

Trabalho 03

Versão: 7 de outubro de 2019

1 Instruções:

- O trabalho é **no máximo em dupla**. Os nomes devem estar no cabeçalho comentado no arquivo principal.
- Será aplicado detector de plágio! Uma vez detectado plágio, a **nota da disciplina** será **zerada**.
- Data de entrega: **22 de outubro de 2018 (terça-feira)** no sistema do moodle até as **10:00**.
- A entrega deve ser no formato ZIP com o seguinte conteúdo:
 - + Código-fonte (bem como os arquivos `.c` e `.h` referentes aos TAD utilizados);
 - + Arquivo Makefile¹ associado para compilar o código.
- O código deve estar escrito em Linguagem C e as únicas bibliotecas permitidas são: `stdio.h`, `stdlib.h`, `string.h` e `math.h` caso contrário o trabalho será invalidado.
- Para validação do código será utilizado um ambiente GNU/Linux com o compilador GCC (versão 7+) passando os seguintes parâmetros²: `gcc -Wall -Wextra -Werror -Wpedantic`
- Para verificação de vazamento de memória e gerência dos ponteiros, será utilizado o software `Valgrind`³.
- Para verificação de correspondência com a saída especificada será utilizado o software `diff`⁴.

2 Correção

Serão utilizados os seguintes critérios para correção:

2.1 Compilação

- Compilação com parâmetros requeridos;
- Makefile funcionando;
- **Se não compilar utilizando o Makefile enviado, o trabalho não será corrigido.**

2.2 Entrada/Saída

- Cada entrada gera uma saída correta.

2.3 TAD / Estrutura de dados

- Implementação correta das estruturas de dados.
- Implementado um TAD por estrutura.
- Implementação correta do conceito de TAD.

2.4 Organização do código

- Funções separadas para cada uma das operações.
- Comunicação por retorno de função.

2.5 Gerência de memória

- Teste de alocação.
- Remoção correta da memória alocada.
- Vazamentos de memória.
- Controle de ponteiros.

Observação: A correção será em boa parte automatizada. Faça o programa funcionar de acordo com as especificações.

¹Veja mais sobre em https://pt.wikibooks.org/wiki/Programar_em_C/Makefiles.

²Leia mais sobre em: <https://gcc.gnu.org/onlinedocs/gcc-6.4.0/gcc/Warning-Options.html>.

³<http://valgrind.org>

⁴<https://www.geeksforgeeks.org/diff-command-linux-examples/>

3 Simulação de uma AGÊNCIA – PARTE III

Nosso objetivo final é desenvolver uma simulação de uma agência de banco simplificada. Tal simulação gerencia o saldo dos clientes envolvidos na simulação através de saques, transferências e depósitos realizados nos guichês da agência.

Para finalizar a implementação da simulação precisamos organizar a implementação e utilização das nossas estruturas de dados. Nesta etapa do desenvolvimento iremos implementar vários relatórios diários, o relatório final e modularizar a utilização de nossas estruturas de dados. Desta forma, deveremos ser capazes de executar simulação que serão definidas através de uma entrada, a simulação deve gerar vários relatórios diários informando a situação de cada guichê e depois um relatório final da simulação geral.

3.1 A simulação

Para cada simulação, informamos que a agência contém M guichês, recebe N clientes durante toda a simulação e D clientes durante um dia. Cada cliente atendido gera os seguintes dados: Código de pessoa física(CPF), operação, valor e CPF de terceiro envolvido na operação. Cada um dos dados tem a seguinte forma:

- CPF cliente: inteiro;
- Operação: caractere ‘D’ para depósito, ‘S’ para saque, ‘T’ para transferência;
- Valor: inteiro;
- CPF terceiros: inteiro;

A entrada dos clientes é gerenciada por uma Fila, onde deve-se organizar todos os clientes em fila e depois enviar uma a uma ao guichê disponível utilizando a estratégia de round-robin iniciando do guichê 0 (zero), ou seja, se o cliente anterior foi alocado no guichê $k \bmod M$, então o presente cliente de ser alocado no guichê $k + 1 \bmod M$ ficando a seguinte ordem dos guichês selecionados: $0, 1, 2, \dots, M - 2, M - 1, 0, 1, 2, \dots, M - 2, M - 1, 0, \dots, 2, \dots, M - 2, M - 1, 0, 1, 2, \dots$

Cada cliente realiza apenas uma operação, mas não existe restrição da quantidade de vezes que um cliente é atendido. Cada guichê atende a um cliente por vez e armazena os documentos de cada cliente em uma pilha. Ao finalizar os atendimentos diários, deve ser impresso o relatório de cada guichê de acordo com a descrição na Seção 3.3. Após os relatórios diários, o sistema deve unir todos os dados referentes aos documentos, processar as transações e imprimir um relatório final. Tal relatório deve ser armazenado em uma lista ordenada pelo CPF.

Ao efetuar uma transação, temos os seguintes comportamentos possíveis:

- Depósito: Acrescentar o valor no saldo do cpf de terceiros (CPFt);
- Saque: Retirar o valor no saldo do cpf do cliente (CPFt);
- Transferência: Retirar o valor do saldo do cpf do cliente (CPF) e acrescentar o valor no saldo do cpf de terceiros (CPFt).

Observe que a operação de saque não deve inserir o cpf de terceiros no relatório.

3.2 Entrada

A entrada é constituída por uma única simulação enviada pela entrada padrão. A primeira linha contém 3 (três) inteiros N, M e D , onde $1 \leq N, M, D \leq 2^{32} - 1$. N representa a quantidade de clientes envolvidos na simulação e M a quantidade de guichês. As próximas N linhas deve conter os seguintes valores CPF , $CPFT$, O e V onde $1 \leq CPF \leq 2^{32} - 1$ representa o CPF do cliente, $1 \leq CPFT \leq 2^{32} - 1$ o CPF do terceiro envolvido, $O \in \{D, S, T\}$ o caractere referente a operação e $1 \leq V \leq 2^{30} - 1$ a quantidade de dinheiro envolvido na transação.

Teremos a seguinte estrutura geral da entrada:

Entrada:
$N \ M \ D$
$CPF_1 \ CPFT_1 \ O_1 \ V_1$
$CPF_2 \ CPFT_2 \ O_2 \ V_2$
\vdots
$CPF_N \ CPFT_N \ O_N \ V_N$

3.3 Saída

A saída é constituída pelos relatórios diários seguido do relatório final que deve ser enviados pela saída padrão (teclado) seguindo a formatação descrita nessa seção.

Relatório diário

Antes do relatório parcial deve ser impressa a mensagem “-: | RELATORIO DIARIO D |:-”, onde D é o inteiro referente ao d -ésimo dia e seguido de um inteiro M referente a quantidade de guichês envolvidos na simulação. Na linha seguinte deve ser impresso o texto “Guiche k : Q_k ” onde k é o número do guichê e um inteiro Q_k referente a quantidade de operações empilhadas no guichê k . Nas próximas P_k linhas deve constar os dados de cada operação na formatação “[CPF_c, CPF_t, O, V]” onde “ CPF_c ” é referente ao CPF do cliente atendido, “ CPF_t ” o cpf do terceiro envolvido na operação, “ O ” o caractere referente a operação e “ V ” o valor envolvido na operação. A ordem e impressão dos dados armazenados em cada guichê deve ser a mesma ordem da remoção da pilha.

Relatório Final

Antes do relatório final deve constar uma linha em branca e na seguinte impressa a mensagem “-: | RELATORIO FINAL |:-”. Na linha seguinte um inteiro K referente a quantidade de CPFs envolvidos na simulação e o relatório deve seguir a formatação “-: [CPF_ℓ : $B_\ell \ C_\ell \ C_\ell$ ”, onde “ CPF_ℓ ” um inteiro que indica o ℓ -ésimo CPF, B_ℓ um inteiro referente a quantidade de operações que envolvem o CPF_ℓ e C_ℓ o saldo final das movimentações referente ao CPF_ℓ .

Observações:

- i) No relatório final deve constar TODOS os CPFs envolvidos na simulação e não apenas os CPFs dos clientes (ignorando cpf de terceiros em saques).
- ii) Todos os clientes iniciam a simulação com saldo igual a zero.

Saída final

Como descrito acima, seguem os seguintes parâmetros para formatação da saída:

- M : Quantidade total de guichês.
- P_k : Quantidade de documentos armazenados no guichê k .
- CPF_c^i : CPF do cliente referente a j -ésima operação desempilhada do guichê i .
- CPF_t^i : CPF de terceiros referente a j -ésima operação desempilhada do guichê i .
- O_j^i : Operação do j -ésima operação desempilhada do guichê i .
- V_j^i : Valor envolvido na j -ésima operação desempilhada do guichê i .
- K : Quantidade total de CPF's distintos envolvidos (cliente e terceiros).
- CPF_ℓ : ℓ -ésimo CPF envolvido na simulação (ordenado).
- B_ℓ : Quantidade operações que envolvendo CPF_ℓ .
- C_ℓ : Saldo final referente ao CPF_ℓ .

Ademais, teremos a estrutura na tabela 3.3:

4 Restrições e informações adicionais

Estruturas de dados & programação

1. Cada guichê deve ser implementado como uma pilha implementada utilizando **alocação simplesmente encadeada**.
2. A lista que armazena o relatório final (ordenado por cpf) deve ser implementada utilizando **alocação duplamente encadeada circular**.
3. A fila de clientes deve ser implementada como uma fila utilizando **alocação sequencial circular com tamanho $\lfloor \sqrt{D} \rfloor$** .
4. Não utilizar variável global!

TAD – Tipos Abstratos de Dados

A estruturas utilizadas na simulação devem ser implementadas utilizando o conceito de TAD em sua totalidade, ou seja, a simulação não deve utilizar nenhuma informação das implementações. É importante observar que no arquivo de simulação não deve-se ter conhecimento das implementações, ou seja, não podemos utilizar instruções do tipo “ $X \rightarrow \text{chave} = \text{valor}$ ” fora do TAD.

Funções

É importante que as funções façam suas operações de modo independente. Para um código bem organizado o mesmo não deve conter variáveis globais e as funções retornam valores que indicam o seu comportamento. É bastante comum definir padrões para o retorno das funções de acordo com os possíveis comportamentos da mesma. Códigos bem documentados receberão um agrado.

Por fim, espera-se que cada estrutura de dados contenha as funções criar, destruir, remover, inserir e buscar (não necessariamente com mesmo nome) quando forem necessárias para ser utilizadas. Ademais, as funções da simulação como enviar para atendimento, enfileirar, imprimir relatórios, etc, também devem ser implementadas separadamente.

Lembre-se: Tudo que é alocado, deve ser desalocado!

☺ Boa diversão!!!! ☺

Saída:

```
-:| RELATORIO DIARIO 1 |:-
M
Guiche 1:  $Q_1$ 
[CPFc11, CP Ft11, 011, V11]
[CPFc21, CP Ft21, 021, V21]
      ⋮
[CPFcQ11, CP FtQ11, 0Q11, VQ11]
      ⋮
Guiche M:  $Q_M$ 
[CPFc1M, CP Ft1M, 01M, B1M]
[CPFc2M, CP Ft2M, 02M, B2M]
      ⋮
[CPFcQMM, CP FtQMM, 0QMM, BQMM]
      ⋮
      ⋮
-:| RELATORIO DIARIO D |:-
M
Guiche 1:  $Q_1$ 
[CPFc11, CP Ft11, 011, V11]
[CPFc21, CP Ft21, 021, V21]
      ⋮
[CPFcQ11, CP FtQ11, 0Q11, VQ11]
      ⋮
Guiche M:  $Q_M$ 
[CPFc1M, CP Ft1M, 01M, B1M]
[CPFc2M, CP Ft2M, 02M, B2M]
      ⋮
[CPFcQMM, CP FtQMM, 0QMM, BQMM]

-:| RELATORIO FINAL |:-
K
-: [ CPF1: B1 C1
-: [ CPF2: B2 C2
      ⋮
-: [ CPFK: BK Ck
```

Tabela 1: Saída da simulação