



Trabalho Sistemas Operacionais – Fase 4 e 5

Integrantes:

- Ramiro Nascimento
- Jéssica Freua
- João Pedro Medeiros
- Matheus

Seção Implementação:

O programa feito pelo grupo apresenta todas as funcionalidades requeridas até a fase 5, com exceção de:

- + Programa BubbleSort
- + Adaptação dos programas ProgMin e Fibonacci para utilizar o Opcode TRAP

Obs: o programa Fatorial está completo e os testes serão feitos majoritariamente sobre ele.

Restrições

Após várias baterias de testes seguidos de ajuste e refatoração do código, o programa encontra apenas 1 bug:

- + Ao preencher todos os frames com programas, caso algum deles solicite mais memória, o programa devolve uma mensagem de erro dizendo que não há memória suficiente (esta parte está executando como esperado), porém a posição de memória que aloca a instrução que solicitou memória é substituída do seu Opcode original pelo Opcode.DATA.

```
----- MENU DE OPCOES S.O. -----  
[1] - Rodar programa existente em memoria  
[2] - Adicionar programa a memoria  
[3] - Remover programa da memória  
[4] - Rodar todos programas em memória  
[5] - Dump de memória de um programa específico  
[0] - Sair
```

1

```
----- Programas disponiveis em memória  
[ 1 ] Programa: FATORIAL Frames:[0, 1]
```

```
RUNNING -> Programa[1 - FATORIAL] PC = 1
```

5

```
RUNNING -> Programa[1 - FATORIAL] PC = 2
```

```
**Interrupção de Sistema:
```

```
**Erro na intrução: [ DATA, 9, -1, 5 ] Nao há  
memória livre. Favor, remova um programa da memória
```

- É escolhido o programa 1 para rodar e na sequencia, quando é solicitado a memória mais frames, a instrução é substituída por DATA e na próxima vez que for tentar rodar esse mesmo processo, essa instrução estará errada.

Teste 1 – Adicionar programa a memória e executá-lo

No menu principal, escolha a opção 2 e em seguida adicione o programa Fatorial (opção 3). Ao escolher o Fatorial, ele será adicionado à memória com o estado READY para ser rodado e o Menu voltará ao principal. Já no menu principal, digite 1 para escolher um dos programas em memória, escolha o único disponível (opção 1) e então é dado o *Run* no programa.

```
----- MENU DE OPCOES S.O. -----
[1] - Rodar programa existente em memoria
[2] - Adicionar programa a memoria
[3] - Remover programa da memória
[4] - Rodar todos programas em memória
[5] - Dump de memória de um programa específico
[0] - Sair
2
----- ESCOLHA O PROGRAMA -----
[1] - ProgMin
[2] - Fibonacci
[3] - Fatorial
[4] - BubbleSort
[0] - Sair
3
```

Resultado esperado:

```
RUNNING -> Programa[2 - FATORIAL] PC = 33
5
RUNNING -> Programa[2 - FATORIAL] PC = 34
RUNNING -> Programa[2 - FATORIAL] PC = 35
RUNNING -> Programa[2 - FATORIAL] PC = 36
READY -> Programa[2 - FATORIAL] PC = 37
RUNNING -> Programa[2 - FATORIAL] PC = 38
RUNNING -> Programa[2 - FATORIAL] PC = 39
RUNNING -> Programa[2 - FATORIAL] PC = 40
RUNNING -> Programa[2 - FATORIAL] PC = 41
READY -> Programa[2 - FATORIAL] PC = 42
RUNNING -> Programa[2 - FATORIAL] PC = 43
RUNNING -> Programa[2 - FATORIAL] PC = 44
RUNNING -> Programa[2 - FATORIAL] PC = 40
RUNNING -> Programa[2 - FATORIAL] PC = 41
READY -> Programa[2 - FATORIAL] PC = 42
RUNNING -> Programa[2 - FATORIAL] PC = 43
RUNNING -> Programa[2 - FATORIAL] PC = 44
RUNNING -> Programa[2 - FATORIAL] PC = 40
RUNNING -> Programa[2 - FATORIAL] PC = 41
READY -> Programa[2 - FATORIAL] PC = 42
RUNNING -> Programa[2 - FATORIAL] PC = 43
RUNNING -> Programa[2 - FATORIAL] PC = 44
RUNNING -> Programa[2 - FATORIAL] PC = 40
RUNNING -> Programa[2 - FATORIAL] PC = 41
READY -> Programa[2 - FATORIAL] PC = 42
RUNNING -> Programa[2 - FATORIAL] PC = 43
RUNNING -> Programa[2 - FATORIAL] PC = 44
RUNNING -> Programa[2 - FATORIAL] PC = 45
RUNNING -> Programa[2 - FATORIAL] PC = 46
READY -> Programa[2 - FATORIAL] PC = 47
RUNNING -> Programa[2 - FATORIAL] PC = 48
120
RUNNING -> Programa[2 - FATORIAL] PC = 49
----- fim do programa
```

Obs: a oscilação de RUNNING para READY, são os estados do Escalonamento. O programa a cada 5 instruções vai pra READY para entrar denovo na fila e ser executado até o final.

Teste 2 – remover programa da memória.

Para remover um programa da memória, usa-se a opção 3 do Menu. Assim que escolher, é mostrado todos os programas em memória, junto do seu nome e quais frames ele ocupa. Escolhendo qual programa você deseja deletar é retornada uma mensagem confirmando a remoção da opção.

Resultado esperado:

```
----- MENU DE OPCOES S.O. -----  
[1] - Rodar programa existente em memoria  
[2] - Adicionar programa a memoria  
[3] - Remover programa da memória  
[4] - Rodar todos programas em memória  
[5] - Dump de memória de um programa específico  
[0] - Sair  
3  
----- Programas disponiveis em memória  
[ 1 ] Programa: FATORIAL Frames:[0, 1]  
-----  
Digite o numero do programa:  
1  
-----  
Programa [ 1 ] [ FATORIAL ] foi excluído com sucesso.
```

Teste 3 – Roda todos os programas da memória.

Aqui poderemos ver melhor a aplicação do ProcessControlBlock em colaboração com o Escalonador. Vamos simular a adição de 2 programas FATORIAL em memória. Em um deles vamos pedir o fatorial de 5 (retornando 120) e no outro o fatorial de 4 (retornando 24).

```
----- Programas disponíveis em memória
[ 1 ] Programa: FATORIAL Frames:[0, 1]
[ 2 ] Programa: FATORIAL Frames:[2, 3]
```

Resultado esperado

```
----- MENU DE OPCOES S.O. -----
[1] - Rodar programa existente em memoria
[2] - Adicionar programa a memoria
[3] - Remover programa da memória
[4] - Rodar todos programas em memória
[5] - Dump de memória de um programa específico
[0] - Sair
4
RUNNING -> Programa[1 - FATORIAL] PC = 1
5
RUNNING -> Programa[1 - FATORIAL] PC = 2
RUNNING -> Programa[1 - FATORIAL] PC = 3
RUNNING -> Programa[1 - FATORIAL] PC = 4
READY -> Programa[1 - FATORIAL] PC = 5
RUNNING -> Programa[2 - FATORIAL] PC = 33
4
RUNNING -> Programa[2 - FATORIAL] PC = 34
RUNNING -> Programa[2 - FATORIAL] PC = 35
RUNNING -> Programa[2 - FATORIAL] PC = 36
READY -> Programa[2 - FATORIAL] PC = 37
RUNNING -> Programa[1 - FATORIAL] PC = 6
RUNNING -> Programa[1 - FATORIAL] PC = 7
RUNNING -> Programa[1 - FATORIAL] PC = 8
RUNNING -> Programa[1 - FATORIAL] PC = 9
READY -> Programa[1 - FATORIAL] PC = 10
RUNNING -> Programa[2 - FATORIAL] PC = 38
RUNNING -> Programa[2 - FATORIAL] PC = 39
RUNNING -> Programa[2 - FATORIAL] PC = 40
RUNNING -> Programa[2 - FATORIAL] PC = 41
READY -> Programa[2 - FATORIAL] PC = 42
RUNNING -> Programa[1 - FATORIAL] PC = 11
RUNNING -> Programa[1 - FATORIAL] PC = 12
RUNNING -> Programa[1 - FATORIAL] PC = 8
RUNNING -> Programa[1 - FATORIAL] PC = 9
READY -> Programa[1 - FATORIAL] PC = 10
RUNNING -> Programa[2 - FATORIAL] PC = 43
RUNNING -> Programa[2 - FATORIAL] PC = 44
RUNNING -> Programa[2 - FATORIAL] PC = 45
RUNNING -> Programa[2 - FATORIAL] PC = 46
READY -> Programa[2 - FATORIAL] PC = 47
RUNNING -> Programa[1 - FATORIAL] PC = 11
RUNNING -> Programa[1 - FATORIAL] PC = 12
RUNNING -> Programa[1 - FATORIAL] PC = 13
RUNNING -> Programa[1 - FATORIAL] PC = 14
READY -> Programa[1 - FATORIAL] PC = 15
RUNNING -> Programa[2 - FATORIAL] PC = 48
24
RUNNING -> Programa[2 - FATORIAL] PC = 49
----- fim do programa
RUNNING -> Programa[1 - FATORIAL] PC = 16
120
RUNNING -> Programa[1 - FATORIAL] PC = 17
----- fim do programa
```

Teste 4 – Dump de memória de um programa específico

Ao escolher a opção 5 de memória, é mostrado quais programas estão em memória. Escolhendo qual deles será feito o dump, o sistema fará o print na tela de cada posição dos frames daquele programa.

```
----- MENU DE OPCOES S.O. -----
[1] - Rodar programa existente em memoria
[2] - Adicionar programa a memoria
[3] - Remover programa da memória
[4] - Rodar todos programas em memória
[5] - Dump de memória de um programa específico
[0] - Sair
5
----- Programas disponiveis em memória
[ 1 ] Programa: FATORIAL Frames:[0, 1, 4, 5]
[ 2 ] Programa: FATORIAL Frames:[2, 3, 6, 7]
-----
Digite o numero do programa:
|
```

Resultado Esperado

```
Digite o numero do programa:
1
-----
---- Frame [0 ]
0: [ LDI, 8, -1, 1 ]
1: [ TRAP, -1, -1, -1 ]
2: [ STD, 9, -1, 50 ]
3: [ LDD, 1, -1, 50 ]
4: [ LDD, 2, -1, 50 ]
5: [ LDI, 0, -1, 18 ]
6: [ JMPIL, 0, 2, -1 ]
7: [ SUBI, 2, -1, 1 ]
8: [ ADDI, 2, -1, 1 ]
9: [ SUBI, 2, -1, 1 ]
10: [ MULT, 1, 2, -1 ]
11: [ SUBI, 2, -1, 1 ]
12: [ JMPIGM, -1, 2, 8 ]
13: [ STD, 1, -1, 50 ]
14: [ LDI, 8, -1, 2 ]
15: [ LDI, 9, -1, 50 ]
---- Frame [1 ]
16: [ TRAP, -1, -1, -1 ]
17: [ STOP, -1, -1, -1 ]
18: [ LDI, 1, -1, -1 ]
19: [ STD, 1, -1, 50 ]
20: [ LDI, 8, -1, 2 ]
21: [ LDI, 9, -1, 50 ]
22: [ TRAP, -1, -1, -1 ]
23: [ STOP, -1, -1, -1 ]
24: [ ____, -1, -1, -1 ]
25: [ ____, -1, -1, -1 ]
26: [ ____, -1, -1, -1 ]
27: [ ____, -1, -1, -1 ]
28: [ ____, -1, -1, -1 ]
29: [ ____, -1, -1, -1 ]
30: [ ____, -1, -1, -1 ]
31: [ ____, -1, -1, -1 ]
---- Frame [4 ]
```

```
---- Frame [4 ]
64: [ ____, -1, -1, -1 ]
65: [ ____, -1, -1, -1 ]
66: [ ____, -1, -1, -1 ]
67: [ ____, -1, -1, -1 ]
68: [ ____, -1, -1, -1 ]
69: [ ____, -1, -1, -1 ]
70: [ ____, -1, -1, -1 ]
71: [ ____, -1, -1, -1 ]
72: [ ____, -1, -1, -1 ]
73: [ ____, -1, -1, -1 ]
74: [ ____, -1, -1, -1 ]
75: [ ____, -1, -1, -1 ]
76: [ ____, -1, -1, -1 ]
77: [ ____, -1, -1, -1 ]
78: [ ____, -1, -1, -1 ]
79: [ ____, -1, -1, -1 ]
---- Frame [5 ]
80: [ ____, -1, -1, -1 ]
81: [ ____, -1, -1, -1 ]
82: [ DATA, -1, -1, 120 ]
83: [ ____, -1, -1, -1 ]
84: [ ____, -1, -1, -1 ]
85: [ ____, -1, -1, -1 ]
86: [ ____, -1, -1, -1 ]
87: [ ____, -1, -1, -1 ]
88: [ ____, -1, -1, -1 ]
89: [ ____, -1, -1, -1 ]
90: [ ____, -1, -1, -1 ]
91: [ ____, -1, -1, -1 ]
92: [ ____, -1, -1, -1 ]
93: [ ____, -1, -1, -1 ]
94: [ ____, -1, -1, -1 ]
95: [ ____, -1, -1, -1 ]
```