

# \* System Programming and Operating System (SPOS) - Test Case Study

Name:- Kaustubh Shrikant Habra.

Class:- Third Year Engineering.

Div:- A Roll Number:- 38

Batch:- T2

Department:- Computer Department.

College:- AISSMS's IOIT.

## Study of Debugging tools like GDB

What is GDB?

GDB, the GNU Project debugger, allows you to see what is going on 'inside' another program while it executes -- or what another program was doing at the moment it crashed.

GDB can do four main kinds of things (plus other things in support of these) to help you catch bugs in the act:

- Start your program, specifying anything that might affect its behavior.
- Make your program stop on specified conditions.
- Examine what has happened, when your program has stopped.
- Change things in your program, so you can experiment with correcting the effect of one bug and go on to learn about another.

These programs might be executing on the same machine as GDB (native), on another machine (remote), or on a simulator.



## History -

GDB was first written by Richard Stallman in 1986 as part of his GNU system, after his GNU Emacs was "reasonably stable". GDB is free software released under the GNU General Public License (GPL). It was modeled after the DBX debugger, which came with Berkeley Unix distributions.

From 1990 to 1993 it was maintained by John Gilmore. Now it is maintained by the GDB Steering Committee which is appointed by the Free Software Foundation.

## What Language does GDB Support?

GDB support the following languages (in):-

- Ada
- Assembly
- C
- C++
- D
- Fortran
- Go
- Objective-C
- OpenCL
- Modula-2
- Pascal
- Rust.



## Technical details:-

### → Features -

GDB offers extensive facilities for tracing and altering the execution of computer programs. The user can monitor and modify the values of programs' internal variables, and even call functions independently of the program's normal behaviour.

GDB target processors (as of 2003) include: Alpha, ARM, AVR, H8/300, Altera Nios/Nios II, System/370, System 390, X86 and its 64-bit extension X86-64, IA-64 "Itanium", Motorola 68000, etc.

### → Remote Debugging -

GDB offers a "remote" mode often used when debugging embedded systems. Remote operation is when GDB runs on one machine and the program being debugged runs on another.

GDB can communicate to the remote "stub" that understands GDB protocol through a serial device or TCP/IP. A stub program can be created by linking to the appropriate stub files provided with GDB, which implement the target side of the communication protocol.

### → Graphical user interface -

The debugger does not contain its own graphical user interface, and defaults to a command-line interface, although it does contain a text user interface.

Some other debugging tools have been designed to work with GDB, such as memory leak detectors.