

Trabalho Prático 1 – Share Center

Neste primeiro trabalho da disciplina de Programação Paralela e Distribuída, você terá que implementar uma aplicação cliente/servidor, o **share center**. O servidor implementa um sistema de compartilhamento de arquivos com no máximo 4 clientes conectados (na mesma máquina ou em máquinas remotas). Um cliente pode então compartilhar arquivos especificando explicitamente quais arquivos deseja compartilhar com o comando `\share` (veja a tabela de comandos abaixo para mais detalhes). O comando `share` irá fazer o *upload* de um arquivo localizado no disco do cliente para o servidor.

Comando do cliente	Significado
<code>\share <nome_do_arquivo></code>	Compartilha um arquivo fazendo upload no servidor.
<code>\list</code>	Solicita a lista de todos os arquivos compartilhados no momento pelo servidor, mostrando-a na tela.
<code>\download <nome_do_arquivo></code>	Especifica o nome de um arquivo que deseja baixar, fazendo o download do servidor.
<code>\emptyqueue</code>	Tenta fazer o envio dos arquivos que estão na fila de espera de um dado cliente. Estes arquivos ainda não foram compartilhados por falta de espaço no servidor.
<code>\leave</code>	Termina a conexão com o servidor neste momento todos os arquivos compartilhados pelo cliente devem ser removidos do repositório.

Quando um cliente deseja saber se um arquivo existe no **Share Center** ele deve usar um comando `\list` que lista os arquivos disponíveis e os respectivos usuários que compartilharam esses, como no exemplo abaixo:

Nome do arquivo	Usuário	Tamanho
livro_de_cpp.pdf	<id_cliente_1>	20 bytes
musica1.mp3	<id_cliente_2>	135 bytes
video1.avi	<id_cliente_1>	120 bytes
zelda_walkthrough.txt	<id_cliente_3>	55 bytes

Para um cliente realizar o *download* de um arquivo ele deve explicitamente informar com o comando `\download <nome_arquivo>` qual arquivo ele deseja baixar. Podendo solicitar vários downloads, porém somente um acontece por vez.

O servidor tem um repositório em disco (usando NFS) que permite no máximo **1 KByte** armazenado ao mesmo tempo. Se um cliente deseja compartilhar um arquivo e não há espaço suficiente, o *upload* (`share`) é enfileirado no cliente até que espaço seja liberado. O comando `\emptyqueue` será usado para esvaziar a fila de cada cliente. Os arquivos somente são removidos do repositório quando o cliente que os havia compartilhado executa o comando `\leave`. Este comando deve ser executado sempre que um cliente for finalizado para evitar inconsistências.

Observações

1) Para implementar este trabalho você deverá utilizar RPC (usando a linguagem de programação C/C++). O sistema operacional fortemente recomendado é o Linux.

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

3) A identificação de cada cliente deve ser feita quando da ativação do mesmo, ou seja vai ser passada como parâmetro na linha de comando que ativará o cliente.

4) É necessário gravar em disco os arquivos no repositório do Servidor (num diretório chamado **RepServ**) e cada cliente deverá ter um diretório para armazenar seus arquivos baixados/compartilhados (diretório com um nome diferente para cada cliente – o mesmo usado para identificar o cliente). Não há necessidade dos diretórios serem criados dinamicamente pelo programa, eles podem ser criados no momento da apresentação.

4) Um arquivo não pode ser maior do que o tamanho máximo estipulado em 1 KB para repositório do servidor. Assim como, a soma total dos arquivos armazenados no repositório também não pode exceder este limite.

5) Independente das extensões dos arquivos (.pdf, .jpg, .mp3), todos arquivos serão do tipo texto para facilitar a transferência das informações.

Avaliação

10% Interface (log no servidor, comandos dos clientes)

50% Funcionamento do Programa (implementação correta das funcionalidades solicitadas)

20% Tratamento de erros (consistência)

20% Documentação* (estrutura 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. 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 consistirá de uma apresentação a ser realizada no LAPRO no dia indicado na página da disciplina (24/05). Nesse dia, todos os participantes do grupo deverão estar presentes. Além disso, 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 poderá ser feito em grupo por no máximo 2 componentes.