



Universidade do Minho

Trabalho Prático

Sistemas Operativos

2020/2021

Data de Entrega:
17/06/2021

Relatório-Grupo 116



Identificação do Grupo nº42

Augusto César Oliveira Campos
MIEI
nº.a93320



Cláudia Peixoto Silva
MIEI
nº.a93177



Introdução

Este projeto consistia em implementar um serviço capaz de transformar ficheiros de áudio por aplicação de uma sequência de filtros. O serviço deveria permitir não só a submissão de pedidos de processamento de ficheiros de áudio como também a consulta das tarefas em execução, e número de filtros disponíveis e em uso.

O serviço é constituído por um servidor e por um cliente que comunicam por pipes com nome. Tanto o cliente (aurras) como o servidor (aurrasd) recebem os parâmetros de funcionamento como argumento da linha de comando.



Decisões tomadas

O projeto está dividido em 2 ficheiro : o ‘aurras’ (o cliente) e o ‘aurrasd’ (o servidor), onde foi necessária a implementação de pipes com nome para realizar a comunicação entre os dois.

Desde modo, o cliente manda o pedido do processo e o servidor executa-o, no enunciado era pedido a existência de limites no uso dos filtros usados para o processamento dos ficheiros, ficando assim um processo em espera. No entanto, apesar de termos conseguido implementar a contagem dos filtros que estão a ser usados num momento, não implementamos a segunda , ou seja o nosso servidor recebe e executa uma infinidade de tarefas, este foi no entanto o único ponto em que falhamos quanto aos pedidos base do enunciado.

A execução de processos do servidor é concorrente ou seja, o servidor consegue executar várias transformações de ficheiros ao mesmo tempo.

Apesar de não ser a melhor implementação, usamos ficheiros temporários para a aplicação dos filtros (outputTMP1 e outputTMP2) e infelizmente, não conseguimos adicionar lidar com o SIGTERM e fechar o programa de forma graciosa nesse caso.

Como alternativa aos ficheiros temporários poderia-se usar pipes anônimos para a comunicação entre os processos que executam os filtros, tirando assim a necessidade destes ficheiros e seguindo mais à risca os pedidos opcionais do enunciado.

Comandos Implementados

Status - Comando que envia o estado atual do servidor.

A execução deste comando corre exatamente como pretendido, listando a informação do instante que está guardada no servidor.

Transform - Comando para a execução das transformações

A execução deste comando não ficou como nós esperávamos, como já referimos anteriormente. Se o limite de filtros for alcançado o pedido é executado na mesma, para além disto a forma de carregar filtros poderia ser mais maleável, se estes são carregados a partir do ficheiro aurrasd.conf a adição de um filtro deveria ser tão simples como adicionar o mesmo ao ficheiro, apesar de o programa suportar a funcionalidade de carregar estes dados, o servidor é nesta versão limitado aos cinco filtros fornecidos pelo enunciado. No entanto, através de uma simples mudança de configuração no projeto esta funcionalidade é facilmente implementada.



Conclusão

Este trabalho ajudou-nos a melhor consolidar os conceitos que foram leccionados nas aulas teóricas no ambiente de system calls e de system control, algo que consideramos muito importante para a nossa formação pois imaginamos que ,independentemente da nossa área de especialização, estas bibliotecas e outras inspiradas nelas irão aparecer no nosso percurso.

Tivemos dificuldades na implementação de algumas funcionalidades e dores de cabeça mascarados de bugs, que nos impediram de realizar certas partes do projeto como nós tínhamos esperado e/ou ponderado.

Apesar destas advertências, entregamos este projeto com a certeza de que ficamos mais capazes de dominar esta área da programação.

