Próximo:Segurança da Função, Anterior:Formulário de declaração, Acima:Funções [Conteúdo][Índice]

13.15 Dizendo ao compilador que uma função está definida

A compilação de bytes de um arquivo geralmente produz avisos sobre funções que o compilador não conhece (consulte Erros do compilador). Às vezes isso indica um problema real, mas geralmente as funções em questão são definidas em outros arquivos que seriam carregados se esse código fosse executado. Por exemplo, compilação de bytes`simple.el`usado para avisar:

```
simple.el:8727:1:Aviso: a função 'shell-mode' não é conhecida por ser
    definiram.
```

Na verdade, `shell-mode`é usado apenas em uma função que executa `(require 'shell)`antes de chamar `shell-mode`, portanto, `shell-mode`será definido corretamente em tempo de execução. Quando você sabe que tal aviso não indica um problema real, é bom suprimir o aviso. Isso torna mais visíveis os novos avisos que podem significar problemas reais. Você faz isso com `declare-function`.

Tudo que você precisa fazer é adicionar uma `declare-function`declaração antes do primeiro uso da função em questão:

```
(declare-função shell-mode "shell" ())
```

Isso diz que `shell-mode`é definido em`shell.el`(a '`.el`' pode ser omitida). O compilador assume que esse arquivo realmente define a função e não verifica.

O terceiro argumento opcional especifica a lista de argumentos de `shell-mode`. Neste caso, não recebe argumentos (`nil`é diferente de não especificar um valor). Em outros casos, isso pode ser algo como `(file &optional overwrite)`. Você não precisa especificar a lista de argumentos, mas se fizer isso, o compilador de bytes pode verificar se as chamadas correspondem à declaração.

Macro: arquivo de função de declaração e *arquivo arglist opcional somente*

> Diga ao compilador de bytes para assumir que a *função* está definida no arquivo *file* . O terceiro argumento opcional *arglist* é `t`, significando que a lista de argumentos não é especificada, ou uma lista de parâmetros formais no mesmo estilo que `defun`. Uma lista de argumentos omitida *tem como* padrão `t`, not `nil`; esse é um comportamento atípico para argumentos omitidos e significa que para fornecer um quarto, mas não um terceiro argumento, deve-se especificar `to` espaço reservado do terceiro argumento em vez do `nil`. O quarto argumento opcional *fileonly* não `nil`significa verificar apenas se o *arquivo* existe, não se ele realmente define a *função* .

Para verificar se essas funções realmente são declaradas onde `declare-function`diz que estão, use `check-declare-file` para verificar todas as `declare-function`chamadas em um arquivo de origem ou use `check-declare-directory`verificar todos os arquivos em e sob um determinado diretório.

Esses comandos encontram o arquivo que deve conter a definição de uma função usando `locate-library`; se não encontrar nenhum arquivo, eles expandem o nome do arquivo de definição em relação ao diretório do arquivo que contém a `declare-function`chamada.

Você também pode dizer que uma função é uma primitiva especificando um nome de arquivo que termina em '`.c`' ou '`.m`'. Isso é útil apenas quando você chama uma primitiva definida apenas em determinados sistemas. A maioria dos primitivos são sempre definidos, então eles nunca lhe darão um aviso.

Às vezes, um arquivo usará opcionalmente funções de um pacote externo. Se você prefixar o nome do arquivo na `declare-function`instrução com '`ramal:`', então será verificado se for encontrado, caso contrário será ignorado sem erro.

Existem algumas definições de função que '`check-declarar`' não entende (por exemplo, `defstructe` algumas outras macros). Nesses casos, você pode passar um argumento non- `nil` *fileonly* para `declare-function`, significando apenas verificar se o arquivo existe, não se ele realmente define a função. Observe que para fazer isso sem precisar especificar uma lista de argumentos, você deve definir o argumento *arglist*`t` como (porque `nil`significa uma lista de argumentos vazia, em oposição a uma não especificada).

Próximo:Segurança da Função, Anterior:Formulário de declaração, Acima:Funções [Conteúdo][Índice]