Sistemas Operativos

Trabalho Prático

Serviço de Indexação e Pesquisa de Documentos

Grupo de Sistemas Distribuídos
Universidade do Minho

10 de março de 2025

Informações gerais

- Cada grupo deve ser constituído por 3 elementos;
- O trabalho deve ser entregue até às 23:59 de 17 de Maio;
- Deve ser entregue o código fonte, *scripts* e um relatório de até 10 páginas de conteúdo (A4, 11pt) no formato PDF (não são contabilizadas capas e anexos para o limite de 10 páginas), justificando a solução, nomeadamente no que diz respeito à arquitetura de processos, funcionalidades, e da escolha e uso concreto dos mecanismos de comunicação;
- O trabalho deve ser realizado tendo por base o sistema operativo Linux como ambiente de desenvolvimento e de execução;
- O trabalho deve ser submetido num arquivo Zip com nome `grupo-xx.zip`, em que `xx` deve ser substituído pelo número do grupo de trabalho (*p.ex.*, `grupo-01.zip`);
- A apresentação do trabalho ocorrerá em data a anunciar, previsivelmente entre os dias 2 e 5 de Junho;
- O trabalho representa 50% da classificação final.
- De notar que a cotação máxima para as diferentes etapas do enunciado depende tanto da correta implementação das mesmas como também da qualidade do relatório e da discussão de cada grupo com a equipa docente.
- A equipa docente irá atualizando uma FAQ do TP com respostas a questões levantadas pelos alunos em: `https://docs.google.com/document/d/1PGrnMWpFf2JQOeKSqVJBsoZ9VUFtKzFZJBzEwSOEXkA/edit?usp=sharing`.

Resumo

Pretende-se implementar um serviço que permita a indexação e pesquisa sobre documentos de texto guardados localmente num computador. O programa servidor é responsável por registar meta-informação sobre cada documento (*p.ex*, identificador único, título, ano, autor, localização), permitindo também um conjunto de interrogações relativamente a esta meta-informação e ao conteúdo dos documentos.

Os utilizadores devem utilizar um programa cliente para interagir com o serviço (*i.e.*, com o programa servidor). Esta interação permitirá que os utilizadores adicionem ou removam a indexação de um documento no serviço, e que efetuem pesquisas (interrogações) sobre os documentos indexados. De notar que o programa cliente apenas executa uma operação por invocação, ou seja, não é um programa interativo (*i.e.*, que vai lendo várias operações a partir do *stdin*).

Servidor e Cliente (12 valores)

Deverá ser desenvolvido um cliente (programa *dclient*) para ser usado pelo utilizador via linha de comandos. Deverá ser também desenvolvido um servidor (programa *dserver*), com o qual o programa cliente irá interagir. O servidor deve manter em memória e em disco a informação relevante para suportar as funcionalidades descritas neste enunciado.

O *standard output* deverá ser usado pelo programa cliente para apresentar as respostas necessárias ao utilizador, e pelo programa servidor apenas para apresentar informação para depuração de erros (*debug*) conforme julgue necessário. Para este tipo de operações pode utilizar a funoão *printf*.

Os programas cliente e servidor deverão ser escritos em C e comunicar via *pipes com nome*. Devem ser utilizadas as chamadas ao sistema lecionadas na cadeira de sistemas operativos para gestão de processos, comunicação entre processos, e interação com ficheiros. Não deve recorrer à execução de programas direta ou indiretamente através do interpretador de comandos (*p.ex.*, *sh*, *bash* ou `system()`). Da mesma forma, não pode utilizar funções como *fopen*, *fwrite*, *fread*, etc. A utilização de bibliotecas (*p.ex.*, Glib) para criação e gestão de estruturas de dados é permitida.

O serviço deverá suportar as seguintes funcionalidades básicas:

Indexação, consulta, e remoção de meta-informação de documentos. Para indexar um novo documento, o utilizador invoca o programa cliente com o comando:

```
dclient -a "title" "authors" "year" "path"
```

- `title`: título do documento.
- `authors`: autor(es) do documento (*p.ex.*, separar por ponto e vírgula (';') quando existem vários autores).
- `year`: ano do documento.
- `path`: caminho relativo do documento, *i.e.,* a partir da directoria base configurada para o serviço.

De notar que o documento deve ter sido criado previamente pelo utilizador, o programa servidor apenas irá indexá-lo. O programa cliente deve comunicar ao programa servidor o pedido de indexação, cuja resposta deverá conter um identificador único do documento, o qual deve ser comunicado ao utilizador. A escolha do identificador fica ao critério de cada grupo. O programa servidor deve indexar a meta-informação de cada documento, utilizando a(s) estrutura(s) de dados que cada grupo ache mais apropriada(s).

Notas: Assuma que o tamanho total dos argumentos da operação anterior não excede os 512 bytes (*p.ex.*, os campos `title` e `authors` têm no máximo 200 bytes cada, o campo `path` tem no máximo 64 bytes e o campo `year` ocupa no máximo 4 bytes).

Ainda, o programa cliente deve permitir consultar (opção *-c*) e remover (opção *-d*) a meta-informação de um documento através do seu identificador *key*.

```
dclient -c "key"
dclient -d "key"
```

O programa cliente deve comunicar ao utilizador o sucesso das operações e, para a operação de consulta, deve imprimir no *stdout* a meta-informação do documento. A operação de remoção não deve apagar o conteúdo do documento em questão, apenas deve remover a sua meta-informação indexada pelo programa servidor.

Pesquisa sobre o conteúdo de documentos. Para além da gestão e consulta de meta-informação, o programa cliente deve também permitir aos utilizadores fazer pesquisas sobre o conteúdo dos documentos indexados.

Em detalhe, deve ser possível (opção *-l*) devolver o número de linhas de um dado documento (*i.e.*, identificado pela sua *key*) que contêm uma dada palavra-chave (*keyword*).

```
dclient -l "key" "keyword"
```

Ainda, deve ser possível (opção *-s*) devolver uma lista de identificadores de documentos que contêm uma dada palavra-chave (*keyword*).

```
dclient -s "keyword"
```

Nota: Para implementar estas operações pode recorrer aos programas *grep* e *wc* que foram discutidos nas aulas práticas.

Notas gerais: O processamento para dar resposta a todas as operações anteriores deve ser realizado pelo servidor. O cliente apenas envia o pedido e espera pela resposta, apresentado-a ao utilizador. Ainda, deve evitar que um cliente fique bloqueado devido a operações de **consulta** e **pesquisa** (opções *-c, -l e -s*) a serem efetuadas por outros clientes.

Otimizações e Avaliação (8 valores)

Partindo da sua implementação base, deve otimizar o programa servidor de forma a incorporar as seguintes funcionalidades.

Pesquisa concorrente. A operação de pesquisa por documentos que contêm uma dada palavra-chave (opção *-s*) deve poder ser efetuada concorrentemente por vários processos. Ao suportar esta operação avançada, a mesma passa a receber um argumento extra, nomeadamente o número máximo de processos a executar simultaneamente.

```
dclient -s "keyword" "nr_processes"
```

Nota: Esta funcionalidade deve ser implementada pelo grupo sem utilizar a capacidade de programas externos para paralelizar a execução. De notar que o grupo pode usar na mesma o programa `grep` para efetuar a pesquisa em cada ficheiro.

Persistência. Assuma, por razões de eficiência da gestão de memória e de durabilidade da informação, que a meta-informação dos documentos geridos pelo programa servidor tem de ser persistida em disco. Assim sendo, garanta que o seu programa servidor guarda uma cópia persistente da meta-informação dos documentos (*i.e.*, podendo esta ser recuperada ao parar e voltar a iniciar o programa). O servidor é parado através de um comando especial do cliente (opção *-f*).

```
dclient -f
```

***Caching*.** Altere o programa servidor para que seja possível controlar o número de entradas de meta-informação em disco que são também guardadas em memória. Para tal, implemente uma *cache* em memória que guarda até *N* entradas de meta-informação (argumento definido no arranque do programa servidor). As políticas para escolher que *items* são mantidos e servidos pela *cache* fica ao critério de cada grupo.

Avaliação experimental. Desenvolva e execute testes que permitam avaliar *i)* o ganho de desempenho ao paralelizar a pesquisa de documentos (opção *-s*) e *ii)* o impacto no desempenho de diferentes configurações (*i.e.*, tamanhos) e políticas de *caching* desenvolvidas pelo grupo. Deve utilizar scripts para automatizar a execução de testes. Ainda, deve escolher cenários de teste (*i.e.*, diferentes configurações, números de documentos, etc.) que permitam perceber melhor o impacto no desempenho em diferentes cenários. O relatório deve incluir uma reflexão sobre os resultados dos testes efetuados.

Nota: Junto com o enunciado, a equipa docente forneceu um conjunto de documentos para testes, e um *script* para automatizar a indexação da meta-informação destes documentos que poderá utilizar juntamente com o seu programa. Cada grupo pode testar com outros conjuntos de documentos que ache relevantes.

Interface e Modo de Utilização

O serviço deverá ser usado do seguinte modo:

- Executar o servidor:

  ```
  $ ./dserver document_folder cache_size
  ```

  Argumentos:

  1. `document_folder`: pasta onde se encontram os documentos a serem indexados.
  2. `cache_size`: número de items (*i.e.*, meta-informação de diferentes documentos) a serem guardados pela *cache* em memória.

- Submeter pedido de indexação de um documento, conforme o exemplo abaixo.

  ```
  $ ./dclient -a "Romeo and Juliet" "William Shakespeare" "1997" "1112.txt"
  Document 1 indexed
  ```

- Submeter pedido de consulta de um documento.

  ```
  $ ./dclient -c 1
  Title: Romeo and Juliet
  Authors: William Shakespeare
  Year: 1997
  Path: 1112.txt
  ```

- Submeter pedido de remoção de um índice.

  ```
  $ ./dclient -d 1
  Index entry 1 deleted
  ```

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

  ```
  $ ./dclient -l 1 "Romeo"
  150
  ```

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

  ```
  $ ./dclient -s "praia"
  [2, 3, 1438]
  ```

- Pesquisar lista de identificadores de documentos que contêm uma certa palavra chave usando vários processos (*p.ex.*, 5).

  ```
  $ ./dclient -s "praia" 5
  [2, 3, 1438]
  ```

- Parar o programa servidor através do programa cliente.

  ```
  $ ./dclient -f
  Server is shuting down
  ```

Makefile

Tenha em conta que a Makefile que se apresenta deverá ser usada como ponto de partida, mas poderá ter necessidade de a adaptar de modo a satisfazer, por exemplo, outras dependências do seu código-fonte. Em todo o caso, deverá manter sempre os objetivos (*targets*) especificados: `all`, `dserver`, `dclient`, e `clean`. Não esqueça que, por convenção, a indentação de uma Makefile é especificada com uma tabulação (*tab*) no início da linha (nunca com espaços em branco).

```
CC = gcc
CFLAGS = -Wall -g -Iinclude
LDFLAGS =

all: folders dserver dclient

dserver: bin/dserver

dclient: bin/dclient

folders:
	@mkdir -p src include obj bin tmp

bin/dserver: obj/dserver.o
	$(CC) $(LDFLAGS) $^ -o $@

bin/dclient: obj/dclient.o
	$(CC) $(LDFLAGS) $^ -o $@

obj/%.o: src/%.c
	$(CC) $(CFLAGS) -c $< -o $@

clean:
	rm -f obj/* tmp/* bin/*
```