Redes de Computadores II: Primeiro Trabalho

Prof. Diego Passos, Universidade Federal Fluminense 2/2015

Descrição e Objetivo

Este trabalho consiste na implementação de um programa (ou um par de programas) que manipule um código de correção de erros baseado em paridade bi-dimensional, conforme visto em sala de aula. Seja na forma de um único programa parametrizável ou de um par de programas, a implementação deve contemplar:

- a codificação (*i.e.*, a inserção de bits de paridade) de um arquivo qualquer especificado pelo usuário;
- a decodificação (*i.e.*, a verificação de integridade, correção e remoção dos bits de paridade) de um arquivo codificado especificado pelo usuário.

A codificação deverá ser realizada da seguinte forma. O codificador deverá dividir o arquivo de entrada em blocos de 8 bytes (64 bits) consecutivos. Para cada bloco, seus bytes devem ser mapeados para linhas de uma matriz de bits **em ordem** (*i.e.*, o primeiro byte corresponde à primeira linha, o segundo byte corresponde à segunda linha e assim sucessivamente).

A partir desta disposição, para cada bloco deverão ser gerados **dois bytes de redundância**: o primeiro byte é composto pelas paridades de cada linha e o segundo byte é composto pelas paridades de cada coluna. A ordem dos bits em cada byte de paridade deve ser contada do bit mais significativo para o menos significativo. Isto é, a paridade da primeira linha da matriz deve ser armazenada no bit mais significativo do primeiro byte, enquanto a paridade da última linha deve ser armazenada no bit menos significativo.

Uma vez calculados, os dois bytes de paridade deverão ser acrescentados ao fluxo de dados **imediatamente antes** dos 8 bytes que compõem o bloco atual.

A Figura 1 ilustra o processo de codificação. Note que deve ser utilizada a **paridade par**, isto é, o número de bits iguais a 1 de uma linha ou coluna acrescida de seu bit de paridade correspondente deve ser par (em outras palavras, se uma linha ou coluna possui um número par de bits 1, seu bit de paridade será 0; o bit de paridade será 1, caso contrário).

Se o número de bytes no arquivo a ser codificado **não for múltiplo de 8**, o codificador deverá **completar matriz de paridade com linhas zeradas**. **Importante:** estes bytes zerados devem ser considerados **apenas para o cálculo da paridade**. Isto é, no arquivo codificado gerado como saída, deverão constar apenas os bits de paridade e os **bytes originais** do arquivo de entrada. Nenhuma outra informação ou cabeçalho deve ser incluída no arquivo codificado.

O processo de decodificação deverá ser feito da seguinte forma. O decodificador deverá ler o arquivo de entrada em **blocos de 10 bytes**: 2 bytes de paridade e 8 bytes de dados.

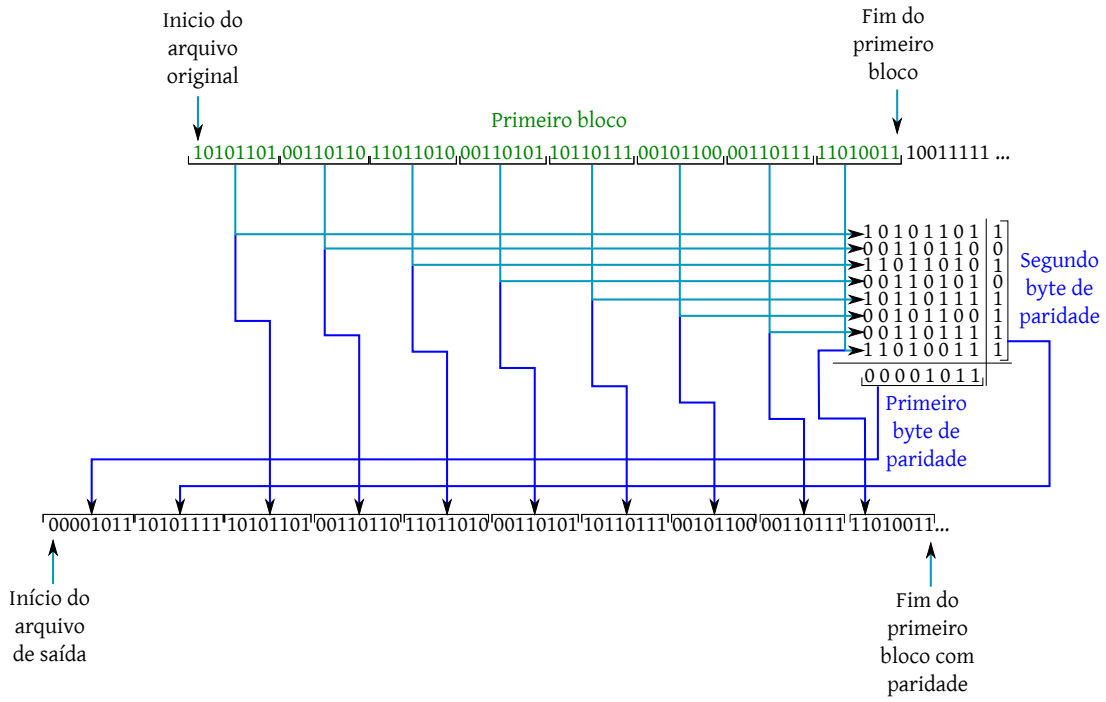

Figura 1: Ilustração do processo de codificação.

Os 8 bytes de dados deverão ser dispostos em uma matriz de paridade, na mesma sequência descrita para o codificador. Os bits de paridade deverão ser calculados e comparados aos bits que constam no arquivo de entrada.

Para cada bloco, o decodificador deverá:

1. Verificar se há erros no bloco, através da comparação dos bits de paridade.
2. Caso um ou mais erros sejam detectados, o decodificador deverá gerar um **aviso impresso na tela**.
3. Caso seja detectado **um único erro**, o decodificador deverá identificar o bit errado e restaurar o valor correto.
4. Caso o bloco esteja correto ou após uma **correção bem sucedida**, o bloco (correto ou corrigido) deverá ser escrito no arquivo de saída **sem os bits de paridade**.
5. Caso haja erros irrecuperáveis, o decodificador deverá ser **abortado, informando a causa da falha** ao usuário.

De forma análoga ao codificador, caso o número de bytes do arquivo de entrada passado ao decodificador não seja **múltiplo de 10**, deve-se assumir que as linhas faltantes da matriz de paridade devem ser completadas com bytes zerados. Assim como para o codificador, estes bytes zerados **não devem ser inseridos no arquivo de saída**, sendo usados apenas para efeito do cálculo da paridade. Note que o último bloco **não deve possuir menos que 3 bytes** (dois bytes de paridade e um byte de dados). Se um arquivo com esta característica for encontrado, o decodificador deverá abortar a execução informando a causa para o usuário.

O trabalho poderá ser feito em grupos de **até 4 alunos**. A linguagem de programação é de livre escolha de cada grupo, assim como detalhes de interface com o usuário (*e.g.*, interface gráfica, interface de linha de comando, obtenção dos nomes dos arquivos de entrada e saída). No entanto, toda a computação dos bits de paridade, verificação da integridade do arquivo e possíveis correções deverão ser implementadas pelo grupo. **Não serão aceitos trabalhos que utilizem bibliotecas/funções de terceiros para estas funcionalidades**.

Requisitos

A implementação deverá cumprir os seguintes requisitos:

1. possuir um módulo codificador e um módulo decodificador, seja em um único programa executável ou em dois programas separados;
2. obter os nomes dos arquivos de entrada e saída a partir do usuário (a maneira pela qual isso é feito não é importante);
3. o codificador deverá calcular os bits de paridade e inseri-los no arquivo de saída, conforme descrito na seção anterior;
4. o decodificador deverá calcular os bits de paridade, compará-los aos bits de paridade do arquivo de entrada, verificar a existência de erros e, se possível, corrigi-los;
5. o decodificador deverá reportar erros encontrados **mesmo que estes sejam recuperáveis**;
6. o decodificador deverá abortar a execução caso encontre um erro não recuperável, informando a causa ao usuário;
7. o decodificador deverá abortar a execução caso o último bloco possua menos que 3 bytes, informando a causa ao usuário;
8. ambos o codificador e o decodificador devem gerar os mesmos resultados independentemente do *endianess* do computador no qual são executados.

Data de Entrega

O prazo para a entrega do trabalho vai até o dia 14 de janeiro, às 22:00. O trabalho deverá ser entregue por e-mail, através do endereço `dpassos@ic.uff.br`. O e-mail deverá conter:

- identificador do trabalho (*e.g.*, “Trabalho 1”);
- lista dos integrantes do grupo;
- código fonte da implementação; e
- instruções de compilação/execução/uso da implementação.

Os e-mails de entrega de trabalho terão seus recebimentos devidamente confirmados. **É responsabilidade do grupo garantir que o trabalho seja recebido, aguardando pela confirmação e reenviando a mensagem caso não a recebam**.

Em caso de dúvidas ou correções relacionadas a esta especificação, também é responsabilidade de cada grupo entrar em contato (seja pessoalmente, ou através do mesmo endereço de e-mail) requisitando esclarecimentos **dentro do prazo de entrega do trabalho**.

Critério de Avaliação

Os trabalhos serão avaliados em uma escala de 0 a 10 pontos. A avaliação será dividida em duas partes:

- Aderência aos requisitos (até 1 ponto por requisito).
- Existência e qualidade das instruções de compilação/execução/uso da implementação (até 2 pontos).

A cada item avaliado, poderão ser atribuídas frações das pontuações máximas. Trabalhos entregues fora da data serão aceitos, **mas com uma penalidade de 1 ponto por dia (ou fração) de atraso**.