Relatório Sistemas Operativos - 1ª meta

Kylix Afonso a2020146228@isec.pt

Gonçalo Salgueirinho a2020142627@isec.pt

21 de novembro de 2021

1 Introdução

Este documento irá servir de justificação e explicação para as estratégias e mecanismos escolhidos e implementados no nosso trabalho prático de Sistemas Operativos e incluira tambem de um breve guia de uso do programa até agora concretizado.

2 Relatório

2.1 Funcionalidades até agora implementadas (obrigatórias à 1ª meta)

- Planificação total do trabalho prático, incluindo a organização da arquitetura do programa, com todos os header files, source files (em princípio não irão ser acrescentados mais ficheiros, à excessão do ficheiro buffer para o named pipe que achamos que não será enviado no momento de submissão), das estruturas de dados (poderão ainda sofrer alterações) e constantes simbólicas.
- Integração de um mecanismo de comunicação básico, baseado na implementação de dois pipes anónimos entre o balcão e o classificador (um pipe do balcão para o classificador - responsável pelo envio direto dos sintomas através do stdin do classificador - e outro do classificador para o balcão - responsável pela obtenção da especialidade e urgência do classificador através do stdout do classificador).
- Implementação de um mecanismo simples de obtenção de valores de variáveis de ambiente, do seu processamento e posterior decisão de usar ou não os valores encontrados (no caso dos valores cumprirem os requisitos, ou, por outras palavras serem valores númericos positivos) ou de os descartar e usar os valores definidos por omissão (constantes que foram definidos no header file "balcao.h").

2.2 Funcionalidades até agora implementadas (não obrigatórias à 1ª meta)

- Makefile com todas as regras e funcionalidades nela necessitadas (não dá relink do utils.o):
  - balcão, médico, cliente, clean e all.
- Verificação de execução do balcão e criação de uma resposta adequada para cada executável:
  - No balcão: não permite executar um balcão caso exista outro balcão em execução.
  - No médico: só permite executar caso o balcão esteja em execução.
  - No cliente: só permite executar caso o balcão esteja em execução.

2.3 Justificação para as principais escolhas de implementação

- Unnamed pipe: esta escolha de implementação é a mais evidente, sendo que: 1) é um mecanismo que permite a leitura de um lado e escrita do outro; 2) é um mecanismo não muito pesado; 3) é aquele que demos nas aulas práticas e teóricas de Sistemas Operativos, sendo que no guião do trabalho prático está mais que explícito que só devemos usar mecanismos de comunicação e outros que tenhamos aprendido nas aulas; 4) para além disto tudo, o classificador só trabalha com stdin e stdout, pelo que named pipes não fariam sentido nem tornavam diretamente possível a comunicação com o classificador.
- Makefile (na 1ª meta): esta foi uma escolha pessoal da nossa parte, fizemo-la pois achamos que o trabalho que dava fazer um Makefile era muito menor do que aquela que usar aliases / escrever manualmente gcc -Wall -Werror -Wextra NOME.c -o NOME cada vez que necessitavamos de compilar / recompilar dava.
- Write / read ao invés de scanf / printf, entre outras funções de stdin / stdout: como em partes específicas do trabalho iria ser necessário usarmos as funções read e write, achamos que para manter um padrão de escrita de código semelhante ao longo de todo o programa, para além das consequências a nível de performance (positivas) e a facilidade de uso destas funções, seria muito melhor usarmos só funções de sistema ao invés de funções standard para input / output, para não falar que um dos objetivos (pelo nosso entendimento) da cadeira de Sistemas Operativos é aprender a usar o sistema POSIX.
- Função isBalcaoRunning (na 1ª meta): foi abordado, numa aula prática o conceito de DIR e dirent, não foi muito aprofundado mas tocou-se no assunto, nós sendo alunos curiosos e empenhados de Informática, fomos fazer pesquisa sobre o assunto, sendo que o conceito não era muito complexo e mais tarde poderemos sempre fazer mudanças caso achemos necessário, decidimos implementar já uma função que devolve um true ou false caso o balcão esteja a correr ou não, respetivamente, de forma a planear já minimamente como será determinado a execução ou não dos executáveis.
- Os vários valores devolvidos pelas funções que possam falhar: neste preciso momento estamos a devolver inteiros como 1, 2, 3 e por aí adiante, de modo a verificar se uma função correu na perfeição ou não. Estes valores em principio serão só para marcar os lugares onde mais tarde iremos estar a disparar sinais de erro que possam ser captados e tratados.

2.4 Trabalho futuro

- Implementação do uso de um named pipe para estabelecer protocolos de comunicação entre os vários "nós"do programa (cliente, balcão e médico).
- Implementação do uso de sinais para uma melhor gerência de recursos e implementaçao de mecanismos de tratamento de erros que possam surgir durante o decorrer do programa, como p.e. quando um read falha.
- Implementação do uso de threads para tratar funcionalidades que no nosso ponto de vista assim o necessitem.

3 Guia de uso prático do MEDICALso

3.1 Makefile

- make all - comando que tem como objetivo compilar todos os source files de modo a obter todos os executaveis necessários ao funcionamento do MEDICALso;
- make balcão - compila somente só o balcao.c e o utils.o (é dependência de todos os objetos);
- make cliente - compila somente só o cliente.c e o utils.o;
- make medico - compila somente só o medico.c e o utils.o;
- make clean - remove todos os objetos, incluindo o utils.o;
- make re - tem como objetivo permitir debugging, no caso, obriga os ficheiros todos a recompilar (tem como dependências o clean e o all).

3.2 Balcão

Para se fazer uso do balcão, é necessário compilá-lo e posteriormente executá-lo, usando, por exemplo, os comandos 'make balcao && ./balcao' (sem os ' '). É preciso notar que, caso não sejam definidas as variáveis de ambiente MAXCLIENTES e MAXMEDICOS, eles irão receber, por defeito, os valores de MAX_CLIENTES_DEFAULT e MAX_MEDICOS_DEFAULT, respetivamente, definidos no balcao.h. Após executarmos o balcão, notamos que a nós se apresenta "[admin]: "e o programa fica à espera, isto dá a ideia que devemos introduzir algum tipo de comando / input, é exatamente isto que queremos, é neste momento do programa que poderemos inserir comandos como "encerra", "freq", entre outros (realisticamente, nesta meta, ainda só se encontra implementado o "encerra"e ainda irá ser alterado mais tarde, quand tivermos a possiblidade de comunicar com o cliente e o médicos através da named pipe). Nesta fase, poderemos também introduzir um outro comando (que irá ser eliminado assim que possível) denominado de "sintomas", introduzindo este comando, é de notar que nos é possível introduzir texto na consola, que através dos pipes anónimos já implementados envia essa informação para o classificador e recebe a especialidade e urgência por ele inferidas. Para sair do mecanismo de comunicação com o classificador, é necessário introduzir o texto "fim". Para sairmos do balcão, introduzimos o comando "encerra".

3.3 Cliente

Para se fazer uso do cliente, é necessário compilá-lo e posteriormente executá-lo (para executar o cliente, o balcao tem obrigatoriamente que estar em execução), usando, por exemplo, os comandos 'make cliente && ./cliente ARGUMENTOS' (sem os ' ' e no espaço dos ARGUMENTOS, inserimos o nome do cliente, caso queiramos introduzir um nome e um sobrenome, por exemplo, teremos que recorrer às haspas pois o programa só considera o argv[1], ou seja o primeiro argumento a ele passado). É de notar que o Cliente ainda está numa fase muito prematura. Após iniciarmos o cliente, ele irá nos pedir para introduzir os nossos sintomas. Após introduzirmos os sintomas, o programa Cliente encerra.

3.4 Médico

Para se fazer uso do medico, é necessário compilá-lo e posteriormente executá-lo (para executar o medico, o balcao tem obrigatoriamente que estar em execução), usando, por exemplo, os comandos 'make medico && ./medico ARGUMENTOS' (sem os ' ' e no espaço dos ARGUMENTOS, inserimos o nome do medico e a especialidade do mesmo, separados por um espaço). É de notar que o Médico ainda está numa fase muito prematura. Logo após iniciarmos o Medico, o programa Medico encerra.

Referências

Até ao momento as nossas principais fontes de referência são as aulas práticas e teóricas da cadeira, no nosso caso lecionadas pelo professor José Luis Nunes e João Durães, respetivamente. Em último caso já consultamos o YouTuber CodeVault que tem videos sobre tópicos diversos, como unnamed pipes, forks, threads, entre outros.