

UNIVERSIDADE DO MINHO

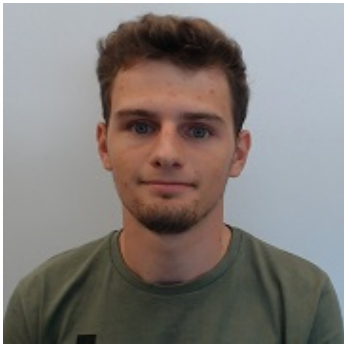
DEPARTAMENTO DE INFORMÁTICA

SISTEMAS OPERATIVOS

GRUPO Nº 89

9 de fevereiro de 2024

1 Autores



Bohdan Malanka
a93300



Diogo Rebelo
a93278



Henrique Alvelos
a93316

Conteúdo

1 Autores	2
2 Introdução	3
3 Arquitetura do projeto	3
3.1 AuxStructs.c	3
3.2 Aurras.c	3
3.3 Aurrasd.c	3
4 Conclusão	4

2 Introdução

O trabalho prático em questão visa primordialmente criar um serviço onde seja possível manipular ficheiros de áudio através da aplicação de uma sequência de filtros. Estes filtros alteram o ficheiro de input fornecido e geram um novo ficheiro com as mesmas alterações.

Um cliente efetua um pedido, o qual vai para fila de espera do servidor. O servidor vai á fila de pedidos de clientes (onde cada cliente tem as suas devidas informações). Com recurso à função de atualização da fila de processos, o servidor determina quantos processos podem ser realizados simultaneamente e efetua *swap* para as primeiras posições da fila dos pedidos que podem ser executados simultaneamente. O critério de seleção para a realização destes pedidos é o limite de filtros que o servidor está preparado para manipular (definido no enunciado).

Além disso, como requisitado, este serviço permite ao cliente observar as tarefas em execução e o status do seu pedido.

3 Arquitetura do projeto

Esta pode ser dividida em três grandes grupos que servem de base para o serviço a desenvolver e que surgem de seguida.

3.1 AuxStructs.c

Estrutura de dados auxiliar onde estão definidas algumas structs e é responsável por guardar dados dos clientes (processo, operação, entre outros), estes que são guardados num Array, de modo a representar uma fila de espera. Além disso, possui outra struct chamada **filtro** que, pelo nome indica, é responsável por guardar um tipo de filtro.

3.2 Aurras.c

Estrutura de dados que representa o cliente. Recebe o pedido, verifica se é válido e, caso seja, cria um **FIFO** para o servidor comunicar, escreve o pedido no **FIFO** do servidor e fica à espera de que o servidor realize o seu pedido. Quando conclui a sua tarefa, o programa termina para o determinado cliente. Para o caso de não receber nenhum pedido, o programa termina, mandando uma mensagem de ajuda. Para o caso de o pedido ser inválido, o programa termina com uma informação de que não é possível executar o pedido.

3.3 Aurrasd.c

Estrutura de dados que representa o servidor. Cria o **FIFO** para que o cliente comunique o pedido. É responsável por receber os pedidos do cliente e, a partir disso, executa-os. Depois, atualiza o estado do servidor e avança para os próximos pedidos.

4 Conclusão

Em suma, este projeto permitiu consolidar a matéria desta unidade curricular e compreender de uma forma mais elucidativa o processo de comunicação que se estabelece em *named pipes*.

Compreende-se que foram utilizados várias formas de comunicação, desde sinais, passando pela relação entre processos pai e filho e *unnamed pipes*, sempre socorrendo-nos do redirecionamento de descritores de ficheiros.

Optou-se por permitir que tanto o servidor como o cliente consigam criar *FIFOs*. Um problema que não conseguimos resolver foi atualizar o estado do servidor, nomeadamente a alteração do número de utilizações dos filtros, mesmo quando acaba a execução de um pedido, fazendo com que a performance não seja a mais desejada