UTFPR – Universidade Tecnológica Federal do Paraná – CM
DACOM – Departamento Acadêmico de Computação
COCIC – Coordenação de Ciência da Computação
Curso: Ciência da Computação
Disciplina/Semestre: BCC34G – Sistemas Operacionais

O trabalho consiste na implementação dos algoritmos de escalonamento de processos: *Round-Robin* (RR), Shortest Job First com Previsão (SJF preemptivo), Prioridade dinâmica com retroalimentação. O objetivo é colocar em prática a teoria vista em sala de aula, e incentivar a pesquisa e estudo sobre algoritmos e estruturas relacionadas à Gerência de Processos de um Sistema Operacional. Logo, deve-se implementar uma aplicação que simule o Gerenciamento de Processos, considerando **filas, BCP e algoritmos de escalonamento**.

Para o SJF, considere que a primeira execução será usada para a previsão inicial. No prioridade dinâmica com retroalimentação, use duas filas de prioridades. A primeira fila (fila A) deve manter os processos que não usaram todo o *quantum* com uma política preemptiva, considerando a prioridade interna do processo. A prioridade é calculada a cada execução como o percentual do *quantum* consumido. A segunda fila segue uma política *round-robin* sem considerar prioridades com *quantum* fixo. Na segunda fila (fila B) devem ser mantidos os processos *CPU bound.* Todos os processos que usam o *quantum* inteiro (fila A ou B) são colocados no final da fila B. Para todos os algoritmos, durante a realização de I/O, deve ser informado em um arquivo de configuração o tempo em ciclos da operação de I/O. Pode-se usar um valor fixo, por exemplo, 2 ciclos, ou um valor aleatório em uma faixa, por exemplo, 2 a 5 ciclos. Segue um exemplo de arquivo de configuração:

Formato do Arquivo de Configuração

```
# round robin
quantum_rr = 2

# jsf com previsao
tempo_min_io = 2
tempo_max_io = 5

# prioridade dinâmica
quantum_fila_a = 3
quantum_fila_b = 4
```

A estrutura de dados que representa um processo em execução, o Bloco de Controle de Processo (BCP), deve conter as seguintes informações:

- ***Pid:*** identificador do processo.
- ***Prioridade:*** prioridade do processo (para ser utilizado no algoritmo que considera prioridade).
- ***Estado:*** estado do processo.
- **Tempos**: de chegada, de início, de uso da CPU, de espera.
- Outros campos de controle e/ou estado que forem necessários.

Para efeito de simulação, pode ser mantida para cada processo uma sequência de eventos (p. ex. bloqueado para I/O) que possibilitem uma dinâmica na execução de um conjunto de processos.

Na análise considerem os cenários de execução:

- **Processos *CPU-Bound*** → processos amarrados à CPU, fazem mais processamento do que entrada/saída.
- **Processos *IO-Bound*** → fazem mais entrada e saída.
- **Misto** → Mistura de processos *CPU-Bound* e *I/O-Bound*.

Além de imprimir a linha do tempo da execução (Diagrama de Gantt), deve-se apresentar as seguintes medições e estatísticas para os cenários propostos:

- **Tempo Total de Espera (TTE)** e **Tempo Médio de Espera (TME)** pelos processos para serem executados;
- ***Throughput* do sistema**: Quantos processos foram executados por unidade de tempo (hora, minuto, segundos).
- **Tamanho máximo, médio das filas** de processos do sistema (carga do sistema).
- **Tempo de resposta médio** por processo.
- E outras informações que julgarem interessantes.

O formato de arquivo de entrada deve seguir o formato especificado: Identificação (PID), Duração da Fase de uso da CPU (DF), Prioridade (PRI), Tempo de Chegada (TC) e Fila de Eventos de Entrada e Saída (FIO).

Formato do Arquivo de Entrada

`# PID – DF – PRI – TC – FIO`
`0 10 4 0 3 5 7  # Processo Id: 0, Duração: 10 ciclos, Prioridade: 4, Tempo de Chegada: 0 e Eventos de I/O: 3, 5 e 7.`
`1 5 2 1  # Processo Id: 1, …, Nenhum evento de I/O.`
`2 2 1 5 1`

Instruções:

1. O trabalho pode ser implementado em qualquer linguagem de programação.
2. Comentar o código-fonte e descrever nos cabeçalhos de arquivos como foi realizada a implementação.
3. Deve ser entregue um relatório explicando os principais aspectos da implementação.
4. O trabalho deve ser apresentado por um membro do grupo a ser selecionado pelo professor.
5. Se, em um mesmo instante, vários processos forem adicionados a fila de aptos, analisar na seguinte ordem: (1) criação (2) retornando da CPU (3) retornando de I/O.
6. Crie um arquivo de teste com no mínimo 3 processos CPU bound, 3 processos IO bound e 3 processos mistos.
7. Serão avaliados:
   a) A correção da implementação em relação ao que foi pedido no trabalho;
   b) A colocação em prática dos conceitos que foram discutidos em sala de aula de forma correta;
   c) A qualidade do projeto e da implementação (descrição da implementação,documentação do código, qualidade da solução);
   d) Apresentação ao professor.