**TRABALHO SISTEMAS OPERACIONAIS I**

**Ciclo de Vida de Processo**

**Integrantes:**

**Cauã Pereira Domingues**

**João Manuel Oliveira Pereira**

**João Paulo Saccomani de Freitas Bortolocce**

**Descrição do Funcionamento do Algoritmo**

**Este algoritmo simula o comportamento de um escalonador de processos de um sistema operacional, utilizando filas para organizar os processos de acordo com o seu estado (pronto para execução ou em espera). O sistema gerencia os processos em diferentes estados, permitindo que sejam adicionados, removidos e visualizados à medida que passam pelo ciclo de vida típico de um processo no sistema.**

**A função central que controla o fluxo da simulação é a função execucao(), que atua como o “cérebro” do sistema. É dentro dessa função que várias chamadas para outras funções são feitas, desempenhando papéis essenciais para o funcionamento correto do simulador. Vamos analisar o propósito de algumas dessas chamadas no contexto geral do algoritmo.**

**1. Gerenciamento das Filas de Processos**

**O algoritmo utiliza duas filas principais:**

**- Fila de Pronto (TpFilaPronto): Esta fila armazena os processos que estão prontos para serem executados.**

**- Fila de Espera (TpFilaEspera): Esta fila mantém os processos que, por algum motivo, foram bloqueados e estão aguardando para serem liberados e, eventualmente, retornarem à fila de prontos.**

**A função execucao() chama funções que manipulam essas filas, como EnqueuePronto() e EnqueueEspera(), para adicionar processos às suas respectivas filas, ou DequeuePronto() e DequeueEspera() para removê-los e tratá-los conforme necessário.**

**Motivo da Chamada:**

**O escalonador precisa constantemente organizar os processos que chegam e que saem das filas, garantindo que sejam corretamente classificados como prontos ou bloqueados. Essas funções são chamadas para inserir e remover processos de forma ordenada, mantendo o fluxo do escalonamento.**

**2. Exibição Visual dos Processos**

**A exibição dos processos é uma parte importante do algoritmo, pois permite ao usuário visualizar o estado atual dos processos nas filas de prontos e espera. Funções como ExibirPronto() e ExibirEspera() são chamadas para desenhar visualmente os processos e suas respectivas prioridades.**

**Motivo da Chamada:**

**Essas funções permitem que o usuário tenha um feedback visual do que está acontecendo no sistema. A visualização do estado dos processos ajuda a compreender melhor como o escalonador está manipulando cada processo e quais estão sendo executados ou bloqueados em determinado momento.**

**3. Execução de Processos**

**O ponto-chave do simulador é a execução dos processos. Na função execucao(), o próximo processo na fila de prontos é retirado e executado. Esse ciclo de execução simula o comportamento real de um escalonador, onde os processos são escolhidos com base em suas prioridades ou pelo tempo restante de execução. Se o processo precisa ser bloqueado ou colocado em espera, ele é transferido para a fila correspondente.**

**Motivo da Chamada:**

**Essa parte do código reflete o verdadeiro comportamento de um sistema operacional, onde o escalonador precisa decidir qual processo será executado a seguir. O processo escolhido é retirado da fila de prontos e, se necessário, movido para a fila de espera.**

**4. Controle de Bloqueios e Desbloqueios**

**O algoritmo também precisa gerenciar quando e por que um processo fica bloqueado. Funções são chamadas para mover processos entre as filas de pronto e espera de acordo com o seu estado (por exemplo, se está aguardando algum evento). Quando um processo bloqueado pode voltar a ser executado, ele é removido da fila de espera e reintroduzido na fila de prontos.**

**Motivo da Chamada:**

**Esse comportamento simula a vida real de processos que podem precisar de recursos externos ou algum evento específico para continuar. A função de bloqueio garante que esses processos não sejam perdidos e sejam reintegrados corretamente ao fluxo de execução quando estiverem prontos novamente.**

**Conclusão**

**A função execucao() é o núcleo que coordena todas as ações do simulador. As chamadas para as funções de manipulação das filas e a exibição dos processos garantem que o escalonamento aconteça de forma ordenada e eficiente. O comportamento do algoritmo imita um sistema operacional real, onde os processos são constantemente movidos entre os estados de prontos, executando e aguardando, enquanto o sistema gerencia esses estados de forma inteligente e eficaz.**