

Modularização de aplicações utilizando biblioteca de vínculo dinâmico, Dynamic-link library (DLL)

Luiz Lino Bertanha de Abreu¹

Adauto Luiz Mancini²

Luis Gustavo Barioni²

Fortemente aderente ao projeto Pecus³ e Animal Change⁴, o documento apresenta uma proposta de modularizar aplicações utilizando biblioteca de vínculo dinâmico (DLL). É proposta uma maneira para que as aplicações de um modo geral possam ter as suas funcionalidades modularizadas em componentes separados usando-se DLLs. Como os módulos são separados, podem ser carregados para o programa principal em tempo de execução. Isto torna o tempo de carregamento do programa mais rápido porque um módulo é carregado somente quando sua funcionalidade é solicitada.

O desenvolvimento de atualizações é mais fácil e estas se aplicam a cada módulo sem afetar outras partes do programa. Por exemplo, o projeto Pecus visa avaliar efeitos de Gases de Efeito Estufa (GEE) de emissão pecuária, para isso é necessário o desenvolvimento de vários modelos de sistemas biofísicos. Cada processo biofísico é direcionado por uma equipe diferente de pesquisadores que avaliam, calibram e modificam a estrutura dos modelos dos componentes. É preciso integrar os componentes desses modelos fazendo a ligação das entradas e saídas de dados dos modelos que foram desenvolvidos de forma independente. Como os modelos podem ser implementados em bibliotecas independentes de vínculo dinâmico, é possível aplicar uma atualização em um dado modelo sem a necessidade de criar ou

¹ Fatec/Americana - lino.bertanha.br@gmail.com

² Embrapa Informática Agropecuária - {adauto.mancini, luis.barioni}@embrapa.br

³ Disponível em: <<https://www.agropediabrasilis.cnpia.embrapa.br/web/pecus>>. Acesso em: 20 set. 2013.

⁴ Disponível em: <<http://www.animalchange.eu/>>. Acesso em: 20 set. 2013.

instalar o programa novamente. Assim, é possível utilizar código e dados que podem ser usados por mais de um programa ao mesmo tempo, promovendo a reutilização de código e o uso eficiente de memória.

Dois tipos de funções, exportado e interno, podem ser usados na criação de uma biblioteca de vínculo dinâmico para a declaração de funções (MICROSOFT DEVELOPER NETWORK, 2013a). As funções exportadas destinam-se a serem chamadas por outros módulos, no programa principal ou em outras bibliotecas, bem como a partir do interior da DLL onde elas são definidas. As funções internas geralmente são destinadas a serem chamadas apenas de dentro da DLL onde são definidas. Para exportar funções da DLL, pode-se adicionar uma palavra-chave para a DLL ou criar um arquivo de definição (.def) do módulo que lista as funções DLL exportadas.

Para usar a palavra-chave, deve-se declarar em cada função que se deseja exportar com a seguinte palavra-chave: `__declspec(dllexport)`

Exemplo: `void __declspec(dllexport) CalcularEmissaoNitrog(void);`

Para usar as funções DLL exportadas no aplicativo, deve-se declarar cada função que se deseja importar com a seguinte palavra-chave: `__declspec(dllimport)`

Exemplo: `void __declspec(dllimport) CalcularEmissaoNitrog(void);`

O modelo crescimento e composição de bovinos da UCDavis (OLTJEN et al., 1986) foi implementado no *framework* MacSim (em fase de elaboração)⁵ e compilado em uma DLL (MICROSOFT DEVELOPER NETWORK, 2013b). Também foi implementada uma interface gráfica em Qt, que carrega a biblioteca dinâmica contendo o simulador e permite executá-lo. A interface tem as opções de construir o simulador, executar o simulador de forma corrida ou passo a passo, e mostrar os valores de entrada ou de saída. Dentre os principais resultados obtidos, observar-se a economia de memória, reduzindo de forma efetiva a troca de memória, visto que muitos processos podem usar uma única DLL simultaneamente, compartilhando uma única cópia da DLL na memória. Também poupa espaço em disco, pois muitos aplicativos podem compartilhar uma única cópia da DLL no disco. Atualizações para a

⁵ MANCINI, A. L.; BARIONI, L. G.; LIMA, H. N.; SANTOS, J. W.; SILVA, R. D. R.; SANTOS, E. H.; DIAS, F. R. T. Arcabouço para desenvolvimento de simuladores de sistemas dinâmicos contínuos e hierárquicos. Em processo de edição pela Embrapa Informática Agropecuária, 2014.

DLL são mais fáceis porque, quando as funções em uma DLL são alteradas, os aplicativos que as utilizam não precisam ser recompilados ou vincularem novamente argumentos de função. Oferece suporte a programas multilíngues: programas escritos em diferentes linguagens de programação podem chamar a mesma função da DLL.

Gostaria de agradecer à Embrapa por ter me proporcionado a oportunidade de desenvolver este trabalho. Agradeço, de forma particular, aos pesquisadores Adauto Luiz Mancini e Luis Gustavo Barioni pelo apoio e orientação durante o desenvolvimento do projeto.

Referências

MICROSOFT DEVELOPER NETWORK. 2013a. **Creating a simple dynamic-link library**. 2013. Disponível em: <[http://msdn.microsoft.com/en-us/library/ms682507\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms682507(v=vs.85).aspx)>. Acesso em: 12 set. 2013.

MICROSOFT DEVELOPER NETWORK. 2013b. **Using shared memory in a dynamic-link library**. 2013. Disponível em: <[http://msdn.microsoft.com/en-us/library/ms686958\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/ms686958(v=vs.85).aspx)>. Acesso em: 20 set. 2013.

OLTJEN, J. W.; BYWATER, R. L.; BALDWIN, R. L.; GARRET, W. N. Development of a dynamic model of beef cattle growth and composition. **Journal of Animal Science**, Champaign, v. 62, n. 1, p. 86-97, Jan. 1986.