Próximo:Variáveis locais do diretório, Anterior:Variáveis locais de buffer, Acima:Variáveis [Conteúdo][Índice]

12.12 Variáveis Locais de Arquivo

Um arquivo pode especificar valores de variáveis locais; O Emacs os usa para criar ligações locais de buffer para essas variáveis no buffer que visita esse arquivo. Veja Variáveis Locais em Arquivos no Manual do GNU Emacs , para informações básicas sobre variáveis locais de arquivos. Esta seção descreve as funções e variáveis que afetam como as variáveis locais de arquivo são processadas.

Se uma variável local de arquivo pudesse especificar uma função arbitrária ou expressão Lisp que seria chamada mais tarde, visitar um arquivo poderia assumir o controle do seu Emacs. O Emacs protege contra isso definindo automaticamente apenas as variáveis locais de arquivo cujos valores especificados são conhecidos como seguros. Outras variáveis locais de arquivo são definidas somente se o usuário concordar.

Para segurança adicional, `read-circle`é temporariamente limitado `nil`quando o Emacs lê variáveis locais de arquivo (consulte Funções de entrada). Isso impede que o leitor Lisp reconheça estruturas Lisp circulares e compartilhadas (consulte Objetos circulares).

Opção do usuário: enable-local-variables

Essa variável controla se as variáveis locais de arquivo devem ser processadas. Os valores possíveis são:

`t`(o padrão)

Defina as variáveis seguras e consulte (uma vez) sobre quaisquer variáveis não seguras.

`:safe`

Defina apenas as variáveis seguras e não faça consultas.

`:all`

Defina todas as variáveis e não faça consultas.

`nil`

Não defina nenhuma variável.

algo mais

Consulta (uma vez) sobre todas as variáveis.

Variável: inibidor-local-variáveis-regexps

Esta é uma lista de expressões regulares. Se um arquivo tiver um nome que corresponda a um elemento dessa lista, ele não será verificado em busca de qualquer forma de variável local de arquivo. Para obter exemplos de por que você pode querer usar isso, consulte Auto Major Mode .

Função: hack-local-variables *e modo handle opcional*

Essa função analisa e vincula ou avalia conforme apropriado, quaisquer variáveis locais especificadas pelo conteúdo do buffer atual. A variável `enable-local-variables`tem seu efeito

aqui. No entanto, esta função não procura o 'modo:'variável local no '-*-' linha. set-auto-modefaz isso, também levando enable-local-variablesem consideração (consulte Auto Major Mode).

Esta função funciona percorrendo o alist armazenado file-local-variables-alisteaplicando cada variável local por vez. Ele chama before-hack-local-variables-hooke hack-local-variables-hookantes e depois de aplicar as variáveis, respectivamente. Ele só chama o antes do gancho se o alist for não- nil; ele sempre chama o outro gancho. Esta função ignora um 'modo' se especificar o mesmo modo principal que o buffer já possui.

Se o argumento opcional *handle-mode* for t, tudo o que esta função faz é retornar um símbolo especificando o modo principal, se o '-*-' ou a lista de variáveis locais especifica um, e nilcaso contrário. Ele não define o modo ou qualquer outra variável local de arquivo. Se *handle-mode* tiver qualquer valor diferente de nilou t, quaisquer configurações de 'modo' no '-*-' ou a lista de variáveis locais são ignoradas e as outras configurações são aplicadas. Se *handle-mode* for nil, todas as variáveis locais do arquivo serão definidas.

Variável: file-local-variables-alist

Essa variável local de buffer contém a lista de configurações de variáveis locais de arquivo. Cada elemento da lista é da forma , onde *var* é um símbolo da variável local e *valor* é seu valor. Quando o Emacs visita um arquivo, ele primeiro coleta todas as variáveis locais do arquivo nessa lista e, em seguida, a função as aplica uma a uma. (*var . value*)hack-local-variables

Variável: antes-hack-local-variables-hook

O Emacs chama esse gancho imediatamente antes de aplicar variáveis locais de arquivo armazenadas em file-local-variables-alist.

Variável: hack-local-variables-hook

O Emacs chama esse gancho imediatamente após terminar de aplicar as variáveis locais de arquivo armazenadas em file-local-variables-alist.

Você pode especificar valores seguros para uma variável com uma safe-local-variablepropriedade. A propriedade deve ser uma função de um argumento; qualquer valor é seguro se a função não retornar nilesse valor. Muitas variáveis de arquivo comumente encontradas têm safe-local-variablepropriedades; estes incluem fill-column, fill-prefixe indent-tabs-mode. Para variáveis de valor booleano que são seguras, use booleanpcomo o valor da propriedade.

Se você deseja definir safe-local-variablepropriedades para variáveis definidas no código-fonte C, adicione os nomes e as propriedades dessas variáveis à lista na seção “Variáveis locais seguras” doarquivos.el.

Ao definir uma opção de usuário usando defcustom, você pode definir sua safe-local-variablepropriedade adicionando os argumentos a (consulte Definições de variáveis). No entanto, um predicado de segurança definido usando só será conhecido quando o pacote que o contém for carregado, o que geralmente é tarde demais. Como alternativa, você pode usar o cookie autoload (veja Autoload) para atribuir à opção seu predicado de segurança, assim: :safe *function*defcustom:safedefcustom

```
;;;###autoload (coloque ' var 'safe-local-variable ' pred )
```

As definições de valor seguro especificadas com autoloadsão copiadas no arquivo de carregamentos automáticos do pacote (loaddefs.elpara a maioria dos pacotes empacotados com o Emacs) e são conhecidos pelo Emacs desde o início de uma sessão.

Opção do usuário: valores variáveis locais seguros

Essa variável fornece outra maneira de marcar alguns valores de variáveis como seguros. É uma lista de células contra , onde *var* é um nome de variável e *val* é um valor que é seguro para essa variável. (`(var . val)`

Quando o Emacs pergunta ao usuário se deve ou não obedecer a um conjunto de especificações de variáveis locais de arquivo, o usuário pode optar por marcá-las como seguras. Fazer isso adiciona esses pares de variável/valor a `safe-local-variable-values`e salva no arquivo personalizado do usuário.

Função: safe-local-variable-p *sym val*

Esta função retorna non- `nil`se for seguro dar a *sym* o valor *val* , com base nos critérios acima.

Algumas variáveis são consideradas *arriscadas* . Se uma variável for arriscada, ela nunca será inserida automaticamente em `safe-local-variable-values`; O Emacs sempre consulta antes de definir uma variável arriscada, a menos que o usuário permita explicitamente um valor personalizando `safe-local-variable-values`diretamente.

Qualquer variável cujo nome tenha uma não `nil` `risky-local-variable`propriedade é considerada arriscada. Ao definir uma opção de usuário usando `defcustom`, você pode definir sua `risky-local-variable`propriedade adicionando os argumentos a (consulte Definições de variáveis). Além disso, qualquer variável cujo nome termine em qualquer um de '`:risky value`defcustom-comando', '`-lista de quadros`', '`-função`', '`-funções`', '`-gancho`', '`-ganchos`', '`-Formato`', '`-formulários`', '`-mapa`', '`-map-alist`', '`-mode-alist`', '`-programa`', ou '`-predicado`' é automaticamente considerado arriscado. As variáveis '`palavras-chave de bloqueio de fonte`', '`palavras-chave de bloqueio de fonte`' seguido por um dígito, e '`palavras-chave sintáticas de bloqueio de fonte`' também são considerados arriscados.

Função: risky-local-variable-p *sym*

Esta função retorna non - `nil`if *sym* é uma variável arriscada, com base nos critérios acima.

Variável: variáveis locais ignoradas

Esta variável contém uma lista de variáveis que não devem receber valores locais por arquivos. Qualquer valor especificado para uma dessas variáveis é completamente ignorado.

O '`Avaliação:`' “variável” também é uma brecha potencial, então o Emacs normalmente pede confirmação antes de manuseá-la.

Opção do usuário: enable-local-eval

Esta variável controla o processamento de '`Avaliação:`' dentro '`-*-`' linhas ou listas de variáveis locais nos arquivos que estão sendo visitados. Um valor de `t`meios processa-os incondicionalmente; `nil`significa ignorá-los; qualquer outra coisa significa perguntar ao usuário o que fazer para cada arquivo. O valor padrão é `maybe`.

Opção do usuário: safe-local-eval-forms

Esta variável contém uma lista de expressões que são seguras para avaliar quando encontradas no '`Avaliação:`' “variável” em uma lista de variáveis locais do arquivo.

Se a expressão for uma chamada de função e a função tiver uma `safe-local-eval-function`propriedade, o valor da propriedade determinará se a expressão é segura para avaliação. O valor da propriedade pode ser um predicado a ser chamado para testar a expressão, uma lista de tais predicados

(é seguro se algum predicado for bem-sucedido) ou `t` (sempre seguro desde que os argumentos sejam constantes).

As propriedades de texto também são brechas em potencial, pois seus valores podem incluir funções a serem chamadas. Portanto, o Emacs descarta todas as propriedades de texto dos valores de string especificados para variáveis locais de arquivo.

Próximo:Variáveis locais do diretório, Anterior:Variáveis locais de buffer, Acima:Variáveis [Conteúdo][Índice]