



## TRABALHO PRÁTICO: SISTEMA DE GERENCIAMENTO DE DISCO

Curso: Engenharia de Produção/Sistemas de Informação

2º semestre de 2018

Disciplina: Algoritmos e Estruturas de Dados I

Data: 08/10/18

Professor: Alexandre Magno de Sousa

Valor: 35 pontos

1. Este trabalho trata do problema de um sistema de gerenciamento de arquivos em disco. Por exemplo, imagine o disco como um array unidimensional chamado **disco** de tamanho igual a **numeroSetores**, em que **tamanhoSetor** indica o número de caracteres armazenados em um setor (Para fins de depuração, teste-o com um número pequeno para ajudá-lo do rastreamento de código). Uma área de setores disponíveis é mantida em uma lista ligada chamada **setores** de estruturas de três campos: dois campos para indicar intervalos de setores disponíveis e um campo **próximo**. Os arquivos são mantidos numa lista ligada **arquivo** de estruturas de quatro campos: nome do arquivo, número de caracteres no arquivo, um ponteiro para uma lista ligada de setores, onde o conteúdo do arquivo pode ser encontrado e o campo **próximo**.
  - (a) Na primeira parte, implemente funções para gravar e para remover arquivos. A gravação de arquivos exige requisitar um suficiente de setores da **area**, se disponível. Os setores não precisam ser contíguos, assim a lista atribuída ao arquivo pode conter diversas células. O conteúdo dos arquivos tem que ser escrito para os setores atribuídos aos arquivos. A remoção de um arquivo exige somente que se removam os nós que correspondam a esse arquivo (um de **arquivos** e o resto de sua própria lista de setores) e que se transfiram os setores atribuídos a esse arquivo de volta a **area**. Nenhuma mudança é feita em **disco**.
  - (b) A fragmentação do arquivo torna vagorosa a sua recuperação. Na situação ideal, um grupo de setores é atribuído a um arquivo. No entanto, depois de muitas operações com os arquivos, isso pode não ser possível. Estenda o programa a fim de incluir uma função **junta()** para transferir arquivos aos setores contíguos, isto é, para criar a situação ilustrada na Figura 1. Os arquivos fragmentados **arquivo1** e **arquivo2** ocupam somente um grupo de setores depois que **junta** é terminado. No entanto, um cuidado especial deve ser tomado para não sobrescrever setores ocupados por outros arquivos. Por exemplo, **arquivo1** exige oito setores; cinco setores estão livres no começo de **area**, mas os setores 5 e 6 estão ocupados por **arquivo2**. Em consequência, um arquivo *f* que ocupa tais setores tem que ser localizado primeiro, varrendo **arquivos**. O conteúdo desses setores precisa ser transferido para posições desocupadas, que exigem atualizar os setores que pertencem a *f* na lista vinculada; somente então podem os setores liberados serem utilizados. Um modo de realizar isso é copiar, da área dentro da qual um arquivo é copiado, pedaços de setores de outro arquivo dentro de um área do disco grande o suficiente para acomodar esses pedaços. No exemplo da Figura 1, o conteúdo de **arquivo1** é primeiro copiado para os setores de 0 a 4, e então a cópia é temporariamente suspensa porque o setor 5 está ocupado. Assim, o conteúdo dos setores de 5 e 6 é movido para os setores 12 e 14, e a cópia de **arquivo1** é retomada.

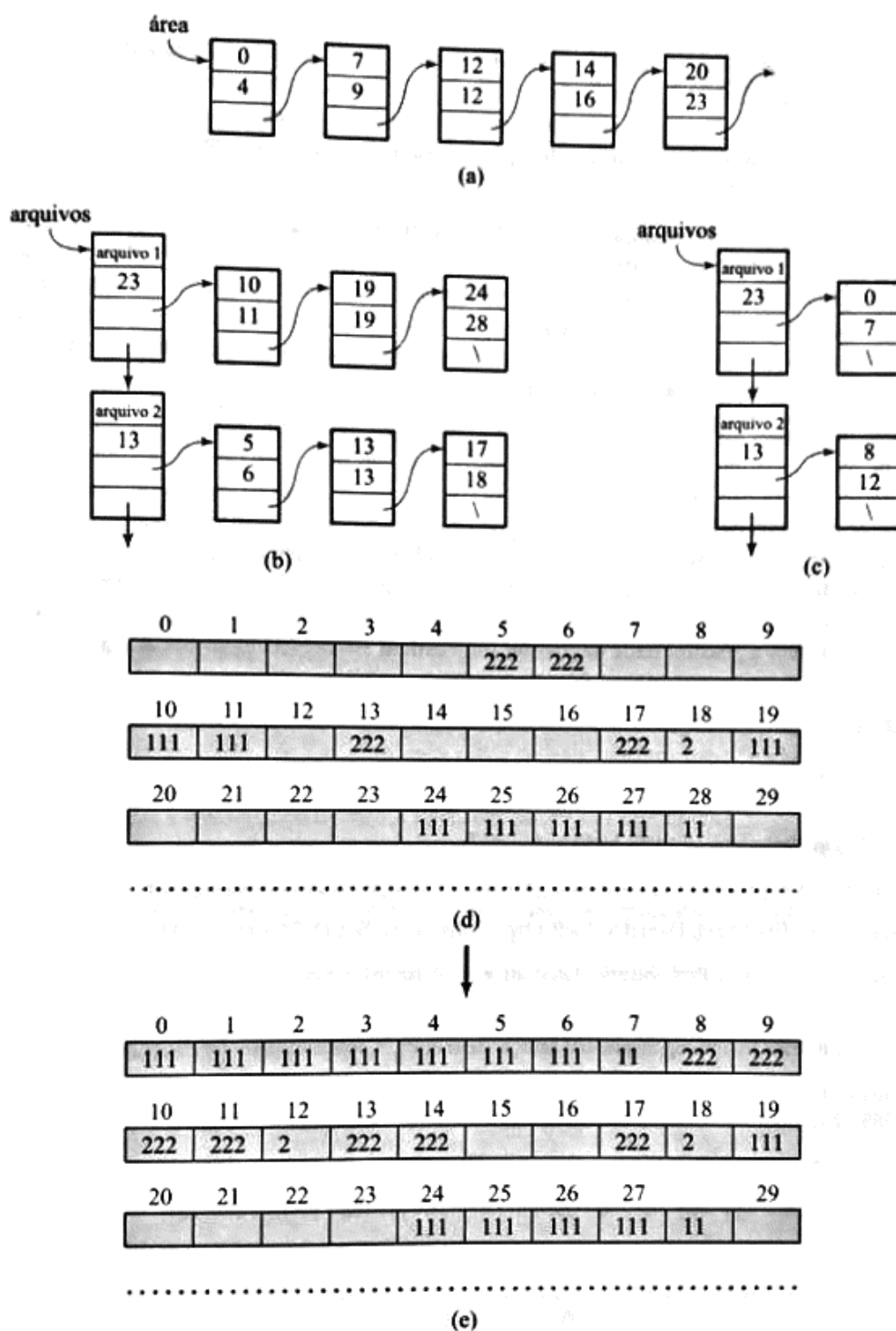


Figura 1: Listas ligadas utilizadas para alocar setores de disco para arquivos: (a) uma área de setores disponíveis; dois arquivos (b) antes e (c) depois de colocá-los em setores contíguos; a situação em setores do disco (d) antes e (e) depois dessa operação.