

Relatório de Sistemas Operacionais II N

Nome: Bruno Ferreira Aires

Nome: Bruno Grohs Vergara

Nome: Erick Larratéa Knoblich

Cartão: 323680

Cartão: 324256

Cartão: 324422

Trabalho Prático: Parte 2

PROF. ALBERTO EGON SCHAEFFER FILHO

Universidade Federal do Rio Grande do Sul
Instituto de Informática

Introdução

Neste relatório, mostraremos como foi continuado o desenvolvimento a partir da implementação prévia realizada na Etapa 1 do trabalho prático da cadeira INF01151 - Sistemas Operacionais II N, cujo objetivo foi a implementação de um serviço semelhante ao Dropbox, permitindo o compartilhamento de arquivos entre diferentes dispositivos por meio da rede. Na Etapa 2, tivemos como objetivo implementar um sistema de replicação passiva, utilizando uma eleição de líder para o caso do servidor considerado principal cair. Será feita uma descrição básica dos ambientes de desenvolvimento, principais métodos e técnicas para a implementação das novas funcionalidades, como algoritmo de eleição escolhido e sincronização entre servidores principal e secundários.

Ambientes de Desenvolvimento

Para o desenvolvimento do trabalho, os testes foram realizados com as seguintes máquinas, todas utilizando gcc com versão C++20 ou superior:

- Sistema operacional Ubuntu 24.04 LTS, Processador Intel Core i5-14600K e 32GB de memória. Desenvolvimento e testes realizados utilizando o próprio S.O.
- Sistema operacional macOS Sonoma v14.6.1, Processador Apple M2 e 8GB de memória. Desenvolvimento e testes realizados utilizando Docker integrado ao Visual Studio Code, executando Ubuntu 22.04 de fundo.
- Sistema operacional Windows 11 Pro, Processador Intel Core i7-12700 e 16GB de memória. Desenvolvimento e testes realizados utilizando a extensão WSL do Visual Studio Code, que executa Ubuntu 22.04 de fundo.

Algoritmo de Eleição de Líder

Como algoritmo de eleição de líder, foi escolhido o algoritmo do anel, em que é criado um círculo virtual entre todos os servidores existentes, onde cada servidor somente se comunica com o seu vizinho seguinte e o servidor com maior ID vence a eleição. O servidor que inicia a eleição é o que detecta a falha do servidor líder através de um sistema de *heartbeat*, onde a cada 15 segundos, o servidor principal envia um tipo de pacote específico para os servidores secundários informando-os de que ele está ativo. Caso uma resposta não seja recebida em tempo viável, uma nova eleição será iniciada.

Cada servidor envia o maior ID que ele tem conhecimento ao servidor vizinho, seja o seu ou o recebido do servidor anterior no círculo. O algoritmo encerra quando o ID recebido pelo servidor for igual ao seu, significando que o maior ID entre todos é o dele.

A escolha por esse algoritmo de eleição baseou-se em principal na sua simplicidade, já que suas características já seriam suficientes para a necessidade aplicada nesse projeto, como por exemplo o uso de poucos servidores secundários, o que impede que a alta complexidade do algoritmo nos piores casos prejudique o desempenho dos servidores e do projeto como um todo.

Implementação da Replicação Passiva

A replicação passiva foi implementada no sistema a partir de um servidor líder, definido inicialmente na criação do servidor em linha de comando utilizando uma *flag*, e servidores de *backup*, também definidos inicialmente em linha de comando usando uma *flag* e informando o IP e porta do servidor líder.

A replicação é realizada a cada ação do cliente e servidor principal, portanto, no momento em que o servidor realiza uma ação, seja de receber um arquivo ou de deletá-lo, essa ação é imediatamente replicada aos servidores secundários. O servidor principal mantém uma lista com todos os servidores de *backup* ativos para utilizar na replicação, além da lista de clientes que já existiam previamente na Etapa 1. Todas essas informações são compartilhadas para os servidores secundários para o caso de falha do servidor primário e correto reestabelecimento da conexão entre todos os clientes.

Para garantir que o servidor principal ainda está operacional, foi implementado um sistema de *heartbeat*, como foi citado anteriormente, e caso não esteja, um novo servidor é eleito como líder para continuar o processo. Após a eleição de um novo líder, é necessário reconectar os clientes com o novo servidor. Para isso, o novo servidor líder envia aos clientes suas informações, que se conectam ao novo servidor líder, recomeçando o processo novamente. É fundamental que os servidores secundários estejam a par com o principal, mantendo informações como clientes conectados, servidores secundários conectados, entre diversas outras para garantir consistência entre todos.

Dificuldades Encontradas

Inicialmente, a maior dificuldade encontrada foi estruturar um plano de como criar a replicação passiva. A tarefa parecia muito complexa, porém, a medida que estudamos melhor o projeto, percebemos maneiras para factualizar as ideias.

Na parte mais técnica, uma das principais dificuldades foi a reconexão dos clientes com o servidor líder após uma eleição, porém foi solucionado ao tornar os clientes "servidores temporários" para receber as informações do novo líder.

A falta de familiaridade com a linguagem C++ continuou sendo um problema, porém menos grave, já que com a experiência obtida no desenvolvimento da Etapa 1 nos acostumamos mais com a linguagem. O restante das funcionalidades foram simples de se implementar, como o algoritmo de eleição e a conexão entre os servidores, por exemplo.