Projeto 1 - Laboratório de Programação

Faculdade de Ciências da Universidade do Porto

Carolina Ferreira, up201905810

João Moura, up201904881

Tiago Eusébio, up201904872

Introdução

Este projeto tem como objetivo desenvolver um quadro de Kanban para a visualização de processos baseados em pipelines, na linguagem C utilizando listas ligadas em memória para manter os dados das tarefas e das pessoas, dado que as relações entre estas devem ser mantidas com novas listas ligadas de apontadores e para ser possível a passagem de tarefas entre as várias listas.

Estrutura de dados:

Para alcançar o propósito do projeto começamos por criar a livraria "listp1" onde implementamos um conjunto de funções, previamente declaradas no header "listp1.h" para a criação e manipulação das listas.

- "cria_lista": cria um header da lista e coloca um ponteiro de lista apontar para ele.

- "lista_vazia": indica se a lista em questão está vazia ou preenchida.

- "procura_nome_lista": procura na lista nós em que a tarefa tenha um nome específico e imprime todas as informações sobre esses nós.

- "procura_lista": procura na lista um nó com um valor de identificador específico e nó anterior a esse.

- "elimina_lista": procura o nó que se pretende eliminar com a função "procura_lista" e elimina-o.

- "prio_lista": tem a mesma funcionalidade da função "procura_lista" mas o valor procurado corresponde ao da prioridade.

- "insere_lista": insere um nó na lista e ordena-o pelo valor da prioridade com auxílio da função "prio_lista".

- "apoio_lista": insere um nó na lista e ordena-o pelo valor do identificador com auxílio da função "procura_lista".

- "pesquisa_nome_lista": pesquisa na lista um nó com um nome específico utilizando a função "procura_nome_lista".

- "pesquisa_lista": pesquina na lista um nó com um valor de identificador específico utilizando a função "procura_lista".

- "imprime_lista": imprime o conteúdo de todos os nós da lista.

- "name_lista": procura na lista o nó anterior ao nó com um nome da pessoa relativa à tarefa específico.

- "insere_listadoing": insere um nó na lista (utilizamos apenas para inserir na lista doing) de forma a ficarem ordenados pelo nome da pessoa responsável pela tarefa,

usando a função “name_lista”.

- ”done_lista”: percorre todos os nós da lista até chegar ao último no qual passa a ser o penultimo, dado que este passou apontar para um novo nó.
- ”insere_listadone”: Insere um novo nó no final da lista done com o auxílio da função ”done_lista”, dessa forma a lista fica ordenada pela data de conclusão (utilizamos só para inserir nós na lista done).

Para definir o tipo “node” e “List” usamos duas estruturas, esta definição tem características recursivas, pois uma lista é um apontador para uma estrutura e um dos membros dessa estrutura é uma lista. Cada nó da Lista tem dois parâmetros a “List next” referente ao proximo nó da lista e a variável “info” que é do tipo “TAREFA” que é também uma estrutura que contém como parâmetros as características de uma tarefa (identificador, prioridade, data_criacao, descricao, pessoa, prazo e data_conclusao), criamos também a estrutura DATA dado que que as variáveis data_criacao, prazo e data_conclusao são desse tipo.

Estrutura geral do programa:

Após declaradas todas as estruturas de dados criamos o ficheiro “main.c” onde implementamos todas as ações pedidas no enunciado com o auxílio da biblioteca criada explicada no ponto anterior. Para isso criamos um switch case com 9 casos cada um correspondente a uma ação.

- Case 1: Serve para inserir tarefas na lista “to_do” anteriormente declarada, onde cada informação é lida a partir do terminal e guardada em cada parâmetro do tipo TAREFA do nó, usando a função ”insere_lista” e ”apoio_lista” (posteriormente explicada). Para registar a data de criação utilizamos a struct tm para que fique como data o exato momento em que a tarefa é inserida na lista.
- Case 2: Inicia uma tarefa, ou seja, passa uma tarefa da lista “to_do” para a lista “doing”, com auxílio da lista ”insere_listadoing”, e dá-nos a opção de inserir uma

data de prazo, de seguida elimina a mesma da lista “to_do” usando a função ”elimina_lista” e aumenta a contagem do max, que é uma variável que conta o número de tarefas que vão sendo inseridas, para caso este chegue ao valor 5, o programa impossibilite a inserção de mais tarefas.

Dado que a nossa lista ordena pelo valor da prioridade e a primeira tarefa a ser realizada deve ser a que tem uma prioridade maior, o nosso programa passa o primeiro nó da lista to_do para a lista doing, visto que é nessa posição que a tarefa prioritária se encontra.

- Case 3: O objetivo deste case é alterar a pessoa responsável numa respetiva tarefa presente na lista Doing, para isso pedimos ao utilizador para inserir o valor do identificador da tarefa e o nome da “nova” pessoa, seguidamente utilizamos a função “pesquisa_lista” que retorna o nó com o valor indicado de identificador e de seguida copia para esse nó o novo nome utilizando a função ”strcpy”.

- Case 4: Este case finaliza a tarefa, o que significa que uma tarefa presente na lista doing é transferida para a lista done, dado um identificador procuramos o nó correspondente utilizando a função ”pesquisa_lista”, usando a função ”insere_listadone” inserimos a tarefa na lista done e com a função ”elimina_lista” removemos da função doing e a data de conclusão é atualizada para o exato momento em que a tarefa passa para a lista ”done” (tal como no case1).

- Case 5: Tem o propósito de refazer a tarefa e para isso recebemos o identificador da tarefa que o utilizador deseja voltar a fazer, procuramos o nó correspondente a essa tarefa na lista ”done” usando a função ”pesquisa_lista” e de seguida movemos essa tarefa para lista to_do, apagamos o nome da pessoa responsável e da data de conclusão e eliminamos da lista done, usando as funções “insere_lista” e “elimina_lista”.

- Case 6: Imprime as listas utilizando a função “imprime_lista”.

- Case 7: Caso seja escolhido este case o utilizador insere o nome de uma pessoa e é impresso as tarefas que ficaram na responsabilidade da mesma, utilizando a função “pesquisa_nome_lista”.

- Case 8: O objetivo é imprimir as tarefas pela ordem de criação. Para isso utilizamos uma função auxiliar ”apoio” onde fomos inserindo as tarefas à medida que foram

inseridas, e neste case imprimimos essa lista com a função "imprime_lista".

- Case 9: Termina o programa e guarda nos ficheiros toda a informação contida atualmente na lista. No ficheiro "to_do.txt" é guardado o conteúdo da lista to_do, no ficheiro "doing.txt" é armazenado o conteúdo da lista doing e a informação da lista done é reservada no ficheiro "done.txt".

Estrutura dos ficheiros:

Utilizamos os ficheiros com a finalidade de armazenar o conteúdo presente nas listas para que numa próxima utilização do programa possamos recuperar e ter acesso às tarefas anteriormente inseridas. Para isso criamos três ficheiros "to_do.txt", "doing.txt" e "done.txt", em cada um deles é guardada a informação relativa à lista associada. Sempre que o utilizador desejar sair do programa (Case 9) a informação presente em cada lista é impressa em cada um dos ficheiros correspondentes, dado que temos 3 ciclos, cada um percorre uma lista e com a função "fprintf" imprimimos todos os dados de cada nó da lista no ficheiro associado.

No ficheiro "to_do.txt" guardamos o identificador, a data de criação, a prioridade e a descrição de cada nó presente na respetiva lista.

No ficheiro "doing.txt" fica armazenada o identificador, a data de criação, a prioridade, a descrição, a pessoa responsável e opcionalmente o prazo de cada nó presente na respetiva lista.

Por fim, no ficheiro "done.txt" reservamos o identificador, a data de criação, a prioridade, a descrição, a pessoa responsável, o prazo (opcional) e a data de conclusão.

Breve explicação de como o programa se executa:

Ao executar o programa é mostrado ao utilizador um menu com várias opções cada uma corresponde a um dos cases explicados anteriormente e têm um número associado. O utilizador deve escolher uma das ações e inserir o número correspondente para que a mesma seja realizada.

Caso ainda não haja tarefas em nenhuma das listas o utilizador deve começar por inserir (opção 1). Ao inserir a tarefa será pedido a prioridade da mesma e a descrição,

pode inserir mais do que uma tarefa seguida.

Após existirem tarefas na lista to_do podemos passar para a lista doing selecionando a opção 2 onde será pedido para indicar o nome da pessoa que ficará responsável pela tarefa e de seguida tem a opção de inserir um prazo para a mesma, caso a resposta seja afirmativa deve inserir a data.

Se desejar alterar a pessoa responsável por uma tarefa presente na lista doing deve inserir o número 3, seguidamente deve indicar o identificador e o nome da pessoa.

Para completar a tarefa, ou seja, passa-la da lista doing para a lista done, utilize a opção 4 onde deve escrever o valor do identificador da tarefa que deseja finalizar.

Caso tenha finalizado uma tarefa e deseja refazê-la selecione a opção 5, onde irá pedir o identificador da tarefa e de seguida passa-la para a lista to_do.

Se escolher a opção 6 todas as tarefas e as informações sobre as mesmas de cada lista são impressas.

A opção 7 imprime todas as tarefas que foram realizadas por uma pessoa especifica, por isso terá primeiramente de indicar um nome.

Para observar todas as tarefas ordenadas pela data de criação deve escolher a opção 8.

Por fim, a opção 9 termina o programa.

Conclusão

No decorrer da realização do projeto tivemos várias dificuldades e nem todas elas foram completamente superadas, por consequência o nosso trabalho foi reescrito enumeras vezes até conseguirmos um resultado final mais positivo.

Devido a estas adversidades sabemos que o nosso programa não está 100% eficaz e que muitos aspetos do mesmo deveriam estar sem imperfeições, contudo todos os elementos do grupo trabalharam para tentar atingir a máxima eficácia do programa.