NAME

trampoline − closures as first-class C functions

SYNOPSIS

#include <trampoline.h>

function = alloc_trampoline(address, variable, data);

free_trampoline(function);

is_trampoline(function)
trampoline_address(function)
trampoline_variable(function)
trampoline_data(function)

DESCRIPTION

These functions implement *closures* as first-class C functions. A closure consists of a regular C function and a piece of data which gets passed to the C function when the closure is called.

Closures as *first-class C functions* means that they fit into a function pointer and can be called exactly like any other C function. *function* **= alloc_trampoline(***address***,** *variable***,** *data***)** allocates a closure. When *function* gets called, it stores *data* in the variable *variable* and calls the C function at *address*. The function at *address* is responsible for fetching *data* out of *variable* immediately, before execution of any other function call.

This is much like **gcc**'s local functions, except that the GNU C local functions have dynamic extent (i.e. are deallocated when the creating function returns), while *trampoline* provides functions with indefinite extent: *function* is only deallocated when **free_trampoline(** *function***)** is called.

is_trampoline(*function***)** checks whether the C function *function* was produced by a call to *alloc_trampoline*. If this returns true, the arguments given to *alloc_trampoline* can be retrieved:

trampoline_address(*function***)** returns *address*,

trampoline_variable(*function***)** returns *variable*,

trampoline_data(*function***)** returns *data*.

SEE ALSO

gcc(1), **stdarg**(3), **callback**(3)

BUGS

Passing the data through a global variable is not reentrant. Don't call trampoline functions from within signal handlers. This is fixed in the **callback**(3) package.

PORTING

The way **gcc** builds local functions is described in the gcc source, file gcc-2.6.3/config/*cpu*/*cpu*.h.

AUTHOR

Bruno Haible <bruno AT clisp DOT org>

ACKNOWLEDGEMENTS

Many ideas were cribbed from the gcc source.