

INTRODUCAO

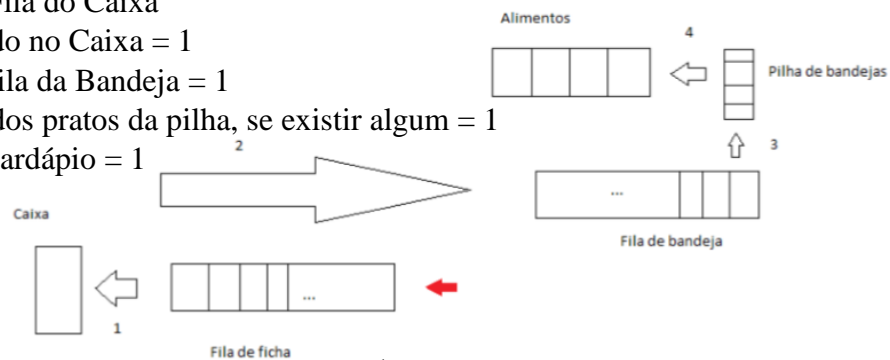
O Restaurante da cantina do ICEX, recebe todos os dias, um grande volume de pessoas para o almoço, que é servido durante 4 horas. Eles possuem um cardápio já definido, com variações apenas dentro de seus quesitos (arroz, feijão, guarnição e salada). Um indivíduo, para se servir desse cardápio, demora em média cerca de 4 minutos.

Um dos maiores problemas enfrentados no almoço é a gestão das filas para o atendimento no restaurante. Sua organização é basicamente realizada em 2 tipos de fila, as filas do caixa e as filas da bandeja. Nessa segunda fila, existe uma dependência que seria uma pilha de pratos, na qual a saída para servir do almoço só é permitida se existir ao menos um prato que possa ser utilizado pelo indivíduo que irá para a etapa de servir.

Como mostra na descrição e figura abaixo, as etapas incluem:

- Entrar na Fila do Caixa
- Ser atendido no Caixa = 1
- Entrar na fila da Bandeja = 1
- Pegar um dos pratos da pilha, se existir algum = 1
- Servir do cardápio = 1

Tempo Total = 4



¹Imagem extraída do enunciado do Trabalho Prático 1

O atendimento da fila do caixa é condicionado ao tempo em que o almoço é servido. A partir desse momento, não será mais permitido a entrada de pessoas para a fila do almoço, sendo servido apenas aos que já estiverem lá.

Sabendo da deficiência existente no tempo de atendimento, foi feito uma simulação, permitindo que, dentro de cada etapa, sejam contempladas as diversas variações de filas que poderiam existir, buscando assim, aquela que de maneira mais ágil e com menor custo operacional, venha a concluir todo o processo no menor espaço de tempo possível.

Para efeitos de estudo do caso, foi estipulado que a cada unidade de tempo, dentro dos 240 minutos de funcionamento, 2 pessoas chegarão ao restaurante para se servir do almoço.

DESENVOLVIMENTO

Com base na abstração dessa ideia, foi feito em linguagem C, um programa que simulasse o funcionamento do restaurante da cantina do ICEX. Esse programa, contém além da função principal, dois tipos abstratos e um módulo com operações pertinentes ao funcionamento do programa.

Tipo Abstrato de Dados

Para uma representação fidedigna ao funcionamento, foram criados dois Tipos Abstratos de Dados (TAD), que basicamente, são constituídos de estruturas que contém as informações necessárias para a simulação e de operações para a manipulação dessas. Segue abaixo os TADs utilizados:

- Fila: Nesse TAD, se encontra o modelo operacional de uma fila de pessoas em sua essência.
 - Estruturas:
 - T_ITEM – Terá as informações necessárias sobre cada indivíduo, que em nosso caso será uma ficha de atendimento e o tempo de entrada na fila.
 - T_FILA – Dentro dessa estrutura, ficarão informações pertinentes a fila e as referências ao seu conteúdo.
 - Operações:
 - *FFVazia* – Cria uma nova fila
 - *VaziaF* – Verifica se já existe algum indivíduo está na fila
 - *Enfileira* – Insere pessoa na fila
 - *Desenfileira* – Remove pessoa da fila
 - *DestroiFila* – Elimina uma fila criada
 - *RetornaPrimeiro* – Verifica quem é o primeiro elemento da fila
 - *VerificaFMenor* – Verifica qual das filas é a menor fila
 - *VerificaFMaior* – Verifica qual das filas é a maior
- Pilha: Esse será o TAD que simulará o comportamento da pilha de pratos existente.
 - Estruturas:
 - T_CONTEUDO – É alojada apenas a informação sobre qual a posição do jogo de talher e prato na pilha de pratos.
 - T_PILHA – Possui as informações sobre a pilha em questão
 - Operações:
 - *FPVazia* – Cria uma pilha de pratos
 - *VaziaP* – Verifica se a pilha está vazia
 - *Empilha* – Insere jogo de talher e prato na pilha
 - *Desempilha* – Remove jogo de talher e prato da pilha
 - *DestroiPilha* – Elimina uma pilha de pratos criada
 - *VerificaPMaior* – Verifica qual a maior pilha de pratos

Modulo Bandeja

Esse módulo constitui as operações necessárias para o programa. Nele estão presentes os comportamentos lógicos para simular o dia-a-dia do restaurante. Situações como, a movimentação entre as filas e pilhas, a dependência entre as filas de pessoas e as pilhas de pratos, a chamada dos processos de inserção ou remoção de cada usuário, e os procedimentos que recebem os parâmetros para inicialização e finalização da execução do programa estão contidos. Suas funções são:

- **VerificaParametro** – Verifica os parâmetros recebidos por linha de comando e os converte para tipos inteiros. Também checa se foram recebidos os parâmetros adicionais de alteração, se não existirem atribui os valores default para essas variáveis do programa.
- **IniciaFila** – Recebe a quantidade de filas de caixa e de bandeja, passada por parâmetro, para criá-las no programa
- **PreenchePilha** – Insere o número máximo de pratos para a primeira execução da pilha criada
- **IniciaPilha** - Recebe a quantidade de pilhas de prato passada por parâmetro para criá-las no programa
- **InserePessoa** – Insere as pessoas em uma das filas de caixa, de acordo com a sua chegada e com o número de filas.
Parametro – Verifica qual das situações é o limite máximo para o número de trocas de fila
- **TrocaFila** – De acordo com o número máximo de inserções/remoções em um dado momento, remove das filas de caixa para inserir nas filas de bandeja
- **ReporPrato** – Dado um instante, repõe o número de pratos na pilha de pratos até o tamanho máximo definido a ela
- **RemovePessoa** – Avalia as primeiras pessoas das filas de bandeja e se já estiverem no momento de saída da fila, caso exista prato para ser utilizado, remove as pessoas
- **Funcionamento** – Caso o tempo de funcionamento do restaurante tenha se encerrado, ele avalia se ainda há pessoas nas filas de bandeja
- **LimpaPonteiros** – Elimina do programa o “lixo” que tiver restado nos ponteiros utilizados.

Programa Principal

O programa principal, é responsável pela inicialização de todo o processo de serviço do restaurante. Ele cria as variáveis necessárias para operar o programa, recebe os parâmetros, e os distribui para serem operados nas funções. É, dele também, o controle do tempo de execução do programa e a exibição dos resultados obtido após a execução.

ANALISE DE COMPLEXIDADE

DestroiFila() – Procedimento que executa um laço enquanto existir célula de item na fila. Ele remove a célula e decrementa no tamanho da fila.

Ordem: $O(n)$

DestroiPilha() – Procedimento que executa um laço enquanto existir campo de conteúdo na pilha. Ele remove a célula e decrementa no tamanho da fila.

Ordem: $O(n)$

VerificaParametro () – Procedimento que recebe os parâmetros por linha de comando. Executa uma comparação para verificar se recebeu mais de 3 argumentos, caso não atribui valores default para as variáveis em questão.

Ordem: $O(1)$

IniciaFila() – Procedimento que executa dois laços para preencher os dois tipos de fila (caixa e bandeja) e a cada iteração, cria uma fila. Ambos os laços variam de 0 até tamanho.

Ordem: $(O(n) + O(n)) = O(n)$

PreenchePilha() – Procedimento que executa um laço que vai de 0 até quantidade de pratos. Nesse procedimento, é empilhado conteúdo na pilha.

Ordem: $O(n)$

IniciaPilha() – Procedimento que executa um laço que vai de 0 até tamanho. A cada iteração cria uma pilha e chama a função *PreenchePilha()*

Ordem: $O(n^2)$

InserPessoa() – Procedimento que executa um laço que vai de 0 até o número de pessoas por vez que chegam a fila. A cada iteração, atribui valores ao item pessoa e enfileira o item.

Ordem: $O(n)$

Parametro() – Função que faz comparações para definir qual o menor dos parâmetros passados.

Ordem: $O(1)$

TrocaFila() – Procedimento que executa um laço que vai de 0 até o menor dos parâmetros escolhidos pela função anterior. Faz comparações, se for o momento, desenfileira da fila caixa e enfileira na fila bandeja.

Ordem: $O(n)$

ReporPrato() – Procedimento que compara se é o momento de repor pratos. Caso for, executa um laço que vai de 0 até tamanho de pratos, e realiza outro laço que vai de 0 até, no máximo, a quantidade de reposição de pratos. A cada iteração, um jogo de talher e prato é inserido.

Ordem: $O(n^2)$

RemovePessoa() – Função executa um laço. A cada iteração, faz comparação para verificar se as pilhas e filas de pratos estão vazias, compara se está no momento de remoção do item, se estiver, desenfileira um item pessoa e desempilha um conteúdo talher(jogo de pratos de talher), atribuindo o valor desenfileirado à variável que contabiliza a média de tempo. Após fim do laço, retorna o valor da média.

Ordem: $O(n)$

Funcionamento() – Procedimento que realiza um laço que vai de 0 até o número de bandejas. A cada iteração, verifica se a bandeja na posição do índice está vazia. Retorna o número de bandejas vazias.

Ordem: $O(n)$

LimpaPonteiros() – Procedimento que executa três laços de 0 até tamanho. A cada iteração, chama procedimento para destruir um dos t_principal das TAD.

Ordem: $(O(n^2) + O(n^2) + O(n^2)) = O(n^2)$

Main() – Chama os procedimentos do modulo bandejão. VerificaParametro(), IniciaFila(), IniciaPilha(). Após executa-los, realiza um laço enquanto a função Funcionamento não retorna o valor de qtd de bandeja. A cada iteração, compara o tempo com o tempo máximo de atendimento para almoço, caso for menor, chama os procedimentos de InserePessoa() e TrocaFila(). Ainda na mesma iteração, incrementa a função RemovePessoa() a variável med_tempo, realiza o procedimento ReporPrato e atribui a função Funcionamento a variável funciona.

Ordem: $(O(1) + O(n) + O(n^2) + (n*[O(n) + O(n) + O(n) + O(n^2) + O(n)]) + O(n))$
 $= O(n^3)$

RESULTADOS

Os testes do programa foram realizados em um computador com processador Intel Atom x5-Z8500, com 4 GB de memória RAM. O Ambiente utilizado, foi a Bash no Ubuntu no Windows, que é uma bash Linux rodando nativamente no Windows 10.

Modo de Execução

Para executar o programa, basta seguir os seguintes passos:

- Descompactar o arquivo TP1.zip na pasta de destino desejada:
 - `cd /"local de destino dos arquivos"`
- Compilar através do arquivo Makefile:
 - `make`
- Inserir os parâmetros desejados no arquivo executável gerado:
 - `./bandejao.out C B P q t r`
 - Parâmetros exigidos:
 - C = Valor inteiro que corresponde a quantidade de filas de caixa desejada
 - B = Valor inteiro que corresponde a quantidade de filas de bandeja
 - P = Valor inteiro que corresponde a quantidade de pilhas de prato
 - Parâmetros alternativos (caso não forem atribuídos, por default 30 12 10)
 - q = Valor inteiro que corresponde a quantidade de pratos na pilha
 - t = Valor inteiro que corresponde ao tempo em que os pratos são repostos
 - r = Valor inteiro que corresponde ao número máximo de pratos repostos

Testes

Após uma bateria exaustiva de testes, foi observado que para uma chegada de 2 pessoas a cada instante de tempo totalizando 240 minutos, 480 pessoas se dispõem a comprar o almoço.

- Execução ./bandejao.out 1 1 1 – O resultado mostra que além de uma elevada média de espera, o tempo de funcionamento do serviço de almoço teve de se estender por muito mais tempo para atender aos que já estavam sob esse serviço.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 151.18
 - Tempo: 550
- Execução ./bandejao.out 1 1 1 30 **5** 10 – Se atribuído um valor de tempo de reposição menor que o default (12), o tempo de atendimento reduz.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 125.75
 - Tempo: 481
- Execução ./bandejao.out 1 1 1 30 **30** 10 – Se atribuído um valor de tempo de reposição maior que o default (12), o tempo de atendimento aumenta consideravelmente.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 536.36
 - Tempo: 1360
- Execução ./bandejao.out 1 1 1 **15** 12 10 – Com a redução na quantidade inicial de pratos, o tempo de espera aumenta ligeiramente.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 165.98
 - Tempo: 569
- Execução ./bandejao.out 1 1 1 **60** 12 10 – Com o aumento na quantidade inicial de pratos, o tempo de atendimento reduz ligeiramente.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 131.71
 - Tempo: 514
- Execução ./bandejao.out 1 1 1 30 12 **5** – Com a redução na reposição dos pratos, o tempo de atendimento também aumenta consideravelmente.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 399.66
 - Tempo: 1085
- Execução ./bandejao.out 1 1 1 30 12 **20** – Com o aumento na reposição de pratos, o tempo de atendimento reduz ligeiramente.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 125.75
 - Tempo: 481

- Execução ./bandejao.out 3 1 1 – Se a quantidade de filas de caixa, aumenta sozinha, o tempo de atendimento se mantém o mesmo da condição (1 1 1).
 - Atendidos: 480
 - Nao Atendidos: 0
 - Media: 151.52
 - Tempo: 551
- Execução ./bandejao.out 1 5 1 – Se a quantidade de filas de bandeja, aumenta sozinha, o tempo de atendimento não sofre grandes alterações, mas menos pessoas serão atendidas.
 - Atendidos: 475
 - Nao Atendidos: 5
 - Media: 146.12
 - Tempo: 543
- Execução ./bandejao.out 1 1 5 – Se a quantidade de pilhas de prato, aumenta sozinha, o tempo de atendimento reduz ligeiramente.
 - Atendidos: 479
 - Nao Atendidos: 1
 - Media: 125.75
 - Tempo: 481
- Execução ./bandejao.out 3 1 3 – Se a quantidade de fila de caixa e pilha de pratos aumentam, o tempo de atendimento reduz ligeiramente.
 - Atendidos: 480
 - Nao Atendidos: 0
 - Media: 126.00
 - Tempo: 482
- Execução ./bandejao.out 3 3 1 – Se a quantidade de fila de caixa e fila de bandeja, aumentam, o tempo de atendimento reduz ligeiramente.
 - Atendidos: 478
 - Nao Atendidos: 2
 - Media: 146.55
 - Tempo: 545
- Execução ./bandejao.out 1 5 5 – Se a quantidade de fila de bandeja e pilha de pratos aumentam, mas são superiores a fila de caixa, o tempo de atendimento é melhorado significativamente.
 - Atendidos: 478
 - Nao Atendidos: 2
 - Media: 5.35
 - Tempo: 241
- Execução ./bandejao.out 3 3 3 – Aumentando igualmente as 2 filas e a pilha, a partir do 3, o tempo de atendimento opera próximo do ideal.
 - Atendidos: 476
 - Nao Atendidos: 4
 - Media: 7.05
 - Tempo: 242

CONCLUSÃO

Após uma execução apurada de testes com base nas diversas configurações possíveis, fica evidente que a melhor solução para o problema de distribuição de, filas de caixa, filas de bandeja e pilha de pratos, que ocorre no Restaurante da Cantina do ICEX é operar no modo ideal. Uma das configurações que provou ser a mais prática e eficiente é a de aumentar a quantidade de filas de bandeja para 6 e a de pilhas de pratos para 3.

Alterar a quantidade de filas de caixa ou outras configurações para reposição de talher e pratos, tempo de reposição ou quantidade inicial de talheres e pratos na pilha de bandeja, mostrou-se pouco eficiente.

- Execução ./bandejao.out **1 6 3** – configuração que provou-se ser a ideal. Pois a partir dela, o tempo mínimo é atingido.
 - Atendidos: 478
 - Não Atendidos: 2
 - Média: 5.34
 - Tempo: 241