

UC Sistemas Operacionais

10/08/2021

Prof. Bruno Kimura
bruno.kimura@unifesp.br
ICT/UNIFESP

TRAB_3: Sincronização transparente de arquivos

Objetivo:	Implementar um mecanismo de sincronização local de arquivos transparente.
Metodologia:	Trabalho em grupo de no máximo 3 (três) alunos para desenvolvimento de protótipo em linguagem C.
Entregáveis:	<ol style="list-style-type: none">1. Protótipo em arquivos em C.2. Vídeo-apresentação:<ul style="list-style-type: none">◦ Apresentar explicação do trabalho desenvolvido: o que, como e porque foi feito da forma como foi.◦ TODOS os integrantes do grupo devem apresentar.◦ O tempo de exposição de vídeo é em torno de 20min e deve ser dividido de forma igualitária, por exemplo, em um grupo de 3 alunos, cada teria entre 6min e 7min de fala.◦ Descrever de forma objetiva as contribuições de cada integrante do grupo no trabalho.◦ Apresentar uma auto-avaliação do grupo sobre o trabalho realizado.
Data de entrega:	18/08/2021
Observação:	A autenticidade do trabalho será verificada. Cópias (entre grupos e/ou de fontes da Internet) implicam em anulação.

Descrição do trabalho:

Ferramentas de sincronização de arquivos são de extrema utilidade, como *google drive*, *icloud*, *dropbox*. Essencialmente, essas ferramentas realizam a verificação periódica da alteração de todos os arquivos em um dado diretório, de modo que somente os arquivos que foram alterados recentemente é que são sincronizados com o diretório destinatário, tipicamente na nuvem.

Neste trabalho, a tarefa é implementar parte das operações de um serviço de sincronização transparente. Considere que a sincronização irá ocorrer localmente, de uma pasta origem */minhapasta* para uma pasta de destino */backup_minhapasta*, ambas na mesma máquina. Embora não seja confiável (do ponto de vista da persistência dos dados) uma sincronização cujo o destino é local, vamos considerá-lo nesta prática apenas para fins de redução de escopo, evitando a programação de comunicação de rede necessária para implementar o *backup* em um nó remoto.

As principais operações previstas são:

- **Verificação periódica dos tempos dos arquivos.** A cada x ms, verifica-se as informações de tempo, t_{origem} , de todos os arquivos na sub-árvore de diretórios de origem (*/minhapasta*) e compara-se com os tempos, $t_{destino}$, das cópias na pasta de destino (*/backup_minhapasta*).
- **Sincronização.** Se na última verificação de t_{origem} , um dado arquivo f teve seus tempos alterados em relação ao tempo $t_{destino}$ da sua cópia, então o arquivo f de */minhapasta*

deve ser sincronizado (nesse caso, sobreposto) em `/backup_minhapasta`.

Para organizar os blocos de um arquivo em um meio de armazenamento persistente, o sistema de arquivos utiliza uma estrutura de dados. Em sistemas Unix, o sistema de arquivo utiliza o *i*-node para indexar os blocos de cada arquivo, reunindo informações como: índice único de *i*-node, modos de operação (leitura/escrita/execução), ID do dispositivo de armazenamento, número de links, ID do usuário proprietário, ID do grupo proprietário, tamanho do bloco, número de blocos alocados, **tempos (acesso, modificação e status/criação)**, ponteiros para blocos diretos e indiretos, etc.

Para implementar seu serviço local de sincronização transparente de arquivos, utilize chamadas de sistemas do Linux para a manipulação do sistema de arquivo. Programação concorrente com o uso de processos leves (*threads*) e/ou processos filhos será importante neste trabalho.