USP – ICMC – SSC

SSC0510 - Arquitetura de Computadores

Professor responsável: *Fernando Santos Osório*
Semestre: 2010/2
Horário: Quarta 21h00

E-mail: fosorio@icmc.usp.br
fosorio@gmail.com
Web: ***http://www.icmc.usp.br/~fosorio /***

TRABALHO PRÁTICO – REC 2010/2

Definição de 23/12/2010 (versão 0.1)

[Descrição Geral]

Este trabalho consiste em implementar um simulador de um controlador de um elevador, usando o microprocessador 6502 (use o mesmo simulador do 6502 que já foi adotado nos demais trabalhos e estudos realizados junto a disciplina – Simulador 6502 de Michal Kowalski). O trabalho consiste em implementar, usando as rotinas de entrada de teclado e saída de tela do simulador do 6502, um programa que permita simular o controle de um elevador, com as funções de chamada do elevador, deslocamento e exibição do estado do elevador.

TRABALHO INDIVIDUAL DA REC:
IMPLEMENTAÇÃO DO CONTROLE DE UM ELEVADOR (simulado)
****

>> ELEVADOR 6502 <<

Usar o Simulador para microprocessador 6502 para desenvolver a aplicação.
Programação em ASM para o 6502 (Simulador Kowalski)

==========

*** Funcionamento do Sistema de Controle do Elevador:**

O sistema implementado usando o 6502 irá servir para controlar um elevador, onde o elevador possui um laço de controle composto de:
> Controle da Movimentação do Elevador (Atuação no Elevador)
> Leitura de Comandos de Entrada: Chamada e Destino

Uma vez iniciada a execução do programa, o elevador inicia no andar 0 (Térreo), com a porta fechada e com a lista de chamadas vazia. Existe um laço de espera de comandos. A medida que o programa for recebendo comandos, entrados via leitura de teclado, e que representam as ações dos usuários do elevador, ele deverá constituir uma fila de chamadas de serviço, e irá executar as operações necessárias a fim de atender estas chamadas. Portanto o elevador irá possuir um sistema de controle composto por um laço que possui basicamente 2 rotinas que se alternam: "controle da movimentação" e "leitura de comandos". Procure fazer um programa que permita que ambas rotinas estejam sempre sendo executadas, lê teclado, e depois executa um pequeno passo do controle de movimentação, volta para ler o teclado, e executa mais um pequeno passo, e assim por diante, o que dará a impressão que estão sendo executadas em paralelo como duas "*threads*" independentes.

*** Descrição do Elevador:**

- O prédio possui um elevador que deve atender **16 andares**, do andar Zero (Térreo ao 15)
- O elevador possui 8 estados, podendo estar a cada momento em uma das seguintes situações:
  1) `WAIT0:` Parado com a porta fechada (sem requisições)
  2) `WAIT1:` Parado com a porta fechada (atendendo uma requisição)
  3) `WAIT2:` Parado com a porta aberta (atendendo uma requisição)
  4) `OPEND:` Abrindo a Porta
  5) `CLOSE:` Fechando a Porta
  6) `MOVUP:` Movendo para cima com a porta fechada
  7) `MOVDW:` Movendo para baixo com a porta fechada
  8) `STOPM:` Parada de movimentação do motor (chegou ao destino)

- O elevador possui os seguintes indicadores de estado (Flags):
  Porta: Aberta (0) ou Fechada (1)
  Motor: Parado (0), Down (1 = Movendo para baixo), Up (2 = Movendo para cima)
  Andar Atual: valor de 0 a 15

Durante a operação o elevador passa por passos (etapas) em sua operação, por exemplo, se estiver no andar 0 parado e for chamado para o andar 3, ele terá que passar pelos seguintes estados a fim de executar a operação requerida: IR ATÉ O ANDAR 3

WAIT0 (Parado sem requisições); WAIT1 (Recebeu requisição);
MOVUP (Andar 0); MOVUP (Andar 1); MOVUP (Andar 2);
STOPM (Andar 3); OPEND; WAIT2; CLOSE; WAIT0 (Parado sem requisições).

É importante destacar que o elevador pode receber novas requisições entre cada uma das mudanças de estado, por exemplo, entre um MOVUP e outro MOVUP ele pode receber mais uma requisição de um usuário para que se mova para algum outro andar. Note que o elevador deve passar tipicamente por ciclos de Fechar a porta (CLOSE), Preparar para mover (WAIT1), Mover (MOVUP ou MOVDW), Parar no andar destino (STOPM), Abrir a porta (OPEND), Espera entrada e saída de passageiros (WAIT2) e volta a Fechar a porta (CLOSE).

*** Informações para o Usuário:**

A situação do elevador deverá ser exibida na tela a cada passo da execução do controle do elevador, ou seja, o estado deste e da lista de requisições. Veja abaixo um exemplo de como devem ser exibidas as informações na tela:

```
Andar Atual: 0      Porta: 0      Motor: 0
Requisições: 0 1X 2X 3  4X 5X 6  7X 8  9  10X 11  12X 13  14  15
Estado Atual: WAIT1
```

O Exemplo acima demonstra que o elevador está parado no andar Zero com a porta fechada e está se preparando para mover (WAIT1) a fim de atender requisições que foram feitas nos andares 1, 2, 4, 5, 7, 10 e 12 (assinalados com um X). Os demais andares não possuem requisições. Esta mensagem apresentando a situação do elevador deve ser exibida na tela a cada ciclo de execução do programa (laço principal), mantendo sempre atualizada a informação sobre o estado do elevador.

*** Comandos de Usuário:**

Os usuários do elevador podem fazer 2 tipos de requisições, chamadas externas, quando o usuário está fora do elevador e chama o elevador através do uso de um botão externo, ou, chamadas internas, quando o usuário está dentro do elevador e deseja enviar o elevador para um andar específico. Portanto teremos requisições consideradas a partir de botões externos e de botões internos que representam andares de destino. No caso de nosso simulador, vamos associar cada botão (seja externo ou seja interno) a uma tecla do teclado, e quando o usuário do simulador apertar uma destas teclas, isto significa que foi feita uma requisição.

1. Botões Externos: andares de 0 a 15
   Teclas: 0 a 9 indicam os andares de 0 a 9.
   Teclas: 'q' a 'y' do teclado alfabético em minúsculas indicam os demais andares
   (Q=10, W=11, E=12, R=13, T=14, Y=15)

2. Botões Internos: andares de 0 a 14
   Teclas: 'a' a 'l' e 'z' a 'm' do teclado alfabético em minúsculas indicam os andares, como descrito abaixo
   A=0, S=1, D=2, F=3, G=4, H=5, J=6, K=7, L=8
   Z=9, X=10, C=11, V=12, B=13, N=14, M=15

Portanto, o usuário poderá a qualquer momento (entre 2 passos do elevador) fazer uma nova requisição, interna ou externa, solicitando que o elevador vá para um determinado andar. As requisições serão feitas usando o teclado, digitando uma das teclas correspondentes aos andares, conforme descrito acima.

*** Prioridades de atendimento de chamadas:**

Neste trabalho é deixado livre ao aluno para definir a "inteligência do elevador", ou seja, a forma como ele dá prioridades para o atendimento das chamadas. Não será avaliada a inteligência ou método de priorização de chamadas, podendo ser implementado um "elevador burro", desde é claro que no final todas as requisições sejam atendidas (ninguém deve ficar "preso" no elevador ou esperando indefinidamente que ele venha atender a chamada). Sugere-se ao aluno que faça a união das chamadas internas e externas em um registro único de requisições (ignorando inclusive múltiplas requisições, marcando apenas para cada andar se existe ou não uma chamada, seja interna ou externa).

==========

ENTREGA DO TRABALHO:

* Caso você tenha optado pelo trabalho de programação em ASM do 6502: Envie em um e-mail anexando os arquivos fontes do projeto de seu trabalho (.65s) ao prof. Osório
(incluir os fontes e a documentação apenas, sem arquivos executáveis! O professor fará a montagem)

E-MAIL TO: **fosorio@gmail.com** (Enviar o original para este email)
EMAIL CC: **work2usp@yahoo.com** (Enviar com cópia para este email)
SUBJECT: **[SSC0510] TP REC - Fonte <nome_aluno>** (Assunto do email)
⇨ No corpo do e-mail identifique o seu nome e seu nro. USP.

* O trabalho é individual. Trabalhos identificados como iguais ou muito similares terão atribuída nota Zero.

* Entregar até a data indicada no Site da Disciplina / Wiki ICMC => 14/02/2011
http://www.icmc.usp.br/~fosorio/ (SSC0510 - Informações sobre a REC)

===== THAT'S ALL FOLKS !!! =====