

Trabalho Prático 2 - Turma 030 - 2023/1

Laboratório de Sistemas Operacionais - Prof. Miguel Xavier



Relatório com gráficos de desempenho - Escalonadores de Disco

Cassiano Luis Flores Michel - 20204012-7

Mateus de Carvalho de Freitas - 20204015-7

Pedro Menuzzi Mascaró - 20103702-5

Gustavo Geyer Arrussul Winkler dos Santos - 19102825-7

Porto Alegre, 16 de outubro de 2023

## Ambiente

Foi solicitado a implementação de um módulo escalonador de disco para o buildroot implementado durante o decorrer das aulas de laboratório de sistemas operacionais. Este módulo deveria funcionar no Linux kernel versão 4.13.9 e deveria implementar a política SSTF (Shortest Seek Time First). Durante o decorrer deste relatório iremos demonstrar como foram implementadas tais tarefas requisitadas para o trabalho prático 2 da disciplina.

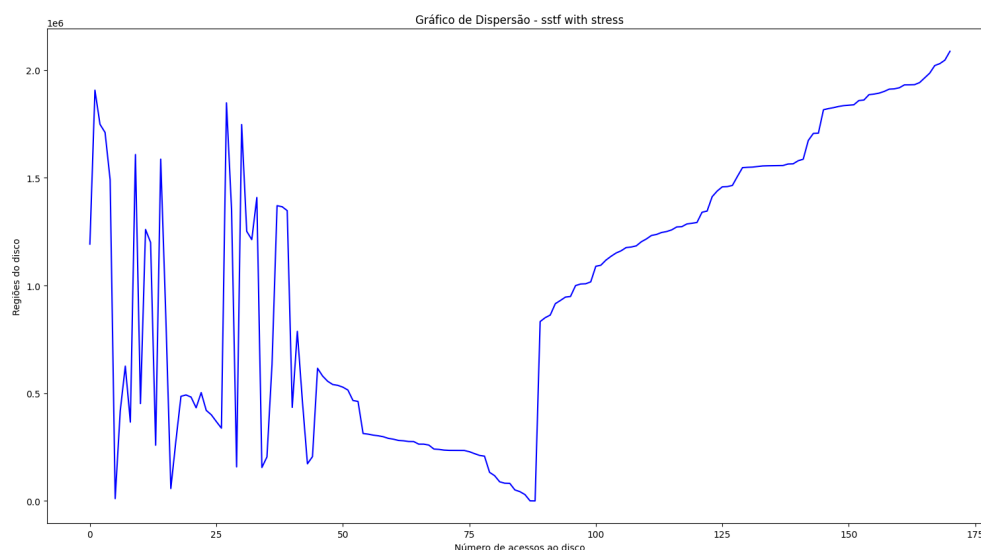
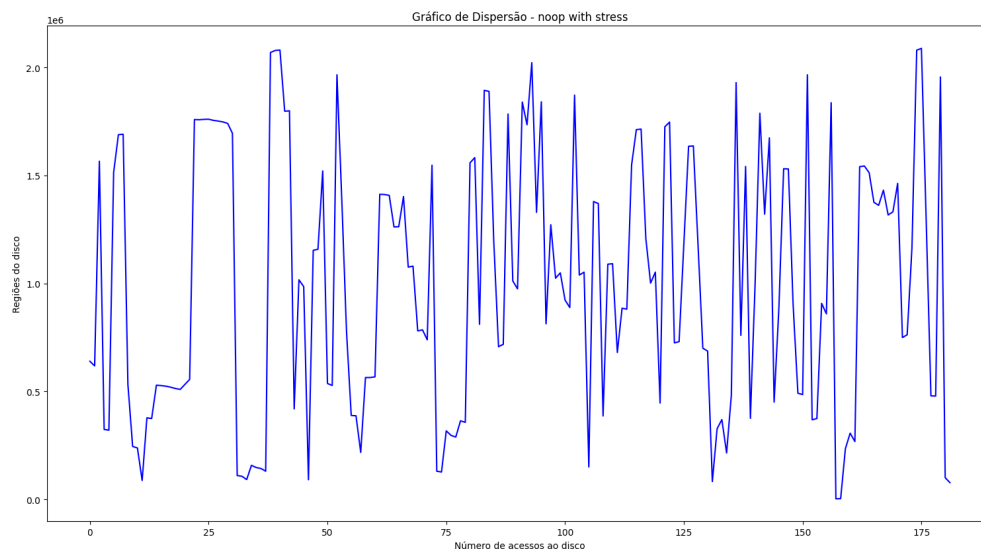
Primeiramente, para que possamos implementar um escalonador de disco, é necessário que façamos uma preparação do ambiente no buildroot, e para isso criamos a imagem de disco assim como descrito no tutorial “2.4: Noções sobre IO-Scheduler e acesso ao disco” disponível no moodle da disciplina.

Tendo o ambiente preparado, copiamos os arquivos base deixados pelo professor na especificação do trabalho, onde fornecem toda estrutura de implementação do módulo de escalonador de disco, nos deixando apenas com a tarefa de implementar a política de SSTF para o mesmo. Para compilar o módulo, foi adicionado um novo diretório no caminho “./buildroot/modules”, representando o módulo SSTF a ser compilado e contendo a estrutura necessária para o escalonador.

A estrutura fornecida possui também um arquivo chamado “sector\_read.c”, no qual é responsável por gerar diversas requisições de acesso ao disco, simulando um ambiente multiprogramado. Por fim, executamos o qemu passando como parâmetro o disco que iremos utilizar (imagem gerada no tutorial 2.4) e registramos o módulo do escalonador para que possamos aplicar os testes necessários para verificação.

Para o correto funcionamento do escalonador SSTF, no qual procura atender sempre a requisição mais próxima da posição atual do leitor, foram adicionadas algumas linhas de código na estrutura original fornecida pelo professor, para que a política de funcionamento pudesse ser comparada com o escalonador NOOP, no qual não possui tratamento das requisições. Foram alterados apenas dois métodos da estrutura original: *sstf\_add\_request()* e *sstf\_dispatch()*.

## Resultados



Podemos observar que o escalonador SSTF só foi efetivo quando de fato a fila de requisições ficou congestionada, visto que nas primeiras requisições ele ainda estava atendendo em ordem de chegada pois assim que chegavam já eram despachadas.

## Considerações

Vale ressaltar que, assim como no trabalho anterior, todas implementações, testes e resultados foram obtidos no nosso ambiente remoto presente no codespace. Além deste relatório, está sendo entregue o arquivo .c com a implementação solicitada e esta também se encontra em nosso repositório do github, dentro da pasta ***labsisop-buildroot-grp-05/programs/tp2***.

Este repositório pode ser acessado através do link abaixo:

<https://github.com/cassiano-flores/labsisop-buildroot-grp-05>