

# MAC0323 - Algoritmos e Estruturas de Dados II

## Segundo semestre de 2020

Segundo Exercício-Programa – Devolução: 11 de outubro de 2020

### Sistema de Multiprocessamento com Prioridades

O objetivo desse exercício-programa é aprimorar o sistema de multiprocessamento do EP1. Neste EP, cada processo terá uma prioridade, e processos com prioridade mais alta terão acesso mais prontamente à CPU.

Se você organizou bem o seu EP1, o EP2 deve consistir essencialmente de implementar uma nova biblioteca para filas, que levará em conta também as prioridades, e substituir uma biblioteca de filas por esta nova, para uma das filas do seu EP1.

## 1 Prioridades

Quando um programa executa em um sistema como por exemplo o Linux, o sistema cria uma instância do programa. Essa instância em execução do programa é chamada de processo. Todo processo requer uma quantidade de recursos, tais como tempo de CPU e RAM, para ser capaz de executar suas tarefas. Cada processo recebe uma prioridade, que determina como é o acesso à CPU para a sua execução. Essa prioridade determina quais processos ganham acesso à CPU mais rapidamente, e quais esperam mais tempo até ganhar acesso à CPU.

Neste EP, a prioridade de um processo será um número inteiro entre 0 e 9, sendo que 9 representa a prioridade mais alta, que dá acesso mais rápido à CPU. Cada processo, ao ser criado, terá uma prioridade escolhida uniformemente ao acaso entre 0 e 9.

Como no EP1, os processos são inicialmente colocados na fila de espera. A cada unidade de tempo, se houver espaço livre na CPU, um processo da fila de espera irá para o fim da fila da CPU. O processo da fila de espera escolhido para ir para a fila da CPU será, alternadamente, o processo de mais alta prioridade ou o processo criado há mais tempo. O restante do funcionamento do sistema é o mesmo do sistema simulado no EP1.

## 2 Fila de entrada com prioridades

Para implementar a fila de entrada com prioridades, com o funcionamento descrito acima, você deverá implementar uma fila ordenada por instante de chegada com uma lista duplamente encadeada circular, com cabeça de lista.

Adicionalmente, os processos que estão na fila de espera devem estar também em uma fila priorizada de máximo, com a prioridade do processo como chave. Esta fila deve ser implementada em um vetor com redimensionamento.

Estas duas estruturas de dados devem estar interligadas, de modo que a remoção de um processo que vai para a fila da CPU seja feita de maneira tão eficiente quanto possível. Evite também duplicar informação.

Quando há mais de um processo com a maior prioridade e devemos escolher um destes para ir para a fila da CPU, deve ser escolhido, dentre os de prioridade máxima, o que está há mais tempo na fila de espera.

Quando o processo escolhido vem da fila priorizada, ele deve ser removido tanto da fila priorizada quanto da fila ordenada por instante de chegada. Quando ele vem da fila ordenada, deve ser removido também da fila priorizada.

A eficiência da sua implementação será fortemente levada em conta na sua nota.

### 3 Ajustes de parâmetros da simulação

Aproveitaremos para fazer alguns ajustes nos parâmetros usados no EP1. Primeiramente, um novo processo chegará ao sistema com probabilidade 0.05. A velocidade de impressão de cada impressora será de 2 linhas por UT.

### 4 Entrada e saída do programa

O seu programa deve ter como entrada a duração da simulação em unidades de tempo (UT) e uma opção de saída completa ou simplificada.

Durante a execução, os dados iniciais de cada processo que chega devem ser impressos. Além disso, para cada processo terminado, as seguintes informações devem ser impressas:

- (a) dados iniciais do processo: tempo de processamento, prioridade e número de linhas a serem impressas);
- (b) tempo total de permanência no sistema;
- (c) tempo gasto em cada uma das filas;
- (d) razão entre o tempo de processamento e o tempo total de permanência no sistema.

No final, seu programa deve imprimir o número total de processos gerados, o número total de processos concluídos, o número de processos em cada uma das filas no final da simulação, assim como a média dos itens (b), (c) e (d). Adicionalmente, imprima a média dos itens (b) e (c), assim como a média do tempo na fila de espera e o número de processos na fila de espera para cada prioridade que apareceu durante a simulação. Para as médias, inclua apenas processos que entraram no sistema a partir da UT 100, para minimizar o efeito do estado inicial vazio do sistema na simulação.

A saída completa deve mostrar também a situação das várias filas a cada unidade de tempo. A fila de espera pode ser mostrada apenas em ordem de tempo de chegada. Não é necessário apresentar a organização da fila de prioridades.

Lembre-se que a saída do seu programa deve ser clara e o seu programa deve estar bem documentado, modularizado e organizado.