Relatório da segunda entrega (Cadeira de Sistemas Operativos) (18/11/2017)

A realização desta entrega foi feita de modo equilibrado. Sendo o grupo constituído por 2 elementos, um dos elementos fez o admin.c e o outro o sevidor.c e o cliente.c.

Para suportar o ficheiro admin.c foi criado o ficheiro estruturas.h que define as estruturas utilizadas.

O admin.c tem tanto a leitura como a escrita dos ficheiros .dat e .txt separados de modo a reduzir a redundância no código. Por outro lado, para suportar a leitura dos ficheiros TXT foi usada uma função desenvolvida na aula 6 denominada de “obter\_substring(…)”.

Foi também imposto um limite de utilizadores e viaturas de modo a facilitar a gestão dos mesmos e validações nos inputs do utilizador para que estes façam sentido em contexto do projeto e que não acabem por estragar a estrutura dos ficheiros, arruinando o funcionamento do programa. Tanto na escrita como na leitura o programa imprime o nome dos ficheiros que lê ou escreve.

Em relação aos ficheiros cliente.c e servidor.c, como ambos partilham de uma função para adicionar a data e hora da ação, esta função apresenta-se num ficheiro timemsg.c que é compilado juntamente com os mesmos, sendo usado um ficheiro chamado timemsg.h para o #include em ambos os ficheiros, contendo a declaração desta mesma função juntamente com apenas #includes relacionados com o tempo.

Em particular no ficheiro cliente.c, apenas armamos os sinais SIGUSR1 e SIGUSR2 a uma mesma função chamada “trata\_sinal(int sinal)” que o que faz é a partir de um switch case, e conforme o sinal (SIGUSR1 ou SIGUSR2) vai escrever uma string diferente na consola.

Por fim no ficheiro servidor.c temos uma constante definida chamada SIZEOFMESSAGE que basicamente é um inteiro que representa o tamanho global das mensagens que estou a escrever (tamanho suficientemente grande para as mensagens que são escritas até ao momento, podendo ser alterada futuramente), temos também uma variável partilhada chamada filho, que serve para guardar o PID do fiscal (filho) em que é diferente em cada um dos processos (Pai e filho), sendo esta variável utilizada para saber qual o PID que o pai vai mandar sinal SIGINT quando o pai recebe um sinal para terminar a partir do teclado (SIGINT), esperando de seguida, com um wait(NULL), que este termine, e por fim apagar o ficheiro servidor.pid, e escreve na consola e no servidor.log as suas ações terminando posteriormente. O fiscal antes de terminar, escreve na consola e no servidor.log que o vai fazer. Ainda no caso do fiscal, este é armado com o tratamento para o sinal SIGALRM, e armado o “alarm(60)” para que este sinal seja lançado dentro de 60 segundos, sendo a mesma função usada dentro do tratamento do sinal, para que isto aconteça de minuto a minuto (até o utilizador matar um dos processos, ou outro processo o faça).

Ao contrario do que foi feito no cliente.c; no servidor.c temos funções separadas para o tratamento de cada sinal. Isto acontece, pois, estes tratamentos não partilham muitas ações idênticas ao contrario do que acontecia no caso do cliente.c. Em relação aos “logs” do servidor, estes são escritos com o auxilio de uma função chamada “write\_log(char message[])” em que lhe damos uma “string” (vetor de char), esta função abre o ficheiro para “append” e escreve essa “string” no mesmo, fechando o ficheiro novamente quando acabar.