

PROJETO 2017
(PCS3446 – Sistemas Operacionais)

Professor: João José Neto

Autor: Victor Funabashi

NUSP: 8992221

Sumário

| | |
|---|---|
| 1. Introdução | 2 |
| 2. Inclusão do ponteiro de 16 bits..... | 2 |
| 3. Modificação das instruções para abrigar endereçamento indireto | 3 |
| 4. Multiprogramação e Extensão da Memória Física | 5 |

1. Introdução

O objetivo deste projeto é a elaboração de uma simulação da operação de um sistema operacional multiprogramado em uma versão modificada da Máquina de von Neumann (MVN).

Para isso, utilizou-se a MVN projetada na disciplina PCS3216 – Sistemas de Programação na linguagem Java, modificando-a para aceitar as necessidades.

2. Inclusão do ponteiro de 16 bits

Para a inclusão do conceito do ponteiro de 16 bits, criou-se uma classe em Java que recebe os dois bytes e já os separa nas duas características:

- z Processo: Os 4 bits mais significativos indicam um dos 16 possíveis processos do programa
- z Endereço: Os outros 12 bits (os menos significativos) representam a posição de memória em que se encontra o processo indicado

Realizou-se os seguintes testes para verificar o funcionamento correto do ponteiro:

- ⊗ Passagem correta: um número hexadecimal de 4 algarismos
- ⊗ Erro tipo 1: Processo inválido – o primeiro caractere não corresponde a um valor hexadecimal
- ⊗ Erro tipo 2: Endereço inválido – os 3 últimos caracteres não correspondem a um número hexadecimal
- ⊗ Erro tipo 3: Passa-se um valor para o ponteiro que não contenha 2 bytes

A Figura 1 contém os valores testados e os resultados. Ressalta-se que os erros aqui considerados muito provavelmente não ocorrerão, devido ao uso deste ponteiro. Contudo foram formas defensivas planejadas, principalmente pelo fato de se utilizar Strings para o número em hexadecimal (sendo simples separar em/concatenar bytes).

```

Entrada: a0d8
Número do processo = 10
Endereço de memória = 0d8

Não existe esse processo
Entrada: ybcd
Número do processo = 0
Endereço de memória = bcd

Não existe esse endereço
Entrada: agf2
Número do processo = 10
Endereço de memória = null

Erro! Tentativa de passar conteúdo errado
Entrada: abd
Número do processo = 0
Endereço de memória = null

Erro! Tentativa de passar conteúdo errado
Entrada: abd05
Número do processo = 0
Endereço de memória = null

```

Figura 1 - Resultado do teste do ponteiro de 16 bits

3. Modificação das instruções para abrigar endereçamento indireto

Como recomendado no enunciado do projeto, eliminou-se a instrução RS (código de instrução B), substituindo-a por IND.

Essa nova instrução transforma a instrução seguinte em endereçamento indireto, por exemplo:

| Endereço de memória | Instrução (memória[endereço]+memória[endereço+1]) | PC |
|---------------------|---|-----|
| 010 | D000 | 012 |
| 012 | B000 | 014 |
| 014 | 43A2 | 3A2 |
| 016 | 3002 | |
| 3A2 | 9066 | 3A4 |

A Figura 2 apresenta o resultado dessa implementação. O programa ("sqrtN.txt") se inicia no endereço 00E. Lá se tem a instrução B000. Isso implica que a instrução seguinte (contida no endereço 010) será agora um ponteiro de 16 bits para o operando. Exatamente o que ocorre, dado que ao fim do que se tem na figura, o endereço atual (PC) é justamente o conteúdo dos 12 bits menos significativos do conteúdo de memória em 010 (região destacada).

```

000      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0 00
001      40 3E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
002      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D0 00
004      90 67 90 69 80 64 90 66 90 68 80 69 50 66 10 62
005      20 62 90 69 80 68 40 64 90 68 80 66 40 65 90 66
006      00 4A C0 32 01 02 00 00 00 00 00 00 00 00 00 00 00
007      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```
acc = 0      |      PC = 00E
```

```

4) adicionar interrupcao
5) desligar o passo-a-passo
Manter
0 que deseja fazer? 0

```

```

000      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0 00
001      40 3E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
002      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D0 00
004      90 67 90 69 80 64 90 66 90 68 80 69 50 66 10 62
005      20 62 90 69 80 68 40 64 90 68 80 66 40 65 90 66
006      00 4A C0 32 01 02 00 00 00 00 00 00 00 00 00 00 00
007      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```
acc = 0      |      PC = 010
```

```

4) adicionar interrupcao
5) desligar o passo-a-passo
Manter
0 que deseja fazer? 0

```

```

000      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 B0 00
001      40 3E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
002      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
003      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 D0 00
004      90 67 90 69 80 64 90 66 90 68 80 69 50 66 10 62
005      20 62 90 69 80 68 40 64 90 68 80 66 40 65 90 66
006      00 4A C0 32 01 02 00 00 00 00 00 00 00 00 00 00 00
007      00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

```
acc = 0      |      PC = 03E
```

Figura 2 - Resultado do teste de instrução de endereçamento indireto

Para se verificar isso, carregou-se a lista inicial de eventos (opção 0), ligou-se o modo passo-a-passo (opção 2), iniciou-se a simulação (opção 4). Em seguida foi-se seguindo os passos (digitando qualquer número que não fosse 4 ou 5), carregou-se, no meio do processo, o programa “sqrtN.txt” (digitou-se o conteúdo entre as aspas) até chegar no instante da Figura 2.

4. Multiprogramação e Extensão da Memória Física

Para essa seção faz-se duas ações: partição (lógica) da memória em 16 bancos (cada banco com 256 espaços cada); tabelas, uma para cada programa alocado, que realizam o mapeamento entre a posição lógica (definida no programa) com a física (posição absoluta) de memória de cada instrução.

Para tal, para cada programa, tem-se um endereço base a partir do qual ele será alocado (000, 100, 200, assim por diante). Assim, caso um programa ultrapasse 256 instruções ele alocará outro banco.

Uma desvantagem desse método é que, caso ele não tenha um número de instruções superior a 256 e próximo de um múltiplo (por exemplo, 512) ter-se-á desperdício de espaço de memória. Em contrapartida, é possível alocar mais de um programa que estivesse com os mesmos endereços lógicos, tendo-se assim, uma memória física aparente maior.

Para se testar este módulo carregou-se dois programas (apresentados na Figura 3).

| | | | |
|---------------|--|--------------------|--|
| 10E | | 00C | |
| 10E B000 | | 00C D000 | |
| 110 413E | | 00E 9035 | |
| 13E D000 | | 010 8032 | |
| 140 9167 | | 012 9037 | |
| 142 9169 | | 014 9034 | |
| 144 8164 | | 016 9036 | |
| 146 9166 | | 018 8037 | |
| 148 9168 | | 01A 5035 | |
| 14A 8169 | | 01C 1030 | |
| 14C 5166 | | 01E 8037 | |
| 14E 1162 | | 020 4032 | |
| 150 2162 | | 022 9037 | |
| 152 9169 | | 024 8034 | |
| 154 8168 | | 026 4033 | |
| 156 4164 | | 028 9034 | |
| 158 9168 | | 02A 4036 | |
| 15A 8166 | | 02C 9036 | |
| 15C 4165 | | 02E 0018 | |
| 15E 9166 | | 030 C030 | |
| 160 014A | | 032 0102 | |
| 162 C032 | | 034 0000 | |
| 164 0102 | | 036 0000 | |
| 166 0000 | | | |
| 168 0000 | | | |
| a - Raíz de n | | b - n ² | |

Figura 3 - Códigos-fonte carregados na MVN

O resultado da carga desses programas na memória deu nas tabelas de mapeamento mostradas na Figura 4.

| | | | | | |
|-----|-----|------|-----|-----|------|
| 00E | 10E | B000 | 10C | 00C | D000 |
| 010 | 110 | 413E | 10E | 00E | 9035 |
| 03E | 13E | D000 | 110 | 010 | 8032 |
| 040 | 140 | 9167 | 112 | 012 | 9037 |
| 042 | 142 | 9169 | 114 | 014 | 9034 |
| 044 | 144 | 8164 | 116 | 016 | 9036 |
| 046 | 146 | 9166 | 118 | 018 | 8037 |
| 048 | 148 | 9168 | 11A | 01A | 5035 |
| 04A | 14A | 8169 | 11C | 01C | 1030 |
| 04C | 14C | 5166 | 11E | 01E | 8037 |
| 04E | 14E | 1162 | 120 | 020 | 4032 |
| 050 | 150 | 2162 | 122 | 022 | 9037 |
| 052 | 152 | 9169 | 124 | 024 | 8034 |
| 054 | 154 | 8168 | 126 | 026 | 4033 |
| 056 | 156 | 4164 | 128 | 028 | 9034 |
| 058 | 158 | 9168 | 12A | 02A | 4036 |
| 05A | 15A | 8166 | 12C | 02C | 9036 |
| 05C | 15C | 4165 | 12E | 02E | 0018 |
| 05E | 15E | 9166 | 130 | 030 | C030 |
| 060 | 160 | 014A | 132 | 032 | 0102 |
| 062 | 162 | C032 | 134 | 034 | 0000 |
| 064 | 164 | 0102 | 136 | 036 | 0000 |
| 066 | 166 | 0000 | | | |
| 068 | 168 | 0000 | | | |

a - mapa de raiz de N

b - mapa de n²

Figura 4 - Mapas de memória (por programa)

Para se testar esse mapeamento, executou-se os dois programas. Dessa forma já se implementou uma forma de *Multithreading*: *Multithreading* com Granularidade Grossa (CGMT), que opera da forma apresentada na Figura 5.

Num processador com suporte à “multithreading”:



Figura 5 - CGMT (Fonte: Slide de aula de PCS3422 - Organização e Arquitetura de Computadores II)

A Figura 6 apresenta a memória com os dois programas devidamente carregados, um em cada banco. Na Figura 6.a tem-se apenas os dois programas carregados.

Por outro lado, a Figura 6.b tem-se após a execução, a qual foi devidamente encerrada. Os dois bytes destacados em (–) tem-se o resultado de “raiz de n”, sendo que o byte à esquerda contém o dado fornecido e o à direita, o resultado.

Já os bytes destacados em (–), o byte à direita contém o dado fornecido e o à esquerda, o resultado (n² em hexadecimal).

A visibilidade, tanto do mapa quanto da memória ocorreu por carregar a lista inicial de eventos (0), iniciar a simulação (4), carregar os dois programas (“sqrtN.txt” e “n2.txt”) e colocar os dados, no caso “9” para o primeiro programa e “4” para o segundo.

| | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 41 | 3E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 91 | 67 | 91 | 69 | 81 | 64 | 91 | 66 | 91 | 68 | 81 | 69 | 51 | 66 | 11 | 62 |
| 005 | 21 | 62 | 91 | 69 | 81 | 68 | 41 | 64 | 91 | 68 | 81 | 66 | 41 | 65 | 91 | 66 |
| 006 | 01 | 4A | C0 | 32 | 01 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 011 | 80 | 32 | 90 | 37 | 90 | 34 | 90 | 36 | 80 | 37 | 50 | 35 | 10 | 30 | 80 | 37 |
| 012 | 40 | 32 | 90 | 37 | 80 | 34 | 40 | 33 | 90 | 34 | 40 | 36 | 90 | 36 | 00 | 18 |
| 013 | C0 | 30 | 01 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 014 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 015 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 016 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 017 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 018 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 019 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

a. Memória pré-execução

| | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 41 | 3E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 91 | 67 | 91 | 69 | 81 | 64 | 91 | 66 | 91 | 68 | 81 | 69 | 51 | 66 | 11 | 62 |
| 005 | 21 | 62 | 91 | 69 | 81 | 68 | 41 | 64 | 91 | 68 | 81 | 66 | 41 | 65 | 91 | 66 |
| 006 | 01 | 4A | C0 | 32 | 01 | 02 | 05 | 09 | 03 | 05 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 011 | 80 | 32 | 90 | 37 | 90 | 34 | 90 | 36 | 80 | 37 | 50 | 35 | 10 | 30 | 80 | 37 |
| 012 | 40 | 32 | 90 | 37 | 80 | 34 | 40 | 33 | 90 | 34 | 40 | 36 | 90 | 36 | 00 | 18 |
| 013 | C0 | 30 | 01 | 02 | 07 | 04 | 10 | 04 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 014 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 015 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 016 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 017 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 018 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 019 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

b. Memória pós-execução

Figura 6 - Memória com os dois programas testados

4.1 Alocação paginada

Para melhor testar-se essa função de extensão da memória, utilizou-se dos mesmos programas, contendo algumas modificações, de modo que o programa de cálculo de \sqrt{n} ocupe duas páginas de memória lógica e n^2 , 3 páginas (Figura 7).

| | |
|----------|----------|
| 10E | 00C |
| 10E B000 | 00C D000 |
| 110 413E | 00E 9135 |
| 13E D000 | 010 8132 |
| 140 9267 | 012 9237 |
| 142 9269 | 014 9134 |
| 144 8264 | 016 9236 |
| 146 9266 | 018 8237 |
| 148 9268 | 01A 5135 |
| 14A 8269 | 01C 1130 |
| 14C 5266 | 01E 8237 |
| 14E 1262 | 020 4132 |
| 150 2262 | 022 9237 |
| 152 9269 | 024 8134 |
| 154 8268 | 026 4133 |
| 156 4264 | 028 9134 |
| 158 9268 | 02A 4236 |
| 15A 8266 | 02C 9236 |
| 15C 4265 | 02E 0018 |
| 15E 9266 | 030 C030 |
| 160 014A | 132 0102 |
| 162 C032 | 134 0000 |
| 264 0102 | 236 0000 |
| 266 0000 | |
| 268 0000 | |

c - Raiz de n

d - n^2

Figura 7 - Códigos-fonte carregados na MVN

A verificação da carga desses programas foi feita analisando suas tabelas de mapeamento (Figura 8). Essas, a menos dos endereços lógicos, são idênticas às contidas na Figura 4. Esse fato era esperado pois a mudança neste teste é para o comportamento de diversas páginas em um mesmo programa.

| | | | | | |
|-----|-----|------|-----|-----|------|
| 00E | 10E | 8000 | 10C | 00C | D000 |
| 010 | 110 | 413E | 10E | 00E | 9135 |
| 03E | 13E | D000 | 110 | 010 | 8132 |
| 040 | 140 | 9267 | 112 | 012 | 9237 |
| 042 | 142 | 9269 | 114 | 014 | 9134 |
| 044 | 144 | 8264 | 116 | 016 | 9236 |
| 046 | 146 | 9266 | 118 | 018 | 8237 |
| 048 | 148 | 9268 | 11A | 01A | 5135 |
| 04A | 14A | 8269 | 11C | 01C | 1130 |
| 04C | 14C | 5266 | 11E | 01E | 8237 |
| 04E | 14E | 1262 | 120 | 020 | 4132 |
| 050 | 150 | 2262 | 122 | 022 | 9237 |
| 052 | 152 | 9269 | 124 | 024 | 8134 |
| 054 | 154 | 8268 | 126 | 026 | 4133 |
| 056 | 156 | 4264 | 128 | 028 | 9134 |
| 058 | 158 | 9268 | 12A | 02A | 4236 |
| 05A | 15A | 8266 | 12C | 02C | 9236 |
| 05C | 15C | 4265 | 12E | 02E | 0018 |
| 05E | 15E | 9266 | 130 | 030 | C030 |
| 060 | 160 | 014A | 132 | 132 | 0102 |
| 062 | 162 | C032 | 134 | 134 | 0000 |
| 064 | 264 | 0102 | 136 | 236 | 0000 |
| 066 | 266 | 0000 | | | |
| 068 | 268 | 0000 | | | |

c - mapa de raíz de N

d - mapa de n²

Figura 8 - Mapas de memória (por programa)

Por fim, averiguou-se o resultado obtido da execução desses programas, do mesmo modo descrito anteriormente, obtendo-se o resultado da Figura 9. Como pode-se reparar, não apenas teve-se os resultados corretos ($\sqrt{9} = 3$ e $4^2 = 16_{10} = 10_{16}$) como a carga foi feita de modo sequencial, como implementado, independente da página em que a instrução se encontra. Isso implica que a alocação foi correta, bem como sua tradução de endereços.

| | | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 41 | 3E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 92 | 67 | 92 | 69 | 82 | 64 | 92 | 66 | 92 | 68 | 82 | 69 | 52 | 66 | 12 | 62 | |
| 005 | 22 | 62 | 92 | 69 | 82 | 68 | 42 | 64 | 92 | 68 | 82 | 66 | 42 | 65 | 92 | 66 | |
| 006 | 01 | 4A | C0 | 32 | 01 | 02 | 05 | 09 | 03 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 007 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 008 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 009 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 00F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 010 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 011 | 81 | 32 | 92 | 37 | 91 | 34 | 92 | 36 | 82 | 37 | 51 | 35 | 11 | 30 | 82 | 37 | |
| 012 | 41 | 32 | 92 | 37 | 81 | 34 | 41 | 33 | 91 | 34 | 42 | 36 | 92 | 36 | 00 | 18 | |
| 013 | C0 | 30 | 01 | 02 | 07 | 04 | 10 | 04 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 014 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 015 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 016 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 017 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 018 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 019 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Figura 9 - Memória pós-execução dos programas

5. Memória Virtual e suas Interrupções

O primeiro passo para se realizar a memória virtual foi implementar uma estrutura de dados que corresponda ao disco. Segundo o enunciado seu tamanho deve ser tal que comporte 16 programas (16 bancos físicos disponíveis), cada um podendo endereçar 16 bancos lógicos de 4kB. Assim que a memória deve ter um tamanho de 1MB, que, para se assimilar à memória, simulou-a como sendo uma matriz de 64kB x 16B

```
String[][] disk; // matriz que simula o disco da maquina
this.disk = new String[65536][16]; // 64kB x 16B
```

O passo seguinte foi simular uma memória parcialmente cheia, isto é, um caso em que determinada página do programa não pode ser alocada. Isto foi necessário pois até o momento a carga de programa era feita completa. E a simulação se deve por estar carregando poucos programas de uma memória recém iniciada, logo vazia.

Para um dos programas anteriores ("sqrtN.txt") foi-se testada essa nova funcionalidade, o resultado (Figura 10) está conforme esperado, o qual, se comparado

com o obtido na seção 4, possui diferenças nas instruções contidas em diferentes páginas (do endereço lógico).

| fisico | logic | value |
|--------|-------|-------|
| 00E | 10E | B000 |
| 010 | 110 | 413E |
| 03E | 13E | D000 |
| 040 | 140 | 9267 |
| 042 | 142 | 9269 |
| 044 | 144 | 8264 |
| 046 | 146 | 9266 |
| 048 | 148 | 9268 |
| 04A | 14A | 8269 |
| 04C | 14C | 5266 |
| 04E | 14E | 1262 |
| 050 | 150 | 2262 |
| 052 | 152 | 9269 |
| 054 | 154 | 8268 |
| 056 | 156 | 4264 |
| 058 | 158 | 9268 |
| 05A | 15A | 8266 |
| 05C | 15C | 4265 |
| 05E | 15E | 9266 |
| 060 | 160 | 014A |
| 062 | 162 | C032 |
| NULL | 264 | 0102 |
| NULL | 266 | 0000 |
| NULL | 268 | 0000 |

| | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 001 | 41 | 3E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 002 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 003 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 004 | 92 | 67 | 92 | 69 | 82 | 64 | 92 | 66 | 92 | 68 | 82 | 69 | 52 | 66 | 12 | 62 |
| 005 | 22 | 62 | 92 | 69 | 82 | 68 | 42 | 64 | 92 | 68 | 82 | 66 | 42 | 65 | 92 | 66 |
| 006 | 01 | 4A | C0 | 32 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

Figura 10 - Carga de apenas uma página de um programa

Conforme informado no enunciado do projeto, a memória virtual (o disco) opera sobre blocos de 4kB, ou seja, o tamanho equivalente a uma memória. Logo, caso uma página não esteja presente, será necessário substituir todo o conteúdo da memória para executar a instrução.

Para isso, o primeiro passo é realizar a carga, no disco, de diferentes páginas do programa em diferentes blocos do disco. A fim de se identificar o bloco que aquela instrução se encontra, será adicionada essa característica à PMT (Page Map Table).

A verificação foi feita em duas etapas: escrita de um programa e escrita de mais de um programa (no caso dois programas).

Para ambos os casos, visando ter melhor visibilidade, apresenta-se apenas os blocos dos discos que são preenchidos e as posições vazias possuem “..”, dado que podem haver blocos preenchidos apenas com “00” (o que impossibilitaria distinguir).

Também para simplificação fez-se que o bloco do disco ao qual a página é alocada é definido pelo algarismo hexadecimal (4 bits) mais significativo da posição lógica de memória.

| | fisico | logic | value | bloco |
|--|--------|-------|-------|-------|
| | 00E | 10E | 8000 | 1 |
| | 010 | 110 | 413E | 1 |
| | 03E | 13E | D000 | 1 |
| | 040 | 140 | 9267 | 1 |
| | 042 | 142 | 9269 | 1 |
| | 044 | 144 | 8264 | 1 |
| | 046 | 146 | 9266 | 1 |
| | 048 | 148 | 9268 | 1 |
| | 04A | 14A | 8269 | 1 |
| | 04C | 14C | 5266 | 1 |
| | 04E | 14E | 1262 | 1 |
| | 050 | 150 | 2262 | 1 |
| | 052 | 152 | 9269 | 1 |
| | 054 | 154 | 8268 | 1 |
| | 056 | 156 | 4264 | 1 |
| | 058 | 158 | 9268 | 1 |
| | 05A | 15A | 8266 | 1 |
| | 05C | 15C | 4265 | 1 |
| | 05E | 15E | 9266 | 1 |
| | 060 | 160 | 014A | 1 |
| | 062 | 162 | C032 | 1 |
| | NULL | 264 | 0102 | 2 |
| | NULL | 266 | 0000 | 2 |
| | NULL | 268 | 0000 | 2 |

[illegible]

| | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 100 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | B0 | 00 |
| 101 | 41 | 3E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 102 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 103 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | D0 | 00 |
| 104 | 92 | 67 | 92 | 69 | 82 | 64 | 92 | 66 | 92 | 68 | 82 | 69 | 52 | 66 | 12 | 62 |
| 105 | 22 | 62 | 92 | 69 | 82 | 68 | 42 | 64 | 92 | 68 | 82 | 66 | 42 | 65 | 92 | 66 |
| 106 | 01 | 4A | C0 | 32 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 200 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 201 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 202 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 203 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 204 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 205 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 206 | .. | .. | .. | .. | 01 | 02 | 00 | 00 | 00 | 00 | .. | .. | .. | .. | .. | .. |

12

| fisico | logic | value | bloco | fisico | logic | value | bloco |
|--------|-------|-------|-------|--------|-------|-------|-------|
| 00E | 10E | B000 | 1 | 10C | 00C | D000 | 0 |
| 010 | 110 | 413E | 1 | 10E | 00E | 9135 | 0 |
| 03E | 13E | D000 | 1 | 110 | 010 | 8132 | 0 |
| 040 | 140 | 9267 | 1 | 112 | 012 | 9237 | 0 |
| 042 | 142 | 9269 | 1 | 114 | 014 | 9134 | 0 |
| 044 | 144 | 8264 | 1 | 116 | 016 | 9236 | 0 |
| 046 | 146 | 9266 | 1 | 118 | 018 | 8237 | 0 |
| 048 | 148 | 9268 | 1 | 11A | 01A | 5135 | 0 |
| 04A | 14A | 8269 | 1 | 11C | 01C | 1130 | 0 |
| 04C | 14C | 5266 | 1 | 11E | 01E | 8237 | 0 |
| 04E | 14E | 1262 | 1 | 120 | 020 | 4132 | 0 |
| 050 | 150 | 2262 | 1 | 122 | 022 | 9237 | 0 |
| 052 | 152 | 9269 | 1 | 124 | 024 | 8134 | 0 |
| 054 | 154 | 8268 | 1 | 126 | 026 | 4133 | 0 |
| 056 | 156 | 4264 | 1 | 128 | 028 | 9134 | 0 |
| 058 | 158 | 9268 | 1 | 12A | 02A | 4236 | 0 |
| 05A | 15A | 8266 | 1 | 12C | 02C | 9236 | 0 |
| 05C | 15C | 4265 | 1 | 12E | 02E | 0018 | 0 |
| 05E | 15E | 9266 | 1 | 130 | 030 | C030 | 0 |
| 060 | 160 | 014A | 1 | NULL | 132 | 0102 | 1 |
| 062 | 162 | C032 | 1 | NULL | 134 | 0000 | 1 |
| NULL | 264 | 0102 | 2 | NULL | 236 | 0000 | 2 |
| NULL | 266 | 0000 | 2 | | | | |
| NULL | 268 | 0000 | 2 | | | | |

a - PMT (sqrtN à esquerda | n² à direita)

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01 | 41 | 3E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | D0 | 00 |
| 04 | 92 | 67 | 92 | 69 | 82 | 64 | 92 | 66 | 92 | 68 | 82 | 69 | 52 | 66 | 12 | 62 |
| 05 | 22 | 62 | 92 | 69 | 82 | 68 | 42 | 64 | 92 | 68 | 82 | 66 | 42 | 65 | 92 | 66 |
| 06 | 01 | 4A | C0 | 32 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 07 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 08 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 09 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | D0 | 00 | 91 | 35 |
| 11 | 81 | 32 | 92 | 37 | 91 | 34 | 92 | 36 | 82 | 37 | 51 | 35 | 11 | 30 | 82 | 37 |
| 12 | 41 | 32 | 92 | 37 | 81 | 34 | 41 | 33 | 91 | 34 | 42 | 36 | 92 | 36 | 00 | 18 |
| 13 | C0 | 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

b - Memória Principal

| | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 001 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 002 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 003 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 004 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 005 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 006 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 007 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 008 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 009 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00A | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00B | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00C | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00D | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00F | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 010 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | D0 | 00 | 91 | 35 |
| 011 | 81 | 32 | 92 | 37 | 91 | 34 | 92 | 36 | 82 | 37 | 51 | 35 | 11 | 30 | 82 | 37 |
| 012 | 41 | 32 | 92 | 37 | 81 | 34 | 41 | 33 | 91 | 34 | 42 | 36 | 92 | 36 | 00 | 18 |
| 013 | C0 | 30 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 100 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | B0 | 00 |
| 101 | 41 | 3E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 102 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 103 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | D0 | 00 |
| 104 | 92 | 67 | 92 | 69 | 82 | 64 | 92 | 66 | 92 | 68 | 82 | 69 | 52 | 66 | 12 | 62 |
| 105 | 22 | 62 | 92 | 69 | 82 | 68 | 42 | 64 | 92 | 68 | 82 | 66 | 42 | 65 | 92 | 66 |
| 106 | 01 | 4A | C0 | 32 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 107 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 108 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 109 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10A | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10B | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10C | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10D | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10F | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 110 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 111 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 112 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 113 | .. | .. | 01 | 02 | 00 | 00 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 200 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 201 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 202 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 203 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 204 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 205 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 206 | .. | .. | .. | .. | 01 | 02 | 00 | 00 | 00 | 00 | .. | .. | .. | .. | .. | .. |
| 207 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 208 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 209 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20A | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20B | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20C | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20D | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20F | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 210 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 211 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 212 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 213 | .. | .. | .. | .. | .. | .. | 00 | 00 | .. | .. | .. | .. | .. | .. | .. | .. |

c - Memória virtual (disco)

Figura 12 - Carga de dois programas em memória e disco

Já o segundo caso, foi feito para verificar a integridade do que estava feito, ou seja, sem sobrescrever conteúdo de memória que poderiam se sobrepor e manter mais de um programa na memória (e no disco). O resultado, como esperado e desejado, pode ser visto na Figura 12.

O passo seguinte foi criar um tratamento para o caso de o bloco necessário não ser o contido na memória. Para isso criou-se uma variável “blocoAtual” que indica a qual bloco do disco corresponde o conteúdo da memória.

Então fez-se os seguintes passos:

- Armazenar o conteúdo da memória no bloco correspondente do disco;
- Puxar o conteúdo do bloco desejado do disco para a memória;

Como o objetivo desta etapa era apenas o tratamento, fez-se como condições iniciais (anteriores à substituição dos blocos):

- Conteúdo da memória vazio (todas as posições com “00”);
- Conteúdo do disco vazio, para visualização se fez
 - Bloco ‘1’ com “***”
 - Blocos restantes com “..” (mas mostrar-se-á somente o bloco ‘0’ pelo uso no teste)
- blocoAtual = 0 e ‘blocoDesejado’ = 1

Ao executar o tratamento, repara-se que foi executado corretamente, pois o conteúdo da memória foi devidamente armazenado no disco, enquanto o conteúdo desejado do disco foi colocado na memória.

```

00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
03  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
04  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
05  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
07  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
08  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
09  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0A  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0B  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0D  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0E  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
11  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
12  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
14  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
15  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
18  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
19  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1A  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1B  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1D  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1E  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1F  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

a – Memória


```

00  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** 
01  ** ** ** *
02  ** ** ** *
03  ** ** ** *
04  ** ** ** *
05  ** ** ** *
06  ** ** ** *
07  ** ** ** *
08  ** ** ** *
09  ** ** ** *
0A  ** ** ** *
0B  ** ** ** *
0C  ** ** ** *
0D  ** ** ** *
0E  ** ** ** *
0F  ** ** ** *
10  ** ** ** *
11  ** ** ** *
12  ** ** ** *
13  ** ** ** *
14  ** ** ** *
15  ** ** ** *
16  ** ** ** *
17  ** ** ** *
18  ** ** ** *
19  ** ** ** *
1A  ** ** ** *
1B  ** ** ** *
1C  ** ** ** *
1D  ** ** ** *
1E  ** ** ** *
1F  ** ** ** *

```

b – Memória

Figura 14 - Estado das Estruturas de Armazenamento posteriori à substituição de conteúdo

Então, para utilizar essa troca de blocos criou-se um evento de código “C”. Para testá-lo fez-se que a lista inicial de eventos teria três eventos: “Início”; “Troca”; “Fim”. Ao executar a simulação obteve-se o resultado da Figura 16. Se comparado com o da Figura 14, pode-se reparar que são idênticos, o que já era esperado.

```

00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
02  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
03  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
04  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
05  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
07  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
08  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
09  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0A  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0B  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0D  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0E  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0F  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
11  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
12  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
13  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
14  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
15  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
16  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
17  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
18  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
19  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1A  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1B  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1D  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1E  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
1F  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```

c – Memória

d – Disco


```

00  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** 
01  ** ** ** *
02  ** ** ** *
03  ** ** ** *
04  ** ** ** *
05  ** ** ** *
06  ** ** ** *
07  ** ** ** *
08  ** ** ** *
09  ** ** ** *
0A  ** ** ** *
0B  ** ** ** *
0C  ** ** ** *
0D  ** ** ** *
0E  ** ** ** *
0F  ** ** ** *
10  ** ** ** *
11  ** ** ** *
12  ** ** ** *
13  ** ** ** *
14  ** ** ** *
15  ** ** ** *
16  ** ** ** *
17  ** ** ** *
18  ** ** ** *
19  ** ** ** *
1A  ** ** ** *
1B  ** ** ** *
1C  ** ** ** *
1D  ** ** ** *
1E  ** ** ** *
1F  ** ** ** *

```

d – Memória

Figura 16 - Estado das Estruturas de Armazenamento posterior à substituição de conteúdo

Com o sistema de execução multipaginado foi necessário salvar em uma variável qual a instrução atual, pois para casos de instrução de acesso à memória em que o endereço a ser acessado está em bloco diferente da instrução lida, ainda é necessário saber a ação. Assim colocou-se também uma variável booleana que indica esse estado, para que não se leia uma outra instrução (totalmente não relacionada, lixo).

| fisico | logic | value | bloco | fisico | logic | value | bloco |
|--------|-------|-------|-------|--------|-------|-------|-------|
| 00E | 10E | B000 | 1 | 10C | 00C | D000 | 0 |
| 010 | 110 | 413E | 1 | 10E | 00E | 9135 | 0 |
| 03E | 13E | D000 | 1 | 110 | 010 | 8132 | 0 |
| 040 | 140 | 9267 | 1 | 112 | 012 | 9237 | 0 |
| 042 | 142 | 9269 | 1 | 114 | 014 | 9134 | 0 |
| 044 | 144 | 8264 | 1 | 116 | 016 | 9236 | 0 |
| 046 | 146 | 9266 | 1 | 118 | 018 | 8237 | 0 |
| 048 | 148 | 9268 | 1 | 11A | 01A | 5135 | 0 |
| 04A | 14A | 8269 | 1 | 11C | 01C | 1130 | 0 |
| 04C | 14C | 5266 | 1 | 11E | 01E | 8237 | 0 |
| 04E | 14E | 1262 | 1 | 120 | 020 | 4132 | 0 |
| 050 | 150 | 2262 | 1 | 122 | 022 | 9237 | 0 |
| 052 | 152 | 9269 | 1 | 124 | 024 | 8134 | 0 |
| 054 | 154 | 8268 | 1 | 126 | 026 | 4133 | 0 |
| 056 | 156 | 4264 | 1 | 128 | 028 | 9134 | 0 |
| 058 | 158 | 9268 | 1 | 12A | 02A | 4236 | 0 |
| 05A | 15A | 8266 | 1 | 12C | 02C | 9236 | 0 |
| 05C | 15C | 4265 | 1 | 12E | 02E | 0018 | 0 |
| 05E | 15E | 9266 | 1 | 130 | 030 | C030 | 0 |
| 060 | 160 | 014A | 1 | NULL | 132 | 0102 | 1 |
| 062 | 162 | C032 | 1 | NULL | 134 | 0704 | 1 |
| NULL | 264 | 0102 | 2 | NULL | 236 | 1004 | 2 |
| NULL | 266 | 0509 | 2 | | | | |
| NULL | 268 | 0305 | 2 | | | | |

d - PMT (sqrtN à esquerda | n² à direita)

| | | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 01 | 41 | 3E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | D0 | 00 |
| 04 | 92 | 67 | 92 | 69 | 82 | 64 | 92 | 66 | 92 | 68 | 82 | 69 | 52 | 66 | 12 | 62 |
| 05 | 22 | 62 | 92 | 69 | 82 | 68 | 42 | 64 | 92 | 68 | 82 | 66 | 42 | 65 | 92 | 66 |
| 06 | 01 | 4A | C0 | 32 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 07 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 08 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 09 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0A | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0B | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0C | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0D | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0E | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 0F | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | D0 | 00 | 91 | 35 |
| 11 | 81 | 32 | 92 | 37 | 91 | 34 | 92 | 36 | 82 | 37 | 51 | 35 | 11 | 30 | 82 | 37 |
| 12 | 41 | 32 | 92 | 37 | 81 | 34 | 41 | 33 | 91 | 34 | 42 | 36 | 92 | 36 | 00 | 18 |
| 13 | C0 | 30 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |

e - Memória Principal

| | | | | | | | | | | | | | | | | |
|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 000 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 001 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 002 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 003 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 004 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 005 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 006 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 007 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 008 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 009 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00A | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00B | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00C | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00D | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 00F | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 010 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | D0 | 00 | 91 | 35 | |
| 011 | 81 | 32 | 92 | 37 | 91 | 34 | 92 | 36 | 82 | 37 | 51 | 35 | 11 | 30 | 82 | 37 |
| 012 | 41 | 32 | 92 | 37 | 81 | 34 | 41 | 33 | 91 | 34 | 42 | 36 | 92 | 36 | 00 | 18 |
| 013 | C0 | 30 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 100 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | B0 | 00 |
| 101 | 41 | 3E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 102 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 103 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | D0 | 00 |
| 104 | 92 | 67 | 92 | 69 | 82 | 64 | 92 | 66 | 92 | 68 | 82 | 69 | 52 | 66 | 12 | 62 |
| 105 | 22 | 62 | 92 | 69 | 82 | 68 | 42 | 64 | 92 | 68 | 82 | 66 | 42 | 65 | 92 | 66 |
| 106 | 01 | 4A | C0 | 32 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 107 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 108 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 109 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10A | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10B | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10C | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10D | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 10F | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 110 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 111 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 112 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 113 | .. | .. | 01 | 02 | 07 | 04 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 200 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 201 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 202 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 203 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 204 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 205 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 206 | .. | .. | .. | .. | 01 | 02 | 05 | 09 | 03 | 05 | .. | .. | .. | .. | .. | .. |
| 207 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 208 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 209 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20A | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20B | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20C | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20D | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20E | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 20F | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 210 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 211 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 212 | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. | .. |
| 213 | .. | .. | .. | .. | .. | .. | 10 | 04 | .. | .. | .. | .. | .. | .. | .. | .. |

f - Memória virtual (disco)

Figura 17 - Execução de dois programas em memória e disco

6. Sistema de Interrupção

O sistema de interrupções a ser incorporado à arquitetura do processador deve implementar um mínimo de funcionalidades dos sistemas de interrupção geralmente encontrados nos processadores usuais:

- Permite/Inibe interrupção - Um bit de estado indica se pedidos de interrupção podem ser atendidas ou não pelo processador. É necessário que haja uma instrução para ligar e desligar este bit. Ao iniciar o processamento, este bit deve estar desligado (interrupção inibida).
- Supervisor/Usuário - o modo "supervisor" é o modo de operação em que o sistema operacional opera: ao início do processamento e durante o atendimento de uma interrupção. Passa-se do modo supervisor para o modo usuário executando-se uma instrução de retorno de interrupção. Passa-se do modo usuário para supervisor: (a) quando se executa uma instrução de chamada de sistema; (b) quando a interrupção estiver permitida, algum pedido interrupção estiver ligado e o processador estiverem modo usuário;
- Um bit de estado do processador indica se está em curso ou não um tratamento de algum pedido de interrupção. Importante: interrupções são sempre tratadas em modo supervisor. Nesta proposta, só uma interrupção pode ser tratada de cada vez, ficando eventuais outros pedidos de interrupção aguardando até que o processador volte ao modo usuário antes que mais um deles possa ser atendido.
- O tratamento de interrupções é feito inicialmente passando o processador para modo supervisor, salvando o endereço de retorno na posição 0 do bloco 0 de memória, e desviando para o programa de atendimento da interrupção, que reside a partir da posição 2 do bloco 0 de memória. Este é também o endereço de partida do sistema, quando acionado pela primeira vez.

7. Entrada e Saída e Suas Interrupções

- Dois bits de estado (em uso ou não; terminou ou não) são usados para cada dispositivo de entrada/saída: um para indicar se o dispositivo está sendo utilizado (ou seja, que uma operação está em curso no dispositivo) ou não, e outro, para indicar que o dispositivo já terminou a última operação iniciada, mas que o processador ainda não fez o respectivo tratamento.
- Pedidos de interrupção da parte dos dispositivos ocorrem quando os bits de estado do dispositivo estiverem ambos ligados ao mesmo tempo (ou seja, estando o dispositivo em uso, terminou a operação nele iniciada anteriormente).
- A rotina de tratamento de interrupção de entrada/saída deve desligar os dois bits de estado após tratar a interrupção, deixando o dispositivo livre e pronto para a sua próxima tarefa.

- As duas operações mais usuais correspondem às tarefas de leitura e de escrita. Dispositivos convencionais efetuam a entrada ou a saída de um byte por vez. O dispositivo deve ter à sua disposição um registrador de oito bits onde o dado que se está transferindo deve ser depositado.
- Se for dispositivo de entrada, o dado lido é depositado nesse registrador pelo dispositivo assim que estiver pronto, e deve ser retirado pela rotina de tratamento de interrupção para ser transferido para a memória do processador.
- Se for dispositivo de saída, esse dado deve ser colocado no registrador do dispositivo antes de o dispositivo ser acionado para efetuar a operação de escrita no meio externo.

8. Chamada de Supervisor e a Interrupção Correspondente

- Deve ser implementada a instrução de chamada de supervisor e pelo menos alguns de seus serviços essenciais.
- A execução de uma instrução de chamada de supervisor se resume essencialmente a provocar um pedido de interrupção, cujo atendimento deve tirar o processador do modo usuário, mudando seu estado para modo supervisor, salvando na posição 0 do bloco 0 de memória o endereço de retorno, e forçando a execução da rotina de tratamento geral de interrupção do sistema, presumidamente presente a partir da posição 2 do bloco 0.
- Implemente as chamadas de sistema essenciais, coletando as necessidades apresentadas neste texto. Um conjunto mínimo é o seguinte:
 - leitura e escrita em dispositivos de entrada/saída,
 - solicitação de final de processamento,
 - operações de create/delete, open/close, read/write aplicados a arquivos de texto.