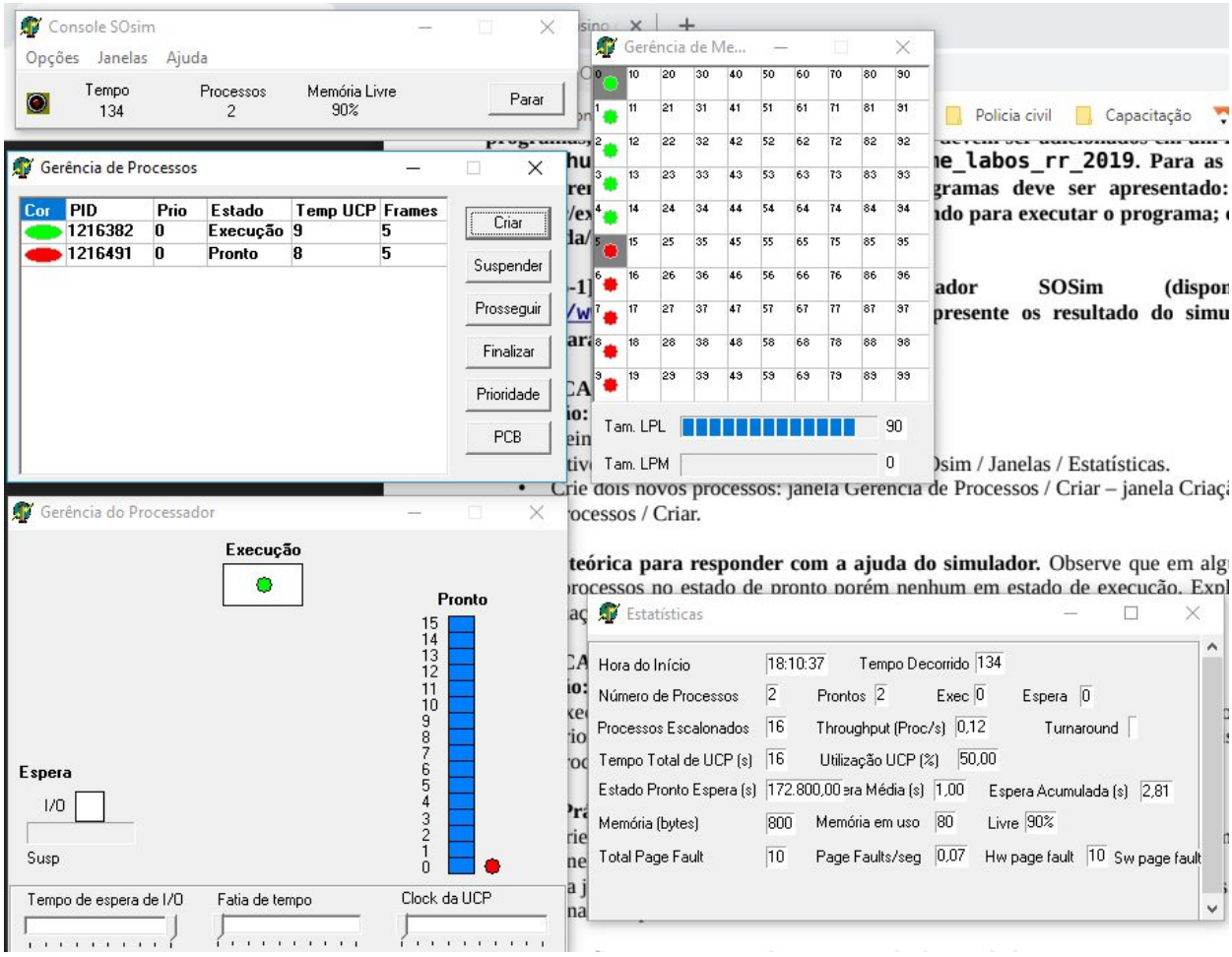
Universidade Federal de Roraima
Departamento de Ciência de Computação

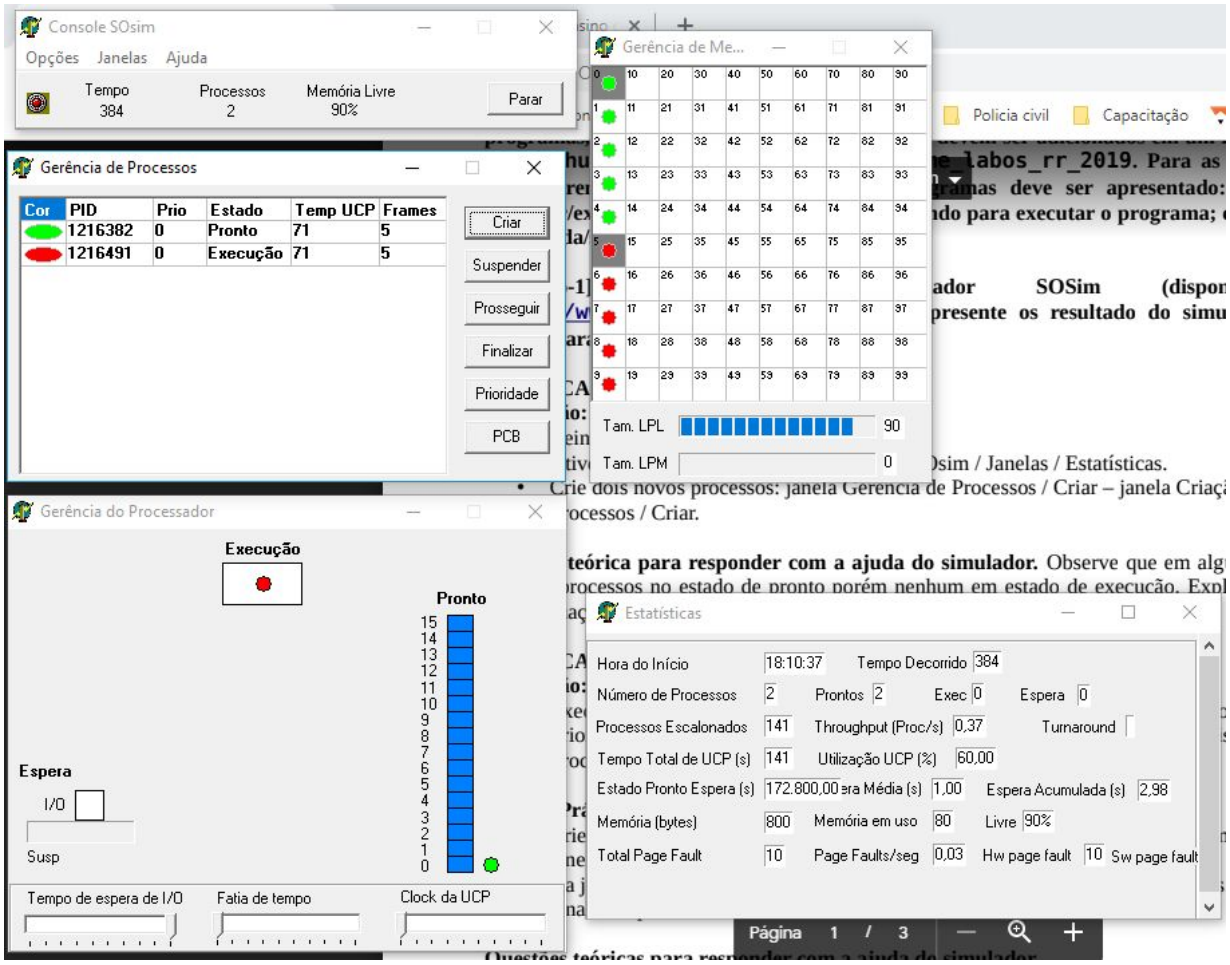
DISCIPLINA: Sistemas Operacionais – DCC403

ALUNO(A): Ewelly Fabiane Cunha de Sousa
NOTA:_____

[Questão-1] Utilizando o simulador SOSim (disponível em `http://www.training.com.br/sosim`) apresente os resultado do simulador e uma análise para cada item abaixo.

(PRÁTICA - A)

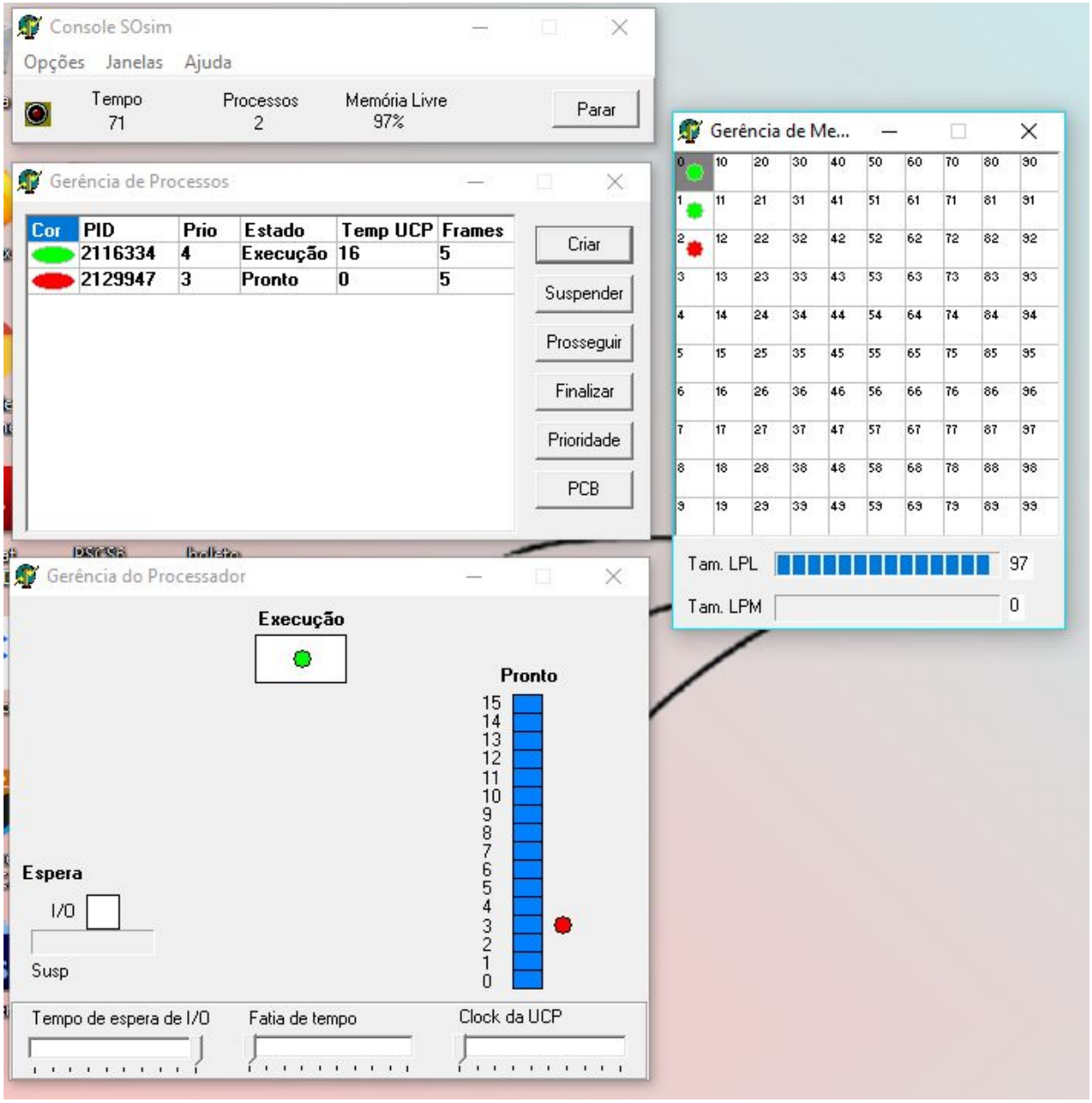


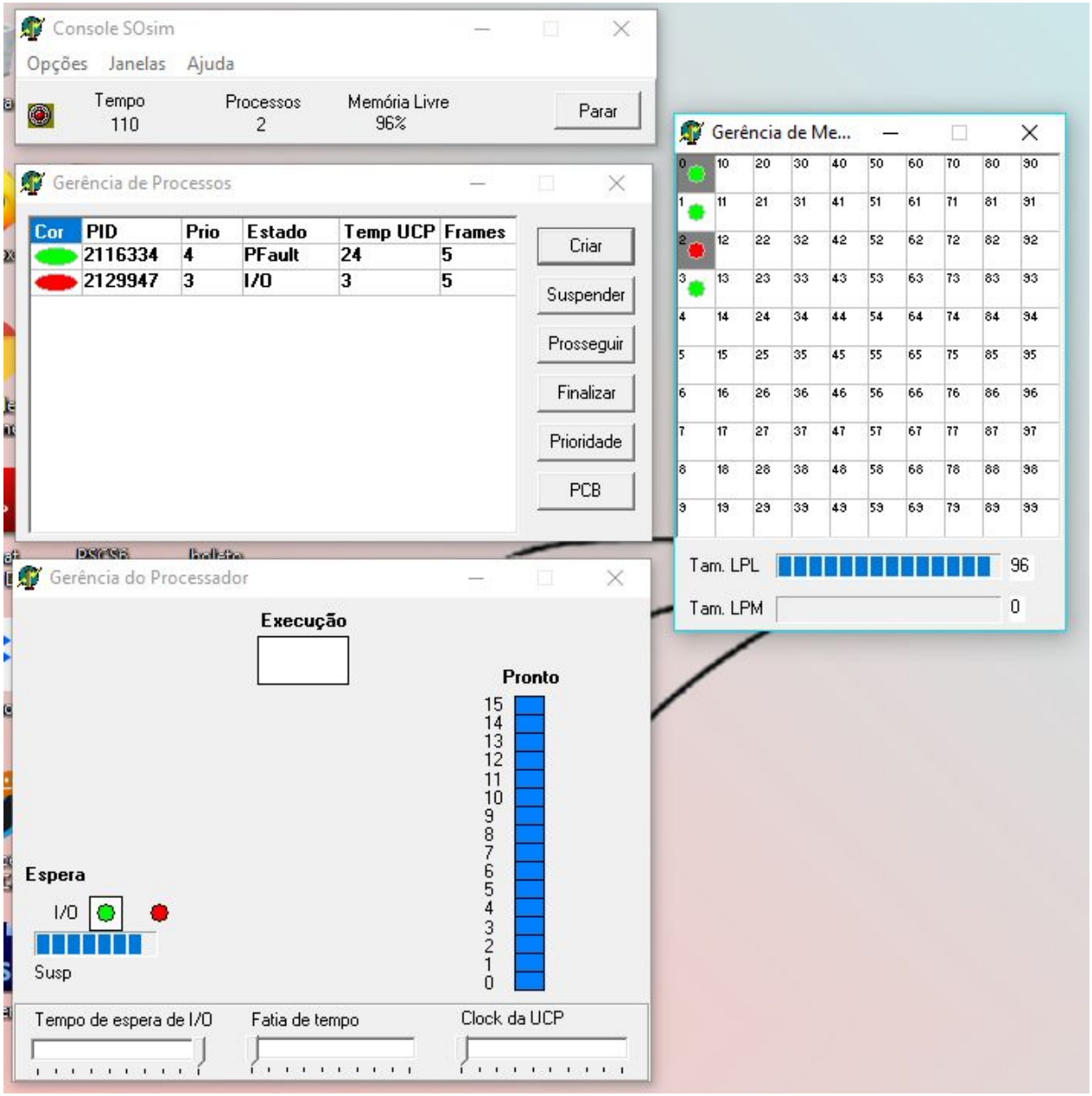


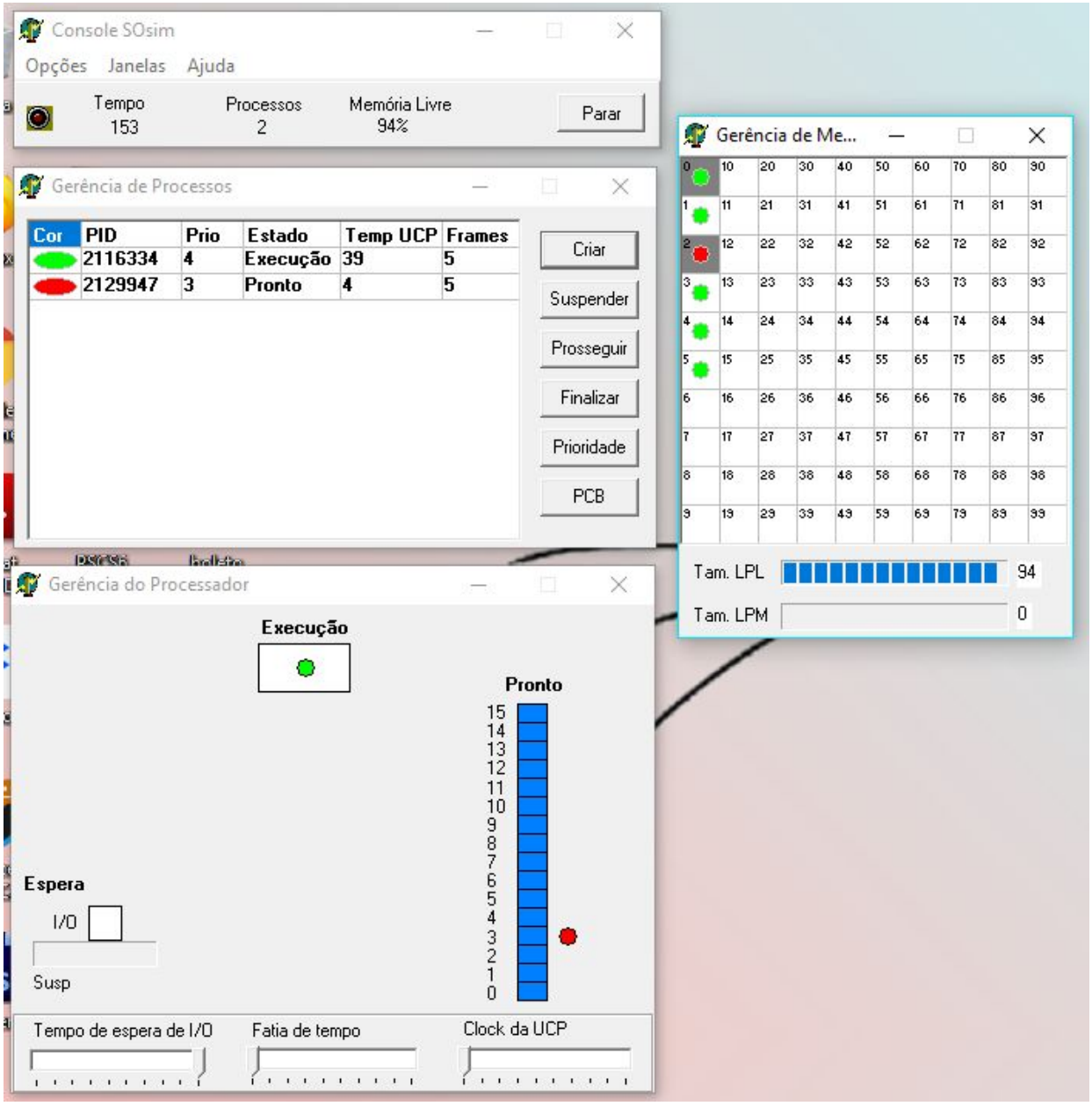
[Questão teórica] Observe que em alguns momentos existem processos no estado de pronto porém nenhum em estado de execução. Explique o porquê dessa situação.

Essa situação ocorre porque nesse momento os processos estão sendo escalonados pôr escalonamento circular, ou seja, conteúdo dos registradores do processo em execução são salvos e logo após carrega-se o conteúdo dos registradores do próximo processo na fila. E esse tipo de escalonamento é organizado de uma maneira que cada um deles possua um determinado tempo da CPU e caso um desses processos não termine dentro do seu tempo ele é colocado no fim da fila e outro tempo é dado para o processo no começo da fila. Nas imagens podemos ver o processo pronto na fila esperando para executar e o mesmo executando quando o outro deixa livre o recurso.

(PRÁTICA - B):


Console SOsim
Opções Janelas Ajuda
Tempo
71
Processos
2
Memória Livre
97%
Parar
Gerência de Processos
Cor PID Prio Estado Temp UCP Frames
2116334 4 Execução 16 5
2129947 3 Pronto 0 5
Criar
Suspender
Prosseguir
Finalizar
Prioridade
PCB
Gerência de Me...
Tam. LPL 97
Tam. LPM 0
Gerência do Processador
Execução
Pronto
Espera
I/O
Susp
Tempo de espera de I/O
Fatia de tempo
Clock da UCP


Console SOsim
Opções Janelas Ajuda
Tempo
110
Processos
2
Memória Livre
96%
Parar
Gerência de Processos
Cor PID Prio Estado Temp UCP Frames
2116334 4 PFault 24 5
2129947 3 I/O 3 5
Criar
Suspender
Prosseguir
Finalizar
Prioridade
PCB
Gerência de Me...
Tam. LPL 96
Tam. LPM 0
Gerência do Processador
Execução
Pronto
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Espera
I/O
Susp
Tempo de espera de I/O
Fatia de tempo
Clock da UCP



Questões teóricas para responder com a ajuda do simulador

• Por que o problema do starvation pode ocorrer?

Ocorre porque foi predefinida uma prioridade de nível 4 para o processo de tipo cpu-bound e uma prioridade estática de 3 para o processo I/O-bound. O que acarreta em deixar o processo de menor prioridade em espera eterna até “morrer de fome”.

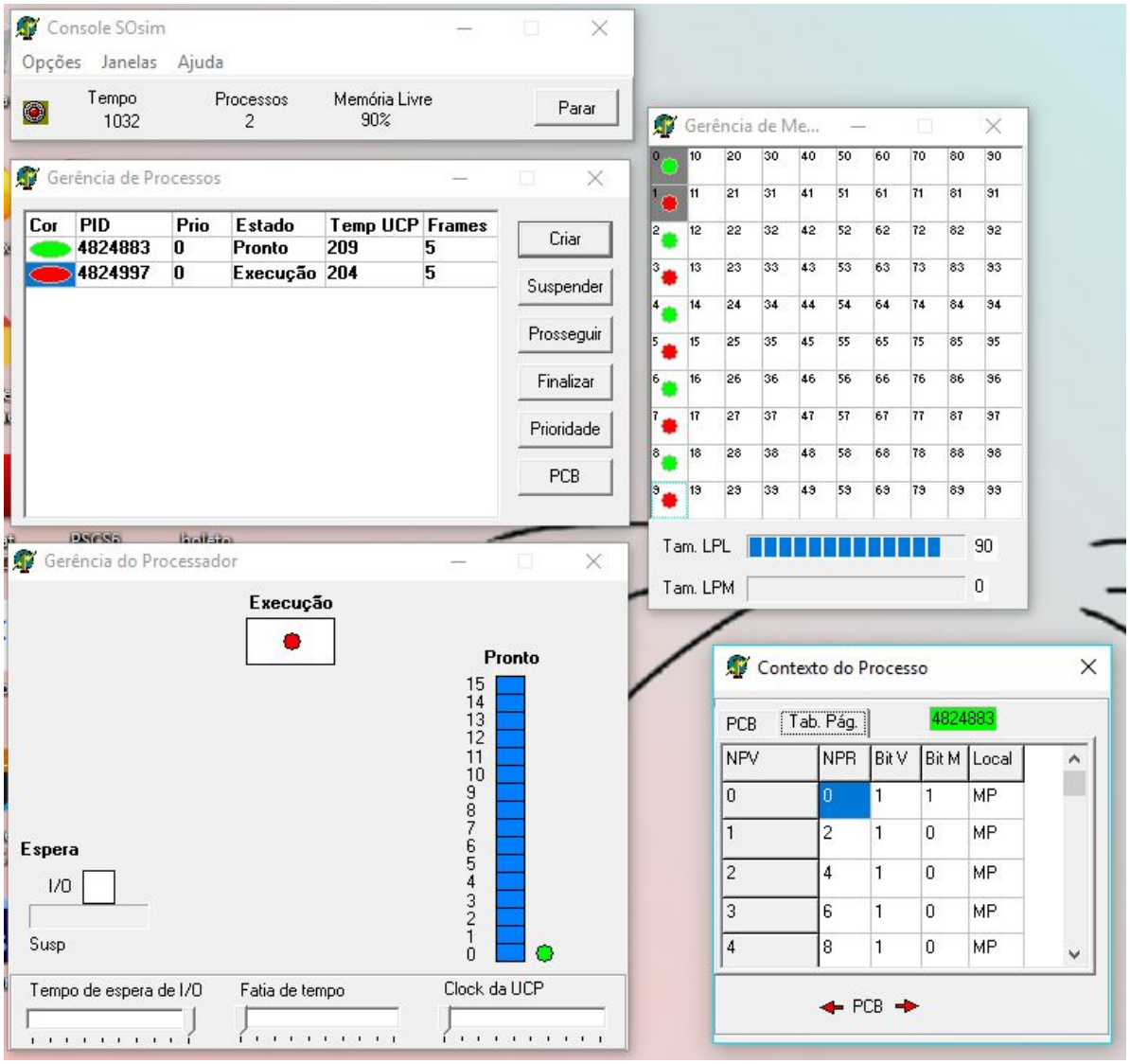
• Cite duas ações que o administrador do sistema pode realizar quando é identificada a situação de starvation em um processo?

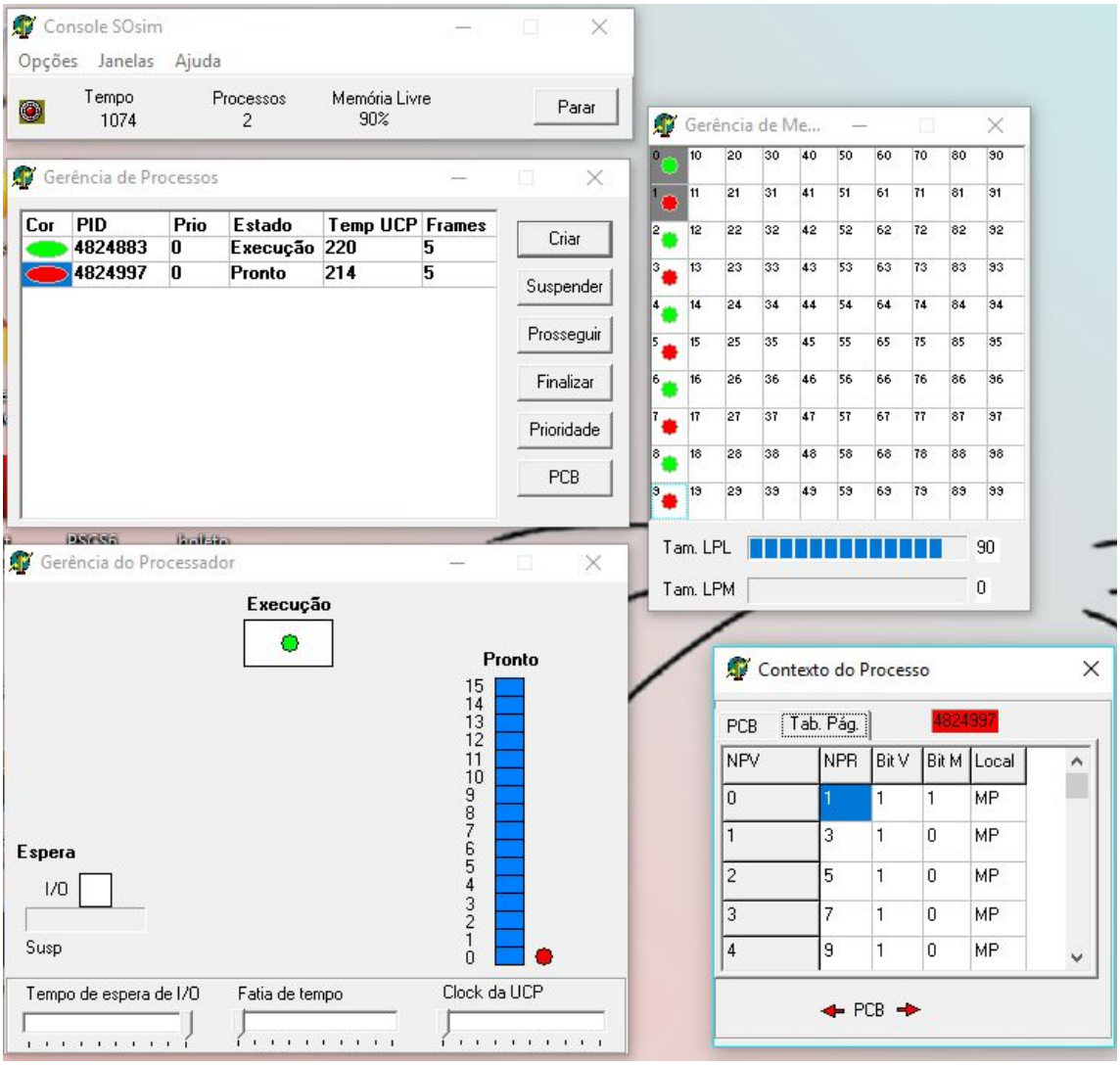
[0] Ignorar o problema kkkkkkk brinks

[1] Usando fila de prioridade onde os dois processos possuem a mesma prioridade ou finalizando ou suspendendo o processo que está causando starvation.

[2] Criar uma fila do tipo FIFO (first in first out) os elementos vão sendo colocados na fila e retirados (processados) por ordem de chegada. A ideia fundamental da fila é que só podemos inserir um novo elemento no final da fila e só podemos retirar o elemento do início.

(PRÁTICA – C) Simulação:


Console SOsim
Opções Janelas Ajuda
Tempo
1032
Processos
2
Memória Livre
90%
Parar
Gerência de Processos
Cor PID Prio Estado Temp UCP Frames
4824883 0 Pronto 209 5
4824997 0 Execução 204 5
Criar
Suspender
Prosseguir
Finalizar
Prioridade
PCB
Gerência de Me...
Tam. LPL 90
Tam. LPM 0
Gerência do Processador
Execução
Pronto
Espera
I/O
Susp
Tempo de espera de I/O
Fatia de tempo
Clock da UCP
Contexto do Processo
PCB
Tab. Pág.
4824883
NPV NPR Bit V Bit M Local
0 0 1 1 MP
1 2 1 0 MP
2 4 1 0 MP
3 6 1 0 MP
4 8 1 0 MP
PCB



Questões teóricas para responder com a ajuda do simulador:

• Qual o espaço de endereçamento real máximo de um processo?

A quantidade total da memória principal e secundária juntas.

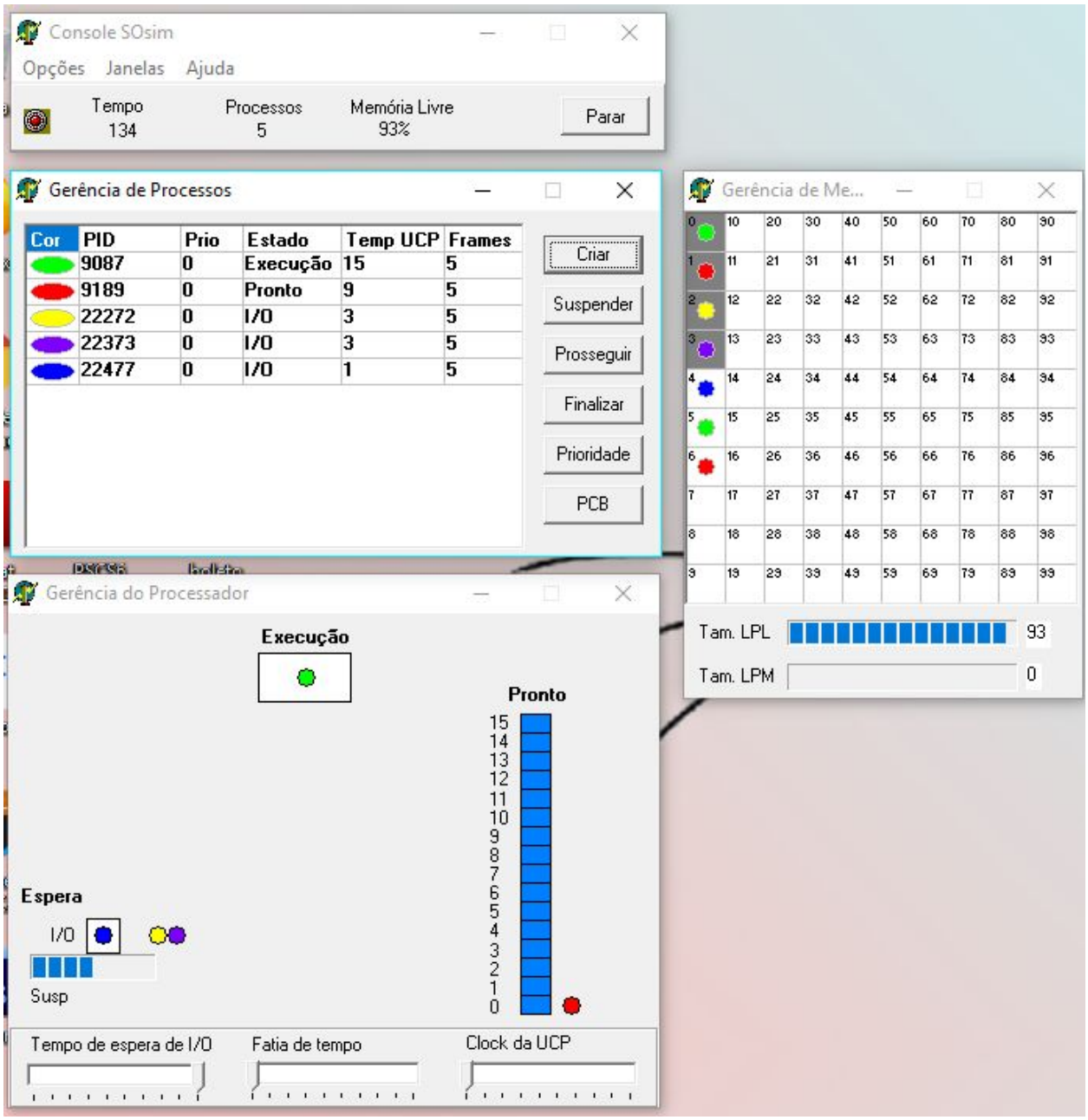
• Qual o espaço de endereçamento real mínimo de um processo?

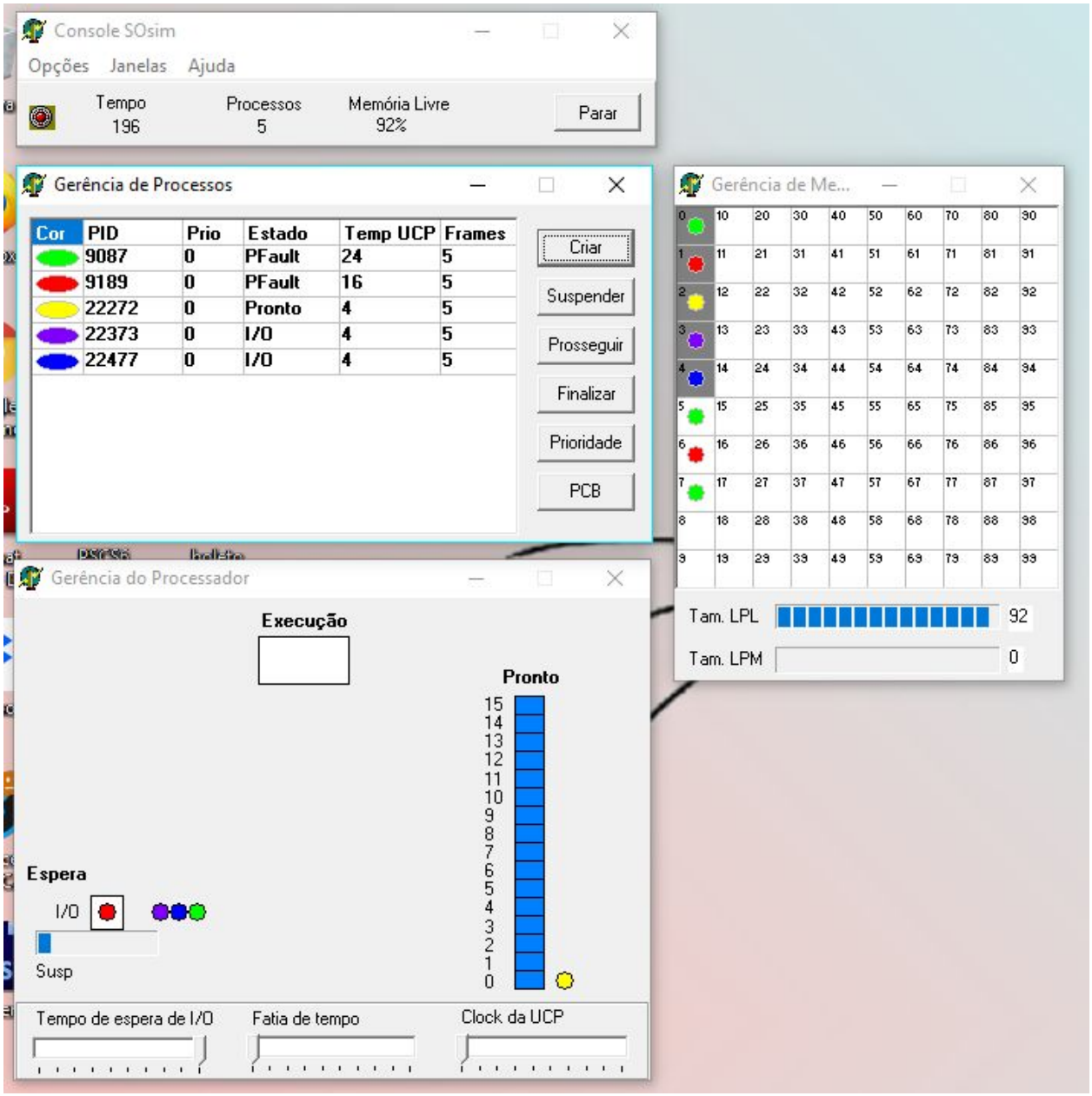
O tamanho mínimo da tabela de mapeamento carregada.

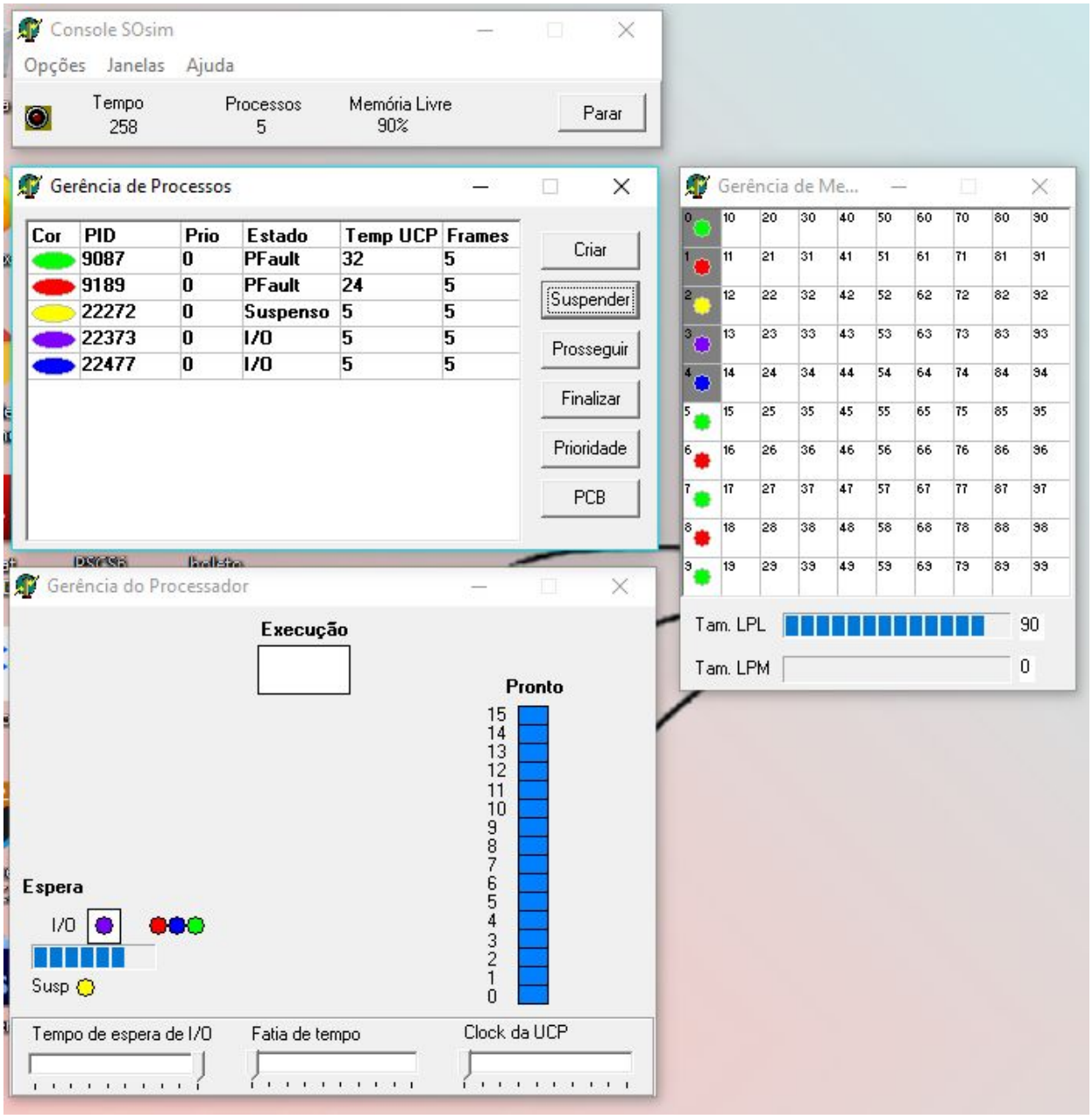
• Qual o tamanho da página virtual?

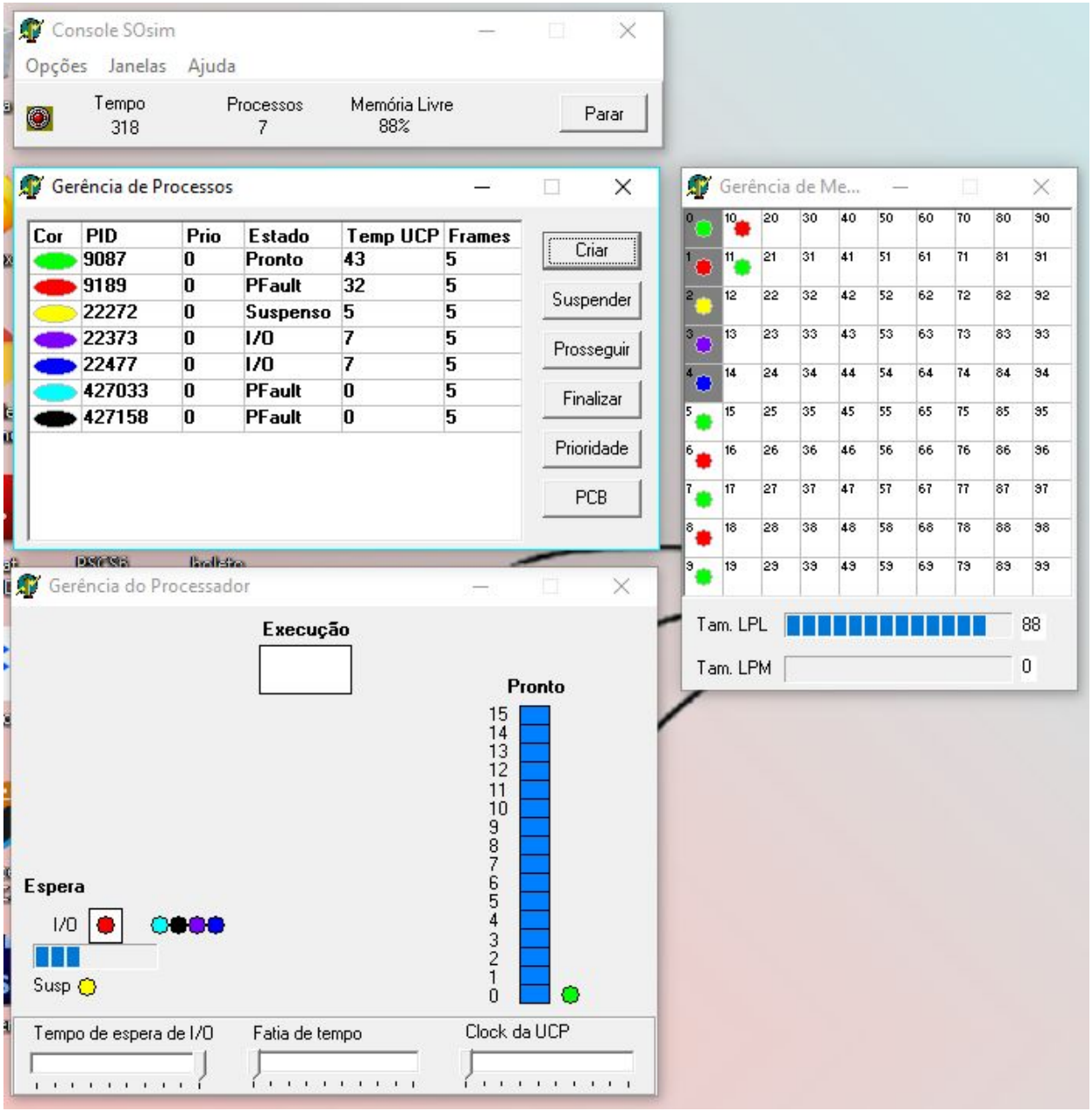
O tamanho pode variar de acordo com o processador utilizado e a arquitetura do hardware, podendo em algumas arquiteturas ser configurado.

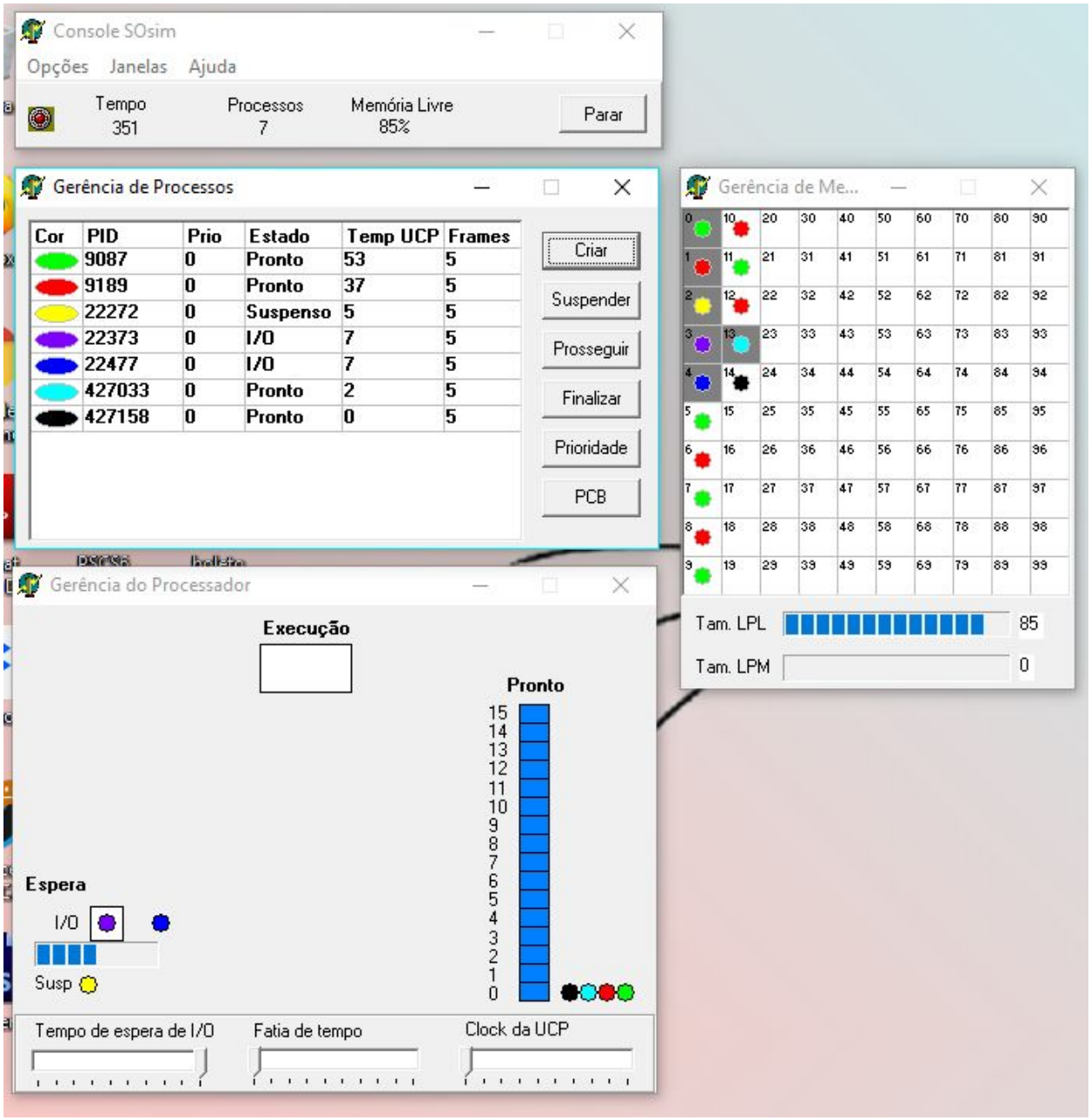
(PRÁTICA – D)


Console SOsim
Opções Janelas Ajuda
Tempo 134
Processos 5
Memória Livre 93%
Parar
Gerência de Processos
Cor PID Prio Estado Temp UCP Frames
9087 0 Execução 15 5
9189 0 Pronto 9 5
22272 0 I/O 3 5
22373 0 I/O 3 5
22477 0 I/O 1 5
Criar
Suspender
Prosseguir
Finalizar
Prioridade
PCB
Gerência de Me...
Tam. LPL 93
Tam. LPM 0
Gerência do Processador
Execução
Pronto
Espera
I/O
Susp
Tempo de espera de I/O
Fatia de tempo
Clock da UCP


Console SOsim
Opções Janelas Ajuda
Tempo
196
Processos
5
Memória Livre
92%
Parar
Gerência de Processos
Cor PID Prio Estado Temp UCP Frames
9087 0 PFault 24 5
9189 0 PFault 16 5
22272 0 Pronto 4 5
22373 0 I/O 4 5
22477 0 I/O 4 5
Criar
Suspender
Prosseguir
Finalizar
Prioridade
PCB
Gerência de Me...
Tam. LPL 92
Tam. LPM 0
Gerência do Processador
Execução
Pronto
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Espera
I/O
Susp
Tempo de espera de I/O
Fatia de tempo
Clock da UCP


Console SOsim
Opções Janelas Ajuda
Tempo
258
Processos
5
Memória Livre
90%
Parar
Gerência de Processos
Cor PID Prio Estado Temp UCP Frames
9087 0 PFault 32 5
9189 0 PFault 24 5
22272 0 Suspenso 5 5
22373 0 I/O 5 5
22477 0 I/O 5 5
Criar
Suspender
Prosseguir
Finalizar
Prioridade
PCB
Gerência de Me...
Tam. LPL 90
Tam. LPM 0
Gerência do Processador
Execução
Pronto
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Espera
I/O
Susp
Tempo de espera de I/O
Fatia de tempo
Clock da UCP


Console SOsim
Opções Janelas Ajuda
Tempo
318
Processos
7
Memória Livre
88%
Parar
Gerência de Processos
Cor PID Prio Estado Temp UCP Frames
9087 0 Pronto 43 5
9189 0 PFault 32 5
22272 0 Suspenso 5 5
22373 0 I/O 7 5
22477 0 I/O 7 5
427033 0 PFault 0 5
427158 0 PFault 0 5
Criar
Suspender
Prosseguir
Finalizar
Prioridade
PCB
Gerência de Me...
Tam. LPL
88
Tam. LPM
0
Gerência do Processador
Execução
Pronto
Espera
I/O
Susp
Tempo de espera de I/O
Fatia de tempo
Clock da UCP



Questão teórica para responder com a ajuda do simulador:

• Quais os critérios utilizados pelo simulador para selecionar o processo a ser transferido para o arquivo de paginação (swap out)?

O processo é selecionado analisando qual processo possui menor chance de utilizar a CPU nos próximos instantes e assim ele manda esse processo para o arquivo de paginação.

• Quando o processo deve ser transferido novamente para a memória principal (swap in)?

Quando o limite de memória principal for referenciada, ou seja é utilizada a política de demanda, desta forma os processos só irão não para a memória principal quando forem realmente necessários.

[Questão-2] Com relação ao problema de Deadlock. Pesquisa e descreva o algoritmo do banqueiro (criado por Dijkstra) que pode ser utilizado para evitar impasses. Sempre que recursos são solicitados, o algoritmo avalia se atender à solicitação levará a um estado inseguro e se isso ocorrer, ela não é atendida. Adicionalmente, escreva o algoritmo do banqueiro em C/C++ e apresente alguns exemplos de sua execução.

O algoritmo funciona para solucionar um impasse explicado em que um banqueiro de uma pequena cidade pode negociar com um grupo de clientes para os quais ele libera linhas de crédito. E esse algoritmo verifica se a liberação de uma requisição é capaz de levar a um estado inseguro. É possível verificar se um estado é seguro ou não observando esta tabela abaixo. Estão disponíveis 7 créditos para serem usados pelos clientes e o banqueiro precisa gerenciar isso. Esse estado mostrado na tabela é seguro, pois, com essa quantidade de recursos disponíveis ele pode dar a um dos clientes a quantidade que ele precisa para atingir o crédito máximo e assim que ele terminar de usar os créditos eles são devolvidos e assim os outros podem usá-lo e desta forma todos conseguem obter o crédito. Caso o crédito disponível fosse, por exemplo, 3 não seria possível ele administrar isso para que todos pudessem usar, considerado assim um caso inseguro. E por fim caso a requisição do cliente seja dada como insegura ela será negada, caso contrário o recurso será liberado.