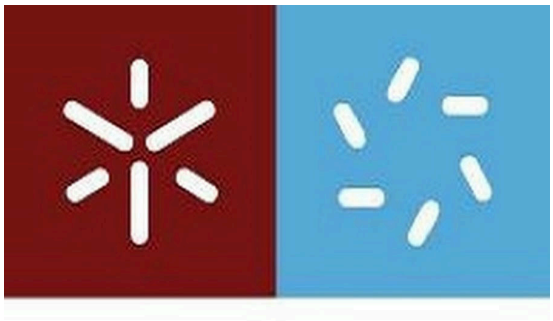

Universidade do Minho
Escola de Ciências

Sistemas Operativos

2024/2025

Trabalho Prático

Serviço de Indexação e Pesquisa de Documentos

Grupo 13

Gonçalo Barroso nº A102931

Pedro Gomes nº A102929

Rafaela Pereira nº A102527

Contents

1. Introdução

No âmbito da unidade curricular de Sistemas Operativos, foi nos proposto implementar um serviço capaz de fazer a indexação e a pesquisa de documentos de texto guardados localmente num computador. Para tal, aplicamos os conhecimentos adquiridos nas aulas lecionadas, sendo estes a gestão de processos, manipulação de ficheiros e comunicação entre processos pai e filho.

2. Funcionalidades

2.1. Indexação de um documento

Ao invocar o comando cliente ***./bin/cliente -a "título" "autores" "ano" "caminho"***, o servidor tenta indexar o documento. Primeiramente, verifica se o documento já se encontra indexado. Caso esteja, apenas será atualizada a data da última vez que o documento foi acedido, o que será útil para a implementação do ***algoritmo LRU***, que explicaremos mais à frente.

Se o documento não estiver indexado, é gerada uma chave para o mesmo, que é então armazenada. Em seguida, o servidor tenta colocá-lo na cache. Caso não exista espaço disponível, utilizamos o algoritmo ***LRU (Least Recently Used)*** para substituir o documento utilizado há mais tempo.

Inicialmente, foi implementado o algoritmo ***FIFO (First In, First Out)***, por "parecer" o mais simples de desenvolver. No entanto, ao longo do desenvolvimento, constatámos que esta abordagem não era a mais eficiente, pois remove o documento mais antigo da cache, independentemente de ter sido acedido recentemente ou não. Isto pode levar à remoção de documentos que ainda são frequentemente utilizados, o que não é desejável.

Por esse motivo, optámos por utilizar o algoritmo LRU, que substitui o documento que não é acedido há mais tempo. Esta estratégia permite uma gestão mais inteligente da cache, mantendo disponíveis os documentos mais relevantes para os utilizadores com base na frequência de acesso.

2.2. Consulta de um documento

A partir do comando ***./bin/dclient -c "key"***, o cliente terá acesso às informações relativas ao documento identificado pela chave fornecida. No servidor, é criado um novo processo filho para executar a pesquisa de forma isolada, invocando a função ***searchKey()***, que realiza a pesquisa do documento através da chave fornecida. Caso o resultado seja vazio, é enviada ao cliente uma mensagem de erro: ***"Documento não encontrado (key=...)"***. Após esta operação, o processo filho liberta a memória alocada e termina de imediato. Optou-se por esta abordagem devido ao isolamento de falhas — desta forma, qualquer erro que ocorra no processo filho não afeta o servidor principal.

2.3. Remoção de um índice

Com o comando ***./bin/dclient -d "key"***, o cliente solicita ao servidor a remoção de um documento correspondente àquela chave. Tal como na consulta de documentos, a operação é realizada de forma concorrente, criando-se um processo filho que irá executar a tarefa. O processo filho tenta, em

primeiro lugar, remover os metadados do documento da cache em memória, através da função ***apagaMeta()***. Caso o documento não esteja presente na cache, o cliente é imediatamente informado ***("Documento não encontrado")***. Se os metadados forem removidos com sucesso, o processo tenta então eliminar o registo persistente através da função removeKey(). Qualquer falha nesta etapa é igualmente comunicada ao cliente. Em caso de sucesso total, é enviada uma mensagem de confirmação da remoção ***("Documento removido")***.

2.4. Pesquisar número de linhas que contêm uma certa palavra chave

Ao executar ***./bin/dclient -l "key" "keyword"***, o servidor devolve o número de linhas de um determinado documento que contêm a palavra-chave indicada. Tal como nas outras operações, o servidor cria um novo processo filho para tratar o pedido. Esse processo filho invoca a função ***searchKeyWords()***, que procura e contabiliza as ocorrências da palavra no documento identificado pela chave. Caso o documento esteja devidamente indexado e a palavra seja encontrada, o número de ocorrências é enviado de volta ao cliente. Caso contrário, é enviada uma mensagem de erro a informar que o documento não está indexado ou que a palavra não foi encontrada ***("Arquivo não indexado ou palavra não encontrada")***.

2.5. Pesquisar lista de identificadores de documentos que contêm uma certa palavra chave

Executando o comando ***./bin/dclient -s "keyword" "nr_processos"***, o servidor irá devolver a lista de identificadores dos documentos que contêm a keyword que foi introduzida pelo utilizador, criando um novo processo filho para realizar esse pedido. O processo invoca a função ***searchAll()***, que irá pesquisar por todos os documentos a palavra chave, devolvendo uma lista com os identificadores onde essa palavra-chave foi encontrada, e, se não for encontrada nenhuma instância dessa keyword devolve ***("Nenhum documento encontrado")***.

Este comando foi otimizado, usando processos concorrentes para uma pesquisa mais rápida, sendo o ***"nr_processos"*** o número máximo de processos que podem estar a executar simultâneamente.

2.6. Parar o programa servidor através do programa cliente.

Em último ao executar o comando ***./bin/dclient -f***, é enviada uma mensagem ao cliente a informar que o servidor está a encerrar ***("Servidor a encerrar...")***. De seguida, são fechadas as conexões através dos descritores de ficheiros client_fd e server_fd, evitando assim vazamentos de recursos. Após o encerramento das ligações, os named pipes ***(SERVER_PIPE e CLIENT_PIPE)*** são removidos do sistema de ficheiros com a função ***unlink()***, garantindo que não permanecem canais de comunicação obsoletos que possam interferir com futuras execuções. Por fim, é impressa uma mensagem de confirmação no terminal do servidor ***("Servidor encerrado.")***, indicando que o encerramento foi bem-sucedido.

3. Conclusão

Com este trabalho prático, tivemos a oportunidade de aplicar de forma mais aprofundada os conceitos abordados ao longo do semestre indo além do que foi explorado nas aulas práticas, demonstrando as nossas capacidades de gestão de processos e de memória, e trabalhar com sistemas de ficheiros em situações mais complexas. Tivemos também algumas dificuldades ao longo do nosso trabalho, tais como implementar o sistema de escalonamento LRU, onde de inicio não tinhamos escolhido essa opção, mas ao longo do tempo reparamos que talvez seria o melhor para este trabalho.