

Classificando imagens dos Simpsons

Leonardo Pontes Baiser¹, Marco Cezar Moreira de Mattos¹,
Rômulo Manciola Meloca¹

¹DACOM – Universidade Tecnológica Federal do Paraná (UTFPR)
Caixa Postal 271 – 87301-899 – Campo Mourão – PR – Brazil

{lpbaiser, marco.cmm,rmeloca}@gmail.com

Abstract. *This report shows the development process of a *****, since the conception, showing the designed solution until its implementation phase. This report shows project decisions and outlined obstacles.*

Resumo.

1. Introdução

2. O Problema

3. Organização da Solução

3.1. Diagramação

4. Implementação

Dado as bibliotecas já implementadas, como mencionado na seção anterior, desenvolveu-se a solução com a linguagem de programação C.

Além das abstrações já citadas, para facilitar o desenvolvimento e torná-lo mais natural, tornando-o mais próximo ao conceito de objetos, utilizou-se apelidos característicos de objetos para estruturas já nomeadas (como por exemplo *Connection* ao invés de *connection_t*) e definições de métodos redundantes (como por exemplo *sendPackage()* em contraposição à *CONN_send()*). Tais fatores, permitiram o encapsulamento de algumas funcionalidades e possibilidade de fácil e rápida manutenção/refatoração do código.

Tendo sido mapeado os *includes* no diagrama de classes, facilmente pode-se visualizar os módulos que os programas *server* e *client* deveriam incluir, mantendo a coerência com a distância que as partes (naturalmente) deveriam ter, com excessão da interface que os conecta, os pacotes definidos na biblioteca *package.h* (que já conta com a inclusão da biblioteca *connection.h*).

Colaborou-se o código com o auxílio do controle de versões git, onde o integrante Rômulo responsabilizou-se pela implementação do arquivo *server.c* e das duas *threads requestHandler.c* e *worker.c*. O núcleo da aplicação foi desenvolvido de maneira conjunta e interativa em relação as partes. O integrante Marco responsabilizou-se por todo o arquivo *client.c* além do *upload* do arquivo em *multi-parts*.

5. Considerações Finais

Considera-se, por fim, que o desenvolvimento de um projeto que conta com várias *threads*, o conceito de produtor-consumidor, um *buffer* compartilhado onde apresentam-se condições de corrida bem específicas, a integração de vários módulos além da comunicação inter-processos via *socket*, permite no mínimo alargar os conhecimentos e fixar o aprendizado de todos esses conceitos vastamente utilizados nas mais diversas aplicações atuais. Nesse espectro, salienta-se a importância de tal desenvolvimento e, sobretudo, a fase de projeto, que tanto agrega para a visualização panorâmica deste ponto em específico da disciplina de Sistemas Operacionais.

Conclui-se que aplicações de determinada escala demandam a produção de vários artefatos, que somente são produzidos após o fiel debruçar-se nas ideias e adiantar-se a respeito de todos os problemas que são solucionados e gerados a partir delas. Toma-se como proveito o pensar em soluções que possam ser integradas a demais programas e o pensar em abstrações e interfaces que possam ser escaladas e reutilizadas.

Exemplo de referência CC01a [1].

Referências

- [1] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.