

Flexible Job Shop Problem

Trabalho Prático

Instituto Politécnico do Cavado e do Ave

Licenciatura em Engenharia em Desenvolvimento de Jogos Digitais

Estrutura de Dados Avançados

Ademar Valente nº23155

Docente: Luís Ferreira

Março 2022

Índice

Introdução – pág.3

Propósitos e Objetivos – pág.5

Estruturas de Dados – pág.6

Testes realizados – pág.9

Conclusão – pág.12

Bibliografia – pág.13

Introdução

O presente relatório procura descrever da forma mais sucinta o processo de realização do trabalho prático lançado pela docência no âmbito de realização da disciplina de Estrutura de Dados Avançados; inserida no curso de Licenciatura de Engenharia em Desenvolvimento dos Jogos Digitais.

O desafio é o de implementar uma solução digital que permita gerar uma proposta de escalonamento para a realização de um determinado produto, que para ser criado utiliza um conjunto específico de máquinas e uma consequente execução com um determinado tempo de execução. Este processo visa minimizar o tempo perdido entre a execução dos diferentes processos e intervenientes nele inseridos. O enunciado do projeto identifica sob a forma de quadro o organigrama de organização da empresa na construção dos seus produtos.

Process Plan	Operation						
	0 1	0 2	0 3	0 4	0 5	0 6	0 7
pr _{1,2}	(1,3) [4,5]	(2,4) [4,5]	(3,5) [5,6]	(4,5,6,7,8) [5,5,4,5,9]			
pr _{2,2}	(1,3,5) [1,5,7]	(4,8) [5,4]	(4,6) [1,6]	(4,7,8) [4,4,7]	(4,6) [1,2]	(1,6,8) [5,6,4]	(4) [4]
pr _{3,3}	(2,3,8) [7,6,8]	(4,8) [7,7]	(3,5,7) [7,8,7]	(4,6) [7,8]	(1,2) [1,4]		
pr _{4,2}	(1,3,5) [4,3,7]	(2,8) [4,4]	(3,4,6,7) [4,5,6,7]	(5,6,8) [3,5,5]			
pr _{5,1}	(1) [3]	(2,4) [4,5]	(3,8) [4,4]	(5,6,8) [3,3,3]	(4,6) [5,4]		
pr _{6,3}	(1,2,3) [3,5,6]	(4,5) [7,8]	(3,6) [9,8]				
pr _{7,2}	(3,5,6) [4,5,4]	(4,7,8) [4,6,4]	(1,3,4,5) [3,3,4,5]	(4,6,8) [4,6,5]	(1,3) [3,3]		
pr _{8,1}	(1,2,6) [3,4,4]	(4,5,8) [6,5,4]	(3,7) [4,5]	(4,6) [4,6]	(7,8) [1,2]		

Fig.1 – Tabela/Problema base do Projeto de trabalho.

O trabalho enunciado divide-se em duas partes, com requisitos diferentes, que procuram solucionar problemas de forma evolutiva, ordenada e dinâmica em termos de aprendizagem. O presente relatório tem o propósito de expor as soluções para os problemas propostos na parte 1 do trabalho prático, que são:

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

Propósitos e Objetivos

A realização do presente trabalho serve o propósito principal de continuidade de aprendizagem na **linguagem de programação C** iniciada na disciplina de Programação Imperativa do semestre passado.

Dadas as dificuldades sentidas anteriormente, os objetivos são os de tirar um aproveitamento máximo na aprendizagem, por forma a ir minimizando as lacunas que ainda são bastante consideráveis à data presente; bem como ir construindo com solidez uma base que me permita gerar soluções para os problemas que me vão sendo propostos e dominar de forma crescente a linguagem específica da elaboração dos trabalhos (neste caso, a linguagem C).

Para além disso, é objetivo adicional ao trabalho proposto o de adquirir conhecimentos relativos a outras ferramentas de trabalho propostas pela docência para a realização deste trabalho. Passa pelos objetivos os de avaliar, armazenar conhecimentos e adquirir competências no manuseamento do **VisualStudio**, **Git**, **GitHub** e **Doxigen**, intervenientes no processo de realização deste projeto.

Posto isto, e de forma sucinta, o objetivo principal passa por, utilizando as competências desenvolvidas e adquiridas acima referenciadas, atravessar com sucesso esta fase 1 por forma a ter oportunidade de realizar a fase 2 e concluir o processo de aprendizagem a que a disciplina propõe com aproveitamento positivo e satisfatório a todas as partes.

Estruturas de Dados

A estrutura de dados do presente projeto assenta num conjunto de estruturas dinâmicas (**structs**). As estruturas adotadas estão interligadas entre si por forma a demonstrarem cada trabalho especificamente, apontando cada uma delas para a seguinte.

De forma resumida, o trabalho possui seis listas ligadas entre si.

Primeiramente a estrutura Máquina (**Machine**), caracteriza por um identificador e por um intervalo de tempo necessário para concluir a sua parte na operação. Nesta estrutura (tal como em todas as outras) a Máquina aponta para uma próxima que possa existir (**nextMachine**).

```
typedef struct Machine
{
    int id;
    int time;
    struct Machine* nextMachine;
}Machine;

extern Machine* machineHeader;
```

De forma consecutiva foi criada uma lista de máquinas (**MachineList**), visto que o enunciado afirma que numa determinada operação podem operar várias máquinas de forma consecutiva e ordenada; lista essa que aponta para uma próxima que possa existir (**nextMachines**).

```
typedef struct MachineList
{
    struct Machine machine;
    struct MachineList* nextMachines;
}MachineList;
```

Como foi dito em cima, uma operação (**Operation**) é uma parte do processo de realização do produto. Possui uma identificação e é realizada por um conjunto de máquinas. Como é uma lista ligada tem sempre um apontador da própria estrutura operação para uma próxima estrutura operação que possa existir (**nextOperation**).

```
typedef struct Operation
{
    int id;
    struct MachineList* machines;
    struct Operation* nextOperation;
}Operation;
```

Tal como com as máquinas, um trabalho encerra em si um conjunto de operações. Assim, ficaria mais fácil a gestão das operações dentro de uma lista ligada de listas de operações (**OperationList**). Como é uma lista ligada tem sempre um apontador da própria de estrutura de operações para uma próxima estrutura de listas de operações que possa existir (**nextOperations**).

```
typedef struct OperationList
{
    struct Operation operation;
    struct OperationList* nextOperations;
}OperationList;
```

Finalizando, chegamos ao trabalho propriamente dito (**Job**). Assim, possui um identificador, aglomera em si o conjunto de dados enunciados acima (conjunto de operações, cada uma delas composta por um conjunto de máquinas), já aglomerados pela lista de operações que o constituem. Finalizado aponta para um próximo que possa existir (**nextJob**).

```
typedef struct Job
{
    int id;
    struct OperationList* operations;
    struct Job* nextJob;
}Job;
```


Foi também criada uma estrutura de auxílio para inserir em ficheiro binário (***JobFile***).

```
typedef struct JobFile  
{  
    int id;  
}JobFile;
```

Testes realizados

Após vários testes, e continuando com a premissa de simplificar a realização do trabalho nesta fase inicial e de crescimento de conhecimentos, foram realizados testes com um job, uma operation e duas machines dentro dela.

As tabelas abaixo apresentados mostram os dados iniciados por código (**criar** e **inserir**), bem como uma alteração efectuada numa unidade de tempo de uma das máquinas (**alterar**).

Em seguida apresenta-se as variantes da amostra apresentada (tempo **mínimo**, **máximo** e **médio**), que faziam parte do pressupostos da execução do trabalho.

Finalmente, é apresentado na imagem a actualização de um job alterado, tendo sido retirada a sua única operação (**eliminar**).

```
-----Flexible Job Shop Problem-----  
  
Jobs:  
ID: 1  
Operações:  
ID: 1  
Máquinas:  
ID: 1 - Tempo: 4  
ID: 3 - Tempo: 5  
  
-----Dados carregados com sucesso!-----
```

Fig.2 – Dados criados e carregados no main.c.

```
-----Flexible Job Shop Problem-----  
  
Jobs:  
ID: 1  
Operações:  
ID: 1  
Máquinas:  
ID: 1 - Tempo: 4  
ID: 3 - Tempo: 15  
  
-----Dados alterados com sucesso!-----
```

Fig.3 – Dados alterados na duração de tempo de uma das máquinas.

```
-----  
  
Tempo Mínimo: 4  
  
-----  
  
Tempo Máximo: 15  
  
-----  
  
Tempo Médio: 9,50  
  
-----
```

Fig.4 – Dados estatísticos dos trabalhos realizados pelas máquinas.

```
-----Flexible Job Shop Problem-----  
  
Jobs:  
ID: 1  
Operações:  
  
-----Operação removida com sucesso!-----
```

Fig.5 – Dados alterados na remoção da operação.

Com isto foi dado como concluído a realização desta primeira parte do trabalho. Através do apresentado todos os pressupostos do enunciado foram realizados com sucesso, tendo a documentação dos dados sido um entrave. Os dados ficam documentados em binário, e acrescentados sempre que se implementam. Esta foi a maior dificuldade encontrada na resolução do problema, e não completamente resolvida, visto que a consulta externa ao programa não ser possível de realizar.

Conclusão

Após a resolução da primeira parte deste trabalho, é possível tirar bastantes ilações sobre o que foi desenvolvido, positivas e negativas. Como ponto introdutório a esta conclusão é importante referir que o trabalho realizado pela docência nas aulas é ponto fundamental para a realização deste trabalho, por ter permitido que a compreensão e elaboração deste projeto tivesse menos entraves do que aqueles que se tinham no ponto inicial da disciplina.

Do ponto de vista pessoal, vinha do semestre anterior com muitas dificuldades em compreender e resolver problemas de programação. Essas dificuldades, apesar de ainda existirem, foram sendo diminuídas ao longo do processo de resolução do trabalho. Para isto contribuíram vários fatores. Para além do trabalho desenvolvido em aula (cuja importância foi referida acima), a própria estrutura de elaboração do trabalho contribuiu bastante para a evolução que senti a realizá-lo.

Por se tratar de um trabalho em que o Professor deu total liberdade para a realização do mesmo em grupo, tive a oportunidade de obter ajuda de várias pessoas e de receber auxílio de vários métodos de resolução do trabalho. Com isto associado ao pacote de conhecimento recebido nos exercícios práticos das aulas, tive a oportunidade de elaborar um código que, apesar de não estar perfeito, identifica o ponto de aprendizagem em que me encontro. Tive auxílio de várias colegas de turma para fazer o que fiz, colegas que estão em diferentes pontos de evolução na sua própria aprendizagem, e que me deram diferentes pontos de vista para o que eu escolhi que me identificasse como executor deste trabalho.

Para além disso, foi importante também perceber o grau de importância que tem uma documentação de código bem estruturada; tendo sido essa uma das minhas principais preocupações para a realização deste desafio. Espero com tudo o que foi realizado até à data avançar para o a segunda parte do trabalho, e ter realizado uma base sólida que para essa mesma parte não seja muito alterada daqui para a frente.

Bibliografia

- Documentação fornecida pela docência via *Moodle*;
- <https://www.doxygen.nl/download.html>
- <https://github.com>
- <https://visualstudio.microsoft.com>
- [Code analysis for C/C++ overview | Microsoft Docs](#)
- PEREIRA, Alexandre. **C e Algoritmos**. 2ª Edição: Edições Sílabo, 2013.