

# Projeto de Sistemas Operativos 2020-21

## Enunciado geral

### LEIC-A/LEIC-T/LETI

Este enunciado apresenta uma visão global do projeto de SO. No futuro serão publicados enunciados detalhados de cada exercício do projeto.

#### Visão global do projeto

O objetivo final do projeto é desenvolver um sistema de ficheiros (*File System*, FS) em modo utilizador (*user-level*) e que mantém os seus conteúdos em memória primária, chamado TecnicoFS.

Os FS em modo utilizador são uma alternativa que tem vindo a ganhar relevância recentemente, já que permitem o desenvolvimento rápido de FS facilmente portáteis e com forte isolamento de falhas, tal como será discutido nas aulas teóricas durante o semestre. Num FS em modo utilizador, as funcionalidades do FS são oferecidas num processo servidor (que, naturalmente, corre em modo utilizador). Outros processos podem chamar funções do FS através de pedidos ao núcleo do Sistema Operativo, que, por sua vez, encaminha esses pedidos ao processo servidor do FS através de um canal de comunicação estabelecido com este. Posteriormente, o retorno da função é devolvido ao cliente invocado pela via inversa.

Ao contrário de FS tradicionais, que guardam a informação em blocos (e.g., num disco magnético ou SSD), o TecnicoFS é desenhado para armazenar o conteúdo dos seus ficheiros e diretorias em memória primária. Pode, por exemplo, ser usado para manter um FS temporário, não persistente, em memória DRAM, beneficiando assim do melhor desempenho desta memória, em comparação com disco/SSD. Pode também aproveitar as novas tecnologias de memória persistente quando usado em máquinas em que estas estejam disponíveis (como, por exemplo, os recentes DIMMs com tecnologia Intel Optane DC).

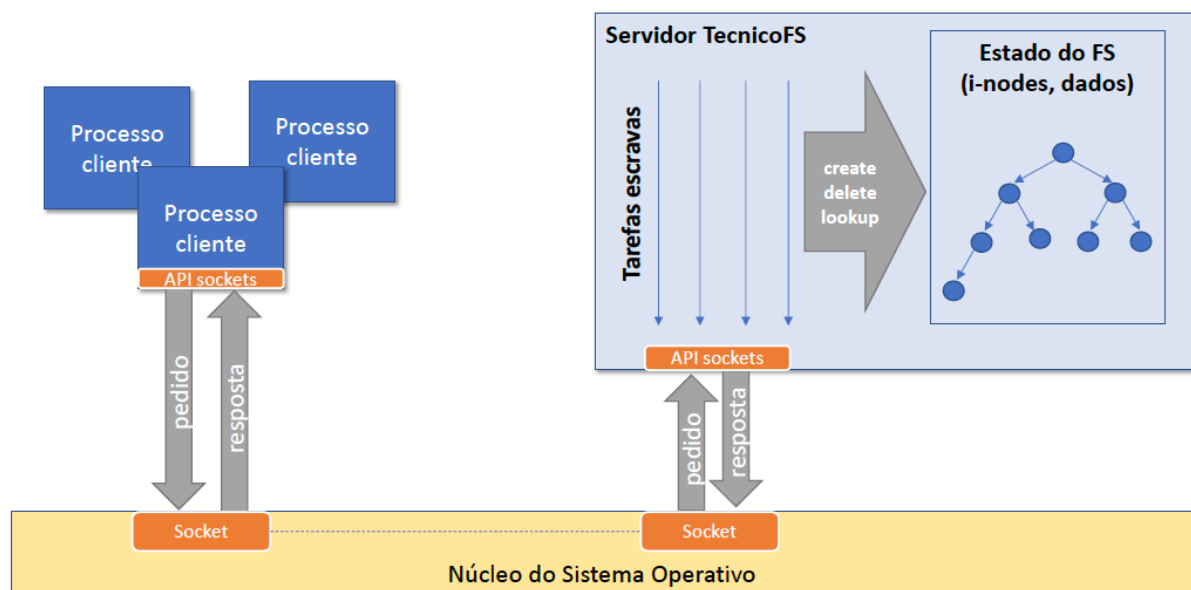


Figura 1: Arquitetura final do TecnicoFS

## Arquitetura

A arquitetura final do TecnicoFS encontra-se ilustrada na Figura 1. Para suportar acessos concorrentes e otimizar o desempenho, as funcionalidades do TecnicoFS são oferecidas, em paralelo, por um conjunto de tarefas escravas no processo servidor do FS.

A API do TecnicoFS oferecerá as seguintes funções básicas:

- Criar, pesquisar, remover e renomear ficheiro/diretoria
- Abrir, ler, escrever e fechar ficheiro

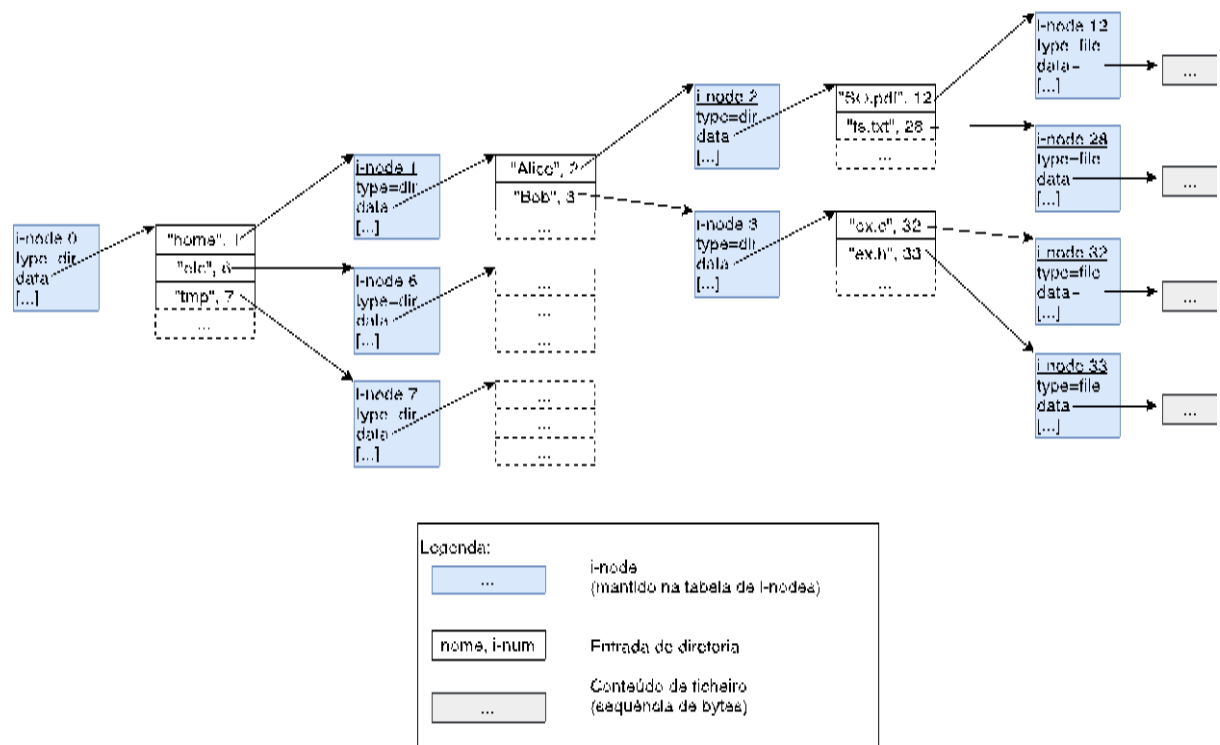


Figura 2: Estado mantido pelo TecnicoFS (exemplo)

## Estado do TecnicoFS

Tal como em FS tradicionais modernos, o conteúdo do FS encontra-se referenciado numa estrutura de dados principal chamada tabela de *i-nodes*, global ao FS. Cada *i-node* representa uma diretoria ou um ficheiro no TecnicoFS, que tem um identificador único chamado *i-number*. O *i-number* de uma diretoria/ficheiro corresponde ao índice do *i-node* correspondente na tabela de *i-nodes*. O *i-node* consiste numa estrutura de dados que descreve os atributos da diretoria/ficheiro (aquilo que normalmente se chamam os seus *meta-dados*) e que referencia o conteúdo da diretoria/ficheiro (ou seja, os *dados*).

Além da tabela de *i-nodes*, existe uma região de dados. Esta região mantém os dados de todos os ficheiros do FS, sendo esses dados referenciados a partir do *i-node* de cada ficheiro (na tabela de *i-nodes*). No caso de ficheiros normais, é na região de dados que é mantido o conteúdo do ficheiro (por exemplo, a sequência de caracteres que compõe um ficheiro de texto). No caso de diretorias, a região de dados mantém a respetiva tabela, que representa o conjunto de ficheiros (ficheiros normais e sub-diretorias) que existem nessa diretoria.

Cada entrada na tabela de uma diretoria consiste num par (*nome*, *i-number*). O conjunto destes pares permite enumerar quais os nomes dos ficheiros incluídos numa diretoria, assim como aceder aos dados e meta-dados desses ficheiros (indexando a tabela de *i-nodes* pelo *i-number* do ficheiro pretendido).

A Figura 1 ilustra o estado mantido pelo TecnicoFS com um exemplo. Neste exemplo, abrir o ficheiro cujo *pathname* é `"/home/Alice/SO.pdf"` implica percorrer os *i-nodes* e as tabelas de diretoria de: diretoria raiz (*i-number* 0); diretoria `"/home"` (*i-number* 1); e diretoria `"/home/Alice"` (*i-number* 2). Após essa pesquisa, o *i-node* do ficheiro desejado será finalmente encontrado. A partir desse *i-node*, o TecnicoFS conseguirá aceder aos dados que compõem o conteúdo desse ficheiro.

## Etapas de desenvolvimento

O TecnicoFS será desenvolvido gradualmente ao longo de 3 exercícios, realizados pelos alunos, e terá uma fase final (não avaliada) que será realizada nas aulas teóricas após a entrega do último exercício. De seguida resumem-se estas etapas.

**Exercício 1:** Desenvolve o servidor do TecnicoFS, composto por uma *pool* de tarefas escravas que executam as operações do FS sobre a diretoria partilhada em memória. Nesta fase, há algumas simplificações importantes: i) apenas são suportadas as operações de criar, pesquisar e remover diretoria/ficheiro; ii) a sincronização dos acessos concorrentes recorre a uma estratégia simplificada com apenas um trinco global; iii) as chamadas a funções do TecnicoFS por parte de processos cliente ainda não são suportadas; em vez disso, as chamadas são simuladas pelo carregamento de um ficheiro contendo sequências de comandos.

**Exercício 2:** Estende a solução anterior com uma sincronização mais fina, o que permitirá maior paralelismo efetivo. Também melhora o carregamento do ficheiro de entrada para permitir que as operações se executem em paralelo com o carregamento inicial do ficheiro. Finalmente, estende a API do FS com a operação de *renomear* ficheiro, com uma semântica atómica.

**Exercício 3:** Estende a solução anterior suportando a invocação de operações do FS por processos cliente. Os processos cliente fazem chamadas ao TecnicoFS enviando mensagens através de um *socket*, que são consumidas pelas tarefas no servidor e que respondem aos clientes com o respetivo resultado. Adicionalmente, implementa as operações em falta da API do TecnicoFS.

## Entrega e avaliação

Em todos os exercícios, os alunos devem submeter um ficheiro no formato *zip* com o código fonte e o ficheiro *Makefile*. O arquivo submetido não deve incluir outros ficheiros tais como binários. Além disso, o comando *make clean* deve limpar todos os ficheiros resultantes da compilação do projeto.

O exercício deve **obrigatoriamente** compilar e executar nos computadores dos laboratórios. O uso de outros ambientes para o desenvolvimento/teste do projeto (e.g., macOS, Windows/WSL) é permitido, mas o corpo docente não dará apoio técnico a dúvidas relacionadas especificamente com esses ambientes.

A submissão é feita através do Fénix, de acordo com os seguintes prazos:

- Exercício 1: 17/outubro, 12h00
- Exercício 2: 18/novembro, 23h59
- Exercício 3: 4/dezembro, 23h59

A avaliação será feita de acordo com o método de avaliação descrito no site da cadeira. O teste prático e a discussão decorrem entre 7 e 19/dezembro.

Os alunos devem consultar regularmente a plataforma piazza. É lá que serão prestados esclarecimentos sobre este enunciado.