da sequência.

FUNCIONAMENTO DO PROGRAMA

Em linhas gerais, o programa deve funcionar da seguinte maneira:

- O programa deve primeiramente ler os arquivos X10Map.txt (arquivo de sensores) e X10OnOff.txt (arquivo do estado de sensores). Certifique-se que esta leitura é feita com sucesso.
- Ler os nomes dos arquivos de sensores instalados e de ativações a ser processado. Certifique-se que a abertura destes arquivos é feita com sucesso.
- Execute o processamento do arquivo de ativações conforme as regras definidas neste documento. O resultado deste processamento serão três arquivos com o resumo das ativações detectadas, ativações dos sensores instalados, sequências detectadas e informações do arquivo. O nome do arquivo resultante deve possuir o mesmo nome que o arquivo processado, exceto que a extensão deve ser igual a **.det** para o resumos de ativações, a **.val** para as ativações dos sensores instalados e **.seq** para o de sequências detectadas. Por exemplo, se o arquivo fornecido pelo usuário é chamado **ativa.txt**, os arquivos resultantes serão **ativa.det**, **ativa.val** e **ativa.seq**. Deve-se registrar o tempo inicial.
 - O programa deve buscar primeiramente um código de sensor válido. Isto é, achar um código que pertença ao arquivo de sensores (X10Map.txt).
 - Ao achar um código válido de sensor, o programa deve buscar um código válido que represente o estado do sensor (X10OnOff.txt). Caso não encontrar a menos de 20ms (regra número 1) na próxima linha, o programa deve voltar a procurar um código de sensor (com o mesmo código lido!).
 - Lembre-se que ativações completas (código do sensor + estado) do mesmo sensor com intervalo menor do que 5 segundos devem ser consideradas como produzidas pelo mesmo evento.
 - Ativações inválidas devem ser ignoradas.
 - O programa deve gerar uma tela centralizada, conforme a imagem a seguir, que mostra os nomes dos arquivos, a linha que está sendo processada (deve ser escrita em vermelho), o número de ativações válidas (deve ser escrita em verde), o número de ativações válidas dos sensores instalados (deve ser escrita em laranja), o número de sequências detectadas (deve ser escrita em azul), sensores foram encontrados (código decimal) e os sensores instalados (código decimal). A lista de sensores encontrados (não pode haver sensores repetidos e os mesmos devem ser mostrados em hexa) deve permitir múltiplas colunas.

Sensores Instalados:		Ativacoes:	
Linha: 9999999	Válidas: 999999	Instaladas: 999999	Sequencias: 999
Sensores Instalados	Sensores Encontrados		

- Ao terminar, o programa deve buscar o tempo final e mostrar o tempo transcorrido.

ESPECIFICAÇÃO QUALITATIVA

O foco durante a construção do programa deve ser a corretude. Gerar exatamente o formato da saída. Não gere saída adicional, pois o teste do programa será com dezenas de milhares de ativações será feito automaticamente. Garanta que o seu programa realize corretamente a codificação.

Seu programa deve rodar eficientemente tal que ele não desperdice comparações. Os arquivos de sensores devem ser transferidos para a memória principal, mas o mesmo não deve ser feito para o arquivo de ativações.

DICAS

Leitura dos Arquivos

Note que os arquivos são muito semelhantes. A mesma proc pode ser utilizada para ler os arquivos.

Leitura de Strings

A leitura de uma string pode ser realizada através da função 0AH da interrupção 21H. Esta função utiliza DX para apontar para a área onde devem ser armazenadas as informações de tamanho máximo, número de caracteres lidos e a string propriamente dita.

Valores dos registradores

- AH = 0AH
- DS:DX = Endereço inicial da área de armazenamento

A área de armazenamento possui a seguinte estrutura:

- BYTE 0 = Quantidade máximas de bytes na área
- BYTE 1 = Quantidade de bytes lidos

- do BYTE 2 até [BYTE 0] + 2 = caracteres lidos

Os caracteres são lidos e armazenados num espaço de memória definido por DS:DX. A estrutura deste espaço indica que o primeiro byte representará a quantidade máxima de caracteres que pode ser lida. O segundo, a quantidade de caracteres lidos e, no terceiro byte, o início onde eles são armazenados.

Quando se atinge a quantidade máxima permitida, ouve-se o som do alto-falante e qualquer caractere adicional é ignorado. Para finalizar a entrada, basta digitar [ENTER].

AVALIAÇÃO

Funcionalidade (Comportamento)	Pontos
Leitura do nome dos arquivos e geração do nome do arquivo com a extensão correta	2
Tela de informações	6
Geração do Arquivo de Ativações Detectadas	18
Geração do Arquivo de Ativações Válidas	14
Geração do Arquivo de Sequências Detectadas	18
Calcula e mostra o tempo decorrido	2
Total	60

Técnico (Implementação)	Pontos
Mesma proc para a leitura dos arquivos de estado e sensores válidos	5
Um único laço para a geração dos arquivos	8
Busca do sensor na memória utiliza a mesma proc para estado e sensores válidos	5
O programa utiliza instruções de deslocamentos e manipulação de strings	5
Cálculo do tempo decorrido entre duas ativações utiliza proc	4
Valida o código do sensor ou estado antes de buscar	3
Total	30

Relatório	Pontos
Atende aos requisitos	10
Total	10