

Próximo:[Escopo Variável](#), Anterior:[Configurando Variáveis](#), Acima:[Variáveis](#) [Conteúdo][Índice]

12.9 Executando uma função quando uma variável é alterada.

Às vezes é útil tomar alguma ação quando uma variável muda seu valor. O recurso de *watchpoint variável* fornece os meios para fazer isso. Alguns usos possíveis para esse recurso incluem manter a exibição em sincronia com as configurações de variáveis e invocar o depurador para rastrear alterações inesperadas nas variáveis (consulte [Depuração de variáveis](#)).

As seguintes funções podem ser usadas para manipular e consultar as funções de observação de uma variável.

Função: add-variable-watcher *symbol watch-function*

Esta função faz com que a função *watch* seja chamada sempre que o *símbolo* for modificado.

Modificações através de aliases (veja [Aliases de Variáveis](#)) terão o mesmo efeito.

watch-function será chamada, logo antes de alterar o valor de *symbol* , com 4 argumentos: *symbol* , *newval* , *operation* e *where* . *símbolo* é a variável que está sendo alterada. *newval* é o valor para o qual será alterado. (O valor antigo está disponível para *watch-function* como o valor de *symbol* , uma vez que ainda não foi alterado para *newval* .) *operação* é um símbolo que representa o tipo de alteração, um dos seguintes: *set*, *let*, *unlet*, *makunbound* ou *defvaralias*. *Onde* é um buffer se o valor local do buffer da variável estiver sendo alterado, nil caso contrário.

Função: remove-variable-watcher *symbol watch-function*

Esta função remove a função *watch* da lista de observadores do *símbolo* .

Função: *símbolo* get-variable-watchers

Esta função retorna a lista de funções do observador ativo do *símbolo* .

12.9.1 Limitações

Existem algumas maneiras pelas quais uma variável pode ser modificada (ou pelo menos parecer ser modificada) sem acionar um *watchpoint*.

Como os *watchpoints* são anexados a símbolos, a modificação dos objetos contidos nas variáveis (por exemplo, por uma função de modificação de lista, consulte [Modificando listas](#)) não é capturada por esse mecanismo.

Além disso, o código C pode modificar o valor das variáveis diretamente, ignorando o mecanismo de *watchpoint*.

Uma pequena limitação desse recurso, novamente porque visa símbolos, é que apenas variáveis de escopo dinâmico podem ser observadas. Isso apresenta pouca dificuldade, pois as modificações nas variáveis léxicas podem ser descobertas facilmente inspecionando o código dentro do escopo da variável (diferentemente das variáveis dinâmicas, que podem ser modificadas por qualquer código, veja [Variable Scoping](#)).

Próximo:[Escopo Variável](#), Anterior:[Configurando Variáveis](#), Acima:[Variáveis](#) [Conteúdo][Índice]