Universidade Federal do Piauí - Campus Senador Helvídio Nunes de Barros – CSHNB

Disciplina: Estrutura de dados I

Relatório de análise da execução com algoritmos recursivos, ordenação e busca. Dupla: João Carlos de Sousa fé e Vitor José Ferreira dos Santos Santana

1. Resumo do projeto

O projeto envolve a resolução de 5 exercícios, onde cada um deles é necessário aplicar é necessário aplicar os conhecimentos de Pilha (estática e dinâmica) e Fila (estática e dinâmica)

2. Introdução

As técnicas utilizadas no projeto definem estruturas de dados aplicadas de maneira clara e objetiva, a fim de manter um código otimizado e modular. As técnicas e algoritmos utilizados foram:

TADS: Tipos abstratos de dados;
Pilhas: Operações de criação estática e dinâmica
Filas: Operações de criação estática e dinâmica

3. Seções Específicas

Detalhes arquiteturais: O sistema operacional utilizado para a realização dos exercícios foi o Windows 10, com o compilador *C MinGw da Microsoft* e com o ambiente de desenvolvimento *Visual Studio Code* e o sistema de versionamento de código *Git e Github*.

Exercícios:

Questão 1: Utilizando-se de ***Fila estática***, a questão utiliza implementar as operações conhecidas de fila: Inserção, Remoção, impressão, se a fila é vazia ou não.

Questão 2: Utilizando-se de ***Fila dinâmica,*** a questão utiliza implementar as operações conhecidas de fila: Inserção, Remoção, impressão, se a fila é vazia ou não. Foram utilizados processos de alocação e desalocação de memória.

Questão 3: Utilizando-se de **Pilha estática**, a questão pede para implementar a verificação e transformação de uma expressão matemática infixa para pós-fixa. Foi necessário fragmentar a string (que é a expressão digitada pelo usuário) em outras strings, resultando em cada elemento da expressão como pilha de elementos. Após isso, foi necessário inverter a pilha e em seguida, dar início a verificação se cada elemento da expressão era válido. Se cada elemento passa se na avaliação, o mesmo era empilhado na pilha de expressões, caso não, o programa é abortado. Em seguida, caso a expressão fosse válida, a pilha de expressões era enviada para a

função “converterPosFixa”, onde a pilha de expressão é distribuída para 2 outras pilhas de operadores e de saída.

Questão 4: Utilizando-se de **Pilha dinâmica,** a questão pede para implementar a verificação e transformação de uma expressão matemática infixa para pós-fixa. Foi necessário fragmentar a string (que é a expressão digitada pelo usuário) em outras strings, resultando em cada elemento da expressão como pilha de elementos. Após isso, foi necessário inverter a pilha e em seguida, dar início a verificação se cada elemento da expressão era válido. Se cada elemento passa se na avaliação, o mesmo era empilhado na pilha de expressões, caso não, o programa é abortado. Em seguida, caso a expressão fosse válida, a pilha de expressões era enviada para a função “converterPosFixa”, onde a pilha de expressão é distribuída para 2 outras pilhas de operadores e de saída.

Questão 5: Utilizando-se de **Fila dinâmica,** foram criadas 3 filas de prioridade, onde a fila de processos de prioridade alta é executada até ficar vazia. O usuário insere os processos na fila de escolhida e escolhendo um tempo de processamento. não é necessário informar o número do processo, pois o mesmo é gerado automaticamente. após inseridos, o escalonador está pronto para executá-los.

4. Resultados da execução

Após as escritas de código de cada exercício, foram realizados vários tipos de testes: testes de validação, entradas de dados variadas e teste das funções individualmente para avaliar desempenho e comportamento com diferentes dados. Depois foram avaliadas as chamadas de todas as funções como um todo, a fim de se saber como se dá a comunicação entre elas e se os retornos e passagem de dados é tido como o esperado.

Exercício 1 (fila estática):

Opção 1	Placa do carro: 111 Entrar ou sair do estacionamento: E - insere o carro na fila Placa do carro: 222 Entrar ou sair do estacionamento: E - insere o carro na fila Placa do carro: 333 Entrar ou sair do estacionamento: E - Insere o carro na fila

	Placa do carro: 222 Entrar ou sair do estacionamento: S - carro sai do estacionamento **Resultado: “*Carro com placa 222 saiu do estacionamento...Total de vezes que foi manobrado: 1 vezes* *Para o carro sair foi preciso manobrar 1 carros* “**
Opção 2	**Resultado:** “ *111* *333* “

Exercício 2 (fila dinâmica):

Opção 1	Placa do carro: 111 Entrar ou sair do estacionamento: E - insere o carro na fila Placa do carro: 222 Entrar ou sair do estacionamento: E - insere o carro na fila Placa do carro: 333 Entrar ou sair do estacionamento: E - Insere o carro na fila Placa do carro: 222 Entrar ou sair do estacionamento: S - carro sai do estacionamento **Resultado: “ Total saíram da frente: 1. total manobrados: 1 “**
Opção 2	**Resultado:** *“111* *333* “

Exercício 3 (Pilha estática):

Opção 1	**Entrada:** “3 a” **saida:** “expressão inválida" ***motivo do erro: caractere que não é operando nem operado é invalido*** **Entrada:** “3 * * 2” **saida:** “expressão inválida" ***motivo do erro: dois operandos de * seguidos*** **Entrada:** “3 * 5 + 1” **saída:** “3 5 * 1 +" **Entrada:** “3 + 5 * 2 + (4 * 1” **saida:** “expressão inválida" ***motivo do erro: total de abertura de parênteses é diferente de parênteses fechados*** **Entrada:** “ 3 + 5 * 2 + (4 * 1 * (5 * 3 + 1 * 2 (5 + 1) * 2))” **saida:** “expressão inválida" ***motivo do erro: falta um sinal de multiplicação da parte “***2 (5 + 1)***”*** **Entrada:** “ 3 + 5 * 2 + (4 * 1 * (5 * 3 + 1 * 2 * (5 + 1) * 2))” **saída:** “3 5 2 * + 4 1 * 5 3 * 1 2 * 5 1 + * 2 * + * +"

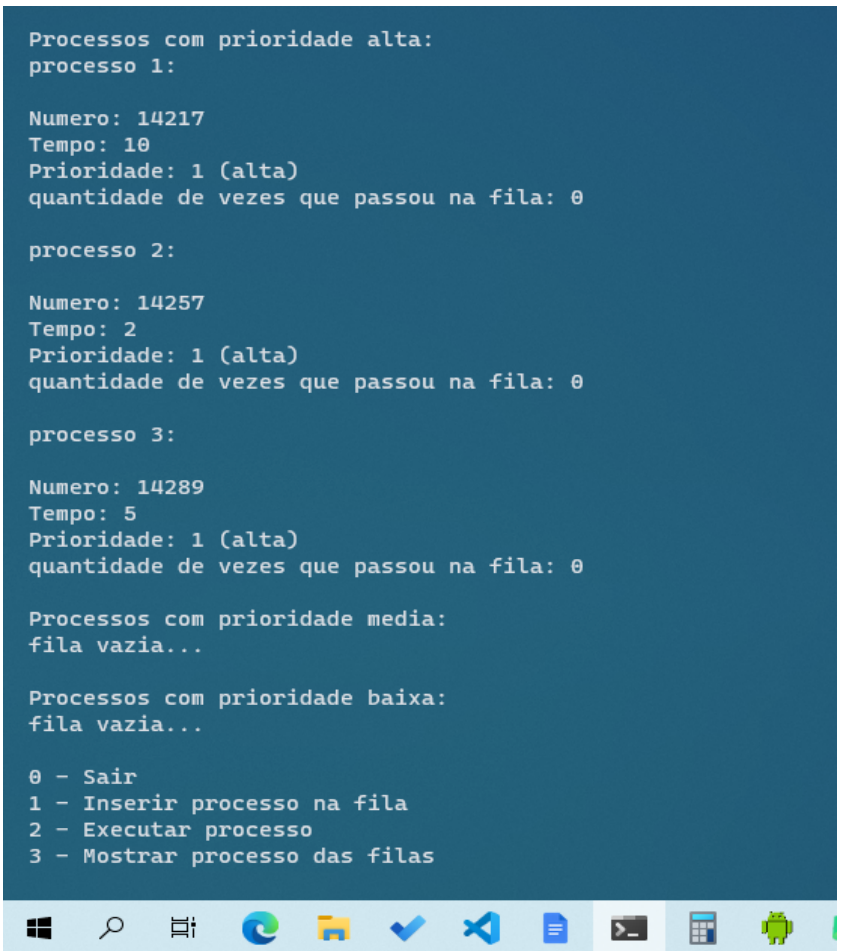
Exercício 4 (Pilha dinâmica):

Opção 1	**Entrada:** “3 a” **saida:** “expressão inválida" ***motivo do erro: caractere que não é operando nem operado é invalido*** **Entrada:** “3 * * 2” **saida:** “expressão inválida" ***motivo do erro: dois operandos de * seguidos***

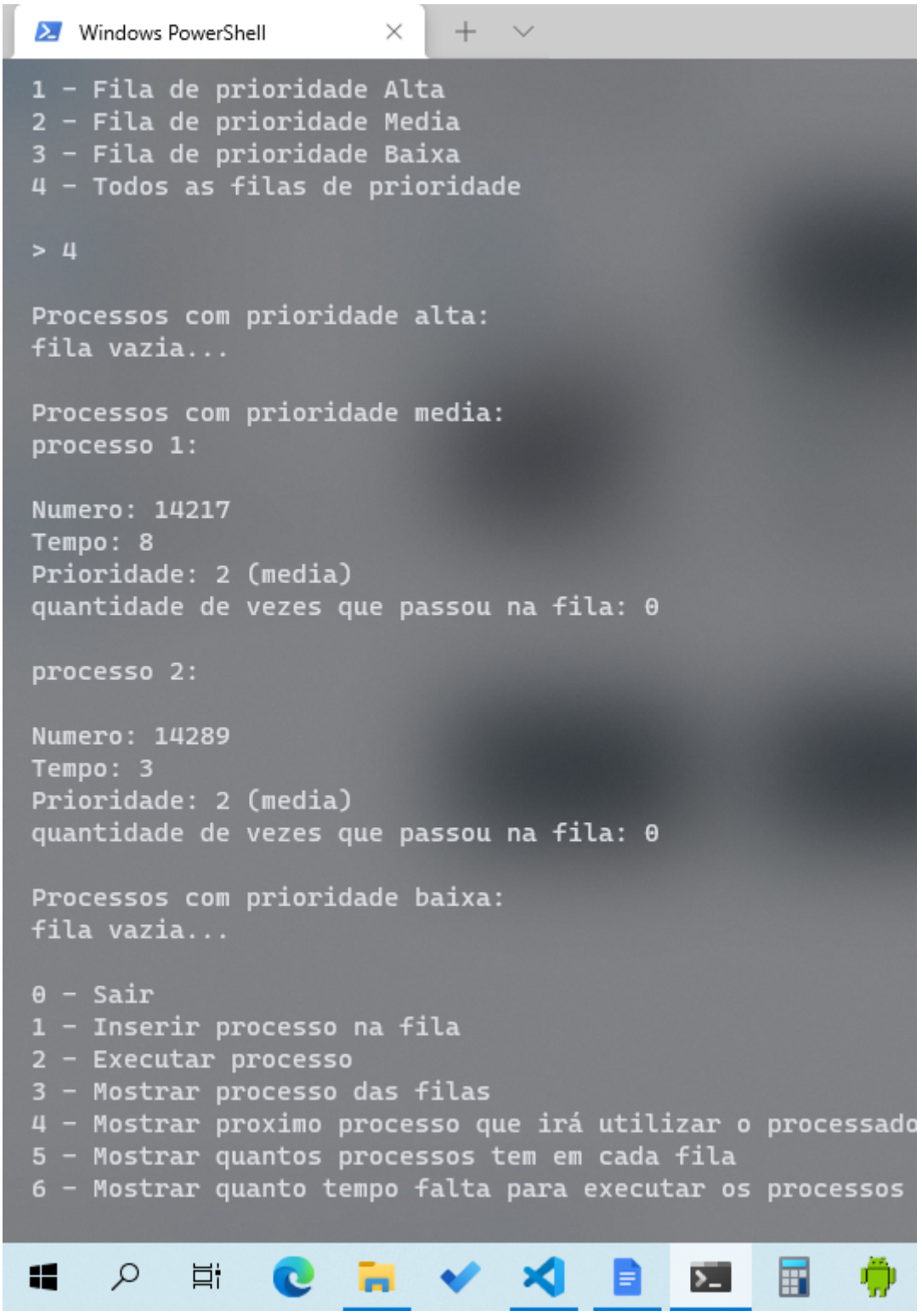
	Entrada: “3 * 5 + 1” **saída:** “3 5 * 1 +" **Entrada:** “3 + 5 * 2 + (4 * 1” **saida:** “expressão inválida" ***motivo do erro: total de abertura de parênteses é diferente de parênteses fechados*** **Entrada:** “ 3 + 5 * 2 + (4 * 1 * (5 * 3 + 1 * 2 (5 + 1) * 2))” **saida:** “expressão inválida" ***motivo do erro: falta um sinal de multiplicação da parte “***2 (5 + 1)***”*** **Entrada:** “ 3 + 5 * 2 + (4 * 1 * (5 * 3 + 1 * 2 * (5 + 1) * 2))” **saída:** “3 5 2 * + 4 1 * 5 3 * 1 2 * 5 1 + * 2 * + * +"

Exercício 5 (Fila dinâmica): Mostrando a fila de processos após inserir os processos na fila de prioridade alta.

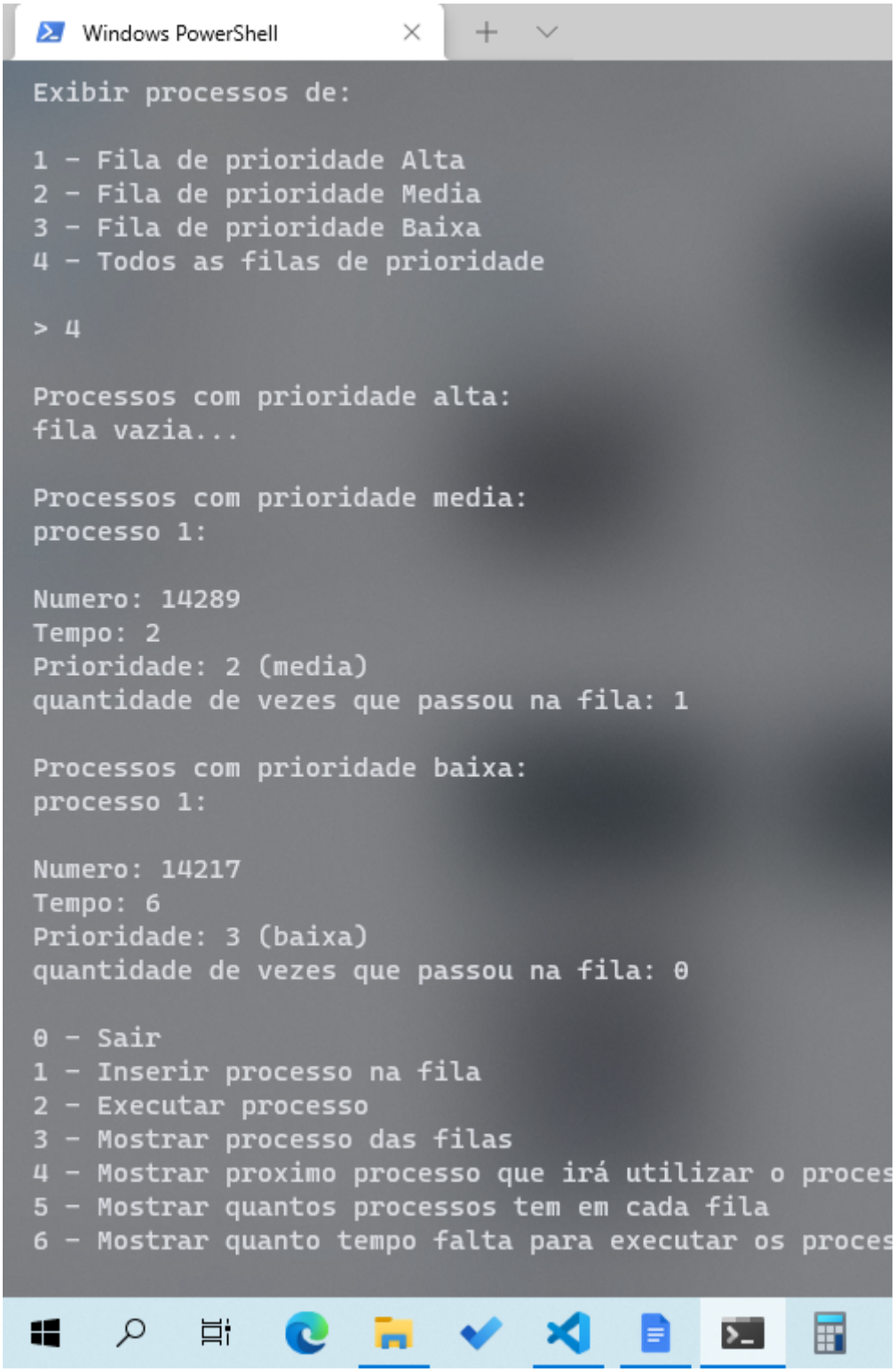
Após 6



execuções…



Após mais 4 execuções…


Windows PowerShell
Exibir processos de:
1 - Fila de prioridade Alta
2 - Fila de prioridade Media
3 - Fila de prioridade Baixa
4 - Todos as filas de prioridade
> 4
Processos com prioridade alta:
fila vazia...
Processos com prioridade media:
processo 1:
Numero: 14289
Tempo: 2
Prioridade: 2 (media)
quantidade de vezes que passou na fila: 1
Processos com prioridade baixa:
processo 1:
Numero: 14217
Tempo: 6
Prioridade: 3 (baixa)
quantidade de vezes que passou na fila: 0
0 - Sair
1 - Inserir processo na fila
2 - Executar processo
3 - Mostrar processo das filas
4 - Mostrar proximo processo que irá utilizar o proces
5 - Mostrar quantos processos tem em cada fila
6 - Mostrar quanto tempo falta para executar os proces

5. Conclusão.

A resolução dos exercícios propostos tiveram várias aplicações de técnicas de programação, aprendizagem e criação de novas técnicas como utilização de estruturas dinâmicas, onde cada uma tinha vantagens e desvantagens (tanto em implementação, quanto desempenho e uso de memória) e aperfeiçoamentos das técnicas já aprendidas. Todas implementações básicas de fila e pilhas foram utilizadas e aproveitadas para uma melhor codificação, resultando numa maior facilidade de implementação e leitura do codigo. Utilizando-se de modularização das funções, resultaram numa maior clareza e solução dos problemas encontrados durante o desenvolvimento, assim como as novas técnicas trouxeram alguns problemas, ao final elas ajudaram no encontro dos erros, permitindo a realização dos testes de maneira fácil e prática.