

# Terceiro Trabalho de Oficina de Computação (2013/1)

## Atenção:

Este trabalho é obrigatório e deverá ser entregue, impreterivelmente, até o dia 23 de julho de 2013. A solução pode ser elaborada em grupos de dois alunos e deverá ser arquivada no diretório “~<professor>/CI067/” onde “~<professor>” deve ser “~alex” (para a TURMA C2) ou “~nicolui” (para a TURMA A). O nome do arquivo terá como prefixo o nome-de-usuário de um dos alunos do grupo no sistema do laboratório e, como extensão, “.tar.gz” para indicar que seu conteúdo possui arquivos empacotados (pelo comando **tar**) e comprimidos (pelo comando **gzip**).

Assim, por exemplo, se você é aluno da Turma C2 e o seu nome de usuário no sistema é grs12 então o nome do arquivo será grs12.tar.gz (dentro do diretório “~alex/CI067/”). Siga as instruções abaixo para entender quais arquivos serão empacotados e comprimidos:

1. Faça tantos arquivos com os programas-fonte que você preferir para permitir a um autor a criação de uma prova de vestibular (ver mais detalhes abaixo), mas um dos arquivos será chamado **grs12-autor.c** e garanta que o código-fonte dos arquivos será compatível com o compilador gcc (GNU C) da servidora de processamento **macalan** do Departamento de Informática;
2. Gere automaticamente um arquivo texto padrão com os dados resultantes da execução do programa de autoria do item anterior (ver mais detalhes abaixo);
3. Faça tantos arquivos com os programas-fonte que você preferir para permitir a um candidato a realização da prova criada pelo autor (ver mais detalhes abaixo), mas um dos arquivos será chamado **grs12-realizador.c**;
4. Coloque os três arquivos acima em um diretório chamado **T3-grs12**;
5. Empacote e comprima esse diretório com os comandos abaixo para criar o arquivo **grs12.tar.gz** assim:

```
tar cf grs12.tar T3-grs12
gzip grs12.tar
```

6. Finalmente, durante a aula anunciada pelo professor, copie o arquivo **grs12.tar.gz** para “~<professor>/CI067/” tal como consta acima.

Não se esqueça de proteger completamente o arquivo criado, de maneira a permitir a leitura do mesmo apenas por você! Isso deve ser feito aplicando **chmod og-rwx grs12.tar.gz** antes de efetuar a cópia com a preservação das permissões (**cp -p grs12.tar.gz ~alex/CI067/**).

Não se preocupe com as permissões do professor que irá corrigir o trabalho. A correção dos trabalhos será parcialmente automatizada, sendo assim, é importante que todos os arquivos com as soluções individuais estejam no diretório citado acima, dentro do prazo estipulado. O procedimento de entrega acima deverá ser feito em sala de aula, na presença do professor, nos dias estipulados no calendário de avaliações de cada turma. Não será permitida a entrega do arquivo por e-mail.

## 1 Descrição geral

O objetivo desse trabalho é construir um pequeno sistema de provas de vestibular com múltiplas opções, assim como a prova aplicada no vestibular da UFPR.

O sistema será basicamente composto de dois programas em linguagem C. O primeiro deles deverá ser capaz de permitir a definição e o armazenamento permanentemente de questões em arquivos. O segundo programa do sistema, quando executado, deverá permitir a realização da prova por parte do aluno. As questões de uma prova deverão ser mantidas em uma lista encadeada, tanto em modo de criação/arquivamento quanto em modo de realização.

Cada questão deve ter a seguinte forma:

Enunciado: Xxxxx xxx xxx ...

- 1) xxxx
- 2) xxxx
- 4) xxxx
- 8) xxxx
- 16) xxxx
- ...
- i) xxxx

Resposta: <<valor igual ao somatório das respostas corretas.>>

Ou seja, a organização das questões deve seguir uma codificação em potências de 2, cuja numeração deve ser gerada automaticamente, de maneira a seguir sempre uma nova ordem aleatória de apresentação das opções a cada vez que a prova é realizada por um aluno diferente. A resposta de cada questão é calculada dinamicamente, de acordo com a ordem aleatória da carga, representando a soma das opções corretas.

## 2 Modos de Operação

O sistema deve ter dois modos de operação: Autoria e Execução, sendo representados por dois programas distintos.

### 2.1 Autoria (criação de novas questões)

Nesse modo, o sistema seria operado pela pessoa que elabora a prova. As seguintes funcionalidades devem estar disponíveis:

- incluir questão: Permite incluir uma questão nova. A inclusão deve pedir o enunciado e permitir editar as alternativas. A edição das alternativas consiste em:
  - inserir nova alternativa requisita no texto e a informação se ela é ou não correta;
  - remover alternativa existente;
  - mudar o texto e/ou a resposta da alternativa.
- Não deve ser necessário especificar o número da alternativa. O sistema deverá ser capaz de gerar automaticamente a numeração das opções, de acordo com a ordem das mesmas.
- Ainda, o sistema deve calcular automaticamente qual a soma correta que especifica a resposta da questão.
- As questões devem ser armazenadas em uma lista encadeada, e cada questão deve ter uma lista encadeada com as opções das questões. A informação de que a opção é correta deve estar armazenada na estrutura da \*opção\* e não na estrutura da questão.
- alterar questão: permite mudar o texto do enunciado e/ou editar as alternativas. A edição de alternativas deve funcionar tal como descrito para inclusão de questão.
- remover questão: remove questão existente, inclusive as alternativas relacionadas.
- salvar prova em arquivo: dado o nome de um arquivo, deve gravar a prova em um arquivo. O formato do arquivo será especificado posteriormente.
- carregar prova de arquivo: dado o nome de um arquivo, carrega as questões do arquivo.

### 2.2 Execução (realização da prova)

No modo de execução, o sistema carrega um arquivo de prova e apresenta (a um vestibulando) as questões sempre numa sequência aleatória, a qual pode ser diferente das sequência com que elas foram armazenadas pelo criador da prova.

Cada questão deve ser apresentada com enunciado e as opções devidamente numeradas segundo uma seqüência também aleatória. O usuário deverá fornecer a soma que julga ser a resposta correta e o sistema vai calcular a nota para cada questão, a qual será baseada na soma correta (do gabarito) e no valor informado para o usuário.

Quando o usuário (vestibulando) terminar a prova ou fornecer -1 como resposta, o sistema pára de apresentar questões e apresenta a nota do usuário.

## 3 Cálculo das notas

A nota de cada questão deve ser a seguinte:

- se pelo menos 2/3 das opções foram respondidas corretamente e não há \*nenhuma\* resposta incorreta, a nota é:  $1.0 * (nOpcoesRespCorretamente / nOpcoesCorretas)$
- senão, a nota é 0.

A nota da prova será o somatório de todas as notas dividido pelo número de questões e multiplicado por 10.

## 4 Formato do arquivo de prova

ATENÇÃO: Os detalhes deste item serão apresentados em sala de aula. Um exemplo de arquivo encontra-se no arquivo [http://www.inf.ufpr.br/alexnd/ofcomp/questoes\\_da\\_prova.txt](http://www.inf.ufpr.br/alexnd/ofcomp/questoes_da_prova.txt)