DCOMP/UFSJ-Prof. Rafael Sachetto Oliveira AEDSII

Primeiro Trabalho Prático

Ciência da Computação 1º Semestre de 2019

1 Objetivo

Neste trabalho, seu objetivo será implementar um simulador de um servidor de emails. O sistema simulado terá suporte ao gerenciamento de contas, assim como à entrega de mensagens de um usuário para outros. O foco da simulação será verificar o funcionamento correto do sistema ao executar as diversas operações do servidor (descritas a seguir) em diferentes situações.

2 Descrição

Agora que você já conhece o conceito de algumas das mais importantes estruturas de dados (fila, pilha e lista encadeada) e como implementá-las em C, podemos começar a usá-las na prática. Suponha que você esteja desenvolvendo um novo servidor de emails para a uma grande empresa para substituir o servidor atual. O servidor será usado por milhões de pessoas, e sabemos que um bom gerenciamento e aproveitamento de memória nesse caso é fundamental. Sua responsabilidade é fazer uma implementação eficiente utilizando seu conhecimento de tipos abstratos de dados e alocação dinâmica de memória.

A empresa precisa que o servidor dê suporte à entrega e consulta de mensagens de email para usuários cadastrados. Além disso, o sistema também deve suportar o gerenciamento (adição e remoção) de contas de usuários. Dessa forma, o servidor que você vai simular deverá suportar as seguintes operações:

- **Cadastrar novo usuário:** esta operação cria uma caixa de entrada para o novo usuário. A nova caixa de entrada deverá estar vazia, ou seja, sem nenhuma mensagem armazenada. Após a criação da conta e enquanto ela não for removida, todas as mensagens enviadas para esse usuário devem ser armazenadas em sua caixa de entrada.
  - **Importante**: você não pode assumir um número pré-determinado de contas de usuário. Isso significa que o sistema não deve ter um número fixo ou máximo de contas de usuário. Cada conta criada deve ser alocada dinamicamente, e a memória utilizada por ela deve ser liberada se a conta for removida.
- **Remover usuário**: esta operação remove um usuário do sistema. Todas as mensagens em sua caixa de entrada devem ser automaticamente apagadas. Após a conta ser removida, mensagens recebidas pelo servidor e destinadas a esse usuário não devem ser aceitas.
- **Entregar email**: esta operação recebe um novo email destinado a determinado usuário. Caso o usuário não exista (ou seja, não tenha criado uma conta ou essa tenha sido removida), a mensagem deve ser recusada e gerar um erro. Se o usuário possuir uma caixa de entrada no servidor, a mensagem deve ser armazenada em ordem de prioridade na caixa de entrada desse usuário. Isso quer dizer que mensagens com prioridade mais alta devem ser armazenadas à frente de mensagens com prioridade mais baixa. Caso dois emails tenham a mesma prioridade, aquele que foi recebido primeiro pelo servidor deverá ficar à frente.
- **Consultar email**: esta operação recebe o identificador de um usuário e deverá retornar a próxima mensagem na caixa de entrada do usuário, segundo a prioridade descrita acima. Caso o usuário não exista, o servidor deve retornar um erro. Em caso de sucesso, a mensagem retornada deverá ser removida da caixa do usuário.

3 O que deve ser feito?

Você deverá implementar um simulador do sistema descrito acima. O simulador deverá suportar todas as quatro operações. Para cada operação processada, será gerada uma saída. Cada operação pode gerar um conjunto definido

de saídas, especificado abaixo. O formato da saída deve corresponder exatamente ao formato especificado, pois será através dela que a correção do simulador será realizada. A entrada do programa de simulação será um arquivo de texto contendo, em cada linha, uma operação. A simulação deve terminar quando o final do arquivo for alcançado. Para cada linha do arquivo de entrada, ou seja, para cada operação, deverá ser impressa uma linha na saída padrão (na tela), conforme especificado abaixo. A seguir, estão especificadas as operações, conforme presentes no arquivo de entrada:

- **CADASTRA ID**
  - Operação de cadastro de novo usuário. O parâmetro ID é um valor inteiro, que tem valor 0 à 100000, e é o identificador do novo usuário.
  - O servidor deve criar uma nova caixa de entrada vazia para o usuário ID. Enquanto essa conta não for removida, mensagens recebidas para o usuário ID deverão ser armazenadas nessa caixa de entrada,segundo a ordem de prioridade já explicada.
  - Somente uma caixa de entrada deve existir para um mesmo ID, ou seja, uma operação de cadastro deve falhar se o ID fornecido já estiver em uso.
  - Saída esperada (em todas elas, ID deve ser substituído pelo valor correspondente):
    - OK: CONTA ID CADASTRADA - caso a conta seja cadastrada com sucesso;
    - ERRO:CONTA ID JA EXISTENTE - caso a conta já exista.
  - Exemplo: CADASTRA 5
- **REMOVE ID**
  - Operação de remoção de conta e usuário. O parâmetro ID é o identificador do usuário cuja conta deve ser removida.
  - A conta ID deve ser removida, e todas as mensagens na caixa de entrada desta conta devem ser apagadas.
  - Após a conta ser removida, deverá ser possível cadastrar uma nova conta com identificador ID.
  - Saída esperada (em todas elas, ID deve ser substituído pelo valor correspondente):
    - OK: CONTA ID REMOVIDA - caso a conta seja removida com sucesso;
    - ERRO: CONTA ID NAO EXISTE - caso a conta não exista.
  - Exemplo: REMOVE 5
- **ENTREGA ID PRI MSG FIM**
  - Operação de solicitação de entrega de uma nova mensagem. O parâmetro ID é o identificador do destinatário da mensagem. Caso esse usuário esteja cadastrado no servidor, a mensagem deve ser armazenada em sua caixa de entrada segundo a prioridades já especificadas.
  - O parâmetro PRI é um inteiro entre 0 e 9 e indica a prioridade da mensagem. Valores mais altos indicam maior prioridade.
  - MSG é a mensagem a ser recebida: um conjunto de palavras separadas por espaço em branco e terminadas com a palavra reservada FIM
  - Saída esperada (em todas elas, ID deve ser substituído pelo valor correspondente):
    - OK: MENSAGEM PARA ID ENTREGUE - caso a mensagem seja entregue com sucesso;
    - ERRO: CONTA ID NAO EXISTE - caso a conta do destinatário não exista
  - Exemplo: ENTREGA 5 1 bom dia meu amigo, como voce esta? FIM
- **CONSULTA ID**
  - Operação de consulta de email. O parâmetro ID é o identificador do usuário para o qual a consulta foi feita.
  - Esta operação deve imprimir a primeira mensagem na caixa de entrada do usuário, segundo a ordem de prioridade já especificada. A mensagem deve então ser removida da caixa de entrada, de tal forma que a próxima consulta para esse mesmo usuário retorne a próxima mensagem segundo as mesmas prioridades.

- Saída esperada (em todas elas, ID deve ser substituído pelo valor correspondente):
  - CONSULTA ID: MSG - imprime a primeira mensagem na caixa de entrada do usuário, segundo a ordem de prioridade. Por exemplo, para o usuário do exemplo anterior e que possua apenas aquela mensagem em sua caixa de entrada: CONSULTA 5: bom dia meu amigo, como voce esta?
  - CONSULTA ID: CAIXA DE ENTRADA VAZIA - caso não haja nenhuma mensagem na caixa de entrada especificada; ERRO: CONTA ID NAO EXISTE - caso a conta não exista.
- Exemplo: CONSULTA 5

4 Quais TADs implementar?

Você deverá implementar pelo menos dois Tipos Abtratos de Dados (TADs): o primeiro para o email (ou mensagem) e o segundo para a caixa de entrada. Você precisará ainda de outros, que implementem estruturas de dados adequadas para armazenar emails na caixa de entrada e para armazenadas as caixas de entrada.

Lembre-se de que as mensagens devem ser armazenadas segundo a ordem de prioridade especificada, de tal maneira que uma consulta seja capaz de retornar a mensagem adequada (aquela de maior prioridade e que tenha sido recebida primeiro). Você deve armazenar somente tantas caixas de entrada quanto sejam as contas cadastradas.

Observações sobre a entrega:

(a) O programa deve ser organizado em módulos.

(b) O programa deve estar bem indentado e comentado

(c) O programa não deve fazer uso de comando goto nem de variáveis globais

(d) Caso apareçam números fixos no código, estes devem ser definidos como constantes.

(e) O trabalho pode ser feito em grupos de, **no máximo**, três pessoas.

(f) A parte de implementação do trabalho deverá ser entregue em um único arquivo compactado, com o nome dos integrantes (por exemplo, Fulano_de_Tal_Beltrano_de_Qual.zip). Indique, em um arquivo "leiame.txt", as instruções para compilação. projeto, envie o arquivo `.dev` também.

(g) Nesse zip não deve haver arquivos executáveis.

(h) Incluir pdf da parte escrita no zip.

(i) A entrega dos arquivos deverá ser feita via moodle e a fórmula para desconto por atraso na entrega é $\frac{2^{d-1}}{0{,}32}\%$, onde d é o atraso em dias. Note que após 6 dias, o trabalho não pode ser mais entregue. Ao final da descrição do trabalho, há outras informações disponíveis sobre a entrega.

(j) **Data de entrega: 25/04/2019**

(k) Valor: 10 pontos

O que deve ser entregue:

- Documentação do trabalho (impressa). Em entre outras coisas, a documentação deve conter:
  1. Introdução: descrição do problema a ser resolvido e visão geral sobre o funcionamento do programa.
  2. Implementação: descrição sobre a implementação do programa.
  3. Resultados e Discussões: dados obtidos das execuções dos algoritmos de ordenação sobre cada um dos conjuntos de dados (incluindo gráficos e tabelas); análise crítica dos dados obtidos em comparação com o esperado.
  4. Conclusão: comentários gerais sobre o trabalho e as principais dificuldades encontradas em sua implementação.
  5. Bibliografia: bibliografia utilizada para o desenvolvimento do trabalho, incluindo sites da Internet, se for o caso.
- Além disso, neste trabalho deve ser enviado ao professor o arquivo fonte. A entrega deverá ser feita via moodle, seguindo as diretrizes informadas no início da descrição deste trabalho.

Comentários Gerais:

- Comece a fazer este trabalho logo, enquanto o problema está fresco na memória e o prazo para terminá-lo está tão longe quanto jamais poderá estar.
- Clareza, identação e comentários no programa também vão valer pontos.
- Avaliarei com maior atenção ainda a parte escrita - incluindo erros de português -. Atenção especial será dada à análise crítica.
- Trabalhos copiados serão penalizados com a nota zero.
- Note que o grande desafio deste trabalho está na avaliação dos vários algoritmos nos diferentes cenários, e nao na implementação de código. Logo, na divisão de pontos, a documentação receberá 50% dos pontos totais. Uma boa documentação o deverá apresentar não ao somente resultados brutos mas também uma discussão dos mesmos, levando a conclusões sobre a superioridade de um ou outro algoritmo em cada cenário considerado, para cada métrica avaliada.