

Henrique Mrás e Mateus Rosa

Implementação

Nosso sistema atual se baseia no T1. Da primeira versão para a atual realizamos algumas mudanças, onde havia o “Gerente Geral”, temos agora “Monitor”. Pontos adicionados, “dec_interrupcao” serve para as interrupções do código onde antes era comandado pela classe CPU nas opções de interrupção. Foi desenvolvido os métodos “depuracao_IO” e “chamada_sis”, onde é feita a inserção de um processo que estava na lista de bloqueados para a fila de prontos caso houver tratamento de IO anteriormente, e a adição de um processo a lista de bloqueados caso tenha uma chamada de sistema respectivamente. A classe Escalonador realiza a troca de processos que estão rodando na CPU, todo o Escalonador representa uma seção crítica em relação as filas de processamento. A classe console é responsável pelo IO, e semáforo para o controle da lista de bloqueados e console. Também houve a retirada do método “legal” e “testOverflow”, e foi realizado testes manuais para não acarretar estouro de memória ou acessos indevidos.

```
100 } try(Thread.sleep(1000)); // usado para que possamos ver a execução, se quiser que vá mais rápida só comentar
```

Nessa parte do código pode ser alterado o tempo de processamento para visualização melhor dos dados.

Funcionamento

De maneira a deixar mais rápido e fácil a execução de testes foi adicionada no menu que adiciona todos os programas na lista de prontos. Após a criação de todos os processos, podemos ver que os programas são executados em segundo plano, o menu continua disponível para realizarmos outras operações, como ver os processos na lista de processos, executando e bloqueados.

```
Selecione a operacao desejada:
1 - Criar processo
2 - Dump processo
3 - Dump memoria
4 - Executar processo
5 - Desalocar processo
6 - Listar processos
7 - Inserir dado TRAP
8 - Inserir todos os Processos
0 - Sair
Operacao: 1
```

```
*****
Selecione a operacao desejada:
1 - Criar processo
2 - Dump processo
3 - Dump memoria
4 - Executar processo
5 - Desalocar processo
6 - Listar processos
7 - Inserir dado TRAP
8 - Inserir todos os Processos
0 - Sair
Operacao: 2
Interrupção detectada: Endereço inválido
Interrupção detectada: Endereço inválido
Interrupção detectada: Endereço inválido
Trap Encerrado
Interrupção detectada: Fim de programa
Interrupção detectada: Endereço inválido
Interrupção detectada: Fim de programa
Interrupção detectada: Fim de programa
```

```
Operacao: 8
frames necessarios: 1, frames livres: 64
frame utilizado: 0
Processo criado com sucesso, ID: 0
frames necessarios: 1, frames livres: 63
frame utilizado: 1
Processo criado com sucesso, ID: 1
frames necessarios: 2, frames livres: 62
frame utilizado: 2
frame utilizado: 3
Processo criado com sucesso, ID: 2
frames necessarios: 2, frames livres: 60
frame utilizado: 4
frame utilizado: 5
Processo criado com sucesso, ID: 3
frames necessarios: 4, frames livres: 58
frame utilizado: 6
frame utilizado: 7
frame utilizado: 8
frame utilizado: 9
Processo criado com sucesso, ID: 8
frames necessarios: 1, frames livres: 57
frame utilizado: 17
Processo criado com sucesso, ID: 9
*****
```

```
Selecione a operacao desejada:
1 - Criar processo
2 - Dump processo
3 - Dump memoria
4 - Executar processo
5 - Desalocar processo
6 - Listar processos
7 - Inserir dado TRAP
8 - Inserir todos os Processos
0 - Sair
Operacao: 2
Processos:
ID: 12
ID: 13
ID: 14
ID: 15
ID: 16
ID: 17
ID: 18
ID: 19
ID: 2
ID: 10
Processos em execucao:
ID: 11
Processos bloqueados:
ID processo para dump: 1
```