

Lista de Exercícios 1

ed20041 – Fabrício Vargas Matos

Fonte: Algumas questões foram extraídas de provas aplicadas na PUCRS

1. Considerando um sistema de gerenciamento de impressão com as seguintes características:
 - Existência de 2 filas de impressão, uma para trabalhos com alta prioridade e outra para trabalhos com baixa prioridade;
 - Para a fila de alta prioridade considerar que nunca poderão existir mais de 30 trabalhos na fila;
 - Para a fila de baixa prioridade não há limite para o número de trabalhos na fila;
 - O gerenciador de impressão só realiza a impressão de um trabalho da fila de baixa prioridade quando não existir nenhum trabalho na fila de alta prioridade.

Definir as estruturas de dados mais adequadas para implementação das filas e escrever os algoritmos dos seguintes módulos do sistema:

- a) Procedimento para impressão de trabalhos – retirada de trabalhos das filas e envio para a impressora. Neste procedimento o algoritmo deverá possuir um laço infinito no qual é executada a função `impressora_disponível()`, já existente, que retorna Verdadeiro sempre que a impressora estiver pronta para impressão de um novo trabalho. Quando a função `impressora_disponível()` retornar Verdadeiro, o procedimento a ser implementado deverá remover um trabalho da fila apropriada (se existir algum trabalho aguardando para impressão) e utilizar o procedimento `envia_imp(nome do arquivo)`, também já existente, para simular o envio do trabalho para a impressora.
- b) Procedimento para inclusão de trabalhos nas filas de impressão. Este procedimento recebe como parâmetros o nome do arquivo a ser impresso, a identificação do "proprietário" do arquivo e a prioridade de impressão do mesmo. A partir destes dados uma solicitação para impressão deverá ser colocada em uma das filas de impressão e o procedimento termina (retorno ao programa chamador).

Não esquecer de definir todos os tipos e variáveis necessários para implementação dos dois procedimentos.

2. Escrever um algoritmo recursivo para determinar a altura de uma árvore binária já construída. O algoritmo deverá receber um ponteiro para a raiz da árvore e deverá retornar sua altura. Apresente uma relação de recorrência para o tempo do algoritmo e resolva a recorrência.
3. Escreva um algoritmo de inserção ordenada em uma lista duplamente encadeada (não é permitido inserir em qualquer posição e reordenar). Apresente um invariante do loop e a pós condição do programa e prove sua correção. Lembre que a pos-condição deve garantir que o novo elemento foi adicionado à seqüência e que ela permanece ordenada.
4. Escreva um algoritmo para determinar se duas árvores binárias são idênticas (conteúdo dos nodos e posições). Caso exista alguma diferença entre as árvores, o algoritmo deverá fornecer uma ou mais mensagens de erro.