

INF01151 – SISTEMAS OPERACIONAIS II N – Turma A
PROF. WEVERTON CORDEIRO
SEMESTRE 2022/1
TRABALHO PRÁTICO PARTE 2: REPLICAÇÃO E ELEIÇÃO DE LÍDER

ESPECIFICAÇÃO DO TRABALHO

A proposta de trabalho prático é implementar um serviço de “gerenciamento de sono” (*sleep management*) de estações de trabalho que pertencem a um mesmo segmento de rede física de uma grande organização. O objetivo do serviço é promover a economia de energia ao incentivar que os colaboradores coloquem suas estações de trabalho para dormir após o término do expediente na organização. Neste caso, o serviço deve garantir que as estações poderão ser acordadas caso o colaborador(a) desejar acessar, remotamente, um serviço na sua estação de trabalho (por exemplo, um serviço de compartilhamento de arquivos, para acessar arquivos pessoais).

A proposta deverá ser desenvolvida em duas etapas. A primeira etapa compreendeu funcionalidades implementadas usando conceitos como *threads*, processos, comunicação e sincronização. 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 User Datagram Protocol (UDP) sockets do Unix.

FUNCIONALIDADES AVANÇADAS

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

1. Replicação Passiva

A estação de trabalho *manager*, caso venha a falhar, levará à indisponibilidade do serviço de gerência de sono. Este é um estado que não é desejável no serviço. Para aumentar a disponibilidade do serviço na ocorrência de falhas do *manager*, um novo *manager* deverá assumir o seu papel e manter o serviço operacional. Note que essa mudança deve ser transparente para os participantes, os quais devem ser notificados de qual o novo *manager*. Para garantir que os participantes estarão disponíveis no novo *manager*, você deverá utilizar um esquema de replicação, informando todas as modificações realizadas aos *managers* secundários. Lembre-se: cada participante do serviço é um *manager* em potencial.

Mais especificamente, você deverá implementar um esquema de Replicação Passiva, onde o *manager* é representado por uma instância primária de *replica manager* (RM), e uma ou mais instâncias de RM secundárias (ou *backup*). Seu esquema precisará garantir que:

- (1) todos os participantes sempre utilizarão a mesma cópia primária (RM primário);
- (2) após cada operação, o RM primário irá propagar o estado aos RMs de backup.

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 *manager*, indicado por linha de comando. Agora, no entanto, como diferentes participantes potencialmente poderão assumir o papel de *manager*, o processo escolhido em questão deverá ser selecionado através do algoritmo de eleição de líder valentão. Quando o *manager* principal falhar ou for colocado para dormir, o algoritmo de eleição de líder deverá ser utilizado para determinar o próximo *manager*. Nesse caso, um dos *participantes* deverá assumir essa função, mantendo um estado consistente do sistema. Para isso, implemente o algoritmo do valentão para eleger um novo RM primário após uma falha ou o *manager* ir dormir. Lembre-se de garantir que seu mecanismo atualize as informações sobre o novo líder nos participantes.

DESCRIÇÃO DO RELATÓRIO

A equipe deverá produzir um relatório fornecendo os seguintes dados:

- Descrição do ambiente de testes: versão do sistema operacional e distribuição, configuração da máquina (processador(es) e memória) e compiladores utilizados (versões).
- Explicação e respectivas justificativas a respeito de:
 - (A) funcionamento do algoritmo de eleição de líder implementado, justificando a escolha;
 - (B) A implementação da replicação passiva na aplicação, incluindo os desafios encontrados;
- Também inclua no relatório uma descrição dos problemas que a equipe 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

O trabalho deve ser feito em grupos de **QUATRO INTEGRANTES**. 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. A ausência de um(a) integrante da equipe no dia da apresentação implicará em conceito zero para a pessoa ausente (salvo por motivos excepcionais como por ex. de saúde, que deverão ser registrados via junta médica da UFRGS).

O trabalho deverá ser entregue até às 23:55 do dia 3 de outubro via Moodle (link para submissão no Tópico 1). O atraso máximo permitido é de uma semana após a data prevista para entrega, com desconto de 20% no conceito final. Nenhum trabalho será aceito após o dia 10 de outubro.

A apresentação dos trabalhos e demonstração prática em laboratório dos sistemas implementados será realizada presencialmente, nos dias 4 e 6 de outubro, conforme agenda a ser divulgada.

Faz parte do pacote de entrega os códigos-fonte da implementação, um tutorial de como compilar e executar os códigos e o relatório em um arquivo ZIP. A implementação deve estar funcional para demonstração durante a apresentação pela equipe, em laboratório. A compilação deverá ser feita via scripts automatizados (por ex., Makefile), de modo a facilitar o processo de avaliação do projeto submetido.

DÚVIDAS, QUESTIONAMENTOS E SUGESTÕES

Dúvidas, questionamentos e sugestões podem ser enviados por e-mail ou via Microsoft Teams.

REFERÊNCIAS

[1] Sen, Siddhartha, Jacob R. Lorch, Richard Hughes, Carlos Garcia Jurado Suarez, Brian Zill, Weverton Cordeiro, and Jitendra Padhye. "Don't Lose Sleep Over Availability: The GreenUp Decentralized Wakeup Service." In 9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12), pp. 211-224. 2012.