

Sistemas Distribuídos

Trabalho Prático 2

Professor: Nuno Preguiça

Trabalho realizado por:

41717 – Rui Filipe Monteiro Queirós

42077 – Henrique Olivera Alves Garcês

Opções implementadas:

Base REST - Crie um servidor que possa ser adicionado ao sistema implementado no trabalho 1 e funcione como um proxy para um serviço de armazenamento remoto de ficheiros - Dropbox.

Sincronização - Implemente um mecanismo de sincronização de um directório local com um directório remoto.

Extra Rest – Crie um servidor adicional que possa ser adicionado ao sistema implementado no trabalho 1 e funcione como proxies para outro serviço de armazenamento remoto de ficheiros - Google Drive.

Cenário de utilização esperado:

O cenário de utilização esperado é através de JAVA RMI com os clientes ligados à mesma rede do servidor.

Dropbox e Google Drive:

A utilização da dropbox funciona para o cliente como outro servidor qualquer que já havia previamente no primeiro trabalho. É apenas necessário fornecer à aplicação um Token de autorização de acesso.

Todos os métodos que estavam disponibilizados para o servidor do primeiro trabalho, também estão disponíveis para a dropbox.

ls – Listagem de directoria – (GET,
"https://api.dropbox.com/1/metadata/sandbox/"+path+"?list=true")

mkdir – Criação de directoria – (POST, POST,
"https://api.dropbox.com/1/fileops/create_folder"+root+dir)

rm – Remove ficheiro – (POST, “https://api.dropbox.com/1/fileops/delete”+root+path)

rmdir – Remove directoria – Semelhante à remoção de ficheiro

cp – Cópia de ficheiro para o servidor – (PUT, “https://api-content.dropbox.com/1/files_put”+root+path)

getFile – Recebe um ficheiro do servidor – (GET, “https://api-content.dropbox.com/1/files”+root+path)

getAttr – Recebe a informação sobre o ficheiro – (GET, “https://api.dropbox.com/1/metadata/” +root+path)

Google Drive

A utilização do Google Drive é semelhante à da dropbox, sendo que também para este é necessário a utilização de um token de acesso para que seja possível aceder à aplicação.

ls – Listagem de directoria – (GET, https://www.googleapis.com/drive/v2/files?trashed=false)

mkdir – Criação de directoria – (POST, https://www.googleapis.com/drive/v2/files)

rm – Remove o ficheiro - DELETE, “https://www.googleapis.com/drive/v2/files/”+ID_do_Ficheiro)

rmdir – Remove a directoria, funciona da mesma forma do anterior

cp – Cópia o ficheiro para o servidor – (POST, “https://www.googleapis.com/upload/drive/v2/files?uploadType=media”)

getFile – Recebe um ficheiro do servidor - GET, “https://www.googleapis.com/drive/v2/files/”+ID_Do_Ficheiro

getAttr – Devolve as informações de um ficheiro - GET, “https://www.googleapis.com/drive/v2/files/”+ID_Do_Ficheiro

Métodos adicionais:

Devido algumas propriedades da Dropbox, nomeadamente a utilização de identificadores de ficheiros, decidimos criar dois métodos auxiliares, o método path que faz um split das directorias e o getID que vai buscar o identificador dos ficheiros.

Sincronização da Dropbox

A sincronização apenas se encontra implementada e funcional com a Dropbox.

Ao executarmos o comando:

```
sync dir server@user:dir
```

O ficheiro vai começar a percorrer as directorias locais e do servidor indicado e comparar os ficheiros. Caso os ficheiros existam na directoria local e não existam na directoria da dropbox, então os ficheiros vão ser copiados para o servidor.

Caso os ficheiros existam nos dois lados, então ele verifica se o ficheiro que se encontra no servidor está diferente do ficheiro local, a nível de data e tamanho, e caso se confirme copia-o para a nossa directoria local acrescentando ao nome do ficheiro “.Local”

Todavia parte da sincronização não funciona, como por exemplo remoções.