PCS 2302/2024 Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

v.1.0 ago. 2012

Aula 07

Implementação – Paradigma OO III
Conceitos de OO em Java e Arquitetura da MVN

Professores:
Anarosa Alves Franco Brandão (PCS 2302)
Marcos A. Simplício Junior (PCS 2302/2024)
Ricardo Luis de Azevedo da Rocha

Monitores: Felipe Leno, Michel Bieleveld e Diego Queiroz

PCS 2302/2024 Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

v.1.0 ago. 2012

Exercícios - Objetivos

- Praticar a interação entre o “hardware” da MVN (código em Java) e o software por ela executado (escrito em linguagem de montagem da MVN)

- Introduzir o uso das instruções Get Data (GD), Put Data (PD) e Operating System (OS), não utilizadas até o momento

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

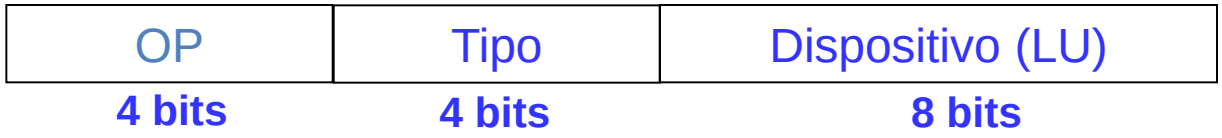
Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

Operações de Entrada-Saída da MVN

Formato da instrução:



OP D (entrada: GD) ou E (saída: PD)
Tipo Tipos de dispositivo:
0 = Teclado
1 = Monitor
2 = Impressora
3 = Disco
Dispositivo Identificação do dispositivo. Pode-se ter vários tipos de dispositivo, ou unidades lógicas (LU). No caso do **disco**, um **arquivo** é considerado uma unidade lógica.

Pode-se ter, portanto, até 16 tipos de dispositivos e, cada um, pode ter até 256 unidades lógicas.

```
Ex:  GD 300  ; Lê 16 bits do disco cuja unidade lógica é 00.
             ; Resultado é colocado no acumulador
PD 100   ; Escreve conteúdo do acumulador no monitor cuja
             ; unidade lógica é 00.
```

Operações de Entrada-Saída da MVN

- Para adicionar/remover dispositivos manualmente na MVN: **comando "s"**
- Para carregar dispositivos automaticamente, pode ser usado o **arquivo "disp.lst"**
  - Sintaxe e exemplos:

```
Tipo    LU   nome_classe                  [nome_arquivo   tipo_acesso]
0       0    mvn.dispositivo.Teclado
1       0    mvn.dispositivo.Monitor
3       0    mvn.dispositivo.Disco     arq.txt          b
3       1    mvn.dispositivo.Disco     arq.txt          e
```

Disco 0: leitura/escrita Disco 1: apenas escrita

  - Obs.: coloque-o na raiz do projeto, se estiver usando o IDE (Eclipse/Netbeans) ou na mesma pasta que o .jar se estiver executando via prompt de comando
  - [tipo de acesso: leitura (r), escrita (w) ou ambos (b)]

Departamento de Engenharia de Computação e Sistemas Digitais
PCS

PCS 2302/2024 Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

v.1.0 ago. 2012

Chamada de Supervisor (SVC) na MVN

- O conjunto de instruções da MVN é bem limitado
  - Mas ela possui a instrução F (**Chamada de Supervisor**, representada pelo **mneumônico OS**), que a princípio não tem uma operação específica associada a ela.
- Para contornar esta limitação da MVN, podemos utilizar a instrução OS para estender a MVN.
  - Para isso, vamos adotar a seguinte estrutura:

OS	Parâmetros	Operação

Exemplo: OS 1FF
(em binário: F1FF)

OS Instrução F (4 bits, ou 1 nibble)

Parâmetros Número de parâmetros (4 bits, ou 1 nibble)

Operação Código de operação [00 a FF] (8 bits, 2 nibbles ou 1 byte)

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

Chamada de Supervisor na MVN

- O valor dos parâmetros a serem passados para a chamada de supervisor devem ser empilhados no endereço de memória anterior ao da chamada.
  - Ex.: suponha que a operação FF tenha como parâmetros um conjunto de unidades lógicas (ULs).
    - Passando a UL 1 como parâmetro:

      UL1 K /0001
      OS /01FF

    - Passando ULs 1 e 2 como parâmetro:

      UL2 K /0002
      UL1 K /0001
      OS /02FF

Observe que o programa precisará “saltar” os parâmetros antes da chamada de supervisor

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

v.1.0 ago. 2012

Exercícios

TYGXXA07E01.asm

- Primitiva da biblioteca elementar da MVN a ser desenvolvida na linguagem de montagem da MVN: **GETLINEF**
  - Função: ler uma linha de um arquivo-texto, visto como um dispositivo do tipo "Disco" pela MVN
  - GETLINEF lê apenas uma linha de texto, ou seja, a leitura termina quando é encontrada a palavra que indica final da linha (EOL) ou final de arquivo (EOF)
    - Pergunta: qual palavra deve ser usada como EOL? E como EOF?
  - Os caracteres contidos nas palavras (2 bytes) da linha são escritos num *buffer* na memória –uma *string*. O EOL e EOF não devem ser escritos no buffer. O final do string deve ser indicado usando uma palavra de finalização (EOS).
    - Pergunta: qual palavra deve ser usada como EOS?

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

Exercícios

TYGXXA07E01.asm

GETLINEF (continuação).

- O buffer tem um tamanho máximo de 10 palavras (20 bytes), de modo que uma linha de tamanho maior do que o especificado é truncada.
- Retorna 1 (*true*) se não chegar ao final do arquivo (EOF); retorna 0 (*false*) se chegar ao final do arquivo.
- Nesta primeira versão de GETLINEF – de escopo limitado – as linhas do arquivo-texto têm número par de caracteres.

Parâmetros: o endereço do *buffer*, a unidade lógica do arquivo-texto e o tamanho do buffer.

Retorno (acumulador): 0 (*false*) se chegar ao final do arquivo e 1 (*true*) no caso contrário.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

Exercícios

TYGXXA07E01.asm

GETLINEF (continuação).

- Use o arquivo TYGXXA07E01_main.asm, que deve ser renomeado de acordo e ligado com o seu programa, para testes.
- Para que a ligação seja possível, use os seguintes nomes para a rotina GETLINEF:
  - Nome da rotina: GETLINEF
  - Parâmetro 1: endereço do *buffer (*GL_END*)*
  - Parâmetro 2: unidade lógica do arquivo (GL_UL)
  - Parâmetro 3: tamanho do buffer (GL_BUF).
- O seu arquivo com a implementação de GETLINEF pode ter outras rotinas desenvolvidas nesta aula ou em aulas anteriores

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

Exercícios

TYGXXA06E02 (Entrega em Aula)

Chamada SVC (em Java).

Altere o código da MVN para que a partir de uma *Chamada de Supervisor* (instrução OS) seja executada a função **reset**. Essa função posiciona o cursor de leitura da unidade lógica de disco em seu início.

Parâmetro: lu – o número da unidade lógica de I/O

Retorno: 0 em caso de sucesso, -1 em caso de erro (número da unidade lógica inválida, inexistente ou somente de escrita)

Instrução de Chamada: F1FF

OBS: O valor do parâmetro a ser passado para a chamada de supervisor deve ser posicionado no endereço de memória anterior ao da chamada.

Restrições: Funciona somente se a unidade lógica estiver aberta para leitura e for uma unidade de disco.

IMPORTANTE: Não existe uma operação explícita de **reset** de arquivo no Java, portanto é necessário pensar em alguma forma para executá-la.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão

Exercícios

TYGXXA06E03 (Entrega em Aula)

Chamada SVC (em Java).

Altere o código da MVN para que a partir de uma *Chamada de Supervisor* (SVC) seja executada a função **position**. Essa função obtém o número bytes que já foram lidos da unidade lógica.

Parâmetro: lu – o número da unidade lógica de I/O

Retorno: número de bytes já lidos, -1 em caso de erro (número da unidade lógica inválida, inexistente ou somente de escrita)

Instrução de Chamada: F1FD

OBS: O valor do parâmetro a ser passado para a chamada de supervisor deve ser posicionado no endereço de memória anterior ao da chamada.

Restrições: Funciona somente se a unidade lógica estiver aberta para leitura e for uma unidade de disco.

PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

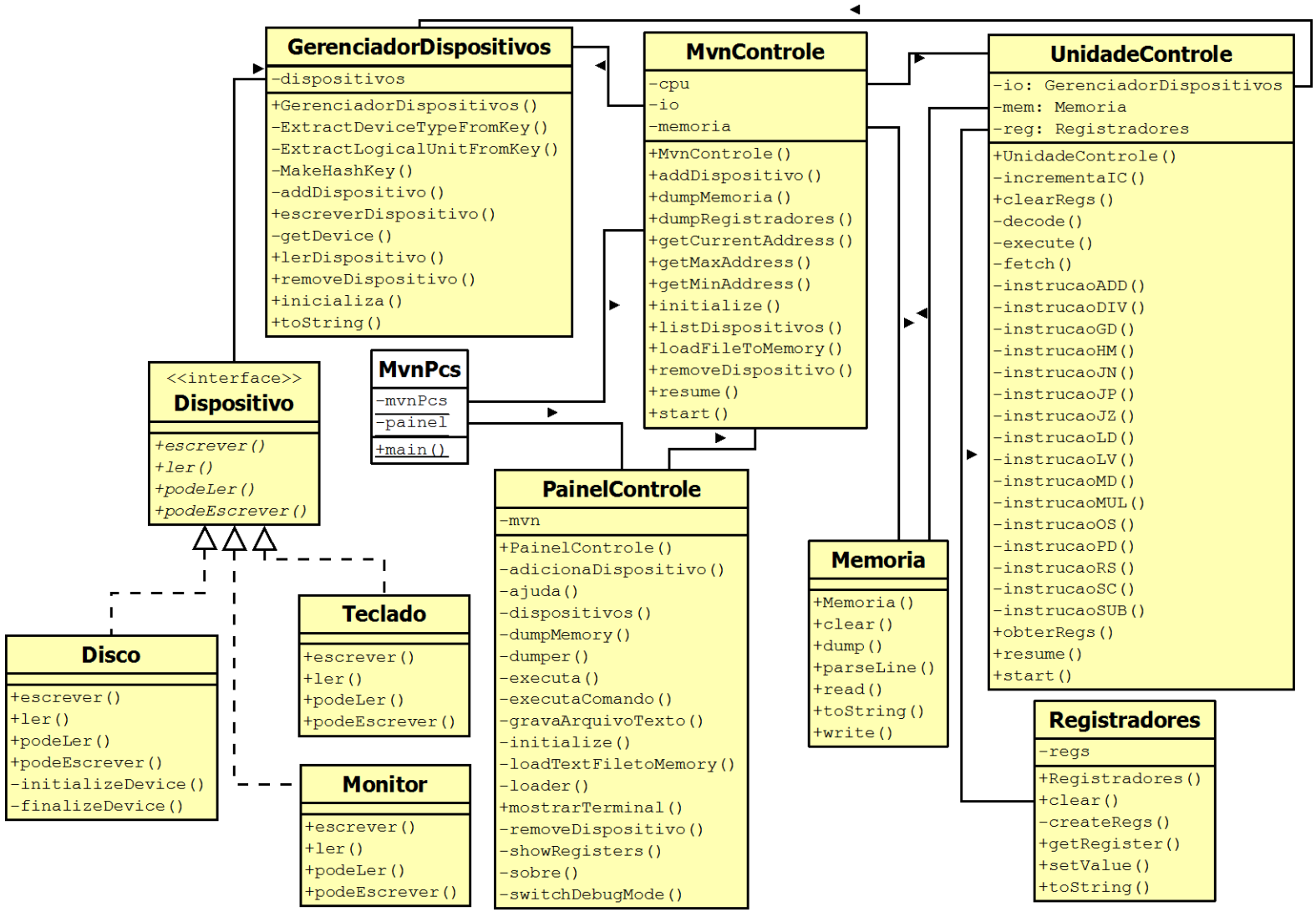
Guilherme M. Gomes
Anarosa A. F. Brandão

v.1.0 ago. 2012

Informações auxiliares

- Utilize os diagramas e códigos apresentados nas aulas anteriores para lhe ajudar a entender o funcionamento da MVN;
- Consulte a documentação das classes da MVN (comentários nos métodos, atributos e classes) para entender o funcionamento da MVN;
- Consulte a documentação do Java para lhe auxiliar no desenvolvimento;
- O código da MVN disponibilizado funciona para carregar arquivos, executar programas e interagir com a memória. O mesmo deve continuar funcionando após a modificação.

Macro Arquitetura da MVN



PCS 2302/2024
Laboratório de Fundamentos da Eng.de Computação

Professores:
Anarosa A. F. Brandão
Marcos A. Simplício Jr
Ricardo Luis A. Rocha

Aula 06:

Implementação Conceitos de OO em Java e Arquitetura da MVN

Autores:

Guilherme M. Gomes
Anarosa A. F. Brandão