Próximo:Carregando não ASCII, Anterior:Carregar sufixos, Acima:Carregando [Conteúdo][Índice]

16.3 Pesquisa de Biblioteca

Quando o Emacs carrega uma biblioteca Lisp, ele procura a biblioteca em uma lista de diretórios especificada pela variável `load-path`.

Variável: caminho de carga

O valor desta variável é uma lista de diretórios para pesquisar ao carregar arquivos com extensão `load`. Cada elemento é uma string (que deve ser um diretório) ou `nil`(que representa o diretório de trabalho atual).

Quando o Emacs é inicializado, ele configura o valor de `load-path` em várias etapas. Primeiro, ele inicializa `load-path`usando locais padrão definidos quando o Emacs foi compilado. Normalmente, este é um diretório algo como

```
"/usr/local/share/emacs/ versão /lisp"
```

(Neste e nos exemplos seguintes, substitua`/usr/local`com o prefixo de instalação apropriado para seu Emacs.) Esses diretórios contêm os arquivos Lisp padrão que vêm com o Emacs. Se o Emacs não conseguir encontrá-los, ele não iniciará corretamente.

Se você executar o Emacs a partir do diretório onde ele foi compilado - ou seja, um executável que não foi instalado formalmente - o Emacs inicializa `load-path`usando o`balbuciar` diretório no diretório que contém as fontes a partir das quais foi construído. Se você construiu o Emacs em um diretório separado das fontes, ele também adiciona os diretórios lisp do diretório build. (Em todos os casos, os elementos são representados como nomes de arquivo absolutos.)

A menos que você inicie o Emacs com o`--no-site-lisp`opção, ele adiciona mais dois`site-lisp`diretórios para a frente de `load-path`. Eles são destinados a arquivos Lisp instalados localmente e normalmente têm o formato:

```
"/usr/local/share/emacs/ versão /site-lisp"
```

e

```
"/usr/local/share/emacs/site-lisp"
```

O primeiro é para arquivos instalados localmente para uma versão específica do Emacs; o segundo é para arquivos instalados localmente destinados ao uso com todas as versões do Emacs instaladas. (Se o Emacs estiver rodando desinstalado, ele também adiciona`site-lisp`diretórios dos diretórios de origem e compilação, se existirem. Normalmente, esses diretórios não contêm `site-lisp`diretórios.)

Se a variável de ambiente `EMACSLOADPATH`estiver definida, ela modifica o procedimento de inicialização acima. O Emacs inicializa `load-path`com base no valor da variável de ambiente.

A sintaxe de `EMACSLOADPATH`é a mesma usada para `PATH`; diretórios são separados por `':'` (ou `';'`, em alguns sistemas operacionais). Aqui está um exemplo de como definir a `EMACSLOADPATH`variável (de um `sh`shell -style):

```
export EMACSLOADPATH=/home/foo/.emacs.d/lisp:
```

Um elemento vazio no valor da variável de ambiente, seja à direita (como no exemplo acima), à esquerda ou incorporado, é substituído pelo valor padrão de `load-path`conforme determinado pelo procedimento de inicialização padrão. Se não houver tais elementos vazios, `EMACSLOADPATH`especifica o arquivo `load-path`. Você deve incluir um elemento vazio ou o caminho explícito para o diretório que contém os arquivos Lisp padrão, senão o Emacs não funcionará. (Outra maneira de modificar `load-path`é usar o `-EU` opção de linha de comando ao iniciar o Emacs; Veja abaixo.)

Para cada diretório em `load-path`, o Emacs verifica se ele contém um arquivo`subdiretórios.el`, e em caso afirmativo, carrega-o. O `subdiretórios.el`O arquivo é criado quando o Emacs é compilado/instalado e contém código que faz com que o Emacs adicione quaisquer subdiretórios desses diretórios ao arquivo `load-path`. Ambos os subdiretórios imediatos e subdiretórios de vários níveis abaixo são adicionados. Mas exclui subdiretórios cujos nomes não começam com uma letra ou dígito, e subdiretórios chamados`RCS`ou`CVS`, e subdiretórios contendo um arquivo chamado`.nosearch`.

Em seguida, o Emacs adiciona qualquer diretório de carregamento extra que você especificar usando o `-EU`opção de linha de comando (consulte Argumentos de ação no Manual do GNU Emacs). Ele também adiciona os diretórios onde os pacotes opcionais são instalados, se houver (consulte Noções básicas de empacotamento).

É comum adicionar código ao arquivo init (consulte Init File) para adicionar um ou mais diretórios ao `load-path`. Por exemplo:

```
(pressione o caminho de carregamento "~/.emacs.d/lisp")
```

Dumping Emacs usa um valor especial de `load-path`. Se você usar um`site-load.el`ou`site-init.el`para personalizar o Emacs despejado (consulte Construindo Emacs), quaisquer alterações feitas por `load-path` esses arquivos serão perdidas após o despejo.

Comando: biblioteca de localização de biblioteca *e chamada interativa de caminho de sufixo opcional*

Este comando encontra o nome preciso do arquivo para a biblioteca da *biblioteca* . Ele procura a biblioteca da mesma maneira `load`, e o argumento *nosuffix* tem o mesmo significado que em `load`: não adicione sufixos `'.elc'` ou `'.el'` *para a biblioteca* de nomes especificada .

Se o *caminho* não for `nil`, essa lista de diretórios será usada em vez de `load-path`.

Quando `locate-library`é chamado de um programa, ele retorna o nome do arquivo como uma string. Quando o usuário executa `locate-library` interativamente, o argumento *chamada interativa* é `t`, e isso informa `locate-library`para exibir o nome do arquivo na área de eco.

Comando: list-load-path-shadows *e stringp opcional*

Este comando mostra uma lista de arquivos Emacs Lisp *sombreados* . Um arquivo sombreado é aquele que normalmente não será carregado, apesar de estar em um diretório em `load-path`, devido à existência de outro arquivo de nome semelhante em um diretório anterior `load-path`.

Por exemplo, suponha que `load-path`esteja definido como

```
("/opt/emacs/site-lisp" "/usr/share/emacs/23.3/lisp")
```

e que ambos os diretórios contêm um arquivo chamado`foo.el`. Então `(require 'foo)`nunca carrega o arquivo no segundo diretório. Tal situação pode indicar um problema na forma como o Emacs foi instalado.

Quando chamada de Lisp, esta função imprime uma mensagem listando os arquivos sombreados, em vez de exibi-los em um buffer. Se o argumento opcional `stringp`não for `nil`, ele retornará os arquivos sombreados como uma string.

Próximo:Carregando não ASCII, Anterior:Carregar sufixos, Acima:Carregando [Conteúdo][Índice]