

## UC Sistemas Operacionais

28/06/2018

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

### LAB\_5: Serviço local de backup transparente

**Metodologia:** Trabalho individual ou em grupo de no máximo 3 (três) alunos a ser desenvolvido em laboratório de informática através de codificação na linguagem C.

**Data de entrega:** 04/07/18

**Forma de entrega:** Código .c deve ser enviado no moodle. Insira como comentário no código o nome e matrícula de cada integrante do grupo.

**Observação:** Somente serão aceitos trabalhos autênticos. Cópias (entre grupos e/ou de fontes da Internet) serão anuladas.

#### Descrição do trabalho:

Para aumentar a persistência de arquivos é comum realizar cópias de segurança (*backup*). Numa eventual remoção indesejada de arquivos, os mesmos podem ser recuperados de forma íntegra a partir das suas cópias. Atualmente, uma ferramenta amplamente utilizada para essa finalidade é a *dropbox*. Essa ferramenta se tornou um serviço de armazenamento e compartilhamento de arquivos que sincroniza uma pasta do sistema de arquivos local em uma nuvem na Internet. Para isso, a ferramenta realiza a verificação periódica da alteração de todos os arquivos na árvore de diretórios observada a partir de uma pasta selecionada. Somente os arquivos que foram alterados recentemente é que são sincronizados com a nuvem.

Nesta prática, a tarefa é implementar parte das operações de um serviço de backup transparente, como o *dropbox*. Contudo, considere que a sincronização irá ocorrer localmente, de uma pasta origem */minhapasta* para uma pasta de destino */backup\_minhapasta*, ambas no mesma máquina. Embora não se possa considerar confiável um *backup* 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 em *backup* transparente são:

- **Verificação periódica dos tempos dos arquivos.** A cada  $x$  ms, verifica-se as informações de tempo,  $to$ , de todos os arquivos na sub-árvore de diretórios de origem (*/minhapasta*) e compara-se com os tempos,  $td$ , das cópias na pasta de destino (*/backup\_minhapasta*).
- **Sincronização.** Se na última verificação de  $to$ , um dado arquivo  $f$  teve seus tempos alterados em relação ao tempo  $td$  da sua cópia, então o arquivo  $f$  de */minhapasta* deve ser sincronizado (nesse caso, sobreposto) em */backup\_minhapasta*. Note que se  $f$  for um arquivo grande, em escala de MB ou GB, e uma pequena fração de seu conteúdo ter sido alterada, não é eficiente copiar o arquivo todo para a pasta de *backup*, mas apenas o bloco referente à fração alterada de seu conteúdo.

Como visto em aula, para organizar os blocos de um arquivo em um meio de armazenamento persistente, o sistema de arquivos utiliza uma estrutura de dados. Em sistemas baseados em 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 *backup* transparente, utilize chamadas de sistemas do Linux para a manipulação do sistema de arquivo, juntamente com uso de *Pthreads*.