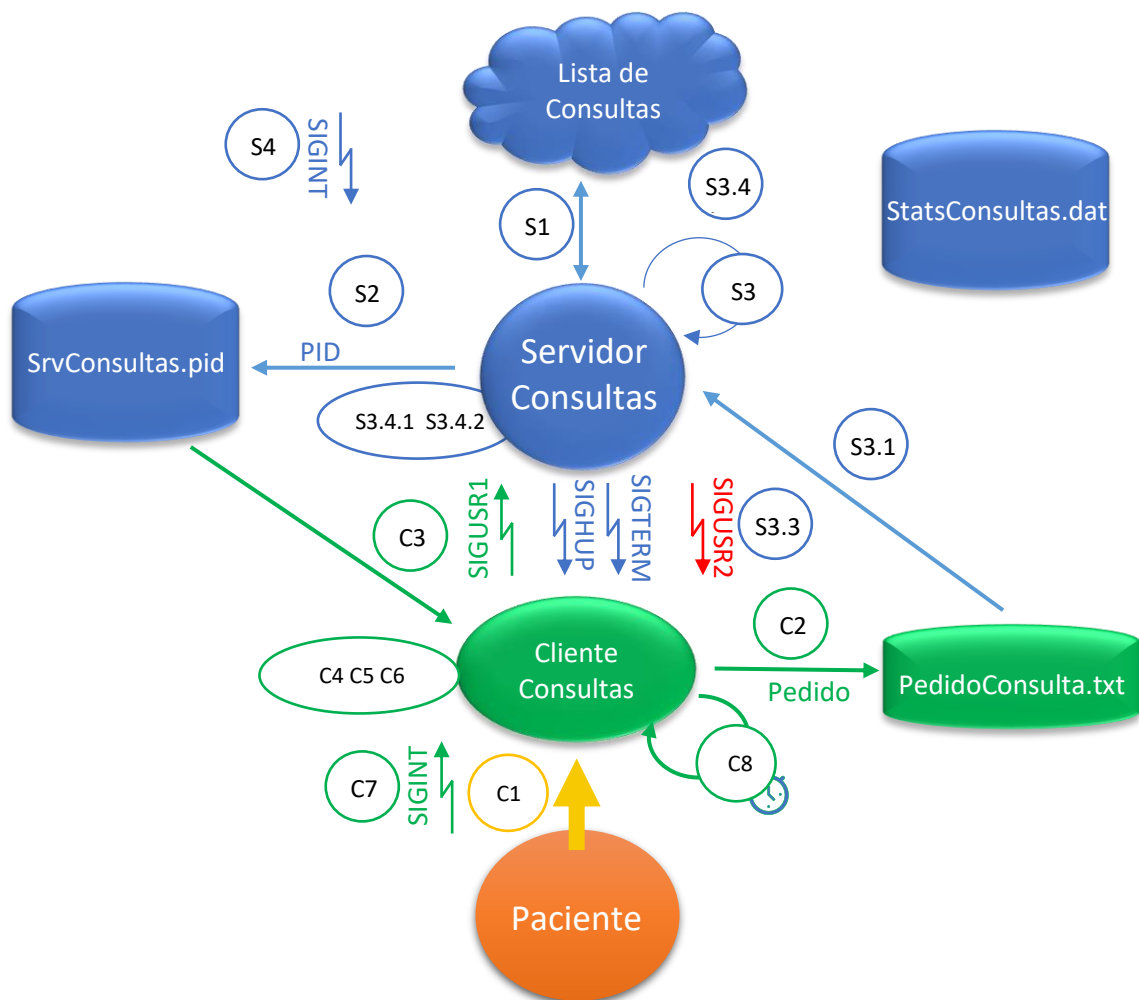


Projeto Cliniq-IUL (Parte 2)

Processos e sinais

Data de entrega: 24 de novembro de 2020

Nesta parte do trabalho, será implementado um modelo simplificado de Triage e Consultas de pacientes no sistema Cliniq-IUL, baseado em comunicação por sinais entre processos, utilizando a linguagem de programação C. Considere o seguinte diagrama, que apresenta uma visão geral da arquitetura pretendida:



Ideia Geral: Pretende-se, nesta fase, simular a realização de consultas no sistema Cliniq-IUL. Assim, teremos dois módulos – **Cliente** e **Servidor**.

Cliente.c

O módulo **Cliente** é responsável pelo encaminhamento dos pacientes. Este módulo será utilizado para solicitar o encaminhamento dos pacientes para as suas consultas. Após esta indicação, o paciente é “encaminhado” e, caso haja disponibilidade, é realizada a consulta, ficando este módulo a aguardar até que a mesma termine.

A estrutura de dados das consultas é a seguinte:

```
typedef struct {  
    int tipo;           // Tipo de Consulta: 1-Normal, 2-COVID19, 3-Urgente  
    char descricao[100]; // Descrição da Consulta  
    int pid_consulta;   // PID do processo que quer fazer a consulta  
} Consulta;
```

Assim, definem-se as seguintes tarefas a desenvolver:

- C1)** Pede ao Paciente (utilizador) que preencha os campos tipo (Tipo de consulta, valores 1, 2 ou 3) e descrição (e.g., “Clínica Geral”, “Ortopedia”, “Pneumologia”).
- C2)** Cria um elemento do tipo **Consulta** com as informações sobre a consulta recolhidas anteriormente, preenchendo o valor pid_consulta com o PID deste processo Cliente. Escreve as informações desse elemento **Consulta** criando um novo ficheiro de texto **PedidoConsulta.txt**;
- C3)** Lê o PID do servidor de Consultas do ficheiro **SrvConsultas.pid** e envia um sinal **SIGUSR1** ao módulo Servidor de Consultas para que este dê início à consulta;
- C4)** Arma e trata o sinal **SIGHUP**, para o caso do servidor indicar que a consulta teve início. Se receber esse sinal, escreve no ecrã a mensagem “Consulta iniciada para o processo <pid_consulta>”, apaga o ficheiro **PedidoConsulta.txt**, e aguarda a conclusão da consulta;
- C5)** Arma e trata o sinal **SIGTERM**, para o caso do servidor indicar que a consulta terminou. Se receber esse sinal, escreve no ecrã a mensagem “Consulta concluída para o processo <pid_consulta>” e termina o processo. Atenção que apenas deverá aceitar este sinal **SIGTERM** se anteriormente tiver recebido um **SIGHUP** para iniciar a consulta, caso contrário assinala um erro;
- C6)** Arma e trata o sinal **SIGUSR2**, para a eventualidade de receber a indicação do servidor de que não é possível realizar a consulta. Se receber esse sinal, escreve no ecrã a mensagem “Consulta não é possível para o processo <pid_consulta>”. Apaga o ficheiro **PedidoConsulta.txt** e termina o processo;
- C7)** O módulo Cliente deve armar o sinal **SIGINT**, para que, no caso do Paciente não querer ficar à espera, possa terminar o pedido com o atalho <CTRL+C>. Nesse caso, escreve no ecrã a mensagem “Paciente cancelou o pedido”. Apaga o ficheiro **PedidoConsulta.txt** e termina o processo;
- C8)** Alínea extra: Como pode ter já percebido, se houver um cliente que faz um novo pedido antes do anterior ter sido satisfeito, o ficheiro **PedidoConsulta.txt** é escrito por cima com o novo pedido. Faça uma validação em **C2)** de forma a que, antes de criar o **ficheiro PedidoConsulta.txt**, verifique se o mesmo já existe ou não. Se já existir, não cria o ficheiro nem envia o sinal **SIGUSR1** ao servidor. Escreve antes uma mensagem de erro no ecrã, e arma um alarme para tentar novamente submeter o pedido após 10 segundos (sem precisar pedir novamente os dados ao utilizador).

Servidor.c

O módulo **Servidor** de Consultas é responsável pela realização das consultas que chegam ao sistema Cliniq-IUL. Este módulo estará normalmente ativo, à espera de pedidos de consulta. As consultas têm duração de 10 segundos. Findo o tempo da consulta, este módulo sinaliza o paciente de que a sua consulta terminou. Este módulo deverá possuir contadores de consultas por tipo, e uma lista com capacidade para agendar 10 consultas:

```
Consulta lista_consultas[10];
```

O módulo de **Servidor** de Consultas é responsável pelas seguintes tarefas:

- S1)** Inicia a lista de consultas com o campo tipo = -1 (“Limpar” a lista de consultas). Deverá ter um contador para cada tipo de consultas e um contador para número de consultas perdidas, todos iniciados a 0;
- S2)** Regista o PID do seu processo no ficheiro **SrvConsultas.pid**;
- S3)** Arma e trata o sinal **SIGUSR1** para que sejam tratados os pedidos de consultas que chegam ao sistema. Quando receber este sinal:
 - S3.1)** Lê a informação do ficheiro **PedidoConsulta.txt**
 - S3.2)** Escreve no ecrã a mensagem “Chegou novo pedido de consulta do tipo <tipo>, descrição <descricao> e PID <pid_consulta>” com os campos preenchidos corretamente;
 - S3.3)** Verifica se a Lista de Consultas tem alguma “vaga”. Se todas as entradas da Lista de Consultas estiverem ocupadas, escreve no ecrã “Lista de consultas cheia”, manda um sinal **SIGUSR2** ao processo <pid_consulta>, e incrementa o contador de consultas perdidas;
 - S3.4)** Insere a consulta na lista de consultas, com os campos devidamente preenchidos, escreve no ecrã “Consulta agendada para a sala <índice_da_lista>”, e incrementa o contador de consultas do tipo correspondente. O módulo Servidor então cria um processo filho (fork). O **processo pai** irá ficar à espera que o processo filho termine e quando isso acontecer, apaga a entrada correspondente da lista de consultas. Por sua vez, **o novo processo filho**:
 - S3.4.1)** Envia um sinal **SIGHUP** ao processo <pid_consulta> correspondente, a indicar o início da consulta;
 - S3.4.2)** Aguarda 10 segundos e escreve no ecrã “Consulta terminada na sala <índice_da_lista>”. Envia um sinal **SIGTERM** ao processo <pid_consulta> correspondente, e termina o processo filho.
- S4)** O módulo **Servidor** de Consultas deve armar e tratar o sinal **SIGINT**, para que possa ser encerrado pelo administrador com o atalho <CTRL+C>. Quando isso acontecer, deverá remover o ficheiro **SrvConsultas.pid** e atualizar um ficheiro binário **StatsConsultas.dat** com as estatísticas de consultas perdidas e nº de consultas de cada tipo, com os valores armazenados nesta sessão, do tipo:

Perdidas	Tipo 1	Tipo 2	Tipo 3
12	354	234	114

Após atualizar as estatísticas, o processo **Servidor** deverá terminar.

Boa Sorte!!!

Os docentes