

## Relatório trabalho prático SO Parte 2

**C1)** Peço o respetivo input ao utilizador e verifico se é válido. Utilizo em ambas *scanf()*.

**C2)** Crio uma Consulta e escrevo as suas informações no ficheiro *txt*. Para isso utilizo *FILE\**, verifico se o ficheiro abriu corretamente e por fim escrevo as informações separadas por *\n* (newline) para mais tarde serem lidas de forma fácil.

**C3)** Verifico que o ficheiro *SrvConsultas.pid* foi aberto corretamente, caso contrário muito provavelmente o *Servidor.c* não está a correr ou houve algum tipo de erro. Se tudo estiver correto, o *pid* do servidor será lido com um *fgets()* e o seu valor atribuído a uma variável global do *Cliente.c*. Por fim é enviado um **SIGUSR1** para o Servidor.

**C4)** Através de *handler\_sighup()* preparo o meu cliente para receber um **SIGHUP** do servidor. Após isso será apresentada a mensagem predefinida, a variável global *'comecou'* será definida como *'t'* (funciona como um true/false) e por fim removo o ficheiro *PedidoConsulta.txt*. *'comecou'* serve para ter a informação caso o servidor tenha dado início à respetiva consulta. *Verificar\_pid\_consulta()* utilizo para garantir que um Cliente não elimina um *PedidoConsulta.txt* que contenha os dados doutra consulta (Por exemplo após receber um **SIGINT**).

**C5)** Em *handler\_sigterm()*, verifico se a consulta começou com *'comecou'*, e consoante o seu valor devolvo a respetiva mensagem ou erro. Por fim *exit(0)* para fechar o cliente.

**C6)** Em *handler\_sigusr2()*, devolvo a mensagem predefinida, e removo o ficheiro *PedidoConsulta.txt*. Fecho o cliente.

**C7)** *handler\_sigint()* confirma que o cliente cancelou o seu pedido, remove o ficheiro *PedidoConsulta.txt* e fecha o cliente.

**C8)** Para esta alínea criei *verificar\_disponibilidade()* que me diz se já existe algum ficheiro *PedidoConsulta.txt* (se não existir é porque já foi aberto no servidor e portanto eliminado pelo cliente). Se existir será criado um *alarm(10)* que irá verificar o mesmo 10 segundos depois. Esta alínea serve como validação à C2) e portanto, quando tudo se verificar corretamente, será executado dessa mesma forma.

**S1)** Adiciono as variáveis globais *tipos[4]* (0-perdidas/1-tipo1/2-tipo2/3-tipo3) para servirem de contadores para o número de consultas de cada tipo. Crio um *for* que percorre a lista de Consultas *lista\_consultas*, fazendo sempre a alteração *lista\_consultas[i].tipo = -1*.

**S2)** Igualo a variável global *pid\_servidor* a *getpid()* para obter o respetivo PID.

**S3)** *handler\_sigusr1* trata o sinal **SIGUSR1**.

**S3.1)** Leio o ficheiro *PedidoConsulta.txt*, verificando antes se não existe qualquer tipo de erro. Dentro dum *for* utilizo *fgets()* para obter a informação do ficheiro linha a linha e no fim associa-a a uma Consulta. Se houver algum erro, o campo tipo de Consulta será definido como -1 e fará com que em *iniciar\_consulta()* nada seja feito.

**S3.2)** *Printf()* da mensagem predefinida com os campos obtidos em S3.1).

**S3.3)** Para esta alínea criei a função *verificar\_vaga()* que vai procurar a primeira posição disponível (onde *tipo == -1*) e devolver o seu índice. Se a lista estiver cheia será devolvido -1, enviado um **SIGUSR2** ao cliente e incrementado o contador de consultas perdidas.

**S3.4.1) e S3.4.2)** Crio um *fork()* onde o processo filho envia um **SIGHUP** ao cliente, faz *sleep(10)* e no fim envia um **SIGTERM**. Enquanto isto, o processo pai espera que o

filho acabe (*wait(NULL)*) e depois apaga a respetiva consulta. Criei *apaga\_consulta()* para procurar por pid na lista de forma a apagar a consulta correta sempre.

**S4)** Para esta alínea criei 2 funções auxiliares, *atualizar\_stats()* (que após algumas verificações abre o ficheiro binário *StatsConsultas.dat* e lê os seus valores, associando-os aos respetivos contadores de consultas do servidor) e *escrever\_stats()* (que é responsável por, antes de fechar o Servidor, escrever os respetivos valores dos contadores de consultas no ficheiro binário *StatsConsultas.dat*).

Trabalho realizado por: Francisco António, nº 92613, EI-B7