



Relatório Fase 1

Estruturas de Dados Avançadas

Aluno

João Carlos da Costa Apresentação

Docentes

Luís Ferreira

João Silva

Licenciatura em Engenharia de Sistemas Informáticos

Barcelos, Abril, 2022

Índice

Introdução.....	3
Propósitos e Objetivos	4
Estruturas de Dados	5
Lista das Máquinas.....	5
Lista MacOp.....	5
Lista das Operações.....	5
Testes realizados	6
Conclusão	11
Bibliografia	12

Introdução

Relatório da 1ª fase do projeto realizado em linguagem C utilizando ferramentas instruídas ao longo do semestre (VS Code, Doxygen, GitHub, ...) na unidade curricular Estruturas de Dados Avançadas integrada no 2 semestre do 1º ano da licenciatura Engenharia de Sistemas Informáticos.

Propósitos e Objetivos

A realização deste projeto tem como propósito a demonstração dos conhecimentos adquiridos dentro e fora de aula, além do empenho para a sua concretização.

Como objetivos, é pretendido o desenvolvimento de uma solução para o problema de escalonamento, com o envolvimento de operações e várias máquinas de forma a minimizar o tempo de produção de um produto.

Para esta 1ª fase é pretendido o seguinte conjunto de funcionalidades:

- Definição de uma estrutura de dados dinâmica para a representação de um job com um conjunto finito de n operações;
- Armazenamento/leitura de ficheiro de texto com representação de um job;
- Inserção de uma nova operação;
- Remoção de uma determinada operação;
- Alteração de uma determinada operação;
- Determinação da quantidade mínima de unidades de tempo necessárias para completar o job e listagem das respetivas operações;
- Determinação da quantidade máxima de unidades de tempo necessárias para completar o job e listagem das respetivas operações;
- Determinação da quantidade média de unidades de tempo necessárias para completar uma operação, considerando todas as alternativas possíveis;

Para a realização do mesmo espera-se o recurso a manipulação de ficheiros e apontadores, e estruturas dinâmicas, além do abordado de makefile e documentação gerada por Doxygen.

Estruturas de Dados

Para a resolução desta fase 1 foi utilizado o recurso a 3 listas duplamente ligadas:

Lista das Máquinas

Lista que contém todas as máquinas do projeto e que como variáveis o seu “id_mac” como identificação e o “time” como representação da quantidade de unidades de tempo para a realização de uma tarefa. Além disso ainda tem o apontador para o próximo e para o anterior nodo da lista. Este será armazenado em um ficheiro para leitura e escrita.

```
typedef struct _listMachine
{
    int time, id_mac;
    struct _listMachine *next, *previous;
} Machine;
```

Lista MacOp

Lista que serve de ligação entre a lista das máquinas e a lista das operações. Esta lista contém o “id_op” identificação de uma operação e o “id_mac” identificação de uma máquina. Um id de uma operação será associado a uma máquina. Além disso ainda tem o apontador para o próximo e para o anterior nodo da lista. Este será armazenado em um ficheiro para leitura e escrita.

```
typedef struct _listMacOp
{
    int id_op, id_mac;
    struct _listMacOp *next, *previous;
} MacOp;
```

Lista das Operações

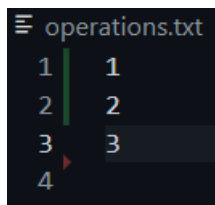
Lista que contém todas as operações do projeto e que como variável o seu “id_op” como identificação. Além disso ainda tem o apontador para o próximo e para o anterior nodo da lista. Este será armazenado em um ficheiro para leitura e escrita.

```
typedef struct _listOperation
{
    int id_op;
    struct _listOperation *next, *previous;
} Operation;
```

Testes realizados

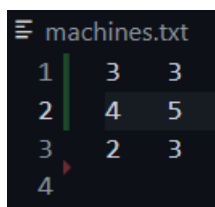
Para a realização destes testes os files encontram-se povoados da seguinte forma:

- Ficheiro das operações que contém os ids das operações em cada linha:



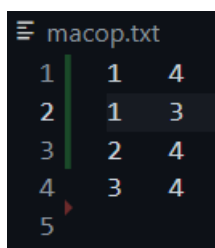
```
operations.txt
1 1
2 2
3 3
4
```

- Ficheiro das máquinas que contém os ids das máquinas na 1ª coluna e respetivo tempo na 2ª coluna:



```
machines.txt
1 3 3
2 4 5
3 2 3
4
```

- Ficheiro de ligação das máquinas com as operações onde a 1ª coluna contém os ids das operações e a 2ª coluna os ids das máquinas:



```
macop.txt
1 1 4
2 1 3
3 2 4
4 3 4
5
```

Seguem-se testes realizados ao código de modo a apresentar a resolução das funcionalidades pretendidas:

Inserção de uma operação

Ao inserir uma operação é escolhido qual máquina/s associa-se e após isso é visível tanto de uma forma visível através de uma função como através dos próprios ficheiros que os conseguem armazenar corretamente.

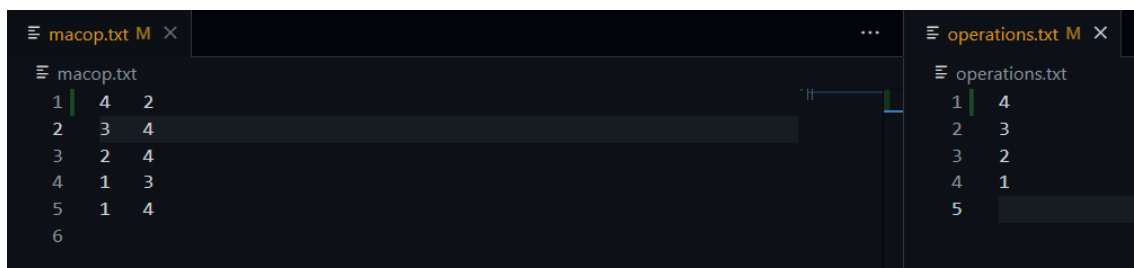
```
-----
      Lista das Maquinas
-----
ID: 2 / TIME: 3
ID: 4 / TIME: 5
ID: 3 / TIME: 3
-----

(Insira numero negativo para terminar nao associar mais maquinas)
Selecione o id da maquina a associar a operacao: 2
```

```
-----
      Lista das Operacoes
-----
ID: 4
ID: 3
ID: 2
ID: 1
-----

      Lista Intermedia
      (Maquinas e Operacoes)
-----
ID Operacao: 4 / ID Maquina: 2
ID Operacao: 3 / ID Maquina: 4
ID Operacao: 2 / ID Maquina: 4
ID Operacao: 1 / ID Maquina: 3
ID Operacao: 1 / ID Maquina: 4
-----

      Lista das Maquinas
-----
ID: 2 / TIME: 3
ID: 4 / TIME: 5
ID: 3 / TIME: 3
-----
Press any key to continue . . .
```



Alterar operação

É selecionado o id 4 criado anteriormente para alterar a/s sua/s máquina/s associada/s e é trocada a máquina 2 pela 4 e é visível a alteração através da função e através dos ficheiros.

```
-----
Lista das Operacoes
-----
ID: 1
ID: 2
ID: 3
ID: 4
-----

Selecione o id da operacao a alterar: 4
```

```
-----
Lista das Maquinas
-----
ID: 3 / TIME: 3
ID: 4 / TIME: 5
ID: 2 / TIME: 3
-----

(Insira numero negativo para terminar nao associar mais maquinas)
Selecione o id da maquina a associar a operacao: 4
```

```
-----
Lista das Operacoes
-----
ID: 1
ID: 2
ID: 3
ID: 4
-----

Lista Intermedia
(Maquinas e Operacoes)
-----
ID Operacao: 4 / ID Maquina: 4
ID Operacao: 1 / ID Maquina: 4
ID Operacao: 1 / ID Maquina: 3
ID Operacao: 2 / ID Maquina: 4
ID Operacao: 3 / ID Maquina: 4
-----

Lista das Maquinas
-----
ID: 3 / TIME: 3
ID: 4 / TIME: 5
ID: 2 / TIME: 3
-----
Press any key to continue . . .
```

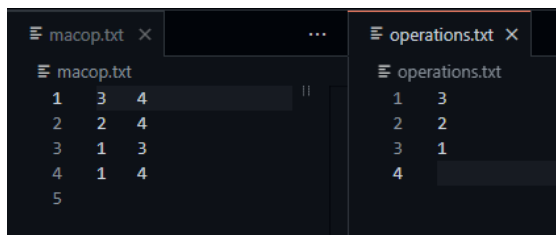
```
≡ macop.txt
1 4 4
2 1 4
3 1 3
4 2 4
5 3 4
6
```


Remoção de uma operação

Agora foi removido a operação 4 da lista e ela deixou de existir nas listas.

```
-----
Lista das Operacoes
-----
ID: 4
ID: 3
ID: 2
ID: 1
-----

Selecione o id da operacao a remover: 4
```



	1	2	3	4
1	3	4		
2	2	4		
3	1	3		
4	1	4		
5				

	1	2
1	3	
2	2	
3	1	
4		

Tempo mínimo da realização do job

A funcionalidade que prevê o tempo mínimo para realizar um job testando todas as possibilidades. Para cada id foi escolher a máquina com menor tempo de trabalho e obteve se o resultado seguinte.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

-----
Lista das Operacoes
-----
ID: 1
ID: 2
ID: 3
-----

Tempo minimo executado pelo job: 13
Press any key to continue . . .
```

Tempo máximo da realização do job

A funcionalidade que prevê o tempo máximo para realizar um job testando todas as possibilidades. Para cada id foi escolher a máquina com maior tempo de trabalho e obteve se o resultado seguinte.

```
-----
Lista das Operacoes
-----
ID: 4
ID: 3
ID: 2
ID: 1
-----

Tempo maximo executado pelo job: 20
Press any key to continue . . .
```

Tempo médio de uma operação

Para testar a funcionalidade final foi inserido o id 1 devido a este ter mais que uma máquina associada e desta forma obteve-se a média do tempo dessa mesma operação.

```
-----  
      Lista das Operacoes  
-----  
ID: 4  
ID: 3  
ID: 2  
ID: 1  
-----  
  
Selecione o id da operacao: 1  
  
Tempo medio executado pela operacao 1: 4.00  
Press any key to continue . . .
```

Conclusão

Com esta 1ª fase foi possível sedimentar todos os elementos aprendidos interna e externamente, assim como o desenvolvimento de código de forma independente e de forma a cumprir todos os objetivos pretendidos. Para a seguinte fase espera-se uma melhoria ao nível da qualidade de código além da implementação das restantes funcionalidades.

Bibliografia

(Luís Ferreira, IV - Structs e Apontadores, 2022)

(Luís Ferreira, VI - Structs e Apontadores, 2022)

(James Aspnes, Notes on Data Structures and Programming Techniques, 2016)