

Pesquisa Torre de Hanói

Aluno: Lucas Ângelo O. M. Rocha

Explicação da solução em C da Torre de Hanói:

Resumo - Na solução, foi utilizada recursividade, para calcular qual o mínimo de movimentos necessários e informar quais seriam eles, com objetivo de solucionar o quebra cabeça Torre de Hanói. Nesse programa, foram utilizados um número fixo de 3 pinos, e um dado número de discos informados pelo usuário.

Explicação do programa - No início, foram declaradas as bibliotecas necessárias, logo em seguida, foi criada a variável "cont" recebendo 0, que será responsável por contar o número de movimentos gerados pela função recursiva, posteriormente, também foi declarado o protótipo da função recursiva trocarDiscos() que será utilizada mais a frente.

Por conseguinte, a criação da função main(), que é responsável por guardar a quantidade de discos que o usuário deseja, na variável "discos", a main() também possui a tarefa de solicitar a função trocarDiscos(), passando (discos, 'A', 'C', 'B'). O "discos", é a quantidade que o usuário digitou, e "A", "C", "B" são os nomes das torres/pinos passados como char.

Com isso, a função void trocarDiscos() é acionada, recebendo os valores primeiramente da main() de (int discos, char origem, char destino, char auxiliar), onde o "int discos" recebe o valor informado pelo usuário, e as variáveis de caracteres recebem da main() 'A' como origem dos discos, 'C' como o destino, e 'B' como auxiliar para movimentos dos discos. Logo em seguida, é criada a condição de parada [if (discos==1)], para a função recursiva finalizar, pois quando o número de discos chegar em 1, não será mais possível continuar efetuando a recursividade da função. Então, será somado o último contador para a variável "cont", e será escrito na tela o número do último movimento, junto com como deve ser o último movimento efetuado, feito por meio dos últimos valores assumidos pelas variáveis "discos", "origem" e "destino".

Portanto, faz-se necessário a utilização do "else" para o verdadeiro funcionamento da recursividade, com isso, é solicitado a própria função trocarDiscos(), agora recebendo a variável "discos" subtraído 1 valor, pois um movimento foi executado, recebendo novamente as variáveis caracteres, porém em ordem diferente, "origem" primeiro, "auxiliar" em segundo, e "destino" em terceiro, trocando assim o pino auxiliar que era antes o último para o segundo lugar, de modo agora, somando mais um valor para a "cont", e sendo possível escrever na tela novamente o movimento que deverá ser efetuado, feito *por meio dos últimos valores assumidos pelas variáveis* "discos", "origem" e "destino".

Dessa maneira, é solicitado novamente a função trocarDiscos(), recebendo a variável "discos" subtraído 1 valor, agora com a variável "auxiliar" ocupando a primeira posição, "destino" na segunda e "origem" última. Por consequência, foi *possível oscilar todas as posições das variáveis caracteres* (que são as torres/pinos), assim, a função trocarDiscos() é reiniciada, até que o valor de discos assuma 1, e ela finalize.

Exemplo 3 discos:

Memória RAM
trocarDiscos(1, B, C, A)
trocarDiscos(2, A, B, C)
trocarDiscos(3, A, C, B)

