

setjmp.h in C

The file header named setjmp.h in C Library defines the macro setjmp (), a longjmp () function, and a variable type jmp_buf, to ignore the regular function call and return the rule, by providing methods mode to perform jumps while maintaining the function call environment.

setjmp.h in C

The file header named setjmp.h in C Library defines the macro setjmp (), a longjmp () function, and a variable type jmp_buf, to ignore the regular function call and return the rule, by providing methods mode to perform jumps while maintaining the function call environment.

The variable is defined in setjmp.h

Here is the type of variable defined in setjmp.h:

jmp_buf : This is an array type used to hold information for the setjmp () macro and longjmp () function.

The macros are defined in setjmp.h

There is only one macro defined in this library:

int setjmp (jmp_buf environment) : This macro stores the current environment inside the environment variable for later use by longjmp (). If this macro returns directly from the macro call, it returns 0; but if it returns from a call to longjmp (), a value other than 0 is returned.

Macro declaration setjmp () in C

Below is the declaration for setjmp () macro.

```
int setjmp (jmp_buf environment)
```

Parameters:

environment - This is an object of type jmp_buf where environment information is stored.

Return value:

This macro returns more than 1 time. First, on its direct call, it always returns 0. When longjmp is called with the information set to the environment, this macro returns again; this time it returns the value passed to longjmp as

the second parameter.

For example:

The following C program illustrates the usage of `setjmp ()` macro.

```
#include
#include
#include

main ()
{
    jmp_buf env;
    int i;

    i = setjmp (env);
    printf ("i =% dn", i);

    if (i! = 0) exit (0);

    longjmp (env, 2);
    printf ("Is it printed now?");
}
```

Compiling and running the above C program will result:

```
i = 0
i = 2
```

The functions are defined in `setjmp.h`

Only one function is defined in `setjmp.h`:

Function `longjmp void (jmp_buf environment, int value)`: This function restores the environment (environment) stored by the closest call to the `setjmp ()` macro in the same function call of the program with the corresponding parameter `jmp_buf`.

Declare the function `longjmp ()` in C

Below is the declaration for `longjmp ()` function in C:

```
void longjmp (jmp_buf environment, int value)
```

Parameters:

`environment` - This is an object of type `jmp_buf` that contains information to store the environment at the call point of `setjmp`.

`value` - This is the value for the `setjmp` expression to estimate.

Return value:

This function does not return any values.

For example:

The following program C illustrates the usage of the `longjmp ()` function in C:

```
#include
#include
#include

main ()
{
    jmp_buf env;
    int i;

    i = setjmp (env);
    printf ("i =% dn", i);

    if (i! = 0) exit (0);

    longjmp (env, 2);
    printf ("Is it printed now?");
}
```

Compiling and running the above C program will result:

```
i = 0
i = 2
```

According to Tutorialspoint

Last lesson: `math.h` in C

Next post: `signal.h` in C

You finished reading the article "**setjmp.h in C**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.