

INF01151 – SISTEMAS OPERACIONAIS II N
SEMESTRE 2020/1
TRABALHO PRÁTICO PARTE 2: REPLICAÇÃO E ELEIÇÃO DE LÍDER

ESPECIFICAÇÃO DO TRABALHO

A proposta de projeto é implementar um serviço de troca de mensagens instantâneas, para permitir a comunicação síncrona entre vários usuários em grupos. A proposta deve ser desenvolvida em duas etapas. A primeira etapa compreendeu funcionalidades implementadas usando conceitos como threads, processos, comunicação e consistência. Na segunda etapa, a equipe deverá estender o serviço com algumas funcionalidades avançadas, onde destaca-se: esquema de replicação com eleição de líder.

A aplicação deverá executar em ambientes Unix (Linux), mesmo que tenha sido desenvolvida em outras plataformas. O projeto deverá ser implementado em C/C++, usando a API Transmission Control Protocol (TCP) sockets do Unix. Quando apropriado, a especificação abaixo define funções e trechos de código que você deverá implementar. No entanto, é possível estender estas funções caso for julgado necessário, destacando as modificações no relatório final a ser entregue.

FUNCIONALIDADES AVANÇADAS

Na segunda parte deste projeto, a aplicação deverá ser estendida para atender os conceitos estudados na segunda metade da disciplina: replicação e segurança.

1. Replicação Passiva

O servidor implementado, caso venha a falhar, levará à indisponibilidade de serviço para seus clientes. Este é um estado que não é desejável no sistema. Para aumentar a disponibilidade do sistema na ocorrência de falhas do servidor principal, um novo servidor deverá assumir o seu papel e manter o serviço de troca de mensagens instantâneas funcionando. Note que essa mudança deve ser transparente para os usuários, e suas mensagens devem ser enviadas nos grupos mesmo após a falha. Além disso, novos usuários que entrem nos grupos devem receber normalmente o histórico de mensagens. Para garantir que os grupos, os usuários pertencentes a cada grupo, e o histórico das mensagens enviadas no grupo¹ estarão disponíveis no novo servidor, você deverá utilizar um esquema de **replicação**, informando todas as modificações realizadas aos servidores secundários.

Mais especificamente, você deverá implementar um esquema de Replicação Passiva², onde o servidor é representado por uma instância primária de réplica manager (RM), e uma ou mais instâncias de RM secundárias (ou backup). Podemos entender este modelo adicionando um front-end (FE) entre a comunicação do cliente (C) e o conjunto de RMs. O front-end é responsável por realizar a comunicação entre os clientes e o serviço de replicação, tornando transparente para o cliente qual é a cópia primária do servidor (Figura 1).

¹ Lembrando que o servidor deve manter o histórico das últimas N mensagens enviadas em cada grupo, sendo N configurável na inicialização do servidor.

² Replicação (Aula da semana 13, e Capítulo 18 do livro COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. – "Distributed Systems: Concepts and Design" (5th edition), Addison-Wesley, 2012).

Seu esquema precisará garantir que:

- (1) todos os clientes sempre utilizarão a mesma cópia primária;
- (2) após cada operação, o RM primário irá propagar o estado dos arquivos aos RMs de backup;
- (3) somente após os backups serem atualizados o primário confirmará a operação ao cliente.

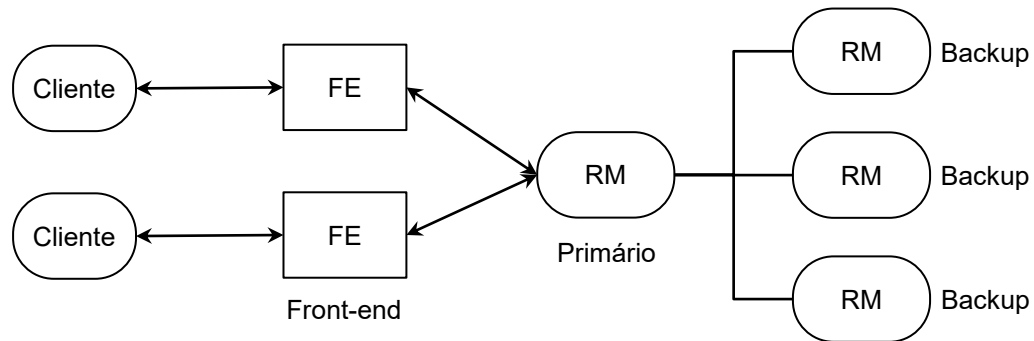


Figura 1 - Esquema de replicação passiva

2. Eleição de Líder

Algoritmos de eleição de líder permitem escolher, dentro de um conjunto de processos distribuídos, qual processo deve desempenhar um papel particular (e.g., coordenador, alocador de recursos, verificador, etc). Algoritmos de eleição de líder são muito usados como parte de outros algoritmos distribuídos, que exigem a escolha de um processo para desempenhar um papel específico. Na primeira parte do trabalho, assumiu-se a existência de um único servidor. Agora, no entanto, como diferentes processos potencialmente poderão assumir o papel de servidor primário, o processo escolhido em questão deverá ser selecionado através de um dos algoritmos de eleição de líder vistos em aula³: algoritmo do anel ou algoritmo do valentão. Quando o servidor principal falhar, o algoritmo de eleição de líder deverá ser utilizado para determinar o próximo servidor primário. Nesse caso, um dos servidores backup deverá assumir essa função, mantendo um estado consistente do sistema. Para isso, implemente um dos algoritmos vistos em aula para eleger um novo RM primário após uma falha. Lembre-se de garantir que seu mecanismo atualize as informações sobre o novo líder nos FE dos clientes.

DESCRIÇÃO DO RELATÓRIO

A equipe deverá produzir um relatório fornecendo os seguintes dados:

- Descrição do ambiente de teste: versão do sistema operacional e distribuição, configuração da máquina (processador(es) e memória) e compiladores utilizados (versões).
- Apresente claramente no relatório uma descrição dos pontos abaixo:
 - O funcionamento do algoritmo de eleição de líder implementado, justificando a escolha;
 - A implementação da replicação passiva na aplicação, incluindo os desafios encontrados;
- Também inclua no relatório problemas que você encontrou durante a implementação e como estes foram resolvidos (ou não).

A nota será atribuída baseando-se nos seguintes critérios: (1) qualidade do relatório produzido conforme os itens acima, (2) correta implementação das funcionalidades requisitadas e (3) qualidade do programa em si (incluindo uma interface limpa e amigável, documentação do código, funcionalidades adicionais implementadas, etc).

DATAS E MÉTODOS DE AVALIAÇÃO

A segunda parte do trabalho deve ser feita considerando as mesmas equipes da primeira parte. As pessoas participantes da equipe devem estar claramente identificadas no relatório e na apresentação. A avaliação do trabalho será pela análise da implementação, do relatório produzido e da apresentação.

³ Algoritmo do anel ou algoritmo do valentão (Aula da semana 10, e Capítulo 15 do livro COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. – "Distributed Systems: Concepts and Design" (5th edition), Addison-Wesley, 2012).

Conforme a consulta à turma durante o primeiro encontro síncrono, a apresentação deverá ser pré-gravada e disponibilizada via YouTube (como vídeo público ou não listado, conforme preferência da equipe). O uso de câmera durante a apresentação de cada participante é opcional. Ao disponibilizar os vídeos das apresentações, os participantes da equipe declaram-se cientes das questões relacionadas ao direito de imagem vigentes para o ERE – UFRGS (Informações sobre Direitos Autorais e de Imagem).

Faz parte do pacote de entrega os códigos-fonte da implementação, um tutorial de como compilar e executar os códigos, o relatório em um arquivo ZIP e o link no YouTube para o vídeo com a apresentação da equipe. A implementação deve estar funcional para a avaliação. A compilação deverá ser feita via scripts automatizados (por exemplo, Makefile). Os links das apresentações serão disponibilizados no Moodle para acompanhamento pela turma. Os vídeos deverão ter duração de 20 minutos.

O trabalho deverá ser entregue até às 23:55 do dia 18 de novembro via Moodle (link para submissão na Semana 10). Após a data de entrega, o trabalho deverá ser entregue via e-mail para weverton.cordeiro@inf.ufrgs.br (subject do e-mail deve ser "INF01151 Turma A: Trabalho Parte 1"). Neste caso, será descontado 02 (dois) pontos por semana de atraso. O atraso máximo permitido é de uma semana após a data prevista para entrega. Isto é, nenhum trabalho será aceito após o dia 25 de novembro. Casos excepcionais poderão ser discutidos com o professor.

DÚVIDAS, QUESTIONAMENTOS E SUGESTÕES

Dúvidas, questionamentos e sugestões podem ser enviadas por e-mail, no fórum da disciplina no Moodle, ou via Microsoft Teams.