

# Relatório do e-fólio A

Sistemas Operativos- 2023/24

Aluno: Daniel Júnior

Número: 2304335

## Introdução

Passos para o desenvolvimento do programa `FDUP.C`, uma ferramenta de linha de comando escrita em C padrão para encontrar ficheiros duplicados dentro de um diretório e suas subdiretórias. O programa foi desenvolvido como parte da avaliação contínua para a unidade curricular de Sistemas Operativos, no ano letivo de 2023-24 – 2 semestre.

## Objetivo

O objetivo do `FDUP.C` é identificar ficheiros duplicados com base num nome fornecido ou em um padrão (pattern) do tipo `"*.C"`, dentro de um diretório especificado pelo utilizador de forma recursiva (no próprio diretoria, e nas diretorias contidas neste). Esta ferramenta é útil rapidamente localizar duplicatas de mesmo nome.

## Metodologia

O programa `FDUP.C` foi desenvolvido utilizando a combinação de quatro comandos Unix/Linux fundamentais (`FIND`, `SORT`, `UNIQ`, `AWK`) para realizar sua tarefa. O processo é dividido em quatro passos principais, cada um correspondendo a um desses comandos, executados sequencialmente através da criação de processos filhos com `FORK()` e substituição do processo pelo comando com `EXECVP()`.

Optei por uma estrutura modular para facilitar a organização do código e na esperança de facilitar uma rápida leitura. Para tal, criei funções específicas para cada tarefa significativa dentro do programa, permitindo que cada parte do processo fosse separada e gerida dentro deste pequeno universo.

### Processo de Desenvolvimento

**Análise de Requisitos:** Analisei cuidadosamente os requisitos fornecidos no enunciado. Pude, então, compreender as funcionalidades que o programa deveria ter e os critérios de sucesso.

**Verificação de Argumentos (`checkArguments`):** A primeira função implementada foi a de verificação dos argumentos de entrada. Esta função assegura que o usuário forneça o número correto de argumentos e que o diretório especificado exista. Assegura que o programa opere sobre uma base correta, e tentamos seguir o princípio de *fail fast*.

**Execução de Comandos (`executeCommand`):** A complexidade de executar os comandos Unix (`FIND`, `SORT`, `UNIQ`, `AWK`) dentro do programa C levou à criação de uma função genérica capaz de executar qualquer comando. Utilizando `FORK()` para criar um processo filho e `EXECVP()` para substituir a imagem do processo pelo comando desejado, essa função originou-se do desejo

te tentar reproduzir minimamente a design pattern Strategy, gerindo a execução e o redireccionamento de saída através de um mesmo método.

**Passos Específicos do Processo:** Para cada um dos comandos Unix utilizados, desenvolveu-se uma função dedicada (`FINDFILES`, `SORTFILES`, `UNIQFILES`, `FORMATOUTPUT`). Essas funções encapsulam os detalhes específicos de cada comando, como os argumentos necessários e o tratamento de arquivos temporários. Esta abordagem não apenas simplifica o fluxo principal do programa, mas também torna o código mais acessível para revisão ou alterações futuras.

**Testes e Validação:** Após a implementação, o programa foi submetido a uma série de testes com diferentes conjuntos de dados para assegurar sua correta funcionalidade. Utilizei-me do Arch Linux que pode ser facilmente utilizado no site <https://copy.sh>. Testes incluíram a verificação da capacidade de encontrar ficheiros duplicados em várias estruturas de diretórios, bem como a manipulação adequada de casos de erro.

## Descrição do Processo

**Busca de Ficheiros:** Utilizando o comando `FIND`, o programa busca todos os ficheiros regulares que correspondem ao padrão especificado, dentro do diretório fornecido, listando o caminho e o nome do ficheiro.

**Ordenação:** Os resultados da busca são então ordenados pelo nome do ficheiro com o comando `SORT`, facilitando a identificação de duplicatas.

**Identificação de Duplicatas:** O comando `UNIQ` é usado para filtrar os nomes de ficheiros duplicados da lista ordenada.

**Formatação da Saída:** Finalmente, o comando `AWK` modifica a ordem de apresentação dos resultados para "`NOME_DO_FICHEIRO CAMINHO_DA_DIRECTORIA`" e imprime a lista final na saída padrão.

## Resultados e Discussão

A execução do `FDUP.C` resulta na impressão na saída padrão da lista de ficheiros duplicados encontrados, formatados conforme especificado anteriormente. Erros no fornecimento de argumentos ou na execução de comandos são tratados de maneira apropriada, com mensagens de erro claras e diretas. Os ficheiros dos passos intermediários, podem ser encontrados na diretoria onde o executável reside para fácil verificação dos resultados intermédios.

## Conclusão

Acredito que o `FDUP.C` cumpre os objetivos mínimos de identificar ficheiros duplicados dentro de um diretório e suas subdiretórios de forma eficiente, demonstrando conceitos de Sistemas Operativos: processos, comunicação entre processos e manipulação de ficheiros e diretórios.

## Anexos

Código-fonte: FDUP.C