

## Atividade 2

### Gerência de Processador

1. Laboratório com o Simulador SOsim: Conforme visto em sala, o Simulador SOsim auxiliará na demonstração de muitas teorias de Sistemas Operacionais. Execute os passos presentes no arquivo GSI519 Aula 04 lab 02 processador.pdf que tratará apenas sobre a Gerência do Processador e responda as questões.

#### Atividade 1 – Escalonamento Circular

##### a) Práticas de Simulação

Configurado o SOsim para escalonamento circular.

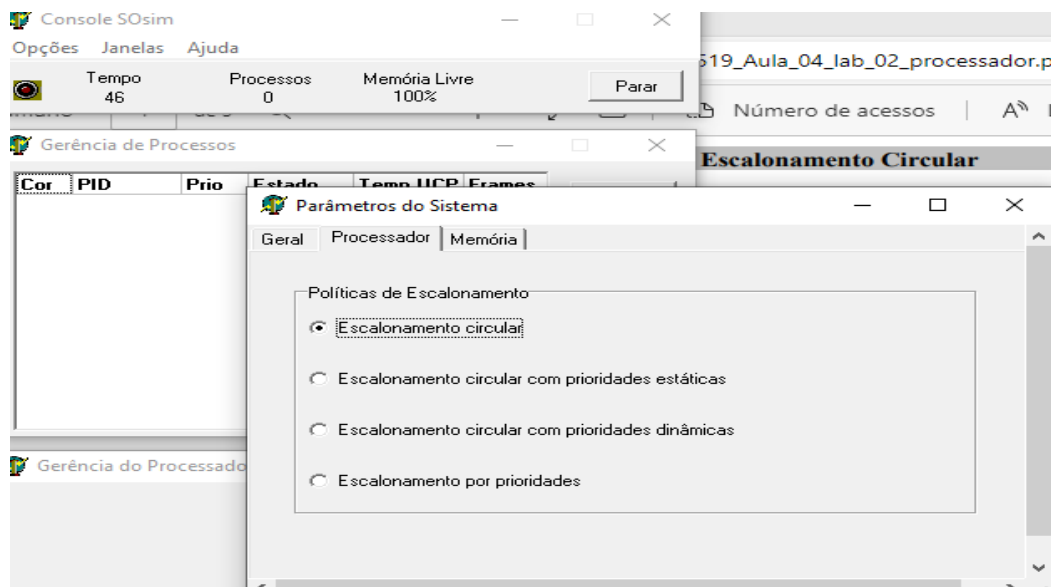


Figura 1: Configurado escalonamento circular.

##### b) Análise Prática

Após 2 minutos, Tempo Processador IO-BOUND : 13s

Após 2 minutos, Tempo Processador CPU-BOUND: 73s

Cor	PID	Prio	Estado	Temp UCP	Frames
Verde	1714594	0	Pronto	73	5
Vermelho	1721936	0	I/O	13	5

Figura 2: Tempo processador dos processos.

Após nova análise, os tempos ficaram: Tempo processo IO = 8s  
Tempo processo CPUBOUND = 86s.

### c) Questão teórica para responder com a ajuda do simulador

Considere a concorrência, nesse tipo de escalonamento, com dois processo CPU-bound que não realizam operações de E/S. Qual o efeito da variação da fatia de tempo sobre o balanceamento no uso do processador?

Resposta =



Cor	PID	Prio	Estado	Temp UCP	Frames
	2931931	0	Execução	11	5
	2934494	0	Pronto	10	5

Figura 3: Tempos processos CPUBOUND.

Os dois processos utilizam de forma igual a UCP, possuindo a mesma prioridade e sendo do mesmo tipo, o efeito é que quanto menor este tempo na fatia de tempo, maior será escalonamento entre estes processos e quanto maior a fatia menor o escalonamento. No exemplo do print acima, na maior fatia de tempo, o processo de cima vai de 0 a 10 e depois o segundo processo vai de 0 a 10, e assim volta para o primeiro que vai de 11s a até 20 e depois volta para o segundo processo, tornando balanceado e idêntico os intervalos. Ou seja, com o aumento de tempo de fatia também há o aumento de tempo total de CPU, que vai de 0 a 10 no maximo de tempo de fatia para cada processo.

### Atividade 2 – Escalonamento Circular com Prioridades Estáticas I

#### a) Práticas de Simulação

Configurado o SOsim para escalonamento circular com propriedades estáticas.

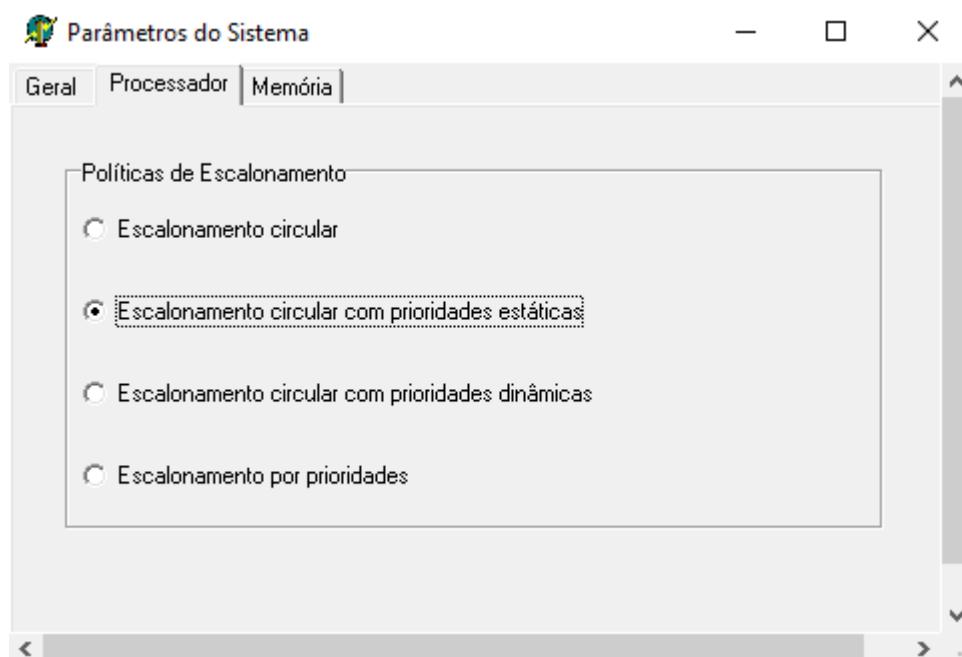


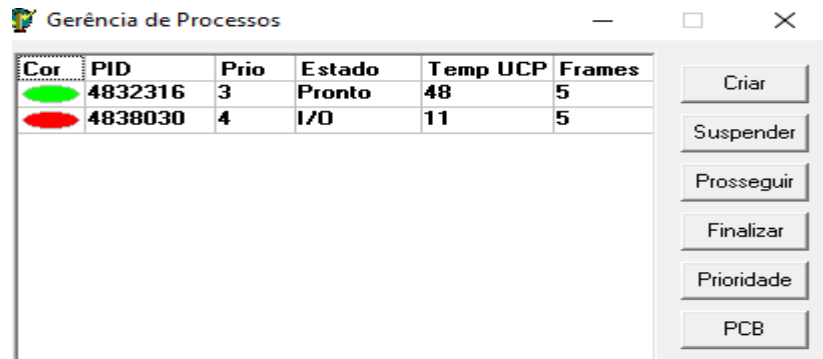
Figura 4: Configurado escalonamento circular propriedades

estáticas.

### b) Análise Prática

Após 2 minutos, Tempo Processador do processo IO-BOUND : 11s

Após 2 minutos, Tempo Processador do processo CPU-BOUND: 48s



Cor	PID	Prio	Estado	Temp UCP	Frames
Verde	4832316	3	Pronto	48	5
Vermelho	4838030	4	I/O	11	5

Buttons: Criar, Suspende, Prosseguir, Finalizar, Prioridade, PCB

Figura 5: Tempo processador dos processos com propriedades estáticas.

### c) Questão teórica para responder com a ajuda do simulador

Quais devem ser os critérios para determinar as prioridades dos processos?

Caso, nesse escalonamento, todos os processos sejam criados com a mesma prioridade, qual o benefício dessa política sobre o Escalonamento Circular?

**Resposta** = Os critérios são a determinação das prioridades para os processos, a importância destes processos, a melhor otimização e maximização do desempenho no sistema, performance do sistema, melhor aproveitamento do processador.

Já sobre o benefício, não há nenhum, pois sobre essa política, o comportamento é parecido, na estática não é modificada durante a existência do processo.

### Atividade 3 – Escalonamento Circular com Prioridades Estáticas II

#### a) Práticas de Simulação

Configurado o SOsim para escalonamento circular com propriedades estáticas.

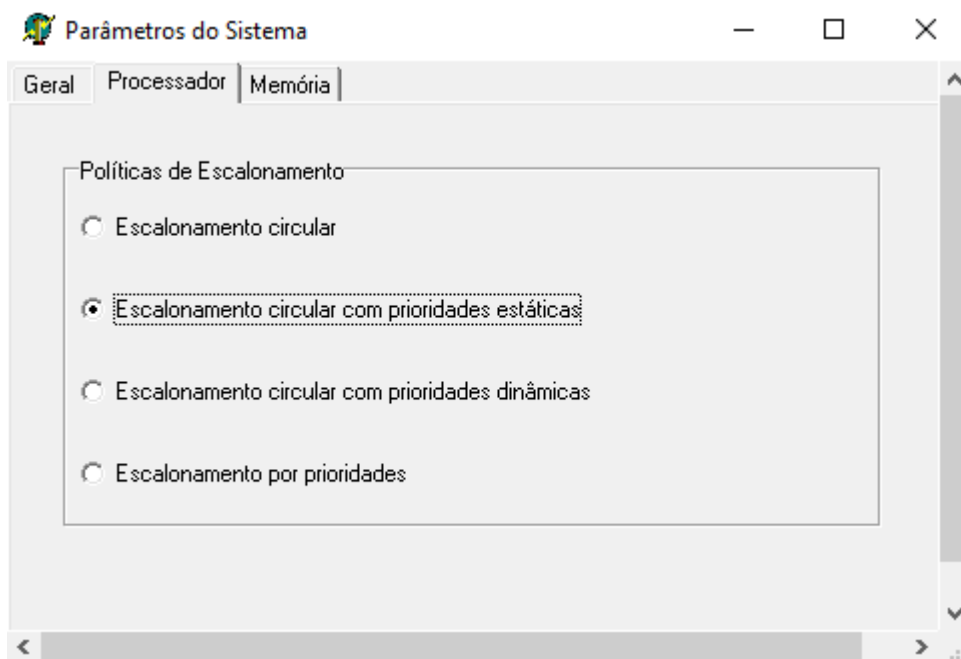


Figura 6: Configurado escalonamento circular propriedades estáticas, atividade 3

## b) Análise Prática

Criado os processos CPU-BOUND com prioridade 4 e IO-BOUND com prioridade 3. Segue o escalonamento e comportamento dos processos na imagem abaixo.

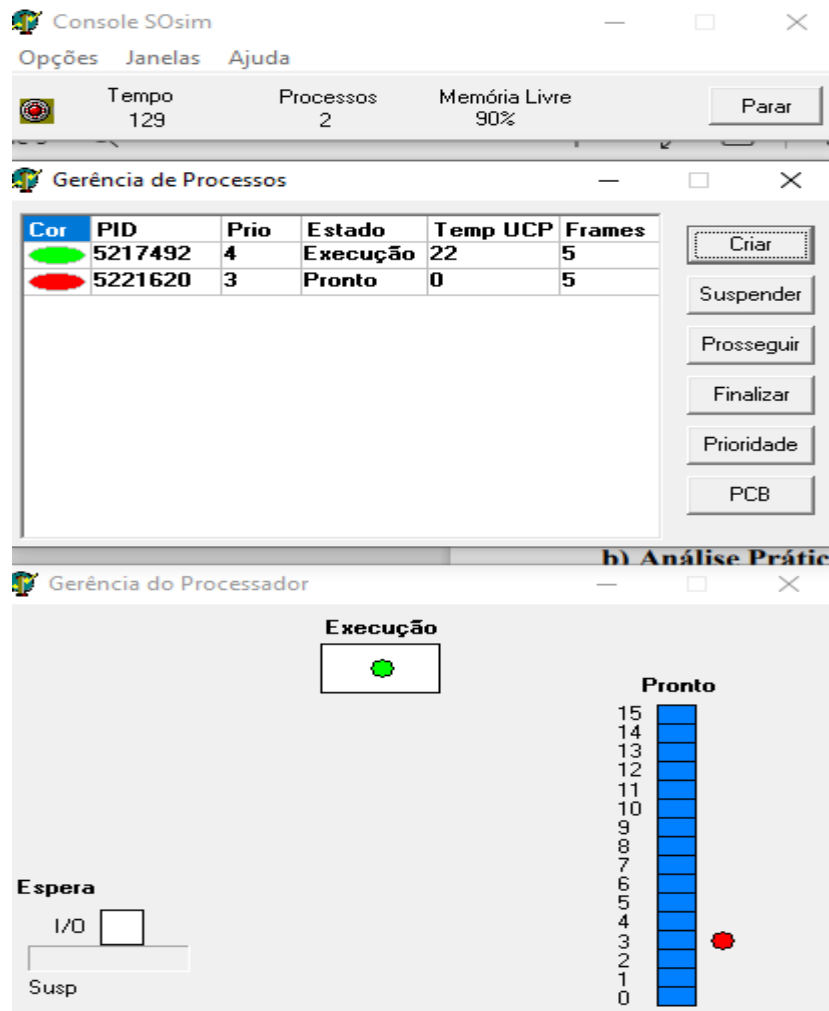


Figura 7: Problema Starvation

Apenas o processo de CPU-BOUND é executado, ele tem prioridade 3 e o de IO prioridade 4 ocorrendo o starvation, tempo de cpu total do processo CPUBOUND 22s na imagem enquanto de IO está zerado.

## c) Questão teórica para responder com a ajuda do simulador

Por que o problema do starvation pode ocorrer?

Cite duas ações que o administrador do sistema pode realizar quando é identificada a situação de starvation em um processo.

**Resposta =** O problema de starvation acontece quando um ou mais processos não conseguem obter recursos do sistema, não havendo continuidade, por exemplo neste caso, o processo que tem a menor prioridade pode acabar não sendo executado ou nunca, pois o outro processo tem a maior prioridade. Exemplo, o processo CPU-BOUND foi sempre executado com prioridade 4 e o IO-BOUND não foi executado nenhuma.

Sobre duas ações que o administrador pode realizar, pode ser a alteração da prioridade do processo que deve ser executado ou suspender a execução do processo que está está

sendo executado para que o outro que está em starvation, possa entrar para a UCP e ser executado, no caso da prática, suspender o CPUBOUND para que o IOBOUND possa ser executado.

#### Atividade 4 – Escalonamento Circular com Prioridades Dinâmica

Configurado o SOsim para escalonamento circular com propriedades dinâmicas.

Foi ligado as caixas de log e estatísticas e aumentado a barra de clock como pedido.

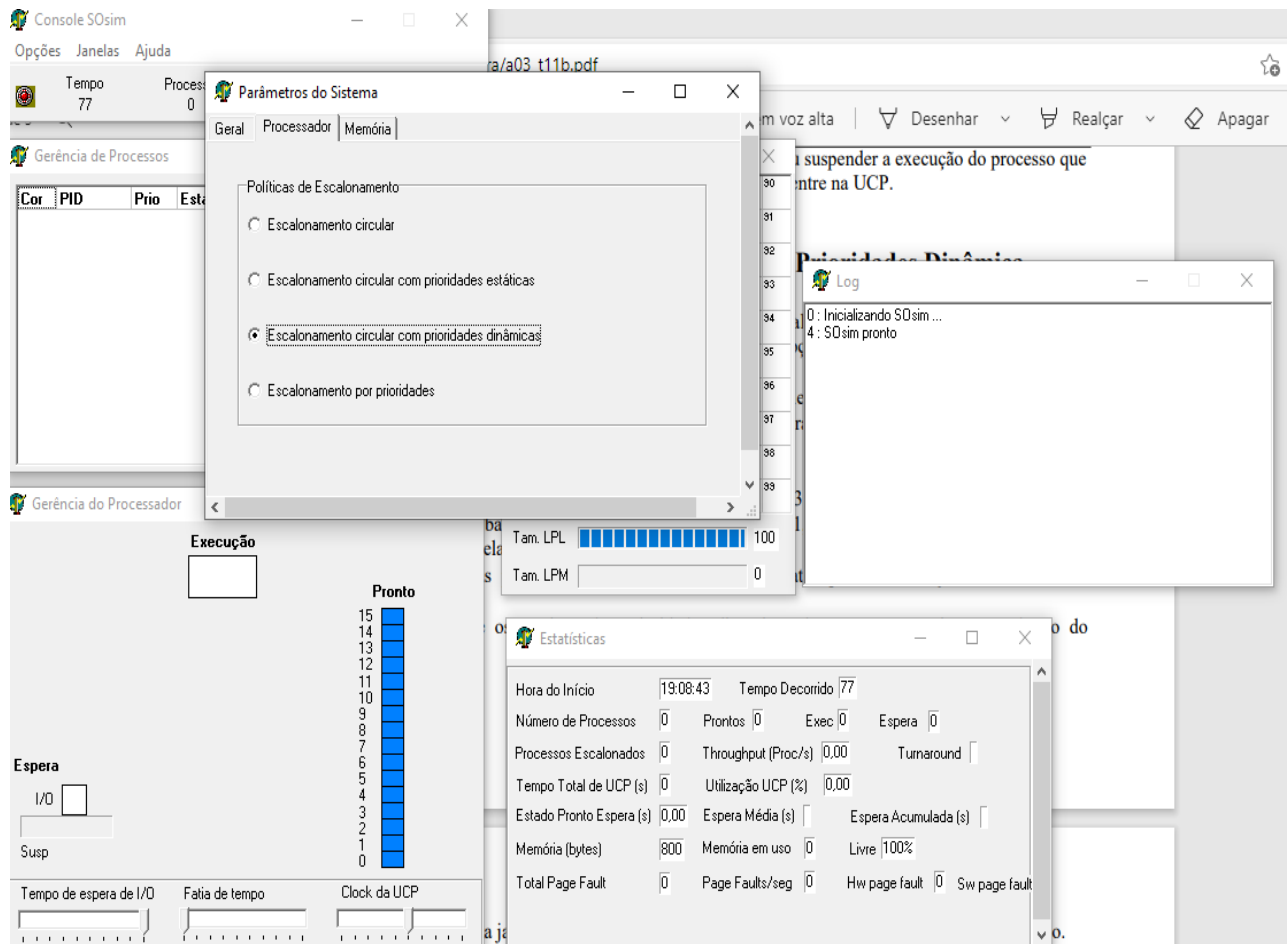


Figura 8: Configurado escalonamento circular propriedades dinâmicas.

#### b) Análise Prática

Foi criado os processos CPUBOUND com prioridade 3 e os 3 processos IOBOUND com prioridade 4, foi observado que o tempo do CPUBOUND registrou 82s enquanto os outros 3 processos IOBOUND registraram 4s. O percentual da utilização da UCP ficou em torno de 50%. Segue a imagem abaixo com os processos criados.

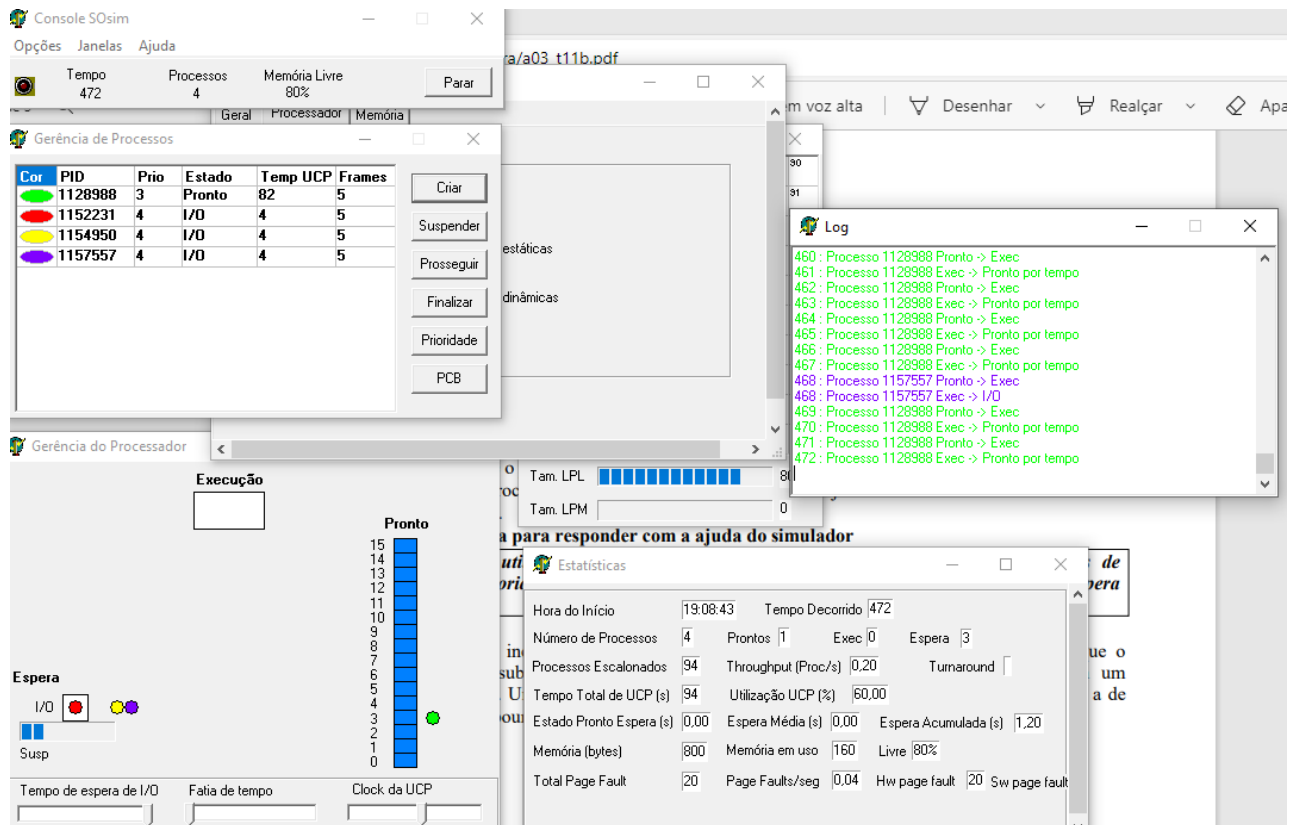


Figura 9: Processo CPUBOUND com prioridade 3 e os 3 processos IOBOUND com prioridade 4.

Com a suspensão do processo CPUBOUND, a taxa de utilização da CPU caiu para 0 a 10% em alguns casos. Segue o print da prática realizada.

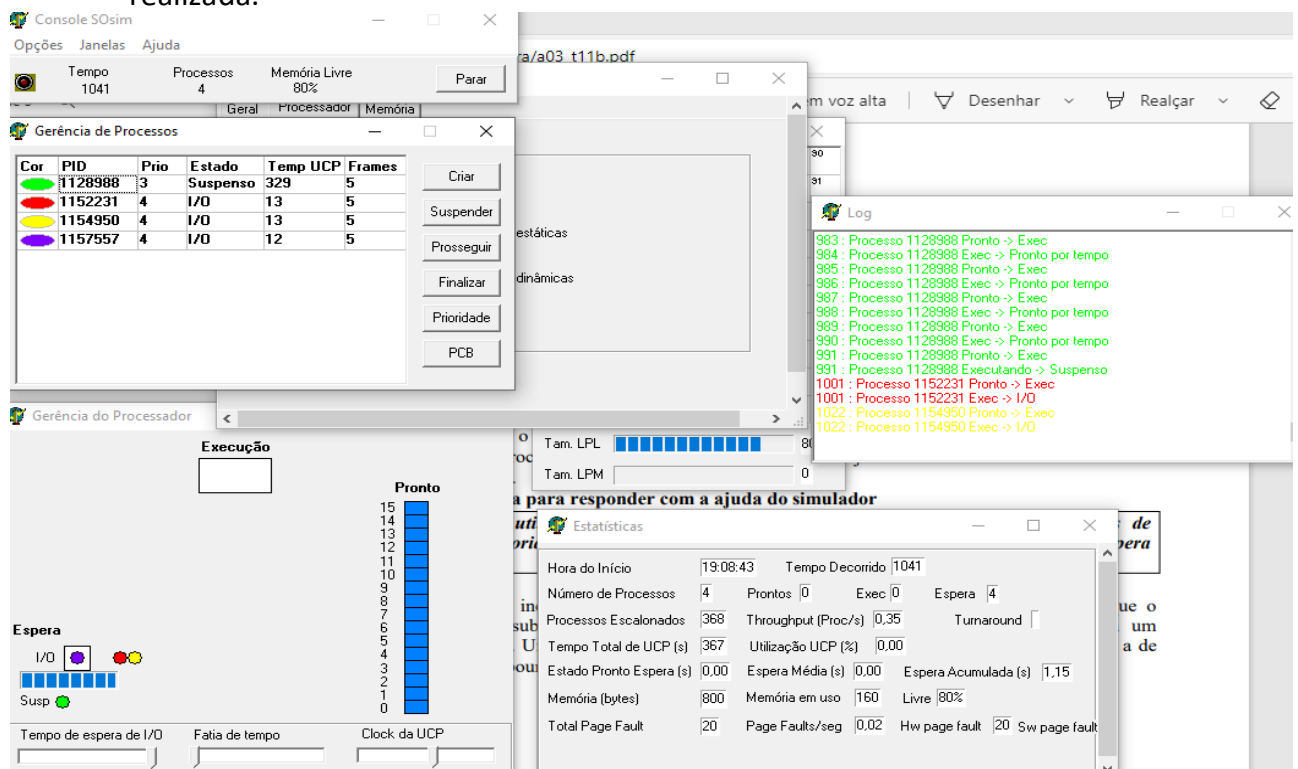


Figura 10: Processo CPUBOUND suspenso como pedido na prática.

### c) Questão teórica para responder com a ajuda do simulador

Qual o critério utilizado pelo sistema operacional para determinar diferentes valores de incremento à prioridade base de um processo quando há uma mudança

do estado de espera para pronto?

**Resposta =** O critério agora depende de cada processo incrementado na ordem da fila, os processos de IO-BOUND possuem a mesma prioridade 4, sendo assim o processo de CPU-BOUND irá ter uma prioridade dinâmica menor que a do processo de IO-BOUND. Sendo no geral, o processo acontece de acordo com a ordem de espera na fila, que no caso foi sendo incrementada em 1.