1 de Junho de 2022

iURI rodrigues (21159)

Trabalho Prático - Fase 2

Estruturas de Dados Avançadas

Conteúdo

[**Introdução** 2](#_Toc105001707)

[**Propósitos e Objetivos** 2](#_Toc105001708)

[**Estruturas de dados** 3](#_Toc105001709)

[**Testes Realizados** 11](#_Toc105001710)

[**Conclusão** 16](#_Toc105001711)

[**Bibliografia** 16](#_Toc105001712)

**Conclusão…………………………………………………………………………………………………………………………….9**

# **Introdução**

Este é o relatório da primeira fase do trabalho prático de Estuturas de dados avançadas.

O código pode se consultado tambem em https://github.com/syke007/EDA--TP2

# **Propósitos e Objetivos**

No âmbito da disciplina de Estruturas de Dados Avançadas foi proposta a realização de um trabalho desenvolvido em linguagem c, de forma a ser desenvolvida uma solução digital para o problema de escalonamento denominado Flexible Job Shop Problem (FJSSP). A solução a implementar deverá permitir gerar uma proposta de escalonamento para a produção de um produto envolvendo várias operações e a utilização de várias máquinas, minimizando o tempo as unidades de tempo necessário na sua produção (makespan)

# **Estruturas de dados**

Neste trabalho foram usados três ficheiros, header.c, functions.c e main.c

No header foram definidas duas estruturas, job e machine. A estrutura machine guarda as maquinas, a operaçao e o tempo tanto como os apontadores \*next e prev\*, a estrutura job guarda o Id do job, apontadores \*next e \*prev mas também apontador \*first e \*last da machine

No ficheiro functions.c foram implementadas todas as funções necessárias

No main.c foi implementado um menu de interação com o utilizador

LoadData

De maneira a ler os dados que vão ser tratados dum ficheiro txt, foi implementada a função LoadData, que abre o ficheiro, lê linha a linha com a formatação dada(job id, operação id, maquina, tempo) de seguida faz a inserção do job na lista(insertion\_job) e depois vai inserir os dados da estrutura machine na lista de jobs , retornando a lista de jobs

Uma imagem com texto, captura de ecrã, ecrã, preto

Descrição gerada automaticamente

WriteData

Para poder salvar os Dados foi desenvolvida esta função que com a permissão de escrever, abre o ficheiro, percorre a lista de jobs, dentro disso vai buscar na lista de maquinas a primeira maquina e percorre essa lista também para no fim escrever no ficheiro no formato : Job ID, operation ID, machine ID, time

Uma imagem com texto

Descrição gerada automaticamente

print

Esta função tal como a anterior percorre as duas listas e no fim mostra os dados q estão a ser lidos

Uma imagem com texto

Descrição gerada automaticamente

Insertion\_job

Para fazer a inserção dos dados optei por inserção à cabeça, alocamos a memoria e de seguida fazemos a inserção dos novos dados

Uma imagem com texto, ecrã, captura de ecrã, fechar

Descrição gerada automaticamente

Insertion\_machine

Para a inserçao das machines também optei por uma inserção á cabeça em que fazemos o mesmo procedimento da anterior, se existir um valor anterior, o next vai apontar para ele

Uma imagem com texto

Descrição gerada automaticamente

Insertion

Esta ultima função de inserção insere a lista de machines dentro da lista de jobs.

Para isso é percorrida a lista até encontrar o job desejado, depois de encontrado, se for o primeiro valor é inserido em first e last, senão apenas alteramos o last

Uma imagem com texto, monitor, captura de ecrã, ecrã

Descrição gerada automaticamente

RemoveJob

De modo a poder remover um job foi implementada esta função que a partir do ID do job que vai ser dado pelo utilizador faz a remoção do mesmo.

Para isso percorremos a lista de jobs até o encontrar, de seguida percorremos a lista de machines, dentro desta e se houver um valor a seguir a primeira posição vai ser igual a esse valor e o anterior vai ser nulo, se não apenas definimos a primeiro e o último como nulo e no fim fazemos um free de modo a remover as maquinas

Uma imagem com texto

Descrição gerada automaticamente

RemoveOperation

Para remover uma operação o processo é o mesmo da função anterior, mas neste caso procuramos pela operation id e ela é removida

Uma imagem com texto

Descrição gerada automaticamente

Verificaçoes

Para fazer verificações ao utilizar as funções implementadas anteriormente, foram desenvolvidas estas três funções que verificam se uma operação ou maquina existem

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Updates

De maneira a poder alterar valores de id de operação, tempo e Id da maquina foram implementadas estas funções onde é percorrida a lista de jobs ate encontrar o Id pretendido, depois é percorrida a lista de machines e é alterado o campo desejado (machine, tempo ou operação) para o que o utilizador inserir

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Melhor tempo

Para fazer a função que calcula o melhor tempo por job, foi definido um array tridimensional, onde é guardado id do job, id da operação, id maquina e tempo

No primeiro for os valores são todos declarados a 0

De seguida percorre as duas listas(jobs e machines), dentro do array tridimensional verifica se aquele campo já tem um tempo, sem não tiver é atribuído o valor, se já tiver, se o tempo do melhor altera se não, fica igual

Uma imagem com texto

Descrição gerada automaticamente

# **Testes Realizados**

Menu Inicial

Uma imagem com texto

Descrição gerada automaticamente

Ao escolher a opção 2 são mostrados os dados lidos do ficheiro

Uma imagem com texto

Descrição gerada automaticamenteUma imagem com texto

Descrição gerada automaticamente

Na opção 1 os dados são escritos num ficheiro txt

Uma imagem com texto

Descrição gerada automaticamente

Na opção três é feita a adição de uma operação



Na opção quatro podemos remover um job, decidi testar remover o job 10 ,e usando novamente a opção de mostrar podemos ver que ela foi removida com sucesso

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto, interior

Descrição gerada automaticamente

Na opção cinco, remove uma operação, removi no job 7, a operação 17



Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Na opção seis podemos alterar o id da operação, e se o job ou a operação não existir dará erro

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto, interior

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Na opção sete podemos alterar o id da maquina, e se o job ou a operação não existir dará erro

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Na opção oito podemos alterar o tempo, e se o job ,a operação ou maquina não existir dará erro

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto, interior, teclado

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Na opção 9 é calculado o melhor caminho pedindo um limite, e gerado um html onde este será apresentado em forma de tabela

Uma imagem com texto

Descrição gerada automaticamente



Uma imagem com mesa

Descrição gerada automaticamente

# **Conclusão**

Em suma, foi um trabalho trabalhoso, mais complicado do que a primeira entrega, algumas coisas poderia ter feito de outra maneira mas em geral penso que apliquei bem os conhecimentos adquiridos ao longo do semestre.

# **Bibliografia**

Para este trabalho foram usados os repositórios git das aulas deste ano bem como do ano anterior, para a correção de erros que iam surgindo, o stackoverflow