

Trabalho Prático 1 – PPD Calendar

Neste primeiro trabalho da disciplina de Programação Paralela e Distribuída, você terá que implementar uma aplicação cliente/servidor, o **PPD Calendar**. O objetivo deste sistema é oferecer um serviço semelhante ao *Google Calendar* ou *Exchange Server*, onde usuários possuem uma agenda do servidor e podem criar eventos nas suas agendas, ou convidar outros usuários para participação em eventos de usuários.

O sistema deverá ser composto por um servidor que deverá gerenciar todos os usuários e suas respectivas agendas. O sistema deve prover a persistência das informações, isto é, caso o servidor será encerrado, ele deve ser capaz de retornar ao último estado (usuários cadastrados e respectivas agendas) quando for reiniciado.

O servidor deve ser iniciado com um usuário administrador com login “admin” e senha “admin”. O administrador pode cadastrar e remover usuários dinamicamente. Somente o administrador pode realizar estas operações. A **operação de cadastro** deve receber como parâmetros o login do usuário e uma senha, e a **operação de remoção** de usuário deve receber como parâmetros o login do usuário. O sistema deve prover a consistência dos dados e retornar erro para problemas como: cadastro de usuário já existente, remoção de usuário inexistente, etc.

Para permitir a troca de senha, o servidor deve prover uma **operação para troca de senha** (disponível para todos os usuários, inclusive admin). Essa operação recebe como parâmetros o login do usuário, a senha antiga e a senha nova. O usuário admin pode trocar a senha de qualquer usuário. Cada usuário pode trocar somente a sua própria senha.

O servidor deve prover operações para inclusão de eventos, remoção de eventos, listagem de eventos e aceitação/rejeição de um convite. Os eventos podem ser de 3 tipos diferentes: particular, geral, com convidados. Um **evento particular** deve conter os seguintes itens: título, data/hora e local. Um **evento geral** deve conter, além dos itens do evento particular, a indicação do dono do evento. Um **evento com convidados** deve conter, além dos itens do evento particular, a indicação do dono do evento e uma lista de convidados (nome dos usuários).

A **operação de inclusão de eventos** deve permitir incluir um evento particular, geral ou com convidados, informando os parâmetros necessários para cada tipo. O evento particular é incluído somente na agenda do usuário. O evento geral é incluído na agenda de todos os usuários (inclusive dos que podem ser criados futuramente pelo admin). O evento com convidados é incluído na agenda do usuário que criou o evento, e também dos convidados.

A **operação de remoção de eventos** deve permitir a remoção de um evento. Os usuários podem remover os seus eventos particulares e também os eventos gerais e com convidados que criou. A tentativa de remoção de um evento geral ou com convidados por um usuário que não criou o evento deve retornar erro.

A **operação de listagem de eventos** deve permitir a listagem dos eventos de um usuário. Deve ser permitido ao usuário utilizar os seguintes filtros na listagem de eventos:

- Tipo de eventos: todos os tipos, particular, geral ou com convidados;
- Data do evento: todas as datas, os eventos a partir de hoje, os eventos de uma data específica;
- Confirmação do evento: todos os eventos, eventos rejeitados, eventos não confirmados, eventos confirmados. (se um usuário rejeitar o evento, o evento é considerado rejeitado; se o evento não estiver confirmado por todos os convidados, o evento é considerado não confirmado)

A **operação de aceitação/rejeição de um convite** deve permitir que um usuário convidado aceite ou rejeite o convite. O usuário deve indicar o evento referente e o aceite ou rejeição.

Observações

1. Para este trabalho você deverá utilizar RPC (usando a linguagem de programação C/C++). O sistema operacional fortemente recomendado é o Linux.
2. Não é permitido o uso de *threads*.

3. Deve haver transparência de localização dos processos envolvidos (servidor e clientes). Os processos podem funcionar tanto local como remotamente.

Avaliação

1. 10% Interface (log no servidor, clareza dos comandos e das informações dos clientes)
2. 20% Funcionamento comprovado da comunicação de pelo menos um cliente com o servidor (**comando connect**)
3. 60% Funcionamento dos demais comandos previamente listados
4. 10% Documentação* (estrutura da implementação, execução, outras questões relevantes)

*A parte da nota relativa à documentação só será considerada caso o funcionamento do programa contemple pelo menos a comunicação RPC entre os processos (item 2). A documentação deve **obrigatoriamente** incluir a descrição de como deve ser executado o programa do grupo e uma explicação sobre a implementação.

A **entrega do trabalho será feita por todos os grupos até o dia 06/05**. Além disso, cada grupo deverá fazer uma apresentação em sala de aula laboratório (no dia da entrega). Nesse dia, todos os participantes do grupo deverão estar presentes (participantes ausentes ficarão sem a nota do trabalho 1. O grupo deverá entregar a documentação e enviar para o e-mail do professor o código fonte do programa após a apresentação. **Trabalhos com atraso serão aceitos mediante um desconto de 2.0 pontos na nota final por dia de aula de atraso.**

Este trabalho deverá ser feito em grupo de 3 componentes.