

# **Processos & Threads**

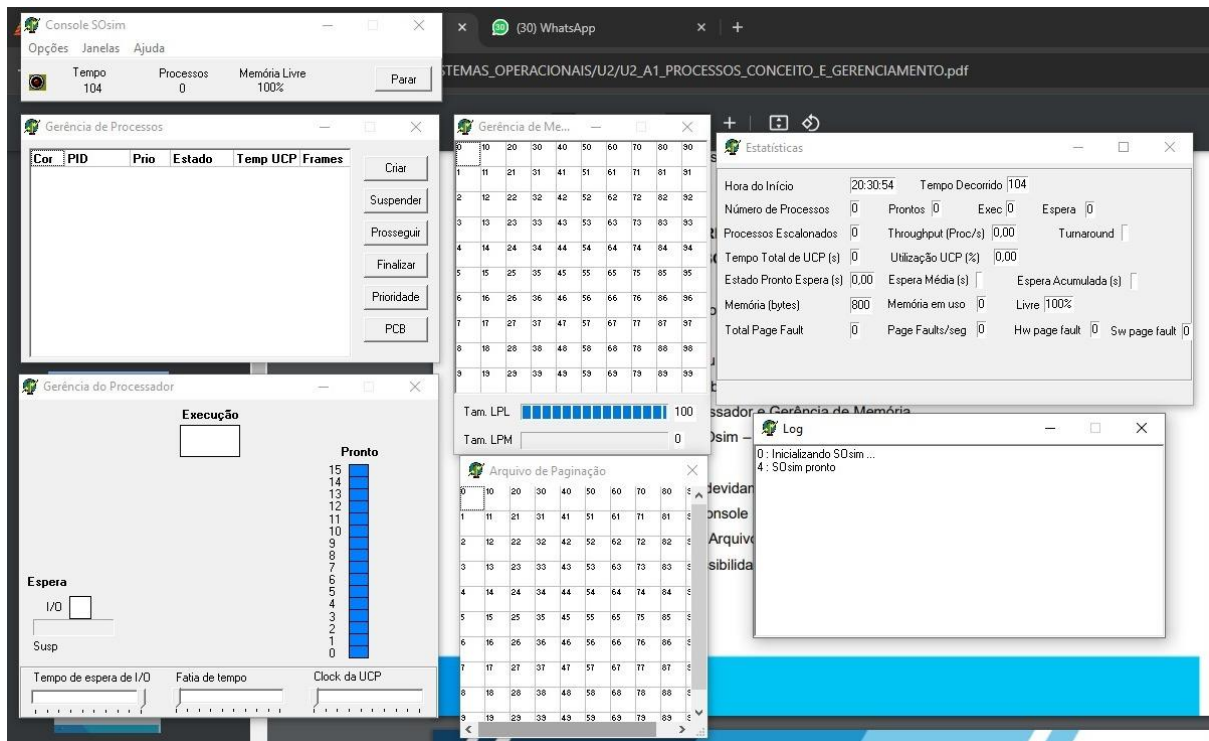
Faculdade Anhanguera

Sistemas Operacionais

Vitor Gabriel Martins Borges

Engenharia de Softwares

## Atividade 1 - Conhecendo o SOsim



## Atividade 2 - Criação de processos

a. Crie um processo tipo CPU-bound: janela Gerência de Processos / Criar.

The screenshot displays the SOSim simulator interface with several windows open. The 'Gerência de Processos' window is the primary focus, showing a table of process information:

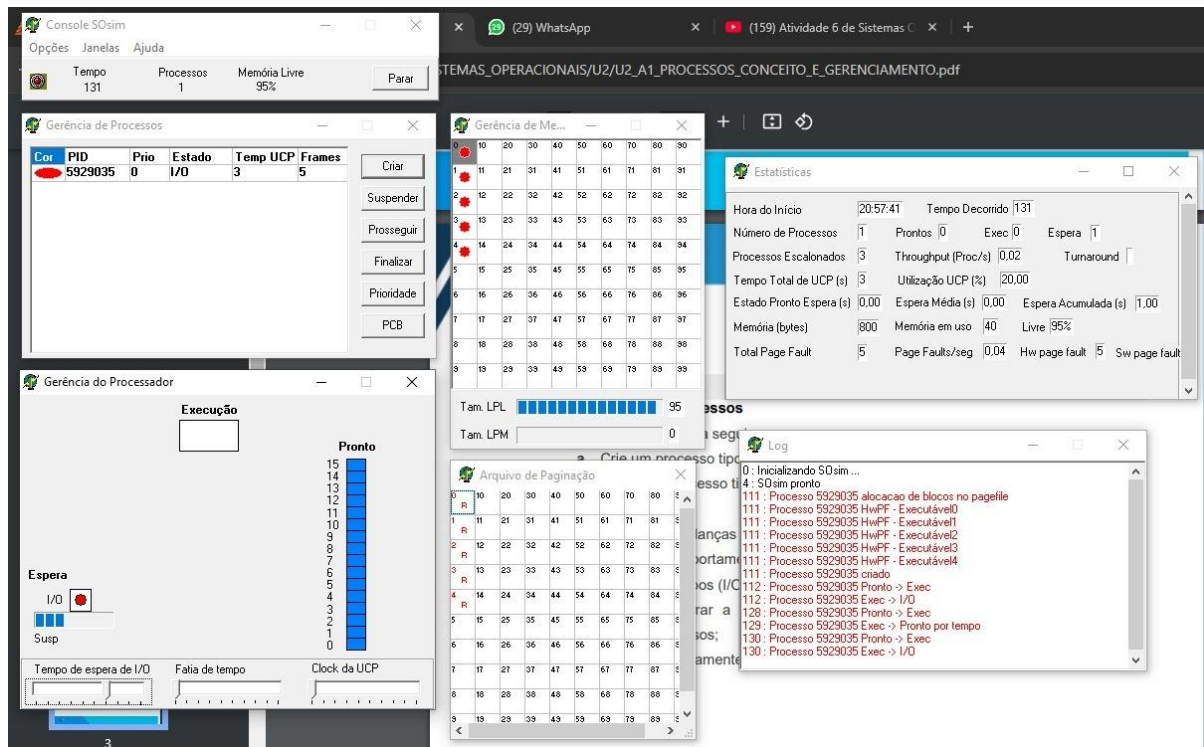
Cor	PID	Prio	Estado	Temp UCP	Frames
Verde	4749512	0	Pronto	10	5

Buttons for 'Criar', 'Suspender', 'Prosseguir', 'Finalizar', 'Prioridade', and 'PCB' are visible. Below this, the 'Gerência do Processador' window shows a bar chart for 'Pronto' and 'Espera' states, with a 'Clock da UCP' at the bottom. The 'Log' window on the right lists process events, such as 'Processo 4749512 Pronto -> Exec' and 'Processo 4749512 Exec -> Pronto por tempo'. The 'Estatísticas' window at the bottom right provides a summary of system metrics:

Hora do Início		Tempo Decorrido	
Número de Processos	1	Prontos	1
Processos Escalonados	10	Exec	0
Tempo Total de UCP (s)	10	Espera	0
Estado Pronto	Espera (s)	Throughput (Proc/s)	0,06
0,00	0,00	Utilização UCP (%)	40,00
Memória (bytes)	800	Espera Média (s)	0,00
Total Page Fault	5	Espera Acumulada (s)	1,00
		Memória em uso	40
		Libre	95%
		Page Faults/seg	0,03
		Hw page fault	5
		Sw page fault	0

The bottom of the screen shows the title 'Atividade 6 de Sistemas Operacionais || - Prática do Simulador SOSim' and a footer with 'Sistemas Operacionais' and 'Me Salva Sistemas Operacionais'.

b. Crie outro processo tipo I/O-bound: janela Gerência de Processos / Criar.



c. Analise:

## 1. Mudanças de estado dos dois processos

O processo **CPU-bound** passa mais tempo no estado "**Executando**", pois usa intensamente a CPU. Ele muda de estado apenas quando é interrompido pelo escalonador ou finaliza sua execução.

O processo **I/O-bound** passa mais tempo no estado "**Espera**", aguardando operações de entrada/saída serem concluídas. Ele executa pequenas frações de CPU antes de voltar à espera.

## 2. Comportamento dos processos e mudanças de contexto

O processo **CPU-bound** sofre menos mudanças de contexto quando usa escalonadores como **FIFO** ou Prioridade, pois tende a executar por longos períodos. No **Round Robin**, ele é frequentemente interrompido.

O processo **I/O-bound** gera mais mudanças de contexto porque executa breves períodos de CPU antes de aguardar **I/O**, liberando a CPU para outros processos.

### **3. Comparação da taxa de crescimento do tempo de processador**

O tempo de processador cresce mais rapidamente para o processo **CPU-bound**, pois ele usa a CPU de forma contínua.

O processo **I/O-bound** tem um crescimento mais lento, pois gasta mais tempo esperando operações de **I/O** do que utilizando a CPU.

### **Observações gerais**

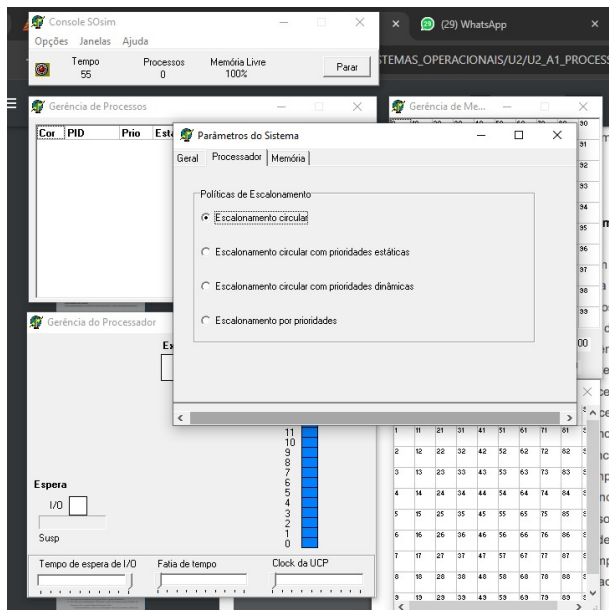
Em sistemas multitarefa, processos **I/O-bound** geralmente têm melhor resposta, pois liberam rapidamente a CPU para outros processos.

O desempenho de um processo **CPU-bound** pode ser afetado dependendo do escalonador escolhido. No **Round Robin**, por exemplo, ele pode sofrer mais interrupções, impactando sua eficiência.

Se o sistema tiver muitos processos **CPU-bound**, a fila de espera da CPU pode crescer, aumentando o tempo de resposta para todos os processos.

## **Atividade 3 - Trabalho com tempo de processador**

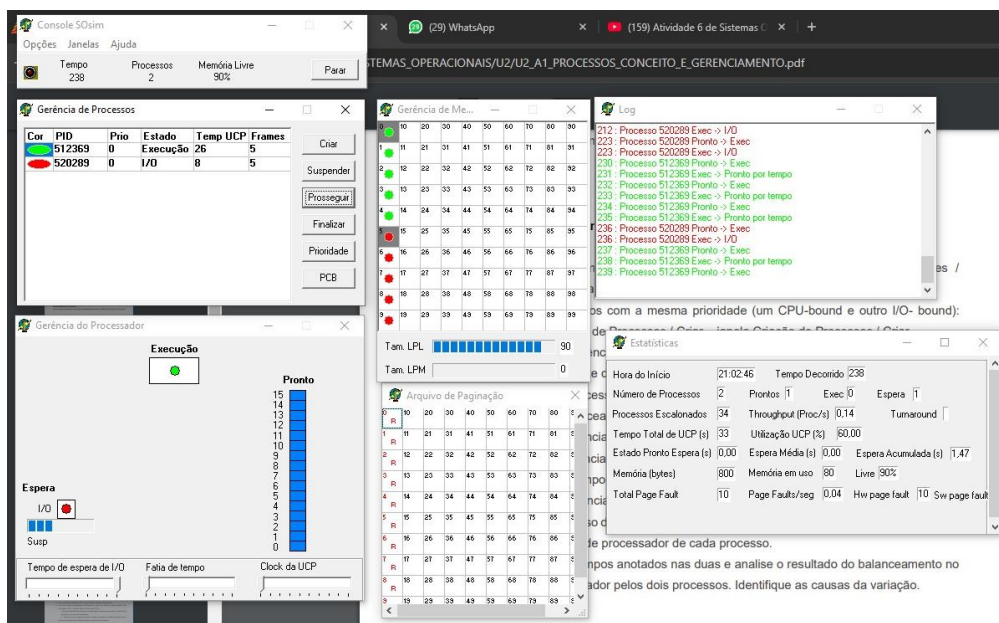
Configure para trabalhar com Escalonamento Circular: janela console SOsim / Opções / Parâmetros do Sistema na guia Processador



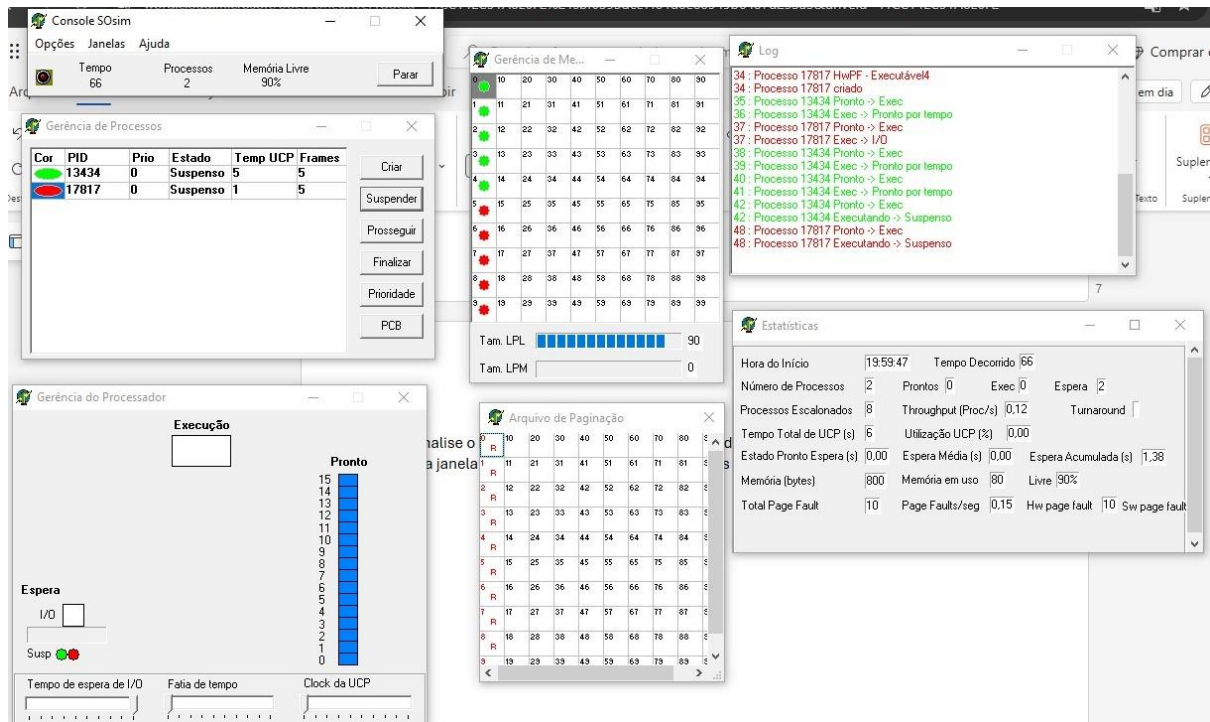
**a.** Criar 2 processos com a mesma prioridade (um CPU-bound e outro I/O- bound):  
janela Gerência de Processos / Criar – janela Criação de Processos / Criar.

**b.** Na janela Gerência de Processos, observe o tempo de processador de cada processo durante dois minutos e as mudanças de estado. Após esse período anote o tempo de processador de cada processo

**c.** Analise o balanceamento no uso do processador pelos dois processos.



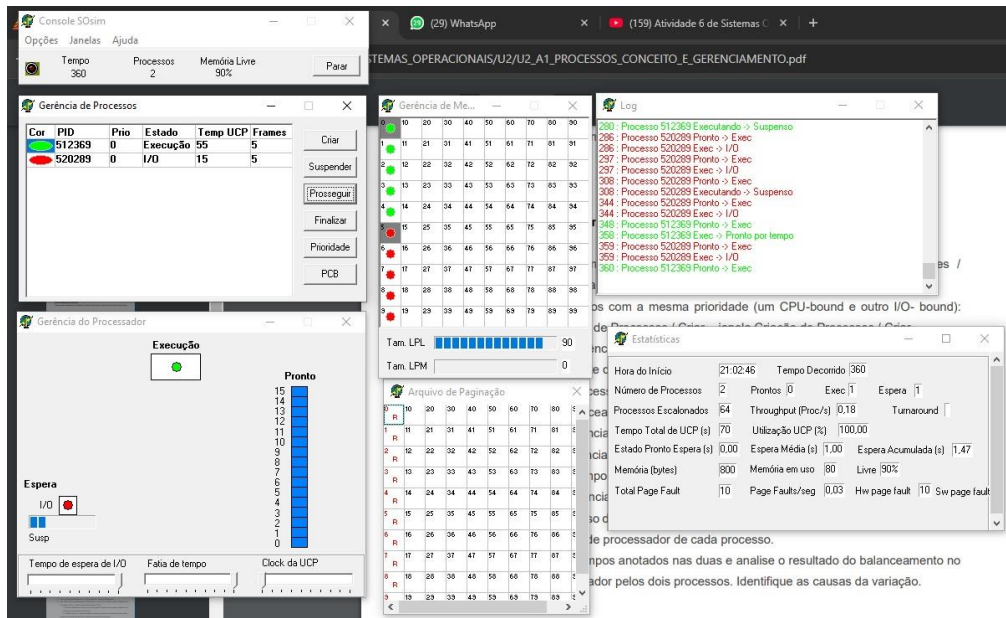
d. Na janela Gerência de Processos suspenda os dois processos.





e. Na janela Gerência de Processador, aumente a fatia de tempo movimentando a barra de Fatia de Tempo.

f. Na janela Gerência de Processos, observe mais uma vez o tempo de processador de cada processo durante dois minutos e as mudanças de estado. Após esse período anote o tempo de processador de cada processo.



g. Compare os tempos anotados nas duas e analise o resultado do balanceamento no uso do processador pelos dois processos. Identifique as causas da variação.

## 1. Comparação dos Tempos

- Na **primeira imagem (238s)**:
  - Tempo decorrido: **238s**
  - Processos escalonados: **34**
  - Uso da CPU: **60%**
  - Processo **512369** (CPU-bound): **Executando - 26s**
  - Processo **520289** (I/O-bound): **Executando - 8s**
- Na **segunda imagem (360s)**:
  - Tempo decorrido: **360s**
  - Processos escalonados: **64**
  - Uso da CPU: **100%**

- Processo **512369: Executando - 55s**
- Processo **520289: Executando - 15s**

Isso mostra que, à medida que o tempo avança, o **uso da CPU aumenta de 60% para 100%**, indicando uma maior eficiência no escalonamento ou que menos tempo foi desperdiçado esperando I/O.

## **2. Análise do Balanceamento**

- Inicialmente (aos 238s), a CPU não estava sendo usada **100% do tempo**, possivelmente porque o processo **520289** passava muito tempo em I/O.
- Depois (aos 360s), o número de processos escalonados aumentou e a **CPU chegou a 100% de utilização**.
- Isso sugere que o escalonamento ajustou melhor a alternância entre os processos, evitando desperdício de CPU.

## **3. Causas da Variação**

### **1. Evolução no uso da CPU:**

- a. No início (238s), a CPU estava sendo utilizada apenas **60% do tempo**, provavelmente porque um dos processos estava frequentemente em I/O.
- b. Com o passar do tempo (360s), a CPU atinge **100% de uso**, indicando que os processos passaram a ser melhor distribuídos.

### **2. Diferença no tipo de processos:**

- a. O processo **512369** continua dominante, pois é **CPU-bound**.
- b. O processo **520289**, sendo **I/O-bound**, depende de operações externas e pode ter causado períodos de ociosidade inicial.

### **3. Fatia de Tempo e Eficiência:**

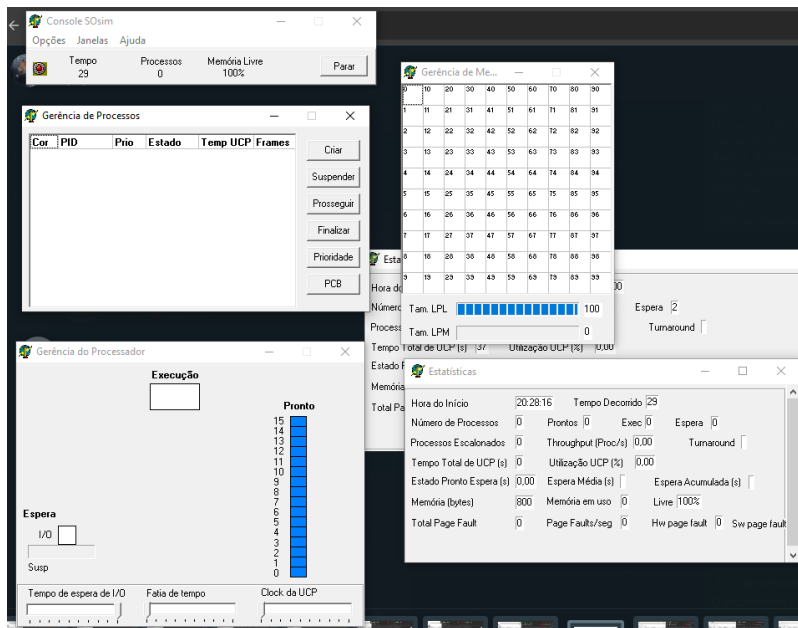
- a. O log mostra momentos em que um processo é interrompido por tempo ("**Pronto por tempo**").
- b. A variação no número de escalonamentos pode indicar que o sistema conseguiu reduzir a espera de processos por I/O, melhorando a utilização da CPU.

## Conclusão

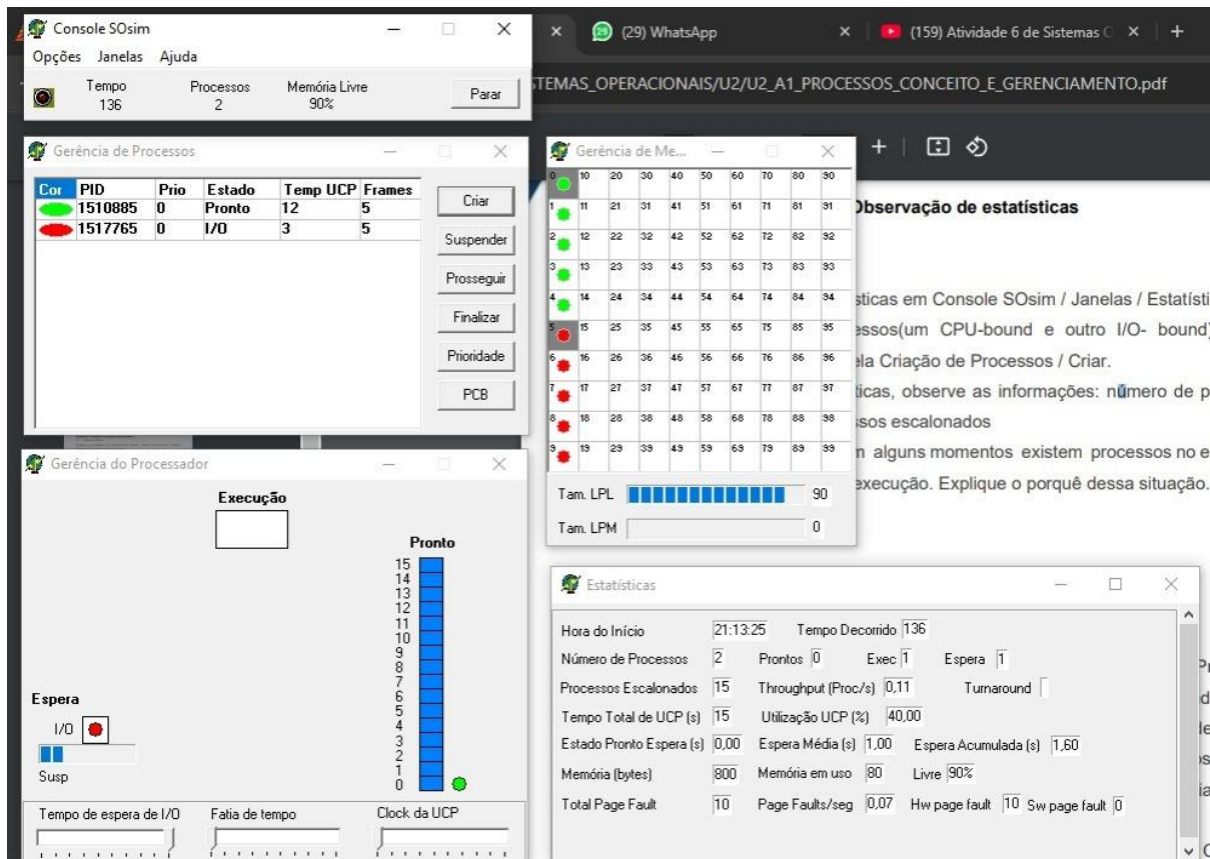
- **No início (238s), o uso da CPU era menor (60%)**, sugerindo que o escalonador ainda estava se ajustando ao comportamento dos processos.
- **No final (360s), a CPU atinge 100% de uso**, indicando que o balanceamento melhorou, reduzindo tempos ociosos.
- A diferença se deve ao **tempo gasto em I/O por um dos processos**, o que impactou a eficiência no início.

## Atividade 4 - Trabalho com Observação de estatísticas

- a. Ative a janela de Estatísticas em Console SOsim / Janelas / Estatísticas.



**b.** Crie dois novos processos(um CPU-bound e outro I/O- bound): janela Gerência de Processos / Criar – janela Criação de Processos / Criar.



I. Na janela Estatísticas, observe as informações: número de processos, estados dos processos e processos escalonados.

II. Observe que em alguns momentos existem processos no estado de pronto, porém nenhum em estado de execução. Explique o porquê dessa situação.

## i. Análise das Estatísticas

Na janela de **Estatísticas**, podemos observar as seguintes informações:

- **Número de Processos:** 2
- **Processos Prontos:** 1
- **Processos em Execução:** 0
- **Processos em Espera:** 1
- **Processos Escalonados:** 15
- **Utilização da CPU:** 40%
- **Tempo total de UCP:** 15s

Esses dados mostram que há um processo pronto para execução, mas nenhum está de fato executando no momento da captura da tela.

## II. Motivo para a ausência de processos em execução

Mesmo havendo um processo pronto, nenhum está sendo executado no exato momento. Isso pode ocorrer por alguns motivos:

### 1. Processo em I/O bloqueando a execução

- a. O processo **1517765** está no estado de **I/O** (entrada/saída), como indicado na tabela de processos e na janela de gerenciamento do processador.
- b. O sistema operacional pode estar aguardando a finalização dessa operação antes de permitir que o outro processo (1510885) entre em execução.

### 2. Fatia de tempo ou escalonamento ainda não ocorreu

- a. O escalonador pode estar aguardando um **novo ciclo de escalonamento** para designar um processo à CPU.
- b. Isso pode ser devido ao algoritmo de escalonamento adotado (exemplo: **Round-Robin, Prioridades**, etc.), que pode estar aguardando a liberação da CPU.

### 3. CPU ociosa temporariamente

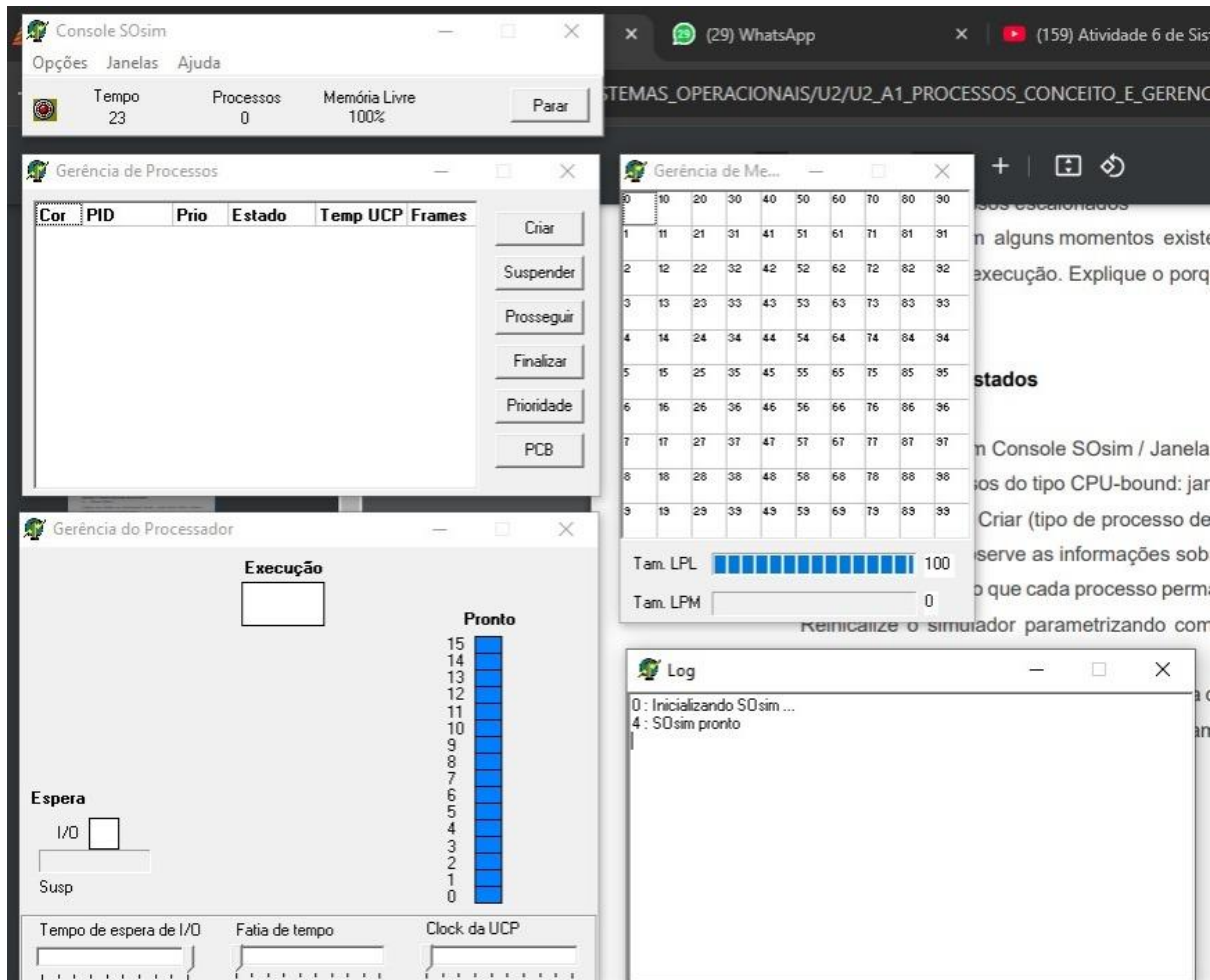
- a. Como a **utilização da CPU está em 40%**, há períodos em que a CPU não está sendo totalmente utilizada.
- b. Pode ter ocorrido um atraso no escalonamento ou um pequeno intervalo entre a finalização de um processo e o início de outro.

## Conclusão

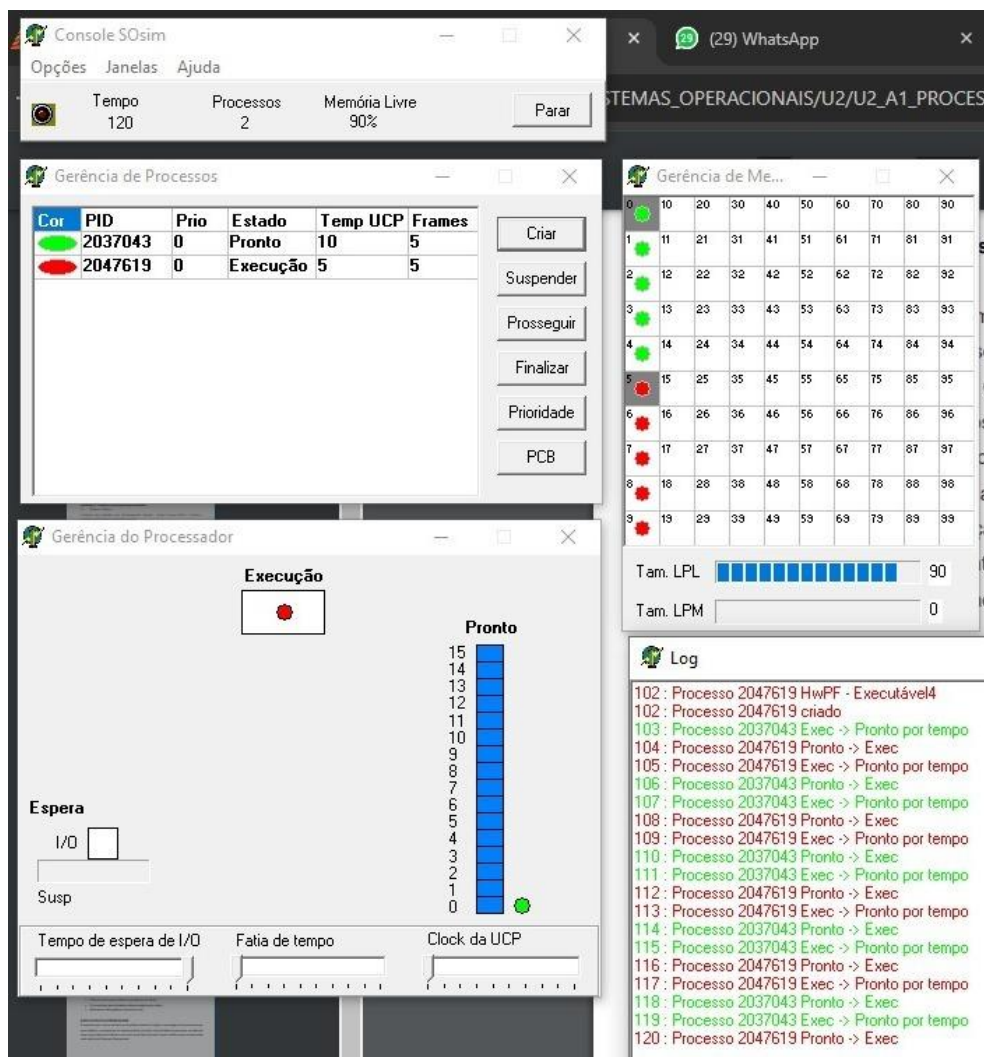
O sistema pode estar aguardando a **finalização da operação de I/O** do processo **1517765**, o que temporariamente impede a execução de outro processo. Além disso, o escalonador pode ainda não ter alocado um processo à CPU naquele instante. Esse comportamento é comum em sistemas que fazem gerenciamento de processos com dependências de I/O.

## Atividade 5 – Mudanças de estados

a. Ative a janela de Log em Console SOsim / Janelas / Log.



**b.** Crie dois novos processos do tipo CPU-bound: janela Gerência de Processos / Cria – janela Criação de Processos / Criar (tipo de processo deve ser CPU-bound).



**I.** Na janela Log, observe as informações sobre as mudanças de estado dos processos observando o tempo que cada processo permanece nos estados de Execução e Pronto.

### **Observações sobre as mudanças de estado**

1. O processo **2047619** começa sendo criado e logo entra em execução.
2. Há várias trocas entre **Execução** → Pronto e Pronto → **Execução**, indicando que o sistema está aplicando **time-sharing** (compartilhamento de tempo de CPU).
3. Cada processo permanece por um período limitado em **Execução**, sendo forçado a voltar para **pronto** por tempo, momento em que o outro processo assume a CPU.



4. Esse comportamento sugere que o **escalonador** está usando um algoritmo preemptivo, provavelmente **Round Robin**, onde cada processo recebe um tempo fixo de **CPU** antes de ser interrompido.

II. Reinicialize o simulador parametrizando com um valor de fatia de tempo diferente e observe as diferenças na janela Log.

The screenshot displays the Console SOsim interface with several windows open:

- Console SOsim**: Shows system status: Tempo 244, Processos 2, Memória Livre 90%, and a Parar button.
- Gerência de Processos**: A table showing process details.
 

Cor	PID	Prio	Estado	Temp UCP	Frames
Verde	2037043	0	Pronto	30	5
Vermelho	2047619	0	Execução	41	5

 Buttons on the right: Criar, Suspender, Prosseguir, Finalizar, Prioridade, PCB.
- Gerência de Memória**: A grid showing memory allocation status for 32 frames (0-31). Frames 0-10 are green (free), and frames 11-31 are red (occupied).
- Gerência do Processador**: Shows the processor state. The 'Execução' box contains a red dot, and the 'Pronto' queue has 15 blue bars. The 'Espera' section has I/O and Suspended checkboxes. At the bottom, there are input fields for 'Tempo de espera de I/O', 'Fatia de tempo', and 'Clock da UCP'.
- Log**: A scrollable list of system events.
 

```

199: Processo 2047619 Exec -> Pronto por tempo
200: Processo 2037043 Pronto -> Exec
201: Processo 2037043 Exec -> Pronto por tempo
202: Processo 2047619 Pronto -> Exec
203: Processo 2047619 Exec -> Pronto por tempo
204: Processo 2037043 Pronto -> Exec
205: Processo 2037043 Exec -> Pronto por tempo
206: Processo 2047619 Pronto -> Exec
207: Processo 2047619 Exec -> Pronto por tempo
208: Processo 2037043 Pronto -> Exec
209: Processo 2037043 Exec -> Pronto por tempo
210: Processo 2047619 Pronto -> Exec
211: Processo 2047619 Exec -> Pronto por tempo
212: Processo 2037043 Pronto -> Exec
213: Processo 2037043 Exec -> Pronto por tempo
214: Processo 2047619 Pronto -> Exec
215: Processo 2047619 Exec -> Pronto por tempo
216: Processo 2037043 Pronto -> Exec
217: Processo 2037043 Exec -> Pronto por tempo
218: Processo 2047619 Pronto -> Exec
228: Processo 2047619 Exec -> Pronto por tempo
229: Processo 2037043 Pronto -> Exec
234: Processo 2037043 Exec -> Pronto por tempo
235: Processo 2047619 Pronto -> Exec
      
```

1. **III.** Analise comparativamente a concorrência de dois processos CPU-bound executando em dois sistemas operacionais que se diferenciam apenas pelo valor da fatia de tempo.

**Fatia de Tempo Maior:**

- a. Um processo pode permanecer mais tempo na CPU antes de ser interrompido.
- b. Menos trocas de contexto, o que reduz a **sobrecarga do sistema**.
- c. O outro processo pode esperar mais tempo antes de ser **executado**.

**Fatia de Tempo Menor:**

- d. Os processos alternam com mais frequência entre "**Pronto**" e "**Exec**".
- e. Maior responsividade (importante para sistemas interativos).
- f. Aumento na sobrecarga do sistema devido a trocas de contexto mais frequentes.

## **Conclusão**

**Se o quantum for muito grande,** um processo pode monopolizar a CPU por mais tempo, prejudicando a concorrência.

**Se o quantum for muito pequeno,** há mais alternância, melhorando a concorrência, mas aumentando a sobrecarga do sistema.

**O ideal depende do contexto:** sistemas interativos favorecem quanta menores, enquanto aplicações intensivas em CPU podem se beneficiar de quanta maiores.