EP3 - CCMI - 2007 - v. 1.1
Editor de Meta-Dados de MP3

Paulo J. S. Silva

Entrega 12 de dezembro de 2007

1 Descrição

Nesse EP, você deve implementar o embrião de um editor de meta-dados (*tags*) de arquivos MP3. O programa será muito simples, mas deve estar claro ao final de seu trabalho qual o caminho que poderia ser tomado para escrever um bom editor desse tipo.

A idéia é bastante simples. Vocês devem escrever um programa capaz de atualizar os meta-dados de aquivos MP3 a partir do nome do arquivo e também de fazer o contrário: renomear o arquivo a partir dos valores armazenados nos meta-dados.

Para isso são necessárias duas informações:

1. Quais são os campos presentes no nome do arquivo, quando se deseja extrair essa informação. Ou o contrário, ou seja, quais campos devem compor o novo nome de arquivo.

2. Qual é o separador dos campos que deve ser usado nos nomes de arquivo.

Por exemplo, considere que o diretório atual contém três arquivos MP3 com os nome abaixo:

```
Ellen Allie & Apparat - Orchestra of Bubbles - 01 - Turbo Dreams.mp3
Ellen Allie & Apparat - Orchestra of Bubbles - 02 - Way Out.mp3
Ellen Allie & Apparat - Orchestra of Bubbles - 05 - Jet.mp3
```

Como vocês podem imaginar, os campos presentes nos nome acima são, em ordem, nome do artista / grupo, nome do disco, número da faixa e nome da música. Alem disso o separador é `" - "`. Usando essa informação é fácil escrever um programa em Python que extrai os campos para atualizar os meta-dados, já que ele sabe qual é a posição de cada campo e qual é o separador.

Além disso, considerado que sabemos como ler os campos dos meta-dados de um arquivo MP3 e que é possível renomear arquivos para uma cadeia de caracteres qualquer, deve ser fácil escrever um programa que usa a informação presente nos campos para gerar nomes como os apresentados acima e atualiza a informação do arquivo.

1.1 Reaproveitando código: bibliotecas

Para fazer o EP você precisa então descobrir como realizar algumas tarefas que parecem não triviais. Você deve fazer uso de bibliotecas escritas por outros que executam as tarefas mais difíceis. Essa é a principal mensagem do EP: *antes de sair escrevendo código, procure se alguém já não resolveu pelo menos parte de seu problema e disponibilizou o seu código em formato de biblioteca.*

Para lidar com arquivos, o seu EP deve usar a biblioteca `os` que já vem com o Python. Já para lidar com os meta-dados de arquivos MP3 use a biblioteca mutagen disponibilizada em `http://www.sacredchao.net/quodlibet/wiki/Development/Mutagen` (quem usa ubuntu pode instalar o pacote python-mutagen). Dêem uma olhada particular no `EasyId3`.

2 Interface

o seu programa será um programa de linha-de-comando. Para rodá-lo o usuário deve ir, em um terminal, para o diretório onde estão os arquivos de música e então chamar o programa de nome `meta.py`[1].

Já a informação sobre qual é a tarefa (alterar o nome do arquivo ou atualizar os meta-dados), quais são os campos e qual é separador deve ser repassada como opção ao programa.

Por exemplo, se queremos que o programa altere os meta-dados a partir dos nomes de arquivo acima devemos usar a seguinte chamada:

```
python meta.py --acao=meta --campos="artista,album,faixa,titulo" --sep=" - "
```

Os parâmetros devem ter os nomes descritos acima.

Já para fazer a operação inversa (re-nomear os arquivos):

```
python meta.py --acao=nome --campos="artista,album,faixa,titulo" --sep=" - "
```

O opção `--acao` define qual a tarefa que deve ser executada, podendo valer `meta` ou `nome`. Se ela não estiver presente, então o valor padrão deve ser `nome`. Já os parâmetros `--campos` e `--sep`, que também são opcionais, devem valer `"faixa,titulo"` e `" - "` quando omitidos. Os possíveis valores de campos são álbum, título, data, gênero, faixa e artista. Eles devem aparecer sem acentos separados por vírgulas apenas.

Use o módulo `optparse`, que já vem com o Python, para passar essas capturar as opções descritas na linha de comando dentro de seu programa.

Uma confusão que pode ocorrer no EP é a presença de acentos. Para facilitar a nossa vida, evitando o problema que pode ocorrer com as diferentes codificações de acentos, você pode assumir que todas as cadeias de caracteres, assim como os nomes de arquivos, não possuem acentos.

[1] Note que o nome `meta.py` é obrigatório

3 Considerações finais

Você deve organizar bem o seu programa, inclusive usando (várias) funções. Não é obrigatório o uso de classes, mas eu creio que isso irá facilitar bem o programa. Procure desenvolver o código de forma sistemática.

Além disso, se você sentir necessidade, comente o seu código. Para novatos, como vocês, é melhor pecar pelo excesso de comentários do que pela falta destes. No *mínimo* comente cada uma das funções da forma descrita em classe.

Ainda, o seu arquivo deve conter um cabeçalho contendo o nome do arquivo, o nome do autor e seu número USP e uma breve descrição dizendo o que o programa faz.

Por fim, lembrem de usar o fórum de discussão na página da disciplina. Isso é fundamental. No fórum posso disponibilizar correções no enunciado ou esclarecer dúvidas e definir detalhes que passaram despercebidos.

Não custa lembrar que o EP é um trabalho *individual*. Você pode discutir com os colegas o programa mas nunca trocar trechos de código. Sugiro nem mesmo olhar código do colega, apenas conversar. Caso encontremos trabalhos com trechos iguais os dois alunos receberão 0 no EP e a média final de EPs ainda receberá um desconto extra de 10%.