Projeto de Sistemas Operativos
2020-21
3º enunciado
LEIC-A/LEIC-T/LETI

O principal objetivo deste exercício é implementar uma aproximação da arquitetura final do TecnicoFS em modo utilizador, ilustrada na Figura 1. Esta solução permitirá que outros processos, externos ao processo servidor do TecnicoFS, possam invocar as operações do mesmo.

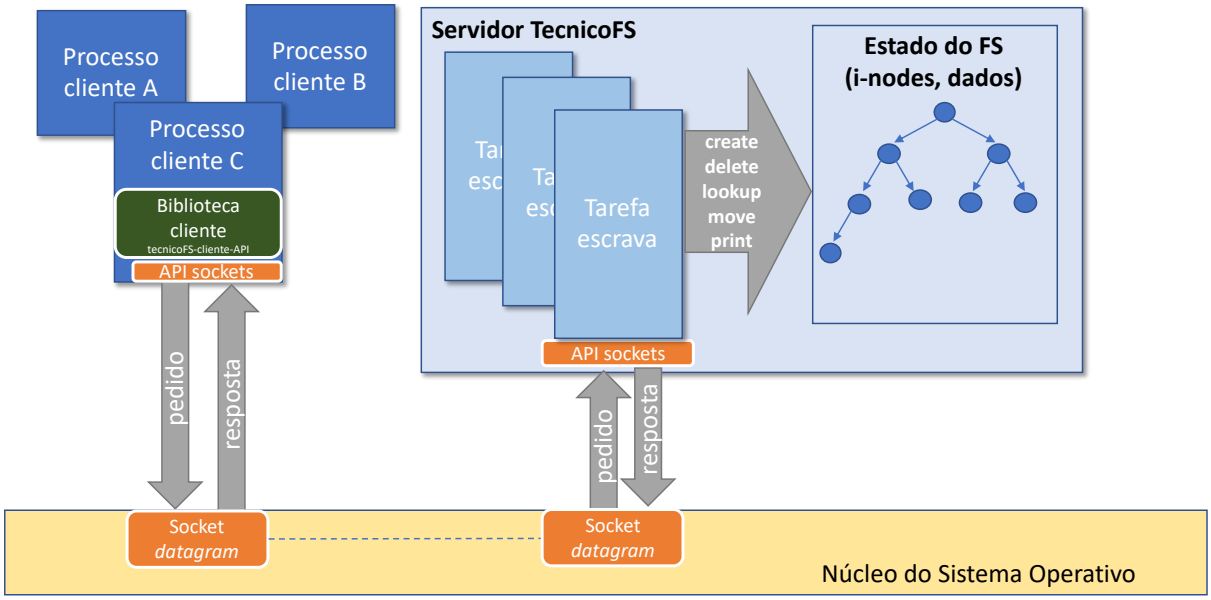

Figura 1: Arquitetura da solução do 3º exercício

1. Comunicação com processos cliente

O servidor TecnicoFS deixa de carregar comandos a partir de um ficheiro. Em vez disso, passa a ter um socket *Unix* do tipo *datagram*, através do qual recebe (e responde a) pedidos de operações solicitadas por outros processos, que designamos de processos cliente.

O executável do TecnicoFS passa a receber os seguintes argumentos de linha de comandos:

```
tecnicofs numthreads nomesocket
```

O novo argumento, *nomesocket*, indica o nome que deve ser associado ao *socket* pelo qual o servidor receberá pedidos de ligação. Os restantes argumentos têm o mesmo significado que nos enunciados anteriores.

Pretende-se desenvolver dois componentes principais: o servidor TecnicoFS e a biblioteca cliente. De seguida descrevemos cada um em detalhe.

1) O **servidor TecnicoFS** deve ter em conta que:

   - A tarefa inicial é responsável por inicializar o *socket.*
   - Cada tarefa escrava recebe pedidos enviados através do *socket*, em vez de os receber a partir do vetor de comandos. Esta mudança torna o vetor de comandos obsoleto, pelo que este **deve ser removido do código do servidor TecnicoFS**.
   - Após receber e executar um pedido de operação, a tarefa escrava deve devolver o resultado da operação numa mensagem enviada pelo mesmo *socket*, endereçada ao *socket* do cliente que originou o pedido.
   - Por simplicidade do projeto, deve assumir-se que o servidor nunca termina.

2) Quanto à **biblioteca cliente**:

   - A biblioteca chama-se *tecnicofs-client-api*, cujo esqueleto (incompleto) é fornecido no código base disponibilizado no site da disciplina.
   - Em teoria, a biblioteca pode ser usada por diferentes programas cliente.
   - A biblioteca implementa o protocolo de comunicação o cliente e o servidor TecnicoFS. Essencialmente, a biblioteca fornece uma função para cada operação do servidor TecnicoFS. Quando chamada, cada função envia um pedido ao servidor TecnicoFS e aguarda pela sua resposta.

Para auxiliar o desenvolvimento e os testes do 3º exercício, é também fornecido um programa cliente de exemplo. Este cliente é sequencial (*single-threaded)*. Essencialmente, este cliente carrega uma sequência de comandos a partir de um ficheiro de entrada e, um por um, invoca-os sobre o servidor TecnicoFS.

Quanto ao **protocolo** que define as mensagens trocadas entre cliente e servidor:

- Os pedidos enviados pelos clientes consistem em *strings* indicando a operação solicitada e o(s) seus(s) argumentos(s), sendo o formato dessas *strings* aquele que foi definido nos enunciados anteriores para cada operação do TecnicoFS.
- A resposta consiste no inteiro que é devolvido pela operação invocada (o valor do inteiro a enviar ao cliente deve ser o mesmo que já era devolvido nos projetos anteriores).

2. Nova operação ‘p’

O TecnicoFS deve passar a suportar uma nova operação que imprime o seu conteúdo atual (através da função já existente *print_tecnicofs_tree*) para um ficheiro de saída indicado como único argumento da operação, com a seguinte sintaxe:

```
p outputfile
```

Tal como as restantes operações do TecnicoFS, a operação ‘p’ pode ser invocada por qualquer cliente.

Um dos requisitos associados à nova operação é o de que o conteúdo guardado no ficheiro deve ser obtido num estado *quiescente* do sistema; ou seja, num momento em que nenhuma das tarefas escravas esteja a executar qualquer operação que modifique o conteúdo do sistema de ficheiros.

Mais concretamente, assuma que a operação 'p' é recebida por uma dada tarefa escrava, *t*, num momento em que uma ou mais tarefas escravas estejam a executar alguma(s) operação(ões) que modifique o estado do TecnicoFS. Nesse caso, a tarefa *t* deve:

- Aguardar até que essa(s) operação(ões) terminem; e, ao mesmo tempo,
- Impor que nenhuma tarefa inicia a execução de novas operações enquanto *t* não terminar de salvaguardar o conteúdo do sistema no ficheiro de saída.

Compete aos alunos pensar e implementar uma solução de sincronização que assegure este requisito, idealmente de forma simples e eficiente.

Experimente

Usando o cliente exemplo fornecido pelo corpo docente, experimente executar os ficheiros de teste (fornecidos na diretoria *inputs*) mas agora invocados (sequencialmente) a partir de um cliente. Acrescente ao final de cada teste um comando 'p' para conferir se o estado final do TecnicoFS é o esperado. Confira também os retornos recebidos pelo cliente.

Experimente agora com 2 ou mais clientes, lançados concorrentemente, cada um correndo um ficheiro de teste distinto. Neste caso o servidor TecnicoFS tratará os comandos recebidos dos diferentes clientes de forma concorrente. Caso o número de tarefas escravas na *pool* do servidor seja idêntico ao número de clientes concorrentes, os pedidos de cada cliente tenderão a ser sempre executados pela mesma tarefa escrava ou por tarefas diferentes? Pense numa forma de instrumentar o código do servidor para obter a resposta a esta questão empiricamente.

Finalmente, numa experiência com 2 clientes, coloque um deles a executar um dos testes (da diretoria *inputs*) e o outro a invocar a operação 'p' frequentemente para diferentes ficheiros. Confirme que os conteúdos guardados nos diferentes ficheiros descrevem sempre estados consistentes do sistema.

Entrega e avaliação

Consultar o enunciado geral.