

Introdução à Ciência da Computação-2

- Fabio Nakano (fabionakano@usp.br)
 - Bloco A-1, segundo andar, sala 77.
 - Horário de atendimento: seg. 14-17h.
- Monitores... aguarde...

Operação

- Teremos listas de exercícios, sem valor para a nota.
- Frequência mínima: 70%, faltas não serão abonadas.
- Avaliação:
 - Duas provas (P1 e P2)
 - Três trabalhos (T1, T2 e T3): exercícios-programas (EPs)
- Prova substitutiva:
 - somente para quem perdeu uma das provas (FECHADA)
 - substitui a que você perdeu
 - envolve todo o conteúdo ministrado na disciplina.
 - DIFÍCIL: USEM SÓ EM CASO DE EMERGÊNCIA!
- Não há EP substitutivo.
- Entrega de Eps pelo COL. Caso o COL esteja com problemas, pode enviar por e-mail. Considerarei o horário de chegada da msg como horário da entrega.
- Prova de recuperação

Datas

- P1: 23/09/2011
- P2: 25/11/2011
- Substitutiva: 02/12/2011
- Recuperação: 15/12/2011
-
- EP1: 17/09/2011
- EP2: 22/10/2011
- EP3: 19/11/2011

Critérios para aprovação

- Média de Provas (MP):
- $MP = (P1 + P2) / 2$
- Média de Trabalhos (MT):
- $MT = (T1 + T2 + T3) / 3$
- Média Final (MF) :
- se $MP \geq 5,0$ e $MT \geq 5,0$ então $MF = (MP + MT) / 2$
- senão $MF = \text{mínimo}(MP, MT)$
- Se $MF \geq 5,0 \rightarrow$ aluno aprovado
- Se $3,0 \leq MF < 5,0 \rightarrow$ recuperação
- Se $MF < 3,0 \rightarrow$ aluno reprovado.

Recuperação

- MFR (Média Final após recuperação):
 - $MFR = \text{máximo}(MF, (MF + REC)/2)$
- REC : envolve todo o conteúdo ministrado na disciplina.
-
- Se $MFR \geq 5,0 \rightarrow$ aluno aprovado
- Se $MFR < 5,0 \rightarrow$ aluno reprovado
-
- 4,9 não é 5,0

ICC2 é bem diferente de ICC1

- O conteúdo é mais teórico, requer mais conhecimento em matemática.
- Os Eps continuam testando habilidade em programar
-
- Dicas
 - Repasse a matéria no mesmo dia.
 - Faça as revisões de provas e eps.
 - Use os horários de atendimento.

Revisão

- Memória e variáveis em memória
- Estruturas básicas
 - Arrays
 - Listas ligadas
- Programação básica
 - Chamadas de função (método) e passagem de parâmetros.
- Programação Orientada a Objetos
- Sequências, séries;
- Arranjos, combinações e permutações
- Logaritmos

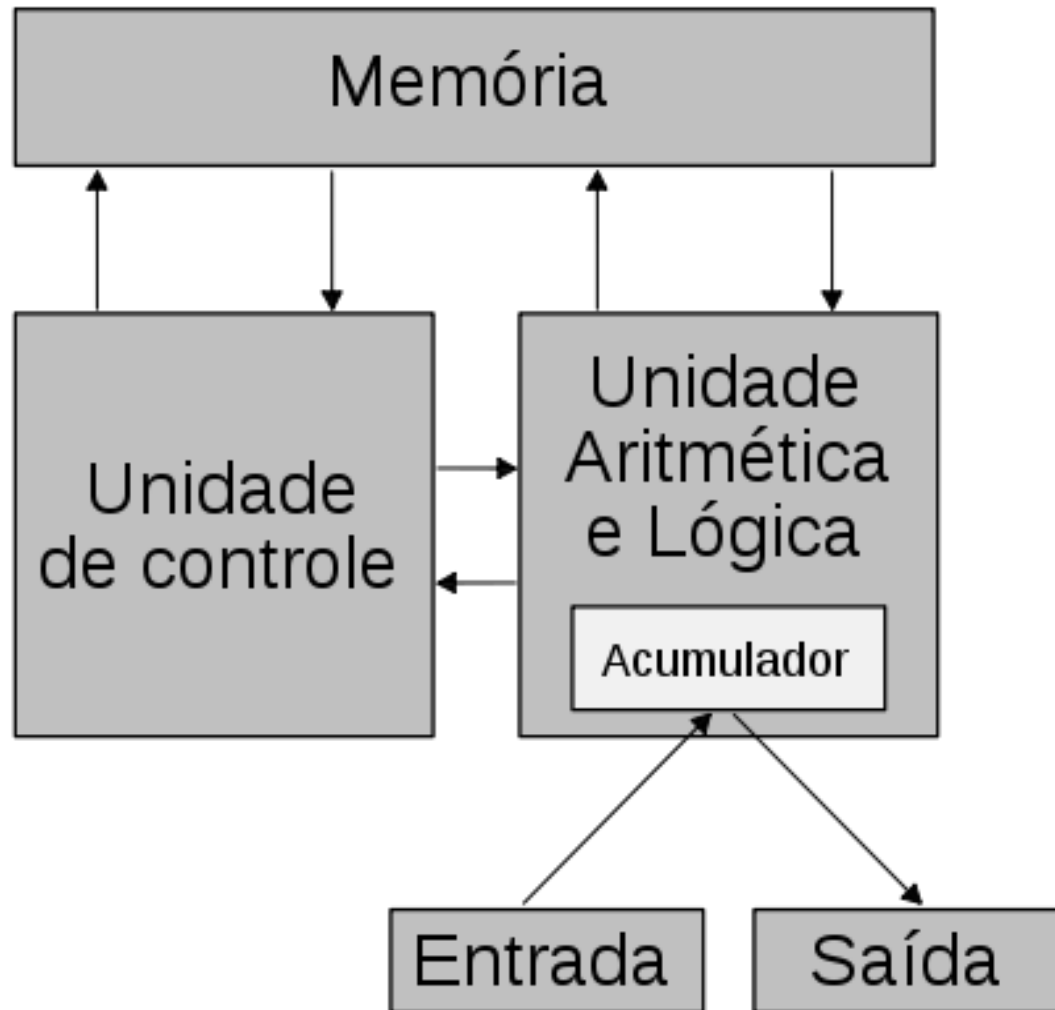
Novidades

- Recursão
- Noções de técnicas de demonstração matemática
- Algoritmos
 - Busca
 - Ordenação
- Noções de análise de algoritmos
 - Complexidade assintótica
- Estruturas de dados
 - Árvores
 - Tabelas de Hash

Bibliografia

- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos - Trad. 2.º Ed. Americana. Ed. Campus, 2002.
- KON, F.; GOLDMAN, A.; SILVA, P.J.S. “Introdução à Ciência de Computação com Java e Orientado a Objetos”, IME - USP, 2005. Disponível em: <http://ccsl.ime.usp.br/introCCJavaOO>
- GOODRICH, M.; TAMASSIA, R. Estruturas de Dados e Algoritmos em Java. Ed. Bookman, 2a. Ed. 2002.
- ZIVIANI, N. – Projeto de Algoritmos, 2ª Ed., Editora Thomson, 2004.
- TENENBAUM, A.M., et all. Data Structures Using C, Prentice-Hall, 1990.
- WIRTH, N. Algorithms + Data Structures = Programs, Prentice-Hall, 1986
- **VAMOS COMEÇAR!!!**

Arquitetura de Von Neumann



Máquina de programa
Armazenado - tanto programas
Quanto variáveis são
Armazenados na memória.

Extraído de Wikipedia.org
para fins didáticos.
Figura original sob licença
GNU Public Documentation

Todo processador funciona de forma parecida

- Programas são sequências de instruções codificadas como números, armazenadas na memória e executadas.
- Variáveis, essencialmente numéricas, são armazenadas na memória.
- Em geral as instruções dos programas são tais que variáveis não são executadas, mas poderiam...

Um programa na memória tem esta “aparência”

Endereço	Conteúdo
n	+1130
n+1	+1240
n+2	+3150
n+3	+4150
n+4	+1150
n+5	+5411
n+6	+1140
n+7	+2150
n+8	+1240
n+9	+5103
n+10	+4140
n+11	+7000

+II EE

Cada instrução é composta por
II: código de operação
EE: endereço de um dos operandos
($00 \leq EE \leq 99$).

Codificação de operações

Código da operação	Mnemônico	Explicação
11	CEA	Copie o conteúdo do endereço EE no acumulador. (AC recebe [EE]).
12	CAE	Copie o conteúdo do acumulador no endereço EE. (EE recebe [AC])
21	SOM	Some o conteúdo do endereço EE com o conteúdo do acumulador e guarde o resultado no acumulador. (AC recebe $[AC] + [EE]$)
22	SUB	Subtraia o conteúdo do endereço EE do conteúdo do acumulador e guarde o resultado no acumulador. (AC recebe $[AC] - [EE]$)
23	MUL	Multiplique o conteúdo do endereço EE com o conteúdo do acumulador e guarde o resultado no acumulador. (AC recebe $[AC] * [EE]$)
24	DIV	Divide o conteúdo do acumulador pelo conteúdo do endereço EE e guarde o resultado no acumulador. (AC recebe $[AC] / [EE]$)
25	MOD	[AC] recebe o resto da divisão $[AC] / [EE]$.
31	LER	Leia um número e guarde-o no endereço EE. (EE recebe o valor lido)
41	IMP	Imprima o conteúdo do endereço EE.
50	NOP	Nenhuma operação é efetuada.
51	DES	Desvie a execução para o endereço EE, i.e. AI recebe EE.
52	DPO	Se o conteúdo do acumulador for maior do que zero, desvie a execução para o endereço EE. (Se $[AC] > 0$, AI recebe EE).
53	DPZ	Se o conteúdo do acumulador for maior ou igual a zero, desvie a execução para o endereço EE. (Se $[AC] \geq 0$, AI recebe EE).
54	DNE	Se o conteúdo do acumulador for menor do que zero, desvie a execução para o endereço EE. (Se $[AC] < 0$, AI recebe EE.)
55	DNZ	Se o conteúdo do acumulador for menor ou igual a zero, desvie a execução para o endereço EE. (Se $[AC] \leq 0$, AI recebe EE).

Onde estão as variáveis?!!

Endereço	Conteúdo	Linguagem de montagem	Explicação por extenso
1	+1130	CEA zero	Copie o conteúdo do endereço 30 no acumulador
2	+1240	CAE soma	Copie o conteúdo do acumulador no endereço 40
3	+3150	leia: LER num	Leia um número e coloque no endereço 50
4	+4150	IMP num	Imprima o conteúdo do endereço 50
5	+1150	CEA num	Copie o conteúdo do endereço 50 no acumulador
6	+5411	DNE fim	Se o conteúdo do acumulador for menor que zero, desvie para o endereço 11
7	+1140	CEA soma	Copie o conteúdo do endereço 40 no acumulador
8	+2150	SOM num	Some o conteúdo do endereço 50 com o conteúdo do acumulador e guarde no acumulador
9	+1240	CAE soma	Copie o conteúdo do acumulador no endereço 40
10	+5103	DES leia	Desvie para o endereço 03
11	+4140	fim: IMP soma	Imprima o conteúdo do endereço 40
12	+7000	PAR	Pare
30	0	zero:	Variável com valor zero
40		soma:	Variável de nome "soma"
50		num:	Variável de nome "num"

Diagramas

- Como representar a memória?
- Como representar variáveis?
-
- ... partindo para objetos mais complexos...

Arrays

```
int[] array={...};
```

-
- `int[] array;`
- `array=new int[20];`
-
- Multidimensionais
- `int[][][] array;`
-
- Referencia útil: <http://www.leepoint.net/notes-java/index.html>
-

Representação em memória, diagramas, referências

- Como representar arrays em memória?
- Como representar variáveis?
-
- Em geral números são tipos primitivos e arrays são tipos abstratos.

Funções/métodos e seus parâmetros

-
- Como os parâmetros são passados para os métodos?

Programas