

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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007


PCS-2302 / PCS-2024
Lab. de Fundamentos de Eng. de Computação

Aula 01

Máquina de Von Neumann

Professores:
Jaime Simão Sichman (PCS 2302)
Ricardo Luis de Azevedo da Rocha (PCS 2024)

1



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007


Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

Roteiro

- Máquina de Von Neumann
 - Problemas práticos da Máquina de Turing
 - Exemplo de uma máquina muito simples no estilo Von Neumann
 - Exemplo de um simulador de uma máquina de Von Neumann (MVN)
- Parte Experimental
 - Pequenos programas em código de máquina MVN

2



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007


Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

Problemas Práticos da Máquina de Turing

- A **Máquina de Turing** se apresenta através de um formalismo poderoso, com fita infinita e apenas quatro operações triviais: ler, gravar, avançar e recuar.
- Isso faz dela um dispositivo detalhista que oferece apenas uma **visão microscópica** da solução do problema que pretende resolver, não permitindo ao usuário usar abstrações.
- Embora a Máquina de Turing Universal permita uma espécie de programação, o seu código é extenso e a sua velocidade final de execução, muito baixa.

3



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007


Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

A idéia da Máquina de Von Neumann (1)

- O **Modelo de Von Neumann** procura oferecer uma alternativa prática, disponibilizando ações mais poderosas e ágeis em seu repertório de operações.
- Isso viabiliza, para os mesmos programas, codificações muito mais expressivas, compactas e eficientes.

4



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007


Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

A idéia da Máquina de Von Neumann (2)

- Para isso, a Máquina de Von Neumann utiliza:
 - Memória endereçável**, usando acesso aleatório
 - Programa armazenados** na memória, para definir diretamente a função corrente da máquina, em lugar do emprego de uma especificação fixa da lógica ou do uso de simulação através de uma máquina universal
 - Dados** representados em posições da memória modificável de acesso aleatório, em lugar da fita seqüencial e dados
 - Codificação numérica **binária** em lugar da unária
 - Instruções variadas e expressivas** em lugar de sub-máquinas específicas para a realização de operações básicas muito frequentes
 - Maior flexibilidade** para o usuário, permitindo operações de entrada e saída, comunicação física com o mundo real e controle dos modos de operação da máquina

5



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007


Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

Elementos da Arquitetura a Simular (1)

- No presente estudo experimental do comportamento dos computadores e dos programas que eles executam, pretende-se simular um *processador muito simples*, porém estruturalmente similar aos disponíveis na prática.
- O processador tem um conjunto de elementos físicos de armazenamento de informações, e o conjunto de dados neles contidos em cada instante constitui o **estado instantâneo** do processamento:
 - Memória Principal**: para armazenar programas e dados
 - Acumulador**: funciona como área de trabalho, para a execução de operações aritméticas e lógicas
 - Outros **registradores auxiliares**: empregados em diversas operações intermediárias no processamento dos programas

6



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann


Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

Conjunto de registradores da Máquina de Von Neumann (MVN)

MAR	Registrador de endereço de memória
MDR	Registrador de dados da memória
IC	Registrador de endereço da próxima instrução
IR	Registrador de instrução
OP	Registrador de código de operação
OI	Registrador de operando de instrução
RA	Registrador de endereço de retorno
AC	Acumulador

IR (16 bits)	
OP (4 bits)	OI (12 bits)

13



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann


Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

[EXECUTE] – Obtenção e Decodificação

EXECUTE - Serve para promover a execução do programa, conforme o modo de operação: execução contínua/uma instrução por vez

- Determinação da Próxima Instrução a Executar
- Fase de Obtenção da Instrução
 - Obter na memória, no endereço contido no registrador de Endereço da Próxima Instrução, o código da instrução desejada
- Fase de Decodificação da Instrução
 - Decompor a instrução em duas partes: o código da instrução e o seu operando, depositando essas partes nos registradores de instrução e de operando, respectivamente.
 - Selecionar, com base no conteúdo do registrador de instrução, um procedimento de execução dentre os disponíveis no repertório do simulador (passo 4).

14



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

[EXECUTE] – Execução de instrução (1)

4) Fase de Execução da Instrução

- Executar o procedimento selecionado em 3, usando como operando o conteúdo do registrador de operando, preenchido anteriormente.
- Caso a instrução executada não seja de desvio, incrementar o registrador de endereço da próxima instrução a executar. Caso contrário, o procedimento de execução já terá atualizado convenientemente tal informação.


4.1) Execução da instrução (decodificada em 3)

- De acordo com o código da instrução a executar (contido no registrador de instrução), executar os procedimentos de simulação correspondentes (detalhados adiante)

4.2) Acerto do registrador de Endereço da Próxima Instrução para apontar a próxima instrução a ser simulada:

- Incrementar o registrador de Endereço da Próxima Instrução.

15



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

[EXECUTE] – Execução de instrução (2)


Registrador de instrução = 0 (desvio incondicional)

- modificar o conteúdo do registrador de Endereço da Próxima Instrução armazenando nele o conteúdo do registrador de operando, e prosseguir no passo 1

Registrador de instrução = 1 (desvio se acumulador é zero)

- se o conteúdo do acumulador for zero, então modificar o conteúdo do registrador de Endereço da Próxima Instrução, armazenando nele o conteúdo do registrador de operando, e prosseguir no passo 1.
- Caso contrário, prosseguir no passo 4.2.

16



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

[EXECUTE] – Execução de instrução (3)


Registrador de instrução = 2 (desvio se negativo)

- se o conteúdo do acumulador for negativo, então modificar o conteúdo do registrador de Endereço da Próxima Instrução armazenando nele o conteúdo do registrador de operando, e prosseguir no passo 1.
- Caso contrário, prosseguir no passo 4.2.

Registrador de instrução = 3 (constante para acumulador)

- Armazenar no acumulador o número relativo de 12 bits contido no registrador de operando, estendendo seu bit mais significativo (bit de sinal) para completar os 16 bits do acumulador
- Prosseguir no passo 4.2.

17



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha
v. 1.2 ago 2007

[EXECUTE] – Execução de instrução (4)


Registrador de instrução = 4 (soma)

- Somar ao conteúdo do acumulador o conteúdo da posição de memória indicada pelo registrador de operando
- Guardar o resultado no acumulador.
- Prosseguir no passo 4.2.

Registrador de instrução = 5 (subtração)

- Subtrair do conteúdo do acumulador o conteúdo da posição de memória indicada pelo registrador de operando
- Guardar o resultado no acumulador.
- Prosseguir no passo 4.2.

18



[EXECUTE] – Execução de instrução (5)

Registador de instrução = 6 (multiplicação)

- Multiplicar o conteúdo do acumulador pelo conteúdo da posição de memória indicada pelo registrador de operando
- Guardar o resultado no acumulador.
- Prosseguir no passo 4.2.

Registador de instrução = 7 (divisão inteira)

- Dividir o conteúdo do acumulador pelo conteúdo da posição de memória indicada pelo registrador de operando
- Guardar a parte inteira do resultado no acumulador.
- Prosseguir no passo 4.2.

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


Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

v. 1.2 ago 2007

19



[EXECUTE] – Execução de instrução (6)

Registador de instrução = 8 (memória para acumulador)

- Armazenar no acumulador o conteúdo da posição de memória cujo endereço é o conteúdo do registrador de operando.
- Prosseguir no passo 4.2.

Registador de instrução = 9 (acumulador para memória)

- Guardar o conteúdo do acumulador na posição de memória indicada pelo registrador de operando.
- Prosseguir no passo 4.2.

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


Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

v. 1.2 ago 2007

20



[EXECUTE] – Execução de instrução (7)

Registador de instrução = A (desvio para subprograma)

- Armazenar o conteúdo do registrador de Endereço da Próxima Instrução, incrementado de uma unidade, no endereço do subprograma
- Armazenar no registrador de Endereço da Próxima Instrução o conteúdo do registrador de operando incrementado de uma unidade
- Prosseguir no passo 1.

Registador de instrução = B (retorno de subprograma)

- Armazenar no registrador de Endereço da Próxima Instrução o conteúdo do registrador de endereço de retorno, e no registrador acumulador o conteúdo da posição de memória apontada pelo registrador de operando.
- Prosseguir no passo 1.

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


Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

v. 1.2 ago 2007

21



[EXECUTE] – Execução de instrução (8)

Registador de instrução = C (stop)

- Modificar o conteúdo do registrador de Endereço da Próxima Instrução armazenando nele o conteúdo do registrador de operando.
- Prosseguir no passo 4.2.

Registador de instrução = D (input)

- Acionar o dispositivo padrão de entrada e aguardar que o usuário forneça o próximo dado a ser lido.
- Transferir o dado para o acumulador.
- Prosseguir no passo 4.2.

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


Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

v. 1.2 ago 2007

22



[EXECUTE] – Execução de instrução (9)

Registador de instrução = E (output)

- Transferir o conteúdo do acumulador para o dispositivo padrão de saída.
- Acionar o o dispositivo padrão de saída e aguardar que este termine de executar a operação de saída.
- Prosseguir no passo 4.2.

Registador de instrução = F (supervisor call)
(não implementado: por enquanto esta instrução não faz nada)

- Prosseguir no passo 4.2.

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


Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

v. 1.2 ago 2007

23



**Diagrama de fluxo do Interpretador
[detalhamento de EXECUTA]**

Executa uma instrução

```

graph TD
    Start([Determinar a próxima instrução a executar]) --> GetMem[Obter a instrução em MEM[IC] e guardar em IR]
    GetMem --> Decode[Decodificar a instrução: OP:=Código de operação<br/>OI:=Operando]
    Decode --> Execute[Executar a instrução]
    Execute --> Next[Determinar a próxima instrução a executar]
  
```

OP (hexa)	Ação a executar
0	IC:=OI
1	Se AC=0 então IC:=OI se não IC:=IC+1
2	Se AC<0 então IC:=OI se não IC:=IC+1
3	AC:=OI ; IC:=IC+1
4	AC:=AC+MEM[OI] ; IC:=IC+1
5	AC:=AC-MEM[OI] ; IC:=IC+1
6	AC:=AC*MEM[OI] ; IC:=IC+1
7	AC:=int(AC/MEM[OI]) ; IC:=IC+1
8	AC:=MEM[OI] ; IC:=IC+1
9	MEM[OI]:=AC ; IC:=IC+1
A	RA:=IC ; IC:=OI
B	AC:=MEM[OI] ; IC:=RA
C	IC:=OI
D	aguarda; AC:= dado de entrada; IC:=IC+1
E	dado de saída := AC ; aguarda ; IC:=IC+1
F	(nada faz por ora) ; IC:=IC+1

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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L. A. Rocha

v. 1.2 ago 2007

24

Conjunto de instruções da Máquina de Von Neumann (MVN)		
Código (hexa)	Instrução	Operando
0	Desvio incondicional	endereço do desvio
1	Desvio se acumulador é zero	endereço do desvio
2	Desvio se acumulador é negativo	endereço do desvio
3	Deposita uma constante no acumulador	constante relativa de 12 bits
4	Soma	endereço da parcela
5	Subtração	endereço do subtraendo
6	Multiplicação	endereço do multiplicador
7	Divisão	endereço do divisor
8	Memória para acumulador	endereço-origem do dado
9	Acumulador para memória	endereço-destino do dado
A	Desvio para subprograma (função)	endereço do subprograma
B	Retorno de subprograma (função)	endereço do desvio
C	Parada	endereço do desvio
D	Entrada	dispositivo de e/s (*)
E	Saída	dispositivo de e/s (*)
F	Chamada de supervisor	constante (**)

(*) ver slide seguinte
(**) por ora, este operando (tipo da chamada) é irrelevante, e esta instrução nada faz.

Conjunto de registradores da Máquina de Von Neumann (MVN)

PCS 23002004
Laboratório de
Fundamentos de
Eng de Computação

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:

Máquina de Von
Neumann

Autores:

Jaime S. Sichman
João José Neto
Paulo S. Maria Silva
Ricardo L.A. Rocha

v. 1.2 - ago 2007

Operações de Entrada e Saída

OP	Tipo	Dispositivo
----	------	-------------

OP
Tipo

D (entrada) ou E (saída)
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.


26

Exemplo de Programa – Prog1	
<p>• Problema: Somar o valor de duas variáveis iniciadas com os valores -125₁₀ e 100₁₀, colocando o resultado em outra variável.</p> <pre> ; prog1.mvn ; Soma os valores de duas posições de memória e guarda o ; resultado em outra posição de memória, parando na ; instrução final. ; 00 0008 ; Ponto de entrada: JMP para as instruções ; Variáveis locais 02 FF83 ; A = 0xFF83 (-125) 04 0064 ; B = 0x0064 (100) 06 0000 ; RESULTADO deverá ser 0xFFE7 (-25) ; Instruções do programa 08 8002 ; Carrega o conteúdo de A no acumulador 0A 4004 ; Adiciona B ao conteúdo do acumulador 0C 9006 ; Armazena o resultado em RESULTADO 0E C00E ; Pára em 0x000E </pre>	

Execução de Programa – Prog1	
<pre> P:\Pswa\POLY\PCS2302\2006\VMN\2006-PSM\java\bin\RunPcs Estado INICIAL Estado RUNSTEP ===== PCS2302/PCS23024 Simulador da Máquina de Von Neumann Versão 3.0 (c)2005 Todos os direitos reservados Inicialize(l) Run(r) Load(l) Show(w) Programa RCLIP(p) Finalize(f): p Estado ENTRADA DE PROGRAMA Nome do arquivo de programa? prog1.mvn Estado RUNSTEP ===== PCS2302/PCS23024 Simulador da Máquina de Von Neumann Versão 3.0 (c)2005 Todos os direitos reservados Inicialize(l) Run(r) Load(l) Show(w) Programa RCLIP(p) Finalize(f): r Estado RUN Mostra o valor dos registradores a cada passo do ciclo RMC (r/h)? s Endereço atual do IC = 8000. Entrar (novos) endereço do IC: Entrar unidade lógica para o disco (ou ENTER para cancelar): IC IR AC MMR MDR OP RA RI OI 0000 Estado RUNSTEP ===== PCS2302/PCS23024 Simulador da Máquina de Von Neumann Versão 3.0 (c)2005 Todos os direitos reservados Inicialize(l) Run(r) Load(l) Show(w) Programa RCLIP(p) Finalize(f): </pre>	

Exemplo de Programa – Prog2 (1)	
<p>• Problema: Implementar uma sub-rotina que subtrai dois inteiros. Os valores dos argumentos estão armazenados em duas variáveis do programa principal. O resultado é armazenado em uma variável do programa principal.</p> <pre> ; prog2.mvn ; Programa de ilustração para JSR ; int subtrair(int x, int y) { ; return x - y; ; } ; 00 0010 ; JMP início das instruções ; Variáveis locais 02 0010 ; A = 0x0010 04 0064 ; B = 0x0064 06 0000 ; RESULTADO Resultado de subtrair() </pre>	

Exemplo de Programa – Prog2 (2)	
<pre> ; Programa principal ; Chamando SUBTRAIR(A, B) ; 10 8002 ; Carrega o conteúdo de A no acumulador 12 903C ; Armazena no parâmetro X 14 8004 ; Carrega o conteúdo de B 16 903E ; Armazena no parâmetro Y 18 A040 ; Chama a sub-rotina SUBTRAIR 1A 9006 ; Armazena o resultado em RESULTADO 1C C01C ; Pára em 0x001C ; Sub-rotina SUBTRAIR ; Parâmetros formais 3C 0000 ; X 3E 0000 ; Y 40 0000 ; Endereço de retorno 42 803C ; Carrega o conteúdo de X 44 503E ; Subtrai Y 46 B040 ; Retorna para o endereço posto em 0x0040 </pre>	



PCS 2302/2024
Laboratório de Fundamentos da Eng. de Computação
Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007
Aula 1:
Máquina de Von Neumann
Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007


Execução de Programa – Prog2

```

PCS2302/PCS2024: Simulador da Máquina de Von Neumann
Versão 3.0 (c)2005 Todos os direitos reservados
Initialize(i) Run(r) Load(l) Show(u) Programa ASCII(p) Finalizar(f): r
Estado RUN
Mostra valor dos registradores a cada passo do ciclo FDE (s/n)? e
Endereço atual do IC = 0000. Entrar (novo) endereço do IC:
Entrar unidade lógica para o disco (ou ENTER para cancelar):
IC IR AC MAR MDR OP RA R2 [R1]
0000 0000 0000 0000 0000 0000 0000 0000
0010 0000 0000 0000 0000 0000 0000 0000
0012 0002 0010 0010 0002 0008 0000 0002 0010
0014 000c 0010 0012 003c 0009 0000 003c 0010
0016 0004 0004 0014 0004 0008 0000 0004 0004
0018 003e 0004 0016 003e 0009 0000 003e 0004
0042 0040 0004 0018 0040 000a 0000 0040 001a
0044 003c 0010 0042 003c 0008 0000 003c 0010
0046 003e 0044 0044 003e 0005 0000 003e 0004
001a 0040 ffac 0046 0040 000b 0000 0040 001a
001c 0006 ffac 001a 0006 0009 0000 0006 ffac
001e c01c ffac 001c c01c 000c 0000 001c c01c
Estado RUNSTEP
=====
PCS2302/PCS2024: Simulador da Máquina de Von Neumann
Versão 3.0 (c)2005 Todos os direitos reservados
Initialize(i) Run(r) Load(l) Show(u) Programa ASCII(p) Finalizar(f):

```

31




PCS 2302/2024
Laboratório de Fundamentos da Eng. de Computação
Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007
Aula 1:
Máquina de Von Neumann
Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007

Algumas práticas de programação (1)

- O conjunto de instruções desta máquina de Von Neumann é extremamente limitado, exigindo alguns artifícios para a obtenção dos efeitos necessários:
 - Não há operações aritméticas. Tudo deve ser feito com operações lógicas.
 - Não há endereçamento indireto nem indexado. Tudo deve ser feito alterando-se convenientemente as instruções disponíveis, no próprio programa, antes de executá-las.
 - Incrementos e decrementos de variáveis devem ser feitos somando-se ou subtraindo-se as constantes desejadas (tipicamente 1 ou 2) às variáveis-alvo.

32




PCS 2302/2024
Laboratório de Fundamentos da Eng. de Computação
Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007
Aula 1:
Máquina de Von Neumann
Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007

Algumas práticas de programação (2)

- Não há instruções específicas para todos os testes. Tudo deve ser feito combinando-se as instruções de desvios condicionais e usando-se lógica invertida quando necessário.
- Convém separar sub-rotinas já testadas e muito usadas, bem como variáveis e constantes, dos programas em desenvolvimento.

33




PCS 2302/2024
Laboratório de Fundamentos da Eng. de Computação
Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007
Aula 1:
Máquina de Von Neumann
Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007

Algumas práticas de programação (3)

- À medida que os programas ficam maiores e/ou tem-se mais de um programa na memória, é importante planejar um **mapa de memória**. Uma sugestão para um mapa bem simples é reservar os endereços 0x0000 – 0x1FFF para área de dados, constantes, tabelas, etc., e os endereços a partir de 0x0200 para programas principais e sub-rotinas.
- Projete sempre no papel seus programas em linguagem de máquina, e simule seu funcionamento no papel antes de utilizar o computador. Economiza-se muito tempo e esforço evitando-se a depuração de erros na base da tentativa e de testes.

34




PCS 2302/2024
Laboratório de Fundamentos da Eng. de Computação
Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007
Aula 1:
Máquina de Von Neumann
Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007

Algumas práticas de programação (4)

- Documente todos os programas desenvolvidos com comentários informativos no código, e no papel, com diagramas de fluxo e com desenhos ilustrativos das estruturas de dados utilizadas e das operações efetuadas. Em programação binária, é muito raro que, passados alguns dias, mesmo o autor consiga lembrar-se exatamente de como funciona o programa que ele próprio criou.
- Projete bem e anote os testes realizados e os resultados esperados. É freqüente ter de repeti-los para as novas versões de um programa em desenvolvimento.

35



PCS 2302/2024
Laboratório de Fundamentos da Eng. de Computação
Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007
Aula 1:
Máquina de Von Neumann
Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007

Exercícios

Comente devidamente os programas desenvolvidos.

- Execute os programas 2302_aula01_prog1.mvn e 2302_aula02_prog2.mvn. Faça testes para valores diferentes de entradas. Guarde o mapa de memória antes e depois da execução.
- Desenvolva uma sub-rotina (e um programa principal para testá-la) 2302_aula01_prog3.mvn para calcular uma potência inteira $k \geq 0$ de um número inteiro dado n , ou seja, n^k . Ensaie para alguns valores de n e de k (variáveis do programa principal), e verifique os resultados.
- Construa um programa 2302_aula01_prog4.mvn que determina se um triângulo com lados a , b , c inteiros positivos é acutângulo, retângulo ou obtusângulo (usar o teorema de Pitágoras).

36



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von
Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007

37

Bibliografia (Programação de Sistemas)

Relíquias Preciosas

Barron, D. W. *Assemblers and Loaders* (3rd. ed.) MacDonald/Elsevier, 1978

Beck, L. L. *System Software - An Introduction to Systems Programming* Addison-Wesley, 1985

Callingaert, P. *Assemblers, Compilers and Program Translation* Computer Science Press, 1979

Donovan, J. J. *Systems Programming* McGraw-Hill, 1972

Duncan, F.G. *Microprocessor Programming and Software Development* Prentice Hall, 1979.

Freeman, P. *Software System Principles* SRA, 1975

Gear, C. W. *Computer Organization and Programming* (3rd. ed.) McGraw-Hill, 1981

Graham, R. M. *Principles of Systems Programming* John Wiley & Sons, 1975

Gust, P. *Introduction to Machine and Assembly Language Programming* Prentice Hall, 1986

Maginnis, J. B. *Elements of Compiler Construction* Appleton-Century-Crofts, Meredith Co., 1972

Presser, L. and White, J. R. *Linkers and Loaders* ACM Comp. Surveys, vol. 4, n. 3, pp. 149-168


Rosen, S. (ed.) *Programming Systems and Languages* McGraw-Hill, 1967

Tseng, V. (ed.) *Microprocessor Development and Development Systems* McGraw-Hill, 1982

Ullman, J. D. *Fundamental Concepts of Programming Systems* Addison-Wesley, 1976

Wegner, P. *Progr. Languages, Inf. Structures and Machine Organization* McGraw-Hill, 1968.

Wohlschlag, J. and McKean, M. *Structured System Programming* Prentice Hall, 1980



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

Professores:
Jaime S. Sichman
Ricardo L.A. Rocha
© 2007

Aula 1:
Máquina de Von
Neumann

Autores:
Jaime S. Sichman
João José Neto
Paulo S. Muniz Silva
Ricardo L.A. Rocha
v. 1.2 ago 2007

38

Bibliografia (interpretação de instruções)

- Leitura complementar

UM SIMULADOR-INTERPRETADOR PARA A LINGUAGEM DE MÁQUINA DO PATINHO FEIO.
(João José Neto, Aspectos do Projeto de Software de um Minicomputador, Dissertação de Mestrado, EPUSP, S. Paulo, 1975, cap.3)