

# Client Server App

## Mini-Projeto 2A

Sistemas Operativos 2020/2021

Turma 7 Grupo 1

Adriano Soares	<a href="mailto:up201904873@edu.fe.up.pt">up201904873@edu.fe.up.pt</a>
Diogo Maia	<a href="mailto:up201904974@edu.fe.up.pt">up201904974@edu.fe.up.pt</a>
Francisco Cerqueira	<a href="mailto:up201905337@edu.fe.up.pt">up201905337@edu.fe.up.pt</a>
Pedro Pereira	<a href="mailto:up201905508@edu.fe.up.pt">up201905508@edu.fe.up.pt</a>
Vasco Alves	<a href="mailto:up201808031@edu.fe.up.pt">up201808031@edu.fe.up.pt</a>

# Índice

<b>Contexto</b>	<b>3</b>
Requisitos Funcionais	3
<b>Estrutura Geral do Código</b>	<b>3</b>
<b>Módulos Utilizados</b>	<b>4</b>
client.h	4
thread.h	4
log.h	4
message.h	4
<b>Contribuição Percentual</b>	<b>5</b>

# Contexto

## *Requisitos Funcionais*

A aplicação relativa ao cliente, requerida nesta primeira entrega, é capaz de solicitar e recepcionar diversas tarefas de diferente carga em paralelo ao servidor disponibilizado pelos professores, através da criação de threads contíguos separados por intervalos pseudo-aleatórios (500 - 1000 ms). Esta recebe o número de segundos que o programa deve permanecer em execução e o nome (absoluto ou relativo) do canal público de comunicação com nome (FIFO) por onde o Cliente envia pedidos ao Servidor, por esta ordem.

O formato requerido será então:

<code>c &lt;-t NSECS&gt; &lt;FIFONAME&gt;</code>
--

- 1) NSECS: número de segundos que o programa deve ficar ativo;
- 2) FIFONAME: caminho relativo ou absoluto do canal público de comunicação com nome (FIFO) por onde o Cliente envia pedidos ao Servidor.

## Estrutura Geral do Código

O código encarregue do cliente divide-se entre quatro módulos, dois destes representando aspectos importantes do problema, sendo estes o código que representa a thread principal do programa, responsável pela criação dos pedidos ao servidor (client.c) e o código remetente a cada um dos pedidos ao servidor (thread.c). Os outros dois módulos são responsáveis por registar as operações dos pedidos na consola e criar a estrutura de dados que permite a transferência de informação entre o cliente e o servidor.

# Módulos Utilizados

## *client.h*

Este módulo contém o método principal (*main*), que é responsável pela gestão da thread principal e que, por sua vez, é responsável pela realização das seguintes tarefas:

1. Armazenar e inicializar variáveis globais que são fundamentais para o módulo *thread.h*, tais como o descritor da FIFO pública, o número de threads geradas, uma *seed* usada para gerar números aleatórios e dois booleanos que indicam se o cliente ou o servidor estão fechados;
2. Verificação do *input* introduzido pelo utilizador;
3. Abertura e possível criação da FIFO pública;
4. Configuração de um *signal handler* para um alarme, de modo a indicar o término do tempo imposto pelo utilizador, que, quando ativado, atualizará o estado do cliente;
5. Geração de threads de pedido com intervalos de tempo pseudo-aleatórios, tendo sempre em conta o estado atual do cliente e servidor.

Finalmente, a thread principal termina quando o tempo de execução do cliente ultrapassar o limite imposto pelo utilizador ou quando o servidor encerrar.

## *thread.h*

O módulo *thread.h* tem apenas uma função, cuja principal funcionalidade é tratar do funcionamento das threads geradas pelo módulo *client.h*. Esta thread é capaz de criar e enviar mensagens através da FIFO pública previamente criada, e recebe o respetivo resultado através de uma FIFO privada criada neste módulo.

## *log.h*

O módulo *log.h* está encarregado de imprimir o conteúdo das mensagens com a operação que a respetiva thread executa.

## *message.h*

Este módulo contém a estrutura responsável por guardar as informações relativas às mensagens trocadas entre o Servidor e o Cliente.

## *Contribuição Percentual*

<b>Nome</b>	<b>Percentagem</b>
Adriano	20%
Diogo	20%
Francisco	20%
Pedro	20%
Vasco	20%