

Anterior:[Ganchos para Carregamento](#), Acima:[Carregando](#) [[Conteúdo](#)][[Índice](#)]

## 16.11 Módulos dinâmicos do Emacs

Um *módulo dinâmico do Emacs* é uma biblioteca compartilhada que fornece funcionalidade adicional para uso em programas Emacs Lisp, assim como um pacote escrito em Emacs Lisp faria.

Funções que carregam pacotes Emacs Lisp também podem carregar módulos dinâmicos. Eles reconhecem módulos dinâmicos observando sua extensão de nome de arquivo, também conhecida como “sufixo”. Este sufixo depende da plataforma.

### Variável: *módulo-arquivo-sufixo*

Essa variável contém o valor dependente do sistema da extensão de nome de arquivo dos arquivos do módulo. Seu valor é `.assim` em hosts POSIX e `.dll` em MS-Windows.

Cada módulo dinâmico deve exportar uma função chamada C chamada `emacs_module_init`, que o Emacs chamará como parte da chamada `load` que requer que carregará o módulo. Deverá também exportar um símbolo nomeado `plugin_is_GPL_compatible` para indicar que seu código está liberado sob a GPL ou licença compatível; O Emacs sinalizará um erro se o seu programa tentar carregar módulos que não exportam esse símbolo.

Se um módulo precisar chamar funções do Emacs, deve fazê-lo através da API (Application Programming Interface) definida e documentada no arquivo de cabeçalho `emacs-module.h` que faz parte da distribuição do Emacs. Consulte [Escrevendo Módulos Dinâmicos](#), para obter detalhes sobre como usar essa API ao escrever seus próprios módulos.

Os módulos podem criar `user-ptr` objetos Lisp que incorporam ponteiros para estruturas C definidas pelo módulo. Isso é útil para manter estruturas de dados complexas criadas por um módulo, para serem passadas de volta para as funções do módulo. Objetos `User-ptr` também podem ter *finalizadores* associados – funções a serem executadas quando o objeto é GC'ed; isso é útil para liberar quaisquer recursos alocados para a estrutura de dados subjacente, como memória, descritores de arquivos abertos, etc. Consulte [Valores do módulo](#).

### Função: *objeto user-ptr*

Esta função retorna o seu argumento `ptr` como um `user-ptr` objeto.

### Função: *arquivo de carregamento do módulo*

O Emacs chama essa primitiva de baixo nível para carregar um módulo do *arquivo* especificado e realizar a inicialização necessária do módulo. Esta é a primitiva que garante que o módulo exporte o `plugin_is_GPL_compatible` símbolo, chame a `emacs_module_init` função do módulo e sinaliza um erro se essa função retornar uma indicação de erro ou se o uso for digitado C-g durante a inicialização. Se a inicialização for bem-sucedida, `module-load` retornará `t`. Observe que o *arquivo* já deve ter a extensão de nome de arquivo adequada, pois essa função não tenta procurar arquivos com extensões conhecidas, ao contrário de `load`.

Ao contrário de `load`, `module-load` não grava o módulo em `load-history`, não imprime nenhuma mensagem e não protege contra cargas recursivas. A maioria dos usuários deve, portanto, usar `load`, `load-file`, `load-library` ou `require` em vez de `module-load`.

Módulos carregáveis no Emacs são habilitados usando a `--with-modules` opção no momento da configuração.

---

Anterior: [Ganchos para Carregamento](#), Acima: [Carregando](#) [[Conteúdo](#)][[Índice](#)]