

Integrantes do grupo

Henrique Barcellos Lima, Iago Raphael Abrao Victoria, João Gabriel Dourado Cervo

Implementação

De acordo com nossos testes, nosso programa implementa todas as características solicitadas sem nenhum tipo de restrição.

Testes

Para acessar os programas do Sistema, foi criado um método Console, onde o usuário pode acessar uma lista de opções(Criar processo, Dump, Desalocar processo...), após selecionar uma opção é possível passar os argumentos que serão usados para chamar o método selecionado anteriormente, por exemplo:

```
Seja bem vindo ao sistema!  
Selecione a operação desejada:  
    1 - Criar processo  
    2 - Dump processo  
    3 - Desaloca processo  
    4 - Dump Memória  
    5 - Executar processo  
    6 - TraceOn  
    7 - TraceOff  
    0 - Sair  
Operação:
```

Se a operação selecionada for a número 1, o usuário receberá mais uma lista de opções correspondentes a sua escolha:

```
Selecione uma ação:  
1- Fibonacci  
2- ProgMinimo  
3- Fatorial  
4- FatorialTrap  
5- FibonacciTrap  
6- Bubble Sort
```

Portanto, a partir dessa interface em forma de console é possível acessar todas as funcionalidades do sistema.

FUNCIONALIDADES

Método "Criar processo": Verifica se há espaço disponível na memória e se o processo cabe na partição, se for o caso, aloca o processo em memória, cria seu pcb e o adiciona na fila de processos prontos e então carrega o programa em memória física.

Dump processo: Recebe o id de um processo e retorna todos seus dados, como pcb, programCounter, registradores e partição alocada.

Desalocar processo: Recebe o id de um processo, o desaloca da memória, o remove da lista de processos e limpa sua partição referente à memória física.

Dump memória: Recebe números de início e fim, e então imprime o que há dentro desse intervalo de memória física.

Executar processo: Recebe um id de processo, configura o setContext da cpu(registradores, programCounter, tamanho da memória...), tira o processo da lista de "prontos" e o adiciona na lista de processos em execução, e então executa a cpu.

TraceOn e TraceOff: São os métodos de debug, que o ativam e desativam, respectivamente.

Execução

Abaixo estão alguns exemplos de programas executando na máquina virtual com o TraceOn

- Fibonacci

```
pc: 0      exec: [ LDI, 1, -1, 0 ]
pc: 1      exec: [ STD, 1, -1, 20 ]
pc: 2      exec: [ LDI, 2, -1, 1 ]
pc: 3      exec: [ STD, 2, -1, 21 ]
pc: 4      exec: [ LDI, 0, -1, 22 ]
pc: 5      exec: [ LDI, 6, -1, 6 ]
pc: 6      exec: [ LDI, 7, -1, 31 ]
pc: 7      exec: [ LDI, 3, -1, 0 ]
pc: 8      exec: [ ADD, 3, 1, -1 ]
pc: 9      exec: [ LDI, 1, -1, 0 ]
pc: 10     exec: [ ADD, 1, 2, -1 ]
pc: 11     exec: [ ADD, 2, 3, -1 ]
pc: 12     exec: [ STX, 0, 2, -1 ]
pc: 13     exec: [ ADDI, 0, -1, 1 ]
pc: 14     exec: [ SUB, 7, 0, -1 ]
pc: 15     exec: [ JMPIG, 6, 7, -1 ]
pc: 6      exec: [ LDI, 7, -1, 31 ]
pc: 7      exec: [ LDI, 3, -1, 0 ]
pc: 8      exec: [ ADD, 3, 1, -1 ]
pc: 9      exec: [ LDI, 1, -1, 0 ]
pc: 10     exec: [ ADD, 1, 2, -1 ]
pc: 11     exec: [ ADD, 2, 3, -1 ]
pc: 12     exec: [ STX, 0, 2, -1 ]
pc: 13     exec: [ ADDI, 0, -1, 1 ]
pc: 14     exec: [ SUB, 7, 0, -1 ]
pc: 15     exec: [ JMPIG, 6, 7, -1 ]
pc: 6      exec: [ LDI, 7, -1, 31 ]
pc: 7      exec: [ LDI, 3, -1, 0 ]
pc: 8      exec: [ ADD, 3, 1, -1 ]
pc: 9      exec: [ LDI, 1, -1, 0 ]
pc: 10     exec: [ ADD, 1, 2, -1 ]
pc: 11     exec: [ ADD, 2, 3, -1 ]
```

```
pc: 11      exec: [ ADD, 2, 3, -1 ]
pc: 12      exec: [ STX, 0, 2, -1 ]
pc: 13      exec: [ ADDI, 0, -1, 1 ]
pc: 14      exec: [ SUB, 7, 0, -1 ]
pc: 15      exec: [ JMPIG, 6, 7, -1 ]
pc: 6       exec: [ LDI, 7, -1, 31 ]
pc: 7       exec: [ LDI, 3, -1, 0 ]
pc: 8       exec: [ ADD, 3, 1, -1 ]
pc: 9       exec: [ LDI, 1, -1, 0 ]
pc: 10      exec: [ ADD, 1, 2, -1 ]
pc: 11      exec: [ ADD, 2, 3, -1 ]
pc: 12      exec: [ STX, 0, 2, -1 ]
pc: 13      exec: [ ADDI, 0, -1, 1 ]
pc: 14      exec: [ SUB, 7, 0, -1 ]
pc: 15      exec: [ JMPIG, 6, 7, -1 ]
pc: 6       exec: [ LDI, 7, -1, 31 ]
pc: 7       exec: [ LDI, 3, -1, 0 ]
pc: 8       exec: [ ADD, 3, 1, -1 ]
pc: 9       exec: [ LDI, 1, -1, 0 ]
pc: 10      exec: [ ADD, 1, 2, -1 ]
pc: 11      exec: [ ADD, 2, 3, -1 ]
pc: 12      exec: [ STX, 0, 2, -1 ]
pc: 13      exec: [ ADDI, 0, -1, 1 ]
pc: 14      exec: [ SUB, 7, 0, -1 ]
pc: 15      exec: [ JMPIG, 6, 7, -1 ]
pc: 6       exec: [ LDI, 7, -1, 31 ]
pc: 7       exec: [ LDI, 3, -1, 0 ]
pc: 8       exec: [ ADD, 3, 1, -1 ]
pc: 9       exec: [ LDI, 1, -1, 0 ]
pc: 10      exec: [ ADD, 1, 2, -1 ]
pc: 11      exec: [ ADD, 2, 3, -1 ]
pc: 12      exec: [ STX, 0, 2, -1 ]
pc: 13      exec: [ ADDI, 0, -1, 1 ]
pc: 13      exec: [ ADDI, 0, -1, 1 ]
```

```

pc: 14      exec: [ SUB, 7, 0, -1 ]
pc: 15      exec: [ JMPIG, 6, 7, -1 ]
pc: 6       exec: [ LDI, 7, -1, 31 ]
pc: 7       exec: [ LDI, 3, -1, 0 ]
pc: 8       exec: [ ADD, 3, 1, -1 ]
pc: 9       exec: [ LDI, 1, -1, 0 ]
pc: 10      exec: [ ADD, 1, 2, -1 ]
pc: 11      exec: [ ADD, 2, 3, -1 ]
pc: 12      exec: [ STX, 0, 2, -1 ]
pc: 13      exec: [ ADDI, 0, -1, 1 ]
pc: 14      exec: [ SUB, 7, 0, -1 ]
pc: 15      exec: [ JMPIG, 6, 7, -1 ]
pc: 6       exec: [ LDI, 7, -1, 31 ]
pc: 7       exec: [ LDI, 3, -1, 0 ]
pc: 8       exec: [ ADD, 3, 1, -1 ]
pc: 9       exec: [ LDI, 1, -1, 0 ]
pc: 10      exec: [ ADD, 1, 2, -1 ]
pc: 11      exec: [ ADD, 2, 3, -1 ]
pc: 12      exec: [ STX, 0, 2, -1 ]
pc: 13      exec: [ ADDI, 0, -1, 1 ]
pc: 14      exec: [ SUB, 7, 0, -1 ]
pc: 15      exec: [ JMPIG, 6, 7, -1 ]
pc: 16      exec: [ STOP, -1, -1, -1 ]
                    Interrupcao intSTOP      pc: 16
Interrupcao: O programa chegou ao fim

```

- ProgMinimo

```

pc: 0       exec: [ LDI, 0, -1, 999 ]
pc: 1       exec: [ STD, 0, -1, 10 ]
pc: 2       exec: [ STD, 0, -1, 11 ]
pc: 3       exec: [ STD, 0, -1, 12 ]
pc: 4       exec: [ STD, 0, -1, 13 ]
pc: 5       exec: [ STD, 0, -1, 14 ]
pc: 6       exec: [ STOP, -1, -1, -1 ]
                    Interrupcao intSTOP      pc: 6
Interrupcao: O programa chegou ao fim

```

- Fatorial

```
pc: 60      exec: [ LDI, 0, -1, 4 ]
pc: 61      exec: [ LDI, 1, -1, 1 ]
pc: 62      exec: [ LDI, 6, -1, 1 ]
pc: 63      exec: [ LDI, 7, -1, 8 ]
pc: 64      exec: [ JMPIE, 7, 0, 0 ]
pc: 65      exec: [ MULT, 1, 0, -1 ]
pc: 66      exec: [ SUB, 0, 6, -1 ]
pc: 67      exec: [ JMP, -1, -1, 4 ]
pc: 64      exec: [ JMPIE, 7, 0, 0 ]
pc: 65      exec: [ MULT, 1, 0, -1 ]
pc: 66      exec: [ SUB, 0, 6, -1 ]
pc: 67      exec: [ JMP, -1, -1, 4 ]
pc: 64      exec: [ JMPIE, 7, 0, 0 ]
pc: 65      exec: [ MULT, 1, 0, -1 ]
pc: 66      exec: [ SUB, 0, 6, -1 ]
pc: 67      exec: [ JMP, -1, -1, 4 ]
pc: 64      exec: [ JMPIE, 7, 0, 0 ]
pc: 68      exec: [ STD, 1, -1, 10 ]
pc: 69      exec: [ STOP, -1, -1, -1 ]
                Interrupcao intSTOP    pc: 69
```

Interrupcao: O programa chegou ao fim