

## Procedimento de compactar

Primeiramente, pensei em fazer o compacta, pegando todos os bytes, juntando em um long long (8 bytes) e enviando-os de uma só vez, mas percebi que isso seria muito trabalhoso, então resolvi usar um long (somabyte) para guardar os bytes e enviá-los sempre que tivesse um byte completo.

Quando fui começar a escrever o programa, percebi que ia ter que saber quando a variável somabyte ia ter um byte completo, e que isso seguia padrões diferentes para cada bpp. Isso me desanimou bastante, pensei até em fazer um compacta para cada bpp, mas não seria muito eficiente. Quando já estava pensando em apagar tudo e refazer juntando tudo em um long long como tinha pensado no início, veio a brilhante ideia de criar uma variável para armazenar o tamanho em bits da variável que armazenava os bytes, pois assim bastava mandar o compacta enviar os bytes sempre que o tamanho fosse maior que 8.

Outros problemas que encontrei foram relacionados aos cálculos de formatar, juntar e enviar os bytes corretamente, para isso usei as operações bit a bit.

Nos comentários do arquivo 'c' há mais detalhes da função.

## Procedimento de descompactar

O processo de descompactar, pensei em fazer igualmente ao de compactar, com diferença apenas na hora de agrupar os bytes (o compacta cortava uma parte deles e o descompacta pega esses pedaços e preenche o que sobrou com zeros ou aleatórios).

Então do mesmo jeito que o anterior, ele pegava os bytes já preenchidos e juntava em um long, enviando sempre que ele enche (sempre que recebe um byte preenchido, pois já está completo).

Nos comentários do arquivo 'c' há mais detalhes da função.

## Descompactar com preenchimento aleatório

- a) As imagens preenchidas com zeros ficam mais “limpas” que as preenchidas com bits aleatórios. Para cada bpp, as imagens terão intensidades de cores diferentes, mas sempre bastante padronizadas e sem ruídos. Quanto menor o bpp, menor a qualidade da imagem, com menos “opções” de intensidade para o arredondamento das cores.
- b) As imagens preenchidas com zeros têm uma intensidade de cor média fixa e mais contínua, sem variações para cada tonalidade, com intensidades sempre inferiores às intensidades da imagem original.
- c) As imagens preenchidas com valores aleatórios ficam com uma textura mais descontínua, com ruído nas cores. Quanto menor o bpp, menor a qualidade da imagem, tendo cada vez mais ruído.
- d) O preenchimento com valores aleatórios dos bits nos pixels faz com que a intensidade dos pixels seja mais descontínua, bem diferente do preenchimento com zeros. Cria intensidades bem diferentes da intensidade média, mas cria uma intensidade média mais próxima da intensidade da imagem original.