

## WHAT IS GCC AND GDB?

**GNU Compiler Collection (GCC)** is a compiler system produced by the GNU Project supporting various programming languages. GCC is a key component of the GNU toolchain and the standard compiler for most projects related to GNU and Linux, including the Linux Kernel. The Free Software Foundation (FSF) distributes GCC under the GNU General Public License (GNU GPL). GCC has played an important role in the growth of the free software, as both a tool and an example.

**GNU Debugger (GDB)** is a portable debugger that runs on many Unix-like systems and works for many programming languages, including Ada, C, C++, Objective-C, Free Pascal, Fortran, Go, and partially others.

### ADDRESS

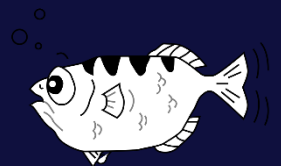
Misión San Miguel 10-5, Zapopan

### CONTACT US

[mrpikachu1997@gmail.com](mailto:mrpikachu1997@gmail.com)  
(52) 33 22 1 3 55 25

## LIST OF GCC AND GDB COMANDS

A brief list of common GCC and GDB commands used during programming.



## COMANDS

`alloc_align (position) (GCC)`

`const (GCC)`

`./main.cpp (GCC)`

`attach (GDB)`

## FUNCTION

The `alloc_align` attribute may be applied to a function that returns a pointer and takes at least one argument of an integer or enumerated type. It indicates that the returned pointer is aligned on a boundary given by the function argument at *position*. Meaningful alignments are powers of 2 greater than one. GCC uses this information to improve pointer alignment analysis.

Calls to functions whose return value is not affected by changes to the observable state of the program and that have no observable effects on such state other than to return a value may lend themselves to optimizations such as common subexpression elimination. Declaring such functions with the `const` attribute allows GCC to avoid emitting some calls in repeated invocations of the function with the same argument values.

Execute the compiled code if the name is not specified

Used to execute a memory process

## EXAMPLE

```
void* my_memalign (size_t, size_t)
__attribute__ ((alloc_align (1)));
```

```
int square (int) __attribute__ ((const));
```

`./example (will execute serial.cpp)`

`attach 9071 (pid 9071 process)`