



# **Execução Agendada de Conjuntos de Programas**

## **Sistemas Operativos**

### **MIETI**



Pedro Rafael Carvalho Miranda

A79662

[a79662@alunos.uminho.pt](mailto:a79662@alunos.uminho.pt)



Pedro Alexandre Morais Alves

A61893

[a61893@alunos.uminho.pt](mailto:a61893@alunos.uminho.pt)



# Índice

<b>1 – Introdução.....</b>	<b>2</b>
<b>2 – Descrição do problema.....</b>	<b>3</b>
<b>3– Conceção .....</b>	<b>3</b>
<b>4– Erros do programa .....</b>	<b>4</b>
<b>5 – Conclusão .....</b>	<b>5</b>



# 1 – Introdução

Com este trabalho prático o grupo pretende consolidar os conhecimentos adquiridos ao longo do semestre, nesta UC, bem como adquirir outros conhecimentos, que não são objeto de estudo, mas que são importantes para a conceção deste serviço de agendamento de tarefas através de uma interação cliente-servidor.



## 2 – Descrição do problema

O objetivo deste trabalho prático passa pela implementação de um serviço que permita agendar uma tarefa. Este deve ainda permitir a um cliente cancelar uma tarefa, bem como visualizar todas as tarefas agendadas, as que já terminaram a sua execução e as que foram canceladas.

Após a execução da tarefa agendada, o utilizador pode consultar toda a informação relativa a uma determinada tarefa, incluindo o resultado desta operação (*standard output*, *standard error* e o valor de saída).

O sistema deverá também oferecer a possibilidade de definir o número máximo de agendamentos a executar concorrentemente, bem como definir o endereço para onde devem ser enviados os resultados das tarefas ainda agendadas.

## 3– Conceção

Para iniciar a execução do serviço, devemos executar o programa da central (servidor) e da agenda (cliente), necessariamente por esta ordem, uma vez que é a central que cria os pipes necessários á comunicação entre os 2 programas, um pipe para o envio dos pedidos (agendar, cancelar, listar e consultar) do cliente para o servidor (clienteservidor) e outro pipe para o envio das respostas aos pedidos anteriormente feitos (servidorcliente).

Para executar um agendamento o cliente fornece a data, hora, caminho do ficheiro executável, e os respetivos argumentos, se forem necessários. Todas estas informações são enviadas ao servidor. No servidor estas informações são separadas, de forma a ser possível calcular o número de segundos até ser ao momento escolhido. Depois disto é preenchida uma posição do *array* de *struct's* Execuções, com todas as informações do agendamento, sendo também colocada esta informação num ficheiro para posterior consulta. Depois é executado um *fork()* para a execução da tarefa agendada. Este processo lança um *SIGALRM*, agendando um alarme com o número de segundos até ao momento escolhido, adormecendo este processo até ao final do alarme, sendo de seguida executando o programa pretendido. Logo que este processo termine é mudado o estado da tarefa para terminada.



Se for pretendido o cancelamento de uma tarefa, o cliente terá de fornecer o ID da tarefa a cancelar á central, onde entre todas as tarefas é procurada a tarefa com o ID introduzido, e assim que encontrada, é enviado um SIGTERM para que o processo termine com um `_exit()`. Sendo depois o estado da tarefa alterado para cancelada.

Se a ação a desenvolver for a consulta da informação de um agendamento, após a inserção do ID é executado o comando `cat` do ID introduzido sendo mostrado ao cliente a informação e resultado do agendamento se ocorrer tudo com sucesso, ou se ocorrer algum erro, é também mostrada a mensagem de erro.

Se o cliente quiser listar todas as tarefas agendadas e que já foram agendadas, o cliente faz o pedido, e o servidor fornece todas as informações relativas a todas as tarefas.

## 4– Erros do programa

Para além do grupo não ter conseguido implementar todas as funcionalidades pretendidas para o serviço, existem algumas limitações ao bom funcionamento do programa:

- Um agendamento tem obrigatoriamente de terminar por “/0”, se isso não suceder o agendamento terminará com um erro no `execv`;
- Depois do cancelamento de uma tarefa, não conseguimos que esta mesma tarefa fique com o estado “cancelada”;
- No final da execução do programa agendado o grupo não conseguiu obter o valor de saída, que seria colocado no ficheiro relativo aquele agendamento.



## 5 – Conclusão

No âmbito da Unidade Curricular de Sistemas Operativos, foi proposto aos alunos que desenvolvessem um serviço de agendamento de tarefas, que permitisse aplicar os conhecimentos apreendidos ao longo do semestre tanto nas aulas práticas como teóricas, com um claro enfoque nas comunicações cliente-servidor em contexto de vida real. Percebemos de que forma é realizada a leitura e escrita em ficheiros, bem como a comunicação de processos com recurso a *fifos* (*pipes* com nome).

Este projeto serviu também para perceber quais as maiores debilidades dos conteúdos lecionados nas aulas (alarmes e manipulação de *strings*).

No final o grupo faz um balanço positivo, do trabalho desenvolvido em torno deste trabalho.