

Sistemas Operativos 21/22

Trabalho Prático - Programação em C para UNIX

O trabalho prático de sistemas operativos consiste na implementação de um sistema para gerir atendimento de clientes em estabelecimentos médicos. O sistema, denominado **MEDICALso**, destina-se a correr em ambiente Unix linha de comandos e irá mediar a interação entre doente, médico e balcão de atendimento. O trabalho aborda acima de tudo os conhecimentos do sistema Unix, sendo necessária a capacidade de desenvolvimento de aplicações para este sistema operativo. Em momento algum será necessário qualquer tipo de conhecimentos de medicina.

O trabalho prático deve ser concretizado em linguagem C, para plataforma Unix (Linux), usando os mecanismos deste sistema operativo abordados nas aulas teóricas e práticas. No que respeita à manipulação de recursos do sistema, deve ser dada prioridade ao uso de chamadas ao sistema operativo¹ face ao uso de funções biblioteca² (por exemplo, devem ser usadas `read()` e `write()` em vez de `fread()` e `fwrite()`). O trabalho foca-se no uso correto dos mecanismos e recursos do sistema e não é dada particular preferência a detalhes de escolhas na implementação de aspetos de carácter periférico à disciplina (por exemplo: a pergunta “devo usar uma lista ligada ou um *array* dinâmico?” terá como resposta um encolher de ombros).

O recurso a bibliotecas que façam por si só parte do trabalho, ou que ocultem o uso dos recursos e mecanismos estudados na disciplina não é permitido. O recurso a mecanismos do sistema Unix que não tenham sido abordados nas aulas é permitido, mas terão que ser devidamente explicados, em particular durante defesa. Não é permitida uma abordagem baseada na mera colagem de excertos de outros programas ou de exemplos. Todo o código apresentado terá que ser entendido e explicado por quem o apresenta, caso contrário não será contado para a nota.

São descritas primeiro as funcionalidades e no final as regras de funcionamento do sistema pretendido.

1. Descrição geral e conceitos principais envolvidos

Neste trabalho deve implementar um sistema para gerir o atendimento de clientes no estabelecimento médico, designado de **MEDICALso**, que encaminha e medeia a interação entre clientes, médicos e balcão de atendimento. Neste enunciado, a palavra *sistema* referir-se-á ao *sistema de atendimento MEDICALso*. Sempre que for necessário referir o sistema operativo, utilizar-se-á a expressão *sistema operativo* ou *Unix/Linux*.

¹ Documentadas na secção 2 das *manpages*

² Documentadas na secção 3 das *manpages*

Programas envolvidos

O sistema pretendido abarca 4 programas principais: cliente, médico, balcão e classificador.

- O **cliente** é o programa usado pelo utente (utilizador que se dirige ao estabelecimento médico). Serve para fazer a interação entre esse utente e o sistema. Existirão tantas instâncias (programas a correr) do cliente como utentes.
- O **balcão** é o programa que recebe pedidos de atendimento por parte de clientes e os encaminha para o médico da especialidade que lhes corresponde, após uma classificação inicial (tipo de pedido, prioridade, etc.). Não é o balcão que faz a classificação do pedido. Este programa age como a receção da urgência de um hospital comum. Existirá apenas uma instância (deste programa) a executar.
- O **classificador** é o programa que recebe o pedido oriundo dos clientes, encaminhado pelo balcão, e os classifica quanto a especialidade e prioridade, encaminhando-o de volta para o balcão para encaminhamento posterior a um médico da especialidade.
- O **médico** é um programa que age como interface de utilizador entre um profissional de saúde (médico especialista) e o sistema. Existirão tantas instâncias deste programa como médicos especialistas. Pode haver mais do que um médico da mesma especialidade e isso não implica nenhuma complexidade adicional.

Mais adiante são dados detalhes adicionais acerca destes programas.

Utilizadores

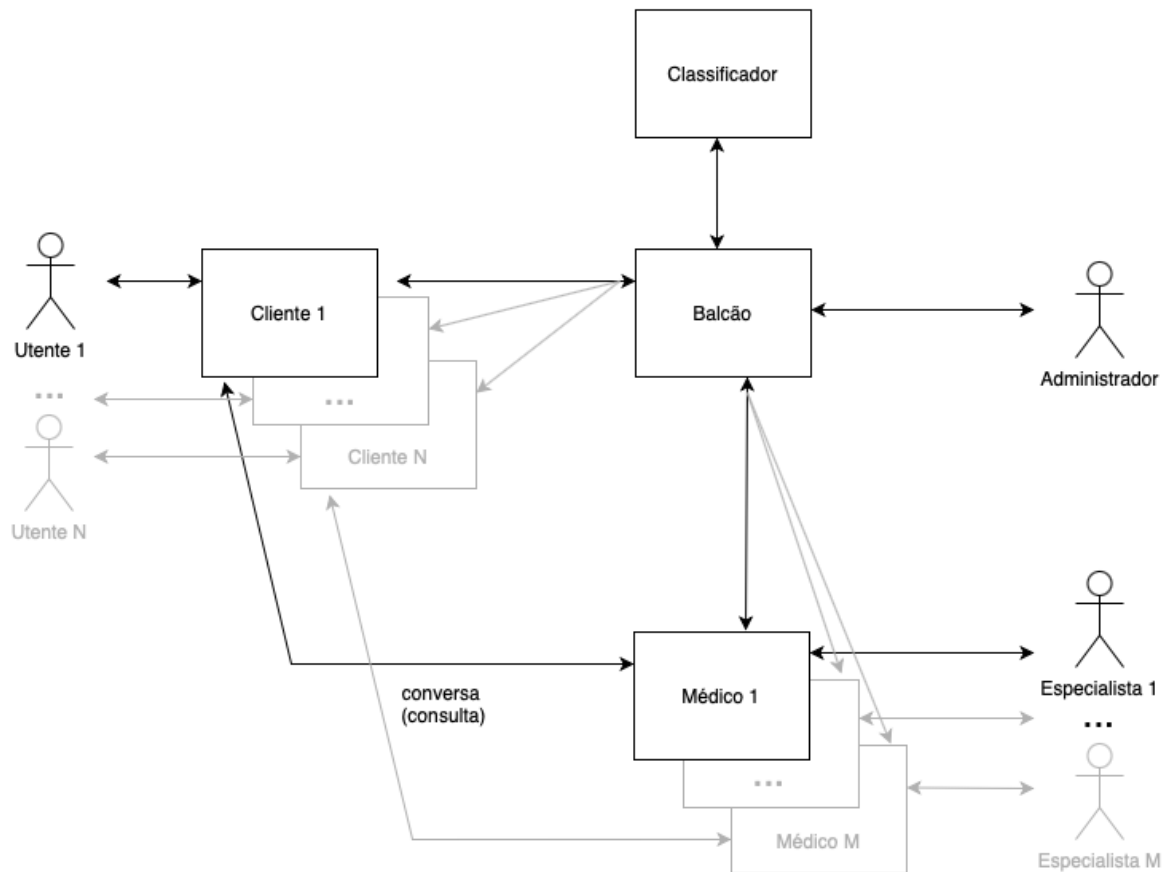
Toda a interface com o utilizador é feita em ambiente de consola (terminal modo-texto). As ações dos utilizadores são especificadas por comandos escritos, e a informação apresentadas pelos vários programas é sempre texto. Não são necessários gráficos nem cores.

Todos os utilizadores do sistema correspondem ao mesmo utilizador do sistema operativo onde os vários programas do sistema são executados. Para simular um novo utilizador será simplesmente aberto um novo terminal. Nenhum dos utilizadores do sistema terá que fornecer password – apenas uma identificação para o distinguir dos demais utilizadores.

Existem três tipos de utilizador neste sistema:

- **Utente.** É aquele que pretende ser atendido. Utiliza o programa cliente, e apenas este programa, para interagir com o sistema. Cada utente executa um cliente.
- **Administrador.** Controla o sistema. Interage com o sistema através do programa balcão. Pode consultar informação relativamente ao sistema e pode forçar a saída de um cliente ou de um médico do sistema. Este “administrador” não tem nada a ver com o administrador (*root*) do sistema operativo.
- **Especialista.** É o profissional de saúde (médico especialista). Para simplicidade de leitura será designado apenas por “especialista” e não deve ser confundido com o “médico”, pois este último, no enunciado, refere-se a um programa e não a um utilizador.

A figura seguinte ilustra a ideia geral do sistema **MEDICALso**. Nota: O número de programas cliente (N) não está relacionado com o número de programas médicos (M).



2. Utilização do sistema

O programa apresenta a seguinte funcionalidade, indicada do ponto de vista de cada tipo de utilizador.

Do ponto de vista do utente – programa cliente

O utente executa o programa cliente e indica: o seu nome e os seus sintomas. O nome é indicado por parâmetro na linha de comandos; os sintomas são indicados respondendo a uma pergunta do programa. Relativamente aos sintomas, o utente escreve o que lhe apetece, sendo que o texto irá depois analisado por um classificador que, mediante algumas palavras-chave, vai atribuir automaticamente uma *especialidade* e uma *prioridade*.

O cliente interage então com o balcão, enviando os sintomas e obtendo de volta as seguintes informações, que mostra ao utente: a área de especialidade em que os sintomas se inserem, a prioridade, o número de utentes nesse instante à frente dele nessa especialidade, e o número de especialistas dessa área neste momento no sistema. O utente permanece em linha, a aguardar que seja encaminhado para um especialista (mesmo que haja zero especialistas nesse momento – podem sempre chegar alguns, entretanto). O utente é livre de desistir a qualquer momento, indicando essa intenção ao cliente, que depois informa o balcão.

Quando chegar a vez de ser atendido, o balcão informa o cliente, que por sua vez informa o utente, e de seguida o cliente passa a interagir com o médico. Nesta parte, a interação é do tipo diálogo: o cliente apresenta texto (pergunta) enviada pelo médico e fica a aguardar que o utilizador escreva a resposta para a enviar ao médico. Se a pergunta recebida pelo médico for “adeus”, então a consulta está terminada e o cliente deverá terminar, informando o utente. Se a resposta enviada pelo utente for “adeus”, então o médico irá assumir que a consulta está encerrada e não interagirá mais com este utente. Quando a consulta termina o cliente não precisa de fazer mais nada de especial. A conversa entre o utente o especialista será, em

princípio pergunta-resposta. No entanto, qualquer um deles poderá fazer duas ou mais perguntas seguidas, ou enviar mais do que uma resposta de seguida.

Exemplo de lançamento do programa cliente

```
$./cliente manuel
```

O utilizador utente apenas interage com o programa cliente.

Do ponto de vista do especialista – programa médico

O utilizador especialista executa uma instância do programa médico e passa a integrar o sistema. Indica ao sistema o nome (uma palavra) e a especialidade (uma palavra) através de parâmetros da linha de comandos. O médico interage com o balcão, o qual regista esta informação para passar a interagir com este médico. Pode haver especialistas com nomes repetidos e podem existir especialidades repetidas. Inclusivamente pode haver especialistas com o mesmo nome e a mesma especialidade e isso não afeta em nada o funcionamento do sistema, dado que o balcão regista a existência de instâncias de médicos em execução e não apenas nomes de utilizadores.

O médico fica a aguardar que o balcão lhe indique que tem um novo utente para atender. A informação de novo utente parte sempre do balcão, sendo recebida também toda a informação de controlo (independente da implementação) que permite ao programa médico estabelecer o diálogo com o utente. Este estabelecimento de diálogo é completamente automático e o utilizador especialista apenas tem que aguardar.

Quando o diálogo com o utente é estabelecido, o especialista deve escrever uma pergunta e aguardar a resposta do utente. No entanto é livre de fazer duas perguntas seguidas, não sendo obrigado a aguardar uma resposta entre ambas as perguntas. A qualquer momento o especialista pode dar por encerrada a consulta escrevendo a palavra “adeus” como texto da sua próxima pergunta (que é enviada ao utente). Quando o programa médico deteta a palavra “adeus” assume o fim da interação com este utente e informa automaticamente o balcão que está disponível para o próximo utente, ficando a aguardar que o balcão lhe diga qual é.

O especialista pode terminar a sua ligação ao sistema a qualquer momento indicando a palavra “sair”, quer esteja a meio de uma consulta (implica automaticamente o fim de consulta), quer esteja a aguardar por um utente. O sistema será avisado que este especialista deixou de estar presente.

Exemplo de lançamento de um programa médico

```
$./medico duarte oftalmologia
```

O especialista pode indicar qualquer especialidade, mesmo uma que nem exista. No entanto, se indicar uma especialidade inexistente ou que o sistema não reconheça, ficará sem receber utentes.

O utilizador especialista apenas interage com o programa médico.

Interação utente-especialista

Trata-se de um diálogo pergunta-resposta. É suposto ser o especialista a indicar a primeira pergunta, depois o utente responde, repetindo este comportamento. No entanto, trata-se de seres humanos que estão por detrás destes programas e podem decidir interagir de outra forma, começando qualquer deles primeiro, ou demorar a responder, permitindo ao outro enviar várias perguntas ou respostas seguidas. Os respetivos programas cliente e médico devem permitir esta característica.

Do ponto de vista do administrador – programa balcão

O administrador inicia o sistema MEDICALso ao lançar o programa balcão. O balcão passa de imediato a gerir todo o sistema de forma automática não sendo necessária mais nenhuma intervenção do administrador (mas este pode intervir, se o desejar).

O balcão informa o administrador sempre que ocorrem os seguintes eventos: novo utente (quem), especialidade do utente determinada pelo classificador (quem e qual), utente começou a ser atendido (quem e por quem), consulta do utente terminou (quem e por quem), especialista juntou-se (quem e qual especialidade), especialista saiu (quem e qual especialidade).

O balcão informa também qual a ocupação das filas de espera (para cada especialidade, quantos utentes em espera). Esta informação ocorre automaticamente, a cada 30 segundos, mas este período pode ser modificado pelo administrador.

O balcão permite ao administrador a escrita de comandos para ajustar e controlar o seu funcionamento. Os comandos podem ser escritos a qualquer altura, a par com todas as restantes atividades do balcão. Os comandos são:

- `utentes` – lista os utentes em fila de espera indicando qual a especialidade e qual a prioridade, e também os utentes atualmente a serem atendidos, em que especialidade e por qual especialista;
- `especialistas` – indica a lista de especialistas conhecidos e estado atual (à espera, a atender o utente X);
- `delut X` – remove o utente em espera X, informando-o através do seu programa cliente (que depois encerra). Esse utente sai do sistema. Válido apenas para utentes que ainda não começaram a ser atendidos;
- `delesp X` – remove o especialista X, informando-o através do seu programa médico (que depois encerra). Válido apenas para especialistas que não estejam a atender nenhum utente no momento;
- `freq N` – passa a apresentar a ocupação das filas de espera de N em N segundos;
- `encerra` – termina o sistema, avisando os clientes, os médicos e o classificador.

Exemplo de lançamento do programa cliente

```
$ ./balcao
```

O utilizador administrador apenas interage com o programa balcão.

Classificação de especialidades

Não existe nenhum utilizador associado à funcionalidade de classificação de sintomas em especialidade e prioridade. Existe um programa “classificador”, que será fornecido pelos professores, que tem como missão receber texto pelo standard input e devolver texto *pelo standard output* com a informação da especialidade e prioridade.

O input é uma só linha por utente com os vários sintomas tal como escrito pelo utente. O output tem a forma de “*especialidade prioridade*” (uma palavra-espaco-um inteiro)

Exemplo de input recebido

```
estomago arde muito
```

Exemplo do output para o input recebido

```
estomatologia 2
```

No exemplo, a especialidade foi deduzida pela palavra “estomago” e a prioridade pela palavra “muito”.

O programa classificador já está feito e será disponibilizado pelos docentes. Não pode ser modificado. É importante saber que:

- O input e o output serão sempre texto em minúsculas, não acentuado e sem pontuação;
- Recebe o seu input pelo *stdin* e envia o seu output pelo *stdout*;
- O grau de prioridade vai de 1 a 3, sendo 3 o mais prioritário;
- As especialidades reconhecidas pelo sistema são: oftalmologia, neurologia, estomatologia, ortopedia e geral. Sintomas desconhecidos encaixam em “geral”.

O programa balcão usa o classificador internamente para classificar os sintomas em especialidades. Isto significa que o balcão terá que lançar o programa classificador e interagir com ele através da única forma que aquele suporta: *stdin* e *stdout*.

3. Funcionamento do sistema

O programa apresenta a seguinte funcionalidade, indicada por cada programa

Balcão

É responsável por gerir o funcionamento do sistema e ainda interagir com o administrador:

- Só aceita executar se ainda não estiver a correr nenhum outro balcão;
- Consegue interagir em simultâneo com clientes, médicos e classificador em simultâneo;
- Ao encerrar avisa todos os intervenientes para encerrarem também;
- Implementa um mecanismo que deteta que um médico terminou abruptamente sem avisar o sistema, removendo a informação dele e avisando o administrador. Este mecanismo baseia-se num “sinal de vida” que o médico tem de enviar ao balcão a cada 20 segundos;
- Executa internamente o classificador e interage com ele para obter as classificações;
- Recebe informações dos clientes: utente chegou com estes sintomas; utente desistiu;
- Recebe informação dos médicos: novo especialista para especialidade X, saída de especialista, fim de consulta;
- Informa o cliente que o utente vai iniciar consulta no especialista X, dando as informações necessárias (dependendo da implementação) para que o cliente possa interagir com o médico correspondente;
- Informa o médico que o especialista vai iniciar consulta ao utente X, dando as informações necessárias (dependendo da implementação) para que o médico possa interagir com o cliente correspondente;
- Deteta que médicos deixaram de dar sinal de vida e age em conformidade;
- Aceita trabalhar com até N clientes e M médicos em simultâneo. N e M são indicados através das variáveis de ambiente MAXCLIENTES e MAXMEDICOS, respetivamente;
- As filas de espera são de 5 utentes para cada especialidade. Este valor é fixo. São suportadas apenas as 5 especialidades acima referidas e este aspeto também é fixo;
- Caso algo não esteja de acordo com os pressupostos o programa deverá sair de forma ordeira.

Classificador

- Recebe texto acerca de um utente e dos seus sintomas e produz texto com o nome do utente, especialidade e prioridade;
- Este programa é fornecido pelos professores. Não pode ser modificado sem a autorização destes.

Cliente

Faz a interação entre o utilizador utente e o resto do sistema. O seu funcionamento é o seguinte:

- Só aceita executar se o balcão estiver em funcionamento;
- Recebe o nome do utente por parâmetros da linha de comando;
- Obtém os sintomas por pergunta direta ao utilizador;
- Interage com o balcão para dar a conhecer o utente e sintomas e recebe de volta a especialidade, prioridade e número de pessoas à sua frente na fila correspondente;
- Recebe do balcão a notificação que deve passar a interagir com um médico para passar à consulta propriamente dita;
- Durante a consulta permite ao utente interagir com o especialista num diálogo direto entre os programas cliente e médico, suportando o envio de respostas seguidas sem uma pergunta pelo meio, ou vice-versa.

Médico

Faz a interação entre o utilizador especialista e o resto do sistema. O seu funcionamento é o seguinte:

- Só aceita executar se o balcão estiver em funcionamento;
- Recebe o nome do especialista e a especialidade por parâmetros da linha de comando;
- Aguarda que o balcão lhe indique que deve iniciar uma consulta com um determinado utente, recebendo os dados necessários para interagir com o cliente que corresponde a esse utente;
- Durante a consulta permite ao especialista interagir com o utente num diálogo direto entre os programas cliente e médico, suportando o envio de perguntas seguidas sem uma resposta pelo meio, ou vice-versa;
- Implementa um mecanismo que indica ao balcão que está a funcionar. Esse mecanismo “dar sinal de vida” atua a cada 20 segundos.

4. Requisitos e restrições de implementação

Mecanismos de comunicação envolvidos e uso

- O mecanismo de comunicação entre clientes, balcão e médicos é o *named pipe*. O número de *named pipes* envolvidos, quem os cria, e a forma como são usados devem seguir os princípios gerais subjacentes a este mecanismo de comunicação tal como exemplificado nas aulas;
- O mecanismo de comunicação entre o balcão e o classificador é o único possível a um programa cuja única interação com o exterior é receber input por *stdin* e enviar output por *stdout*. O programa classificador será mantido em execução – apenas uma instância – durante toda a execução do balcão: a mesma instância servirá para classificar todos os utentes;
- Só podem ser usados os mecanismos de comunicação que foram abordados nas aulas;
- Em momento algum poderá usar ficheiros regulares como “mecanismo” de comunicação.

Desempenho de atividades em simultâneo

- Existem diversos pontos neste trabalho que obrigam os programas envolvidos a dar atenção a várias coisas em simultâneo. Existem diversas estratégias para abordar estas questões. Para cada uma das situações deve ser usada aquela que é a mais adequada à situação em questão.

Terminação dos programas

- Quer seja feita a pedido do utilizador, ou por não estarem reunidos os pressupostos para o programa (a variável de ambiente não existe, o balcão não está a acorrer, etc.), ou por situação de erro em *runtime*, o programa deve terminar, avisando tanto quanto possível os restantes de forma a manter a coerência do sistema.

Interface com o utilizador

- Não deve ser usada outra forma que não a indicada no enunciado. Nomeadamente, não podem ser usadas lógicas de menus em vez da lógica de palavras escritas (“comandos”);
- A informação que é recebida por parâmetros de linha de comandos ou por variáveis de ambiente tem mesmo que ser recebida dessa forma.

Implementação de estruturas de dados

- Devem ser simplificadas ao máximo dado que o foco da disciplina são os recursos e mecanismos do sistema e não a programação em si. Recorde-se que são indicadas simplificações muito importantes quanto à complexidade de dados: existem no máximo N clientes, M médicos, X especialidades, e Y lugares em cada fila. Deve aproveitar estas simplificações.

Persistência de dados

- Não existe qualquer persistência de informação entre execuções distintas do serviço, e o mesmo se aplica às execuções dos clientes e médicos. Não existe registo permanente de utentes nem de listas.
- Se quiser, pode tornar esta informação persistente (em ficheiro), mas tal é opcional, sem valorização associada, e fortemente desaconselhada.

Validações e simplificações indevidas

- Deve precaver a validade das operações efetuadas pelos programas, de acordo com a estratégia que for implementada. Esta questão decorre das características da solução que é escolhida e não de algo que seja dito no enunciado. Por exemplo, se em algum momento a estratégia seguida necessitar de mecanismos de sincronização, então essa necessidade deve ser garantida, não podendo ser invocada a lógica “o enunciado não fala sobre isso” ou “isso é uma situação que raramente acontecerá”.

Alternativas de implementação

- Existem alguns aspetos que admitem mais do que uma estratégia ou implementação (por exemplo, a questão do “sinal de vida”). Deve optar por uma estratégia lógica, que não desperdice recursos e saber justifica-la.

6. Considerações adicionais

- Toda a ação do sistema aqui descrito decorre na mesma máquina e na mesma conta de utilizador. Serão usadas consolas/terminais diferentes para cada cliente, médico e mais uma para o serviço.
- O balcão tem que conseguir lidar com várias tarefas em simultâneo. Os clientes e os médicos também. Deve pensar nisto.
- Podem existir diversas situações e potenciais situações de conflito ou erro que não são explicitamente mencionados no enunciado. Estas situações devem ser identificadas e os programas devem lidar com elas de uma forma controlada e estável. Um programa terminar abruptamente perante uma destas situações não é considerada uma forma adequada de lidar com o cenário.
- Caso se detetem incoerências neste enunciado, as mesmas serão corrigidas e publicadas. Poderá haver lugar a documentos adicionais no *moodle*, nomeadamente um *Frequently Asked Questions* (FAQ) com as perguntas mais frequentes e respetivas respostas.

7. Regras gerais do trabalho, METAS e DATAS IMPORTANTES

Aplicam-se as seguintes regras, descritas na primeira aula e na ficha da unidade curricular (FUC):

- O trabalho pode e deve ser realizado em grupos de dois alunos (grupos de três não são admitidos e qualquer pedido nesse sentido será sempre negado ou ignorado).
- Existe defesa obrigatória. A defesa será efetuada em moldes a definir e anunciados através dos canais habituais na altura em que tal for relevante. A defesa será presencial exceto se houver ordens em contrário pela presidência da escola.
- Existem a meta intermédia e a meta final, tal como descrito na FUC. As datas e requisitos das metas são indicados mais abaixo. Em todas as metas a entrega é feita via *moodle* através da submissão de um único arquivo zip³ cujo nome respeita o seguinte padrão⁴:

so_2122_tp_meta1_nome1_numero1_nome2_numero2.zip

(evidentemente, meta1, nome e número serão adaptados à meta, nomes e números dos elementos do grupo)

A não adesão ao formato de compressão indicado ou ao padrão do nome do ficheiro será penalizada, *podendo levar a que o trabalho nem sequer seja visto*.

Metas: requisitos e datas de entrega

Meta intermédia: **21 de Novembro**

Requisitos:

- Planear e definir as **estruturas de dados** responsáveis por gerir as definições de funcionamento no cliente, médico e serviço. Definir os vários *header files* com constantes simbólicas que registem os valores por omissão comuns e específicos do cliente e servidor bem como as estruturas de dados relevantes.
- Implementar a parte do balcão relativa à classificação da especialidade e respetiva prioridade. Para tal deve assumir que os dados dos sintomas são obtidos diretamente pelo utilizador administrador do programa balcão.
- Implementar no balcão a obtenção dos valores das variáveis de ambiente e concretizar as estruturas de dados do balcão.

Data de entrega: Domingo, 21 de Novembro, 2021 Sem possibilidade de entrega atrasada.

Meta 2: **16 de Janeiro**

Requisitos:

- Todos os requisitos expostos no enunciado.
- *makefile* que possua os *targets* de compilação "all" (compilação de todos os programas), "cliente" (compilação do programa cliente), "balcao" (compilação do programa balcão), "medico" (compilação do programa médico) e "clean" (eliminação de todos os ficheiros temporários de apoio à compilação e dos executáveis).

Data de entrega: Domingo, 16 de Janeiro, 2022. Sujeito a ajustes caso haja alterações no calendário escolar.

³ Leia-se "**zip**" - não é *arj*, *rar*, *tar*, ou outros. O uso de outro formato poderá ser **penalizado**. Há muitos utilitários da linha de comando UNIX para lidar com estes ficheiros (zip, gzip, etc.). Use um.

⁴ O não cumprimento do formato do nome causa atrasos na gestão dos trabalhos recebidos e será **penalizado**.

Na **meta intermédia** deverá ser entregue um documento (pdf, incluído no ficheiro zip) com duas páginas descrevendo os pormenores da implementação e principais opções tomadas.

Na **meta final** deverá ser entregue um relatório (pdf, também no zip). O relatório compreenderá o conteúdo que for relevante para justificar o trabalho feito, deverá ser da exclusiva autoria dos membros do grupo. Caso venha a ser divulgado, entretanto um guia de elaboração do relatório, então este deverá seguir as indicações dadas.

8. Avaliação do trabalho

Para a avaliação do trabalho serão tomados em conta os seguintes elementos:

- **Arquitetura do sistema** – Há aspetos relativos à interação dos vários processos que devem ser cuidadosamente planeados de forma a apresentar-se uma solução elegante, leve e simples. A arquitetura deve ser bem explicada no relatório para não existirem mal-entendidos.
- **Implementação** – Deve ser racional e não deve desperdiçar recursos do sistema. As soluções encontradas para cada problema do trabalho devem ser claras e bem documentadas no relatório. O estilo de programação deve seguir as boas práticas. O código deve ter comentários relevantes. Os recursos do sistema devem ser usados de acordo com a sua natureza.
- **Relatório** – O relatório deve descrever convenientemente o trabalho. Descrições meramente superficiais ou genéricas de pouco servirão. De forma geral, o relatório descreve a estratégia e os modelos seguidos, a estrutura da implementação e as opções tomadas. Podem vir a ser dadas indicações adicionais sobre a sua elaboração. O relatório deve ser entregue juntamente com o código no arquivo submetido via *moodle* da meta em questão.
- **Defesa** – Os trabalhos são sujeitos a defesa individual, durante a qual será verificada a respetiva autenticidade. Pode haver mais do que uma defesa caso subsistam dúvidas quanto à autoria dos trabalhos. A nota final do trabalho é diretamente proporcional à qualidade da defesa. Elementos do mesmo grupo podem ter notas diferentes consoante o desempenho e grau de participação individuais que demonstraram na defesa.
Apesar da defesa ser individual, ambos os elementos do grupo devem comparecer ao mesmo tempo. A falta à defesa implica automaticamente a perda da totalidade da nota do trabalho.
Este ano será usado um software que automatiza e auxilia a deteção de fraude dos trabalhos entregues. O regulamento da escola descreve o que acontece nas situações de fraude (ex., plágio).
- Os trabalhos que não funcionem serão fortemente penalizados independentemente da qualidade do código-fonte ou arquitetura apresentados. Trabalhos que nem sequer compilam terão uma nota extremamente baixa.
- A identificação dos elementos de grupo deve ser clara e inequívoca (tanto no arquivo como no relatório). Trabalhos anónimos não são corrigidos.
- Qualquer desvio quanto ao formato e forma nas submissões (exemplo, tipo de ficheiro) dará lugar a penalizações.

Importante: O trabalho deve ser realizado por ambos os elementos do grupo. Não são aceites divisões herméticas em que um elemento faz uma parte e apenas essa, nada sabendo do restante. Se num grupo tiver havido uma participação desigual, deverá informar o professor que lhe faz a defesa. Se não o fizer e for detectada essa desigualdade, ambos os alunos ficarão prejudicados em vez de apenas o que trabalhou menos.