Trabalho de Sistemas Operacionais - Escalonamento por prioridades

Universidade Federal de Alfenas | Professor Fellipe Rey

Pedro Henrique Botelho da Silva - RA: 2023.1.08.027

Diogo da Silva Moreira - RA: 2023.1.08.003

1. OBJETIVO

Este trabalho tem como objetivo a implementação de um escalonador por prioridades, considerando a execução dos processos em filas de diferentes níveis de prioridade. As filas de maior prioridade, neste caso, aquelas com números mais altos, devem ter seus processos executados primeiro. Os demais processos, para evitar *starvation*, terão suas prioridades elevadas após determinado tempo de espera, determinado pelo usuário.

2. LINGUAGEM UTILIZADA

O trabalho foi desenvolvido utilizando a linguagem *Python*, na versão 3.12.4. A escolha dessa linguagem se deve à familiaridade dos desenvolvedores com ela, além de garantir a entrega de um trabalho simples, que pode ser executado em qualquer ambiente, inclusive em *IDEs* online, como o Online Python.

2.1. INSTALAÇÃO DO PYTHON

A instalação do Python pode ser realizada baixando o executável disponível na página oficial de downloads. Em sistemas baseados em *Unix*, o *Python* geralmente já vem instalado por padrão, permitindo que o *script* seja executado sem problemas em qualquer versão superior à 3.5, uma vez que não exige bibliotecas adicionais.

3. IMPLEMENTAÇÃO

A implementação solicita ao usuário que defina valores para o *quantum*, o tempo máximo que um processo pode permanecer sem ser executado, a quantidade de filas de prioridade, a quantidade de processos em cada uma das filas, e os tempos de execução de cada processo. Além disso, cada processo possui um PID e um contador de tempo sem execução. Esse controle serve para garantir a manutenção dos processos, evitando *starvation*, de modo que nenhum processo fique mais tempo do que o definido pelo usuário sem ser executado.

O escalonador altera a prioridade de processos que não estão sendo executados antes que eles excedam o tempo máximo definido. Ele monitora as filas de prioridade, começando pela de maior prioridade, para localizar processos, removê-los da fila para processamento, e verificar se outros processos precisam ter suas prioridades elevadas. Caso positivo, realiza a promoção; caso contrário, nenhuma ação é tomada. Ao final do processamento, o processo é devolvido ao final de sua fila específica. Um processo que teve sua prioridade aumentada permanecerá temporariamente na nova fila até ser executado uma única vez, sendo então devolvido à sua fila original.

4. SAÍDA

O programa possui uma saída dinâmica, exibindo, a cada intervalo de tempo relevante, as ações realizadas. Além disso, as filas de processos são constantemente atualizadas, indicando na tela sempre que um processo sofre alteração de prioridade ou é processado.

É importante notar que, em nossa implementação, um processo em execução está dentro do processador e, por isso, não é exibido na fila de processos. Isso simula o cenário real, no qual o processo é removido da fila para ser executado. Assim, é possível observar que ele retorna à fila quando o escalonador passa a execução para outro processo.

5. EXECUÇÃO

Para executar o programa, basta rodar o *script* em uma *IDE* online ou, no terminal, digitar:

```
python3 main.py
```