Integrantes:

Eduardo André Leite | Matrícula: 00287684.

Henrique Borges Manzke | Matrícula: 00326970.

Pedro Arejano Scheunemann | Matrícula: 00335768.

1. **Introdução.**

Este relatório apresenta a implementação da segunda parte do projeto de "Gerenciamento de Sono" de estações de trabalho em um mesmo segmento de rede física. O objetivo deste serviço é garantir que as estações de trabalho, ao serem colocadas em modo de sono, ainda permitam que os colaboradores possam acordá-las remotamente para acessar serviços específicos, através de uma estação líder. Nesta segunda etapa, foi adicionado um esquema de replicação passiva da tabela de participantes do serviço, e o algoritmo do valentão para eleição de líder.

1. **Descrição do ambiente de testes.**

Todos os testes foram realizados no laboratório 202, prédio 43424, do Instituto de Informática. Todas as máquinas com Ubuntu, e compilador gcc.

1. **Explicação e justificativa do que foi implementado.**

**3.1. Algoritmo de eleição de líder.**

Foi decidido implementar o algoritmo do valentão, já que é projetado para lidar com falhas e recuperações de processos de forma eficaz, dito que quando um processo falha ou se torna inacessível, os outros processos são capazes de iniciar uma nova eleição e eleger um novo líder rapidamente, além de evitar a eleição de múltiplos líderes simultaneamente. Mas o principal motivo é por ser relativamente simples de entender e implementar.

Assim que um manager não é detectado na rede, a estação inicia uma eleição. Esta estação que inicia a eleição envia pacotes do tipo ELECTION para as outras estações – que possuem IP maior – na porta PORT\_ELECTION\_SERVICE e espera ser respondida por cada uma. Se não receber nenhuma resposta, como nos casos em que não enviou nenhuma mensagem (quando a lista de participantes está vazia), ela se autointitula líder. Se recebeu uma resposta do tipo ANSWER, fica esperando uma mensagem de algum líder, dentro de um timeout. Nesse caso, se não receber mensagem do líder, reinicia a eleição.

Ademais, as estações participantes recebem e respondem pacotes do tipo ELECTION que chegaram na porta PORT\_ELECTION\_SERVICE. Temos um loop para esperar receber mensagens de ELECTION nesse endereço, e enviar de volta uma mensagem do tipo ANSWER. Depois de enviar a resposta, iniciamos uma eleição, caso já não tivermos em uma.

* 1. **Replicação passiva.**

Na replicação passiva, uma cópia exata da tabela é mantida nas estações participantes (servidores secundários), mas elas apenas acompanham as atualizações feitas na estação manager (servidor primário). Isso é vantajoso, já que, em caso de falha da estação manager, uma das estações participantes pode ser promovida rapidamente para assumir o papel, através da eleição de um novo líder, utilizando a cópia da tabela que é mantida.

Tanto no serviço de descoberta, quanto no de monitoramento, a estação manager envia a tabela nas mensagens, e as estações participantes a recebem.

Como estamos lidando com troca de mensagens via sockets UDP, elas possuem um tamanho limitado, e é necessário convencionar este tamanho, visto que é preciso especificar o tamanho da área de recepção. Nesse contexto, a principal dificuldade para implementar a replicação passiva, foi encontrar uma solução para enviar a tabela na struct da mensagem, que é um objeto com tamanho variável. Tivemos que criar uma lógica para serializar e desserializar a tabela, para assim conseguirmos definir um tamanho máximo para as mensagens.

**4. Relato dos problemas encontrados durante a implementação e como estes foram resolvidos.**

Nosso grupo havia implementado a estrutura básica deste trabalho antes de realizarmos os testes, uma vez que nenhum dos três membros tinha como testar o algoritmo fora do ambiente da faculdade. Entretanto, subestimamos a dificuldade da segunda parte da entrega (após termos tido um desempenho positivo na primeira), o que nos levou a adiar os testes práticos do programa para o dia 11/09. Infelizmente, durante os testes, o programa apresentou um erro de "segmentation fault" ao tentar ler a tabela enviada pelo líder do serviço. Isso pelo problema que já foi apontado como dificuldade para a implementação da réplica passiva no item 3.2 deste relatório. Nesse momento, decidimos que seria necessário transformar o pacote enviado em uma estrutura de tamanho fixo. A tarefa de serialização e desserialização exigida para essa modificação acabou consumindo bastante tempo, forçando-nos a abandonar o laboratório da faculdade e concluir a implementação em nossas residências, onde não tínhamos mais a capacidade de realizar testes adicionais.