Uma imagem com Tipo de letra, Gráficos, texto, design gráfico

Descrição gerada automaticamente

Departamento de Engenharia Informática e de Sistemas Instituto Superior de Engenharia de Coimbra Licenciatura em Engenharia Informática

Sistemas Operativos 2024/2025

Relatório de Trabalho Prático

Martim Oliveira 2022132041

Ricardo Duarte 2022137878

Índice

[Programas 3](#_Toc185170264)

[Manager.c 4](#_Toc185170265)

[Feed.c 4](#_Toc185170266)

[Utils.h 5](#_Toc185170267)

[Manager 6](#_Toc185170268)

[User 6](#_Toc185170269)

[Topic 6](#_Toc185170270)

[Mensagem 7](#_Toc185170271)

[TData 7](#_Toc185170272)

[Conclusão 7](#_Toc185170273)

Introdução

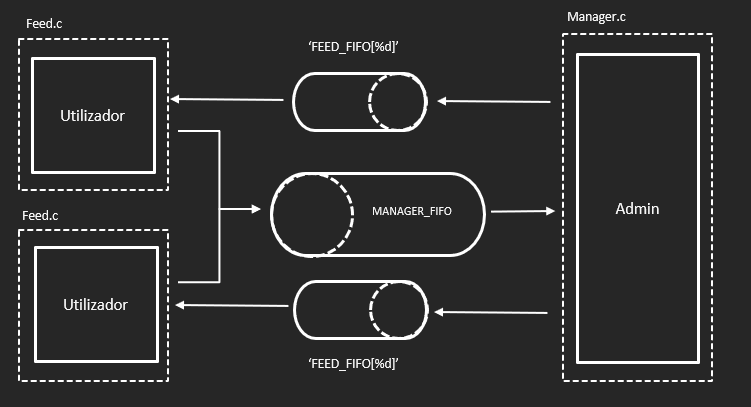
O trabalho prático aborda os conhecimentos do Sistema Unix, em linguagem C, através da utilização de mecanismos deste Sistema Operativo, estes abordados nas aulas teóricas e práticas, tais como sinais, pipes, threads, mutex’s, entre outros.

O trabalho consiste em uma plataforma de envio e receção de mensagens, estas organizadas por tópicos. Um utilizador pode enviar mensagens para um determinado tópico e receber mensagens do mesmo.

Um programa “manager” faz a gestão dos tópicos e mensagens, o outro programa “feed” efetua o envio e receção de mensagens, a comunicação é feita através de (named pipes), este programa “feed” é usado pelos utilizadores, podendo existir diversos “feeds”.

O nosso programa “manager” aceita um máximo de 10 utilizadores, 20 tópicos e cada tópico pode ter no máximo 5 mensagens persistentes, estas que ficam guardadas e enquanto tiverem tempo são vistas por novos utilizadores, quando o programa é encerrado, se existir

alguma mensagem persistente é guardada num ficheiro.

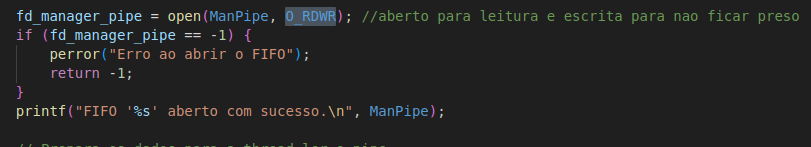


# Programas

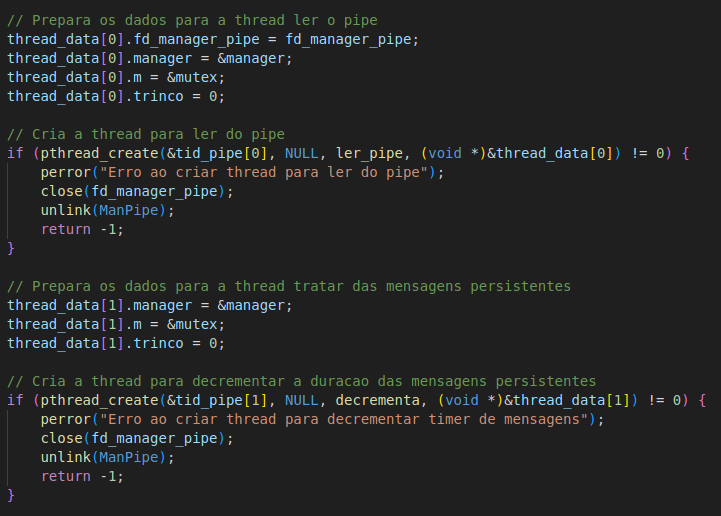
Tal como dito na Introdução, temos os programas “manager.c” e “feed.c”, e .h (headers) adjacentes, mas para além desses 2, temos outros que explicaremos mais à frente, tais como, “utils.h”, “processocom.c” e “makefile”.

## Manager.c

Tal como dito o programa manager, faz o processamento de toda a informação, tal como Utilizadores, tópicos e mensagens.

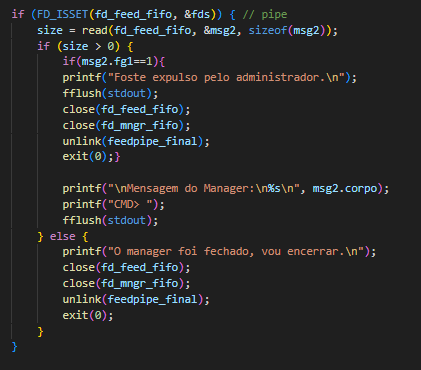
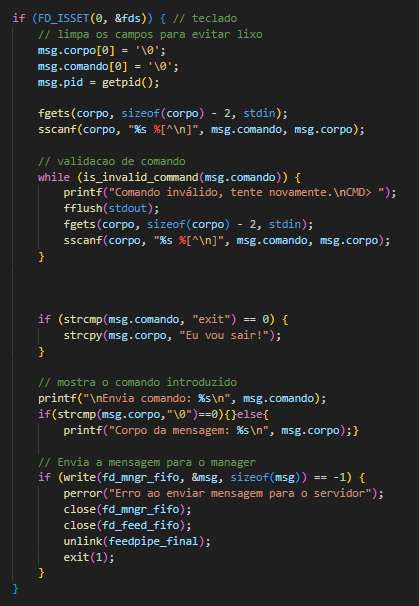
Para efetuar o processamento, o manager usa 5 funções, main(principal), acorda, abre pipes, decrementa e ler pipe, para além dessas ainda conta com a “ajuda” do ficheiro processocom, tendo este outras funções, tais como, “guarda\_mensagens\_persistentes”, “load\_mensagens\_persistentes”, “processa\_comando\_manager”, “processa\_comando\_feed”, estas ajudam com a informação recebida.  
  
Inicialmente o main declara as suas variáveis, para futura utilização. Após isso, tenta abrir o ficheiro onde tem as mensagens e tópicos guardados, se existirem guarda. Após isso cria o pipe para receber informação, liga no modo O\_RDWR(escrita e leitura) de modo a não ficar bloqueado quando não tem utilizadores, apesar de nunca escrever.

Após, cria 2 threads, uma para leitura dos pipes e a outra para decrementar as mensagens.



Após isso fica em ciclo, à espera de comandos introduzidos pelo “admin”, quando este escreve “close” termina o programa.

## Feed.c

O programa Feed.c é bastante simples, fazendo apenas a leitura de inputs vindas do teclado/stdin para enviar para o programa Manager, e, com uso de selects faz a leitura de mensagens vindas do Manager, sejam estas: mensagens de outros feeds ou respostas do Manager a comandos vindos do Feed.

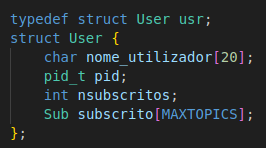
## Utils.h

O ficheiro utils.h serve para definir variáveis a usar no feed e manager, inclusão de bibliotecas e definição de estruturas.

### Manager

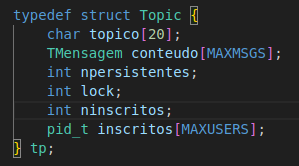
Estrutura do Manager, esta tem as estruturas relativamente aos utilizadores e aos tópicos, esta é a estrutura “principal”, já que contém as estruturas necessárias para o projeto.

### User



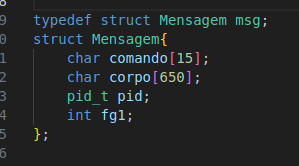
Estrutura relativamente aos Utilizadores, esta guarda na estrutura Sub, o nome do tópico e o pid do utilizador subscrito. Mesmo que não esteja subscrito a nenhum tópico a estrutura guarda o pid do utilizador para usar também nos pipes do utilizador

### Topic



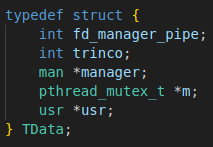
Estrutura onde guarda as mensagens do tópico, se este está disponível, e os pid’s dos utilizadores subscritos.

### Mensagem



Estrutura que serve de comunicação entre os programas, os 2 programas, manager e feed, têm acesso a esta estrutura e usam-na para comunicar entre si.

### TData



Estrutura para o uso das threads, aqui está um ponteiro para mutex, importante, para o programa manager, conseguir efetuar o processamento de 2, ou mais comandos, quer seja do pipe ou pelo admin, sem este ficar “preso”

# Conclusão

O desenvolvimento deste trabalho prático permitiu a aplicação de diversos conceitos fundamentais do sistema Unix, incluindo sinais, pipes, threads e mutex’s, em uma solução prática e funcional. A implementação dos programas “manager.c” e “feed.c”, juntamente com os “headers” auxiliares e o arquivo “makefile”, demonstrou a importância da integração entre diferentes componentes e técnicas de programação para a criação de um sistema robusto e eficiente.

A plataforma desenvolvida atendeu aos requisitos definidos, permitindo o envio e a receção de mensagens organizadas por tópicos, com persistência de dados e controle eficiente de recursos, como o número máximo de utilizadores, tópicos e mensagens. Além disso, a possibilidade de interação simultânea entre utilizadores, com suporte a múltiplas threads, garantiu a escalabilidade e o correto funcionamento da plataforma.

Por fim, este trabalho não apenas consolidou o conhecimento técnico, mas também reforçou significativamente a compreensão das capacidades e desafios do desenvolvimento em ambientes Unix.