

1. Implementar, durante o semestre, um programa que simule os algoritmos de escalonamento de processos FCFS, SJF Não-Preemptivo, SJF Preemptivo, Prioridade Não-Preemptivo, Prioridade Preemptivo e Round-Robin.

Para isso, considere as seguintes especificações:

- a. O programa terá como entrada um conjunto de processos, com cada processo contendo os seguintes atributos:
 - i. O seu tempo de execução
 - ii. O seu tempo de chegada (menos para o FCFS e Round Robin)
 - iii. A sua prioridade (exclusivo para o algoritmo Prioridade)
- b. Considere uma lista de processos de entrada (com número de processos na lista configurável através de uma variável estática ou dinâmica), onde cada atributo pode ser definido pelo usuário (através de leitura pelo teclado) ou de forma aleatória (através de uma biblioteca de geração de número aleatórios).
- c. O programa deve imprimir na tela o histórico de execução dos processos (qual processo executou em qual instante de tempo), o tempo de espera de cada processo e o tempo médio de espera (soma dos tempos de espera de cada processo dividido pelo número de processos). Um exemplo de saída do programa está disponibilizada no moodle.
- d. A lista de processos deve ser implementada em uma lógica que garanta que o mesmo conjunto de processos possa servir como entrada para os diferentes algoritmos de escalonamento implementados, para fins de comparação dos resultados de tempo médio de espera.
- e. Deve ser possível reatribuir a lista de processos de entrada através de uma opção digitada pelo usuário, sem precisar reiniciar o programa.

Cronograma e Forma de Entrega:

- Os arquivos desenvolvidos deverão ser submetidos para um repositório GITHUB compartilhado com o e-mail jean.hamerski@restinga.ifrs.edu.br, e entregues até o final do semestre, através de envio de e-mail para jeah.hamerski@restinga.ifrs.edu.br, informando a URL do repositório e anexando um arquivo de texto informando como executar o programa desenvolvido.

Critérios de avaliação:

- Ao todo, são 6 algoritmos a serem desenvolvidos, sendo que cada algoritmo vale 1,67 pontos na nota final desta avaliação (valendo 10,0).
- O trabalho é individual, porém os(as) estudantes podem se auxiliar durante o desenvolvimento. O(A) estudante deve ter domínio completo do código desenvolvido. O professor poderá solicitar explicação do código desenvolvido durante o desenvolvimento ou após a entrega, para fins de avaliação.

Observações:

- A teoria por trás de cada um dos algoritmos citados é a que vimos na disciplina, baseada na apostila base.
- Para implementação dos programas, pode ser usada qualquer linguagem de programação, assim como pode ser orientado a objeto ou não.
- Um código em Java não orientado a objeto, que pode ser utilizado como base para início da implementação, é disponibilizado no seguinte repositório do github: <https://github.com/jeanhamerski/so-base.git>
- Este trabalho terá o mesmo peso das provas do semestre, sendo que a média do semestre será calculada da seguinte maneira:
 - Média do Semestre = (Prova1 + Prova2 + Trabalho)/3