

GDB - DEBUGGING SYMBOLS

http://www.tutorialspoint.com/gnu_debugger/gdb_debugging_symbols.htm

Copyright © tutorialspoint.com

A **Debugging Symbol Table** maps instructions in the compiled binary program to their corresponding variable, function, or line in the source code. This mapping could be something like:

- Program instruction \Rightarrow item name, item type, original file, line number defined.

Symbol tables may be embedded into the program or stored as a separate file. So if you plan to debug your program, then it is required to create a symbol table which will have the required information to debug the program.

We can infer the following facts about symbol tables:

- A symbol table works for a particular version of the program – if the program changes, a new table must be created.
- Debug builds are often larger and slower than retail *non – debug* builds; debug builds contain the symbol table and other ancillary information.
- If you wish to debug a binary program you did not compile yourself, you must get the symbol tables from the author.

To let GDB be able to read all that information line by line from the symbol table, we need to compile it a bit differently. Normally we compile our programs as:

```
gcc hello.cc -o hello
```

Instead of doing this, we need to compile with the `-g` flag as shown below:

```
gcc -g hello.cc -o hello
```

Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js