C LIBRARY FUNCTION - SIGNAL

http://www.tutorialspoint.com/c_standard_library/c_function_signal.htm

Description

The C library function **void** $*signal(intsig, void(*funcint))int$ sets a function to handle signal i.e. a signal handler with signal number **sig**.

Declaration

Following is the declaration for signal function.

```
void (*signal(int sig, void (*func)(int)))(int)
```

Parameters

- **sig** − This is the signal number to which a handling function is set. The following are few important standard signal numbers −

macro	signal
SIGABRT	*SignalAbort* Abnormal termination, such as is initiated by the function.
SIGFPE	*SignalFloating − PointException* Erroneous arithmetic operation, such as zero divide or an operation resulting in overflow *notnecessarilywithafloating − pointoperation*.
SIGILL	*SignalIllegalInstruction* Invalid function image, such as an illegal instruction. This is generally due to a corruption in the code or to an attempt to execute data.
SIGINT	*SignalInterrupt* Interactive attention signal. Generally generated by the application user.
SIGSEGV	*SignalSegmentationViolation* Invalid access to storage − When a program tries to read or write outside the memory it is allocated for it.
SIGTERM	*SignalTerminate* Termination request sent to program.

- **func** − This is a pointer to a function. This can be a function defined by the programmer or one of the following predefined functions −

SIG_DFL	Default handling − The signal is handled by the default action for that particular signal.
SIG_IGN	Ignore Signal − The signal is ignored.

Return Value

This function returns the previous value of the signal handler, or SIG_ERR on error.

Example

The following example shows the usage of signal function.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <signal.h>
```

```
void sighandler(int);

int main()
{
   signal(SIGINT, sighandler);

   while(1)
   {
      printf("Going to sleep for a second...\n");
      sleep(1);
   }
   return(0);
}

void sighandler(int signum)
{
   printf("Caught signal %d, coming out...\n", signum);
   exit(1);
}
```

Let us compile and run the above program that will produce the following result and program will go in infinite loop. To come out of the program we used CTRL + C keys.

```
Going to sleep for a second...
Going to sleep for a second...
Going to sleep for a second...
Going to sleep for a second...
Going to sleep for a second...
Caught signal 2, coming out
```