



Relatório

Estruturas de Dados Avançadas

Trabalho Prático – Flexible Job Shop Problem

Henrique Azevedo – 23488

Docente – Luís Ferreira

Licenciatura em Engenharia em Desenvolvimento de Jogos Digitais

Junho 2022

Conteúdo

Introdução	3
Contextualização	3
Propósitos e Objetivos	4
Estruturas de Dados	5
Implementação	5
Nodes das Listas e Dados das Structs.....	5
Structs de Armazenamento para Ficheiros	6
Testes Realizados	6
Conclusão	7
Bibliografia	8

Introdução

Contextualização

Neste relatório é pretendido apresentar os propósitos, objetivos, as estruturas de dados e os testes realizados em relação à segunda fase do trabalho prático durante a unidade curricular de Estruturas de Dados Avançadas no âmbito do curso da Licenciatura em Engenharia de Desenvolvimento de Jogos Digitais.

O trabalho prático consiste no desenvolvimento de uma solução digital para o problema de escalonamento denominado “Flexible Job Shop Problem” (FJSSP).

A solução consiste em um Software responsável pela produção de um determinado produto, envolvendo vários “Jobs”, que são constituídos por um conjunto de operações que fazem uso de uma quantidade limitada de máquinas. O software também é responsável por minimizar o tempo necessário de produção.

Abaixo na tabela, está demonstrado o process plan para este problema, a quantidade e os valores necessários para a construção do projeto.

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]		

Tabela 1. Process plan para um problema de escalonamento com dimensão 8x7 e 8 máquinas

Propósitos e Objetivos

O propósito e objetivo deste trabalho prático consiste em sedimentar os conhecimentos relativos a definição e manipulação de estruturas de dados dinâmicas na linguagem de programação C. Neste trabalho é possível reforçar e aplicar os conhecimentos adquiridos ao longo do semestre.

Além da aplicação de conhecimento a nível de linguagem, é possível também desenvolver boas práticas de programação. A nível técnico um código limpo, o incentivo ao uso de comentários e o auxílio do doxygen têm o objetivo de facilitar a compreensão do código para qualquer utilizador.

Estruturas de Dados

No desenvolvimento da solução do problema apresentado, foram encontrados vários desafios, relacionados com a organização e gestão de múltiplas variáveis. Estas variáveis teriam de ser armazenadas e utilizadas de forma bastante prudente, para evitar perda e mistura de informação e ter fácil acesso às mesmas.

Todos os dados, introduzidos ou carregados, são guardados em ficheiros, em qualquer altura da execução da solução. Os dados são guardados em ficheiros de texto. Deste modo, o utilizador pode facilmente manipular os dados da forma desejada, através dos ficheiros de texto.

Os dados uma vez carregados são devidamente organizados nas estruturas dinâmicas às quais os mesmos pertencem. Existem vários tipos de dados, e várias estruturas que os armazenam devidamente.

As Listas Ligadas foram as estruturas dinâmicas utilizadas na primeira fase do projeto, nesta segunda fase, com o aumento dos dados, foi escolhido implementar uma Tabela Hash, de modo a conseguir ter os Jobs mais organizados e um process plan mais claro.

Implementação

Nodes das Listas e Dados das Structs

Caracterizam as estruturas responsáveis pela criação das listas ligadas, que contêm todos os dados relacionados com os jobs, operações e máquinas.

```
typedef struct Job {  
    int id; //id do job  
    struct Operation* operation; //Todas as operações do Job  
    struct Job* next; //Ligação ao proximo job  
}Job;
```

Figura 2 – Representação de um Trabalho

Structs de Armazenamento para Ficheiros

São estruturas muito semelhantes aos nodes das listas, com a diferença de que estas apenas armazenam informação, sem apontadores para os próximos elementos, sendo assim utilizados como meio de armazenamento.

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

Figura 3 – Representação de um “JobFile”

Structs de Jobs para Armazenar em HashTable

Estruturas também muito parecidas com os nodes de cada um dos Jobs das Listas, que apenas exclui o apontador para o próximo job, evitando assim armazenar dados repetidamente na HashTable.

```
typedef struct JobIn {  
    int id; //id do job  
    struct Operation* operation; //Todas as operações do Job  
}JobI;
```

Struct da HashTable

Testes Realizados

Todas as funções do projeto foram, devidamente, testadas e corrigidas, no caso de encontrar algum problema. A inserção de dados pode ser efetuada de variadas maneiras e tudo de forma bastante simples e para qualquer utilizador.

Os testes foram sendo realizados com o decorrer do projeto e sempre adicionando algo mais completo e, de certa forma, mais correta e fácil de entender.

Conclusão

A proposta desta segunda fase do projeto foi cumprida quase na totalidade, com exceção da funcionalidade 8 que não foi implementada. No entanto, no restante processo a maioria das dificuldades foram superadas e os objetivos devidamente atingidos. Nesta segunda fase, a organização e clareza dos dados foi o objetivo fulcral, sem perder nenhum conteúdo e ter tudo bem estruturado.

A matéria lecionada ao longo do semestre ficou muito bem consolidada com este projeto e enriqueceram muito as competências de programação, visto que este foi um trabalho exigente.

Bibliografia

- Conteúdos e Material disponibilizado no Moodle.
- <https://www.geeksforgeeks.org/linked-list-set-1-introduction/>
- <https://www.youtube.com/watch?v=2Ti5yvumFTU>
- <https://www.youtube.com/watch?v=5pYYy3Z0m9I>
- https://www.youtube.com/watch?v=eSkIGt_70-U
-