

RELATÓRIO DE FINAL DE SEMESTRE

Renan Greca

O *LAN para Sincronização de Diretórios* (daqui em diante abreviado como *LSD*) será um programa multiplataforma para manter diretórios sincronizados através de uma rede local (LAN - *Local Area Network*). Este relatório descreve o início do projeto, desde a motivação da ideia até os desenvolvimentos atuais.

A ideia do *LSD* veio da necessidade de manter diretórios em dois computadores pertencentes ao mesmo usuário sincronizados para oferecer conveniência. Por exemplo, ao chegar em casa com seu *laptop*, os documentos novos neste seriam automaticamente copiados para o *desktop* do usuário. Portanto, essa será a funcionalidade principal do *LSD*: (1) Executa-se duas instâncias do programa em dois computadores em rede, cada uma recebendo um diretório como argumento; (2) Os diretórios são comparados e suas diferenças são analisadas pelo programa; (3) Arquivos são copiados de um diretório para outro até que ambos sejam iguais.

O *LSD* será desenvolvido na linguagem de programação Python, escolhida pelo seu desenvolvimento rápido e facilidade de acesso a recursos de rede. Com sockets TCP, é possível efetuar comunicação entre duas instâncias do programa através da rede local. Por enquanto, dois protótipos estão sendo desenvolvidos: Um deles analisa um diretório e armazena a lista de arquivos (inclusive de sub-diretórios) e as datas de criação e modificação desses; outro envia arquivos específicos de um computador a outro através da rede. Quando ambos protótipos estiverem sólidos, eles serão combinados para realizar a proposta do *LSD*.

Abaixo estão algumas possíveis situações que podem ocorrer ao se comparar diretórios (vamos chamá-los de A e B), e como pretende-se lidar com elas no *LSD*:

1. Um arquivo é encontrado no diretório A, mas não no diretório B. Neste caso, basta copiar esse arquivo de A para B. Há a possibilidade de que o arquivo já existia e

foi excluído de B. Portanto, é necessário comparar as informações de cada diretório com a de si mesmo durante a execução anterior do programa.

2. Um arquivo existe em ambos, mas não é igual. Deve-se comparar as datas de alteração contidas nos metadados dos arquivos. Há a possibilidade de um ser apenas a atualização do outro, mas pode ser que ambos sejam atualizações de uma versão comum anterior. Na eventualidade de um conflito, o programa deverá indagar o usuário sobre como prosseguir.
3. Um arquivo foi renomeado em A mas não sofreu alterações em B. Deve-se verificar qual foi era o nome original do arquivo renomeado e então renomeá-lo também em B.

Durante o desenvolvimento do trabalho, outras situações podem ocorrer durante testes. Elas serão tratadas de forma adequada.

Nos próximos meses, o foco do projeto será concluir e integrar os protótipos para poder efetuar a sincronização. Após isso, num possível passo adicional, será considerada a hipótese de fazer com que o programa detecte automaticamente mudanças no diretório para fazer a sincronização em tempo real.