



Εισαγωγή στο GDB

Ανδρόνικος Χαραλάμπους



Περιεχόμενα

- Βασικές εντολές του gdb
 - Εκτέλεση προγράμματος
 - Breakpoints και Watchpoints
 - Μετακίνηση μέσα στο πρόγραμμα
- Πως κάνουμε debug ένα πρόγραμμα
 - Αντιμετώπιση πιθανών σφαλμάτων(Segmentation fault, Floating point exceptions)
 - Αντιμετώπιση λογικών λαθών
- References



Βασικές εντολές του GDB

Για να χρησιμοποιήσουμε το gdb ακολουθούμε τα εξής βήματα:

- `g++ -g example1.cpp -o example1`
- `gdb example1` ή `gdb example1 < inputFile` (Για να περάσουμε είσοδο στο πρόγραμμα)

```
b103ws33:/home/students/cs/2016/achara01/Documents/camp2019>g++ -g example1.cpp -o example1
b103ws33:/home/students/cs/2016/achara01/Documents/camp2019>gdb example1
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-114.el7
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/students/cs/2016/achara01/Documents/camp2019/example1...done.
(gdb) □
```



Βασικές εντολές του GDB

Εκτέλεση προγράμματος

run ή r	Εκτέλεση του προγράμματος
kill	Τερματισμός προγράμματος



Βασικές εντολές του GDB

Breakpoints και Watchpoints

<code>break <where></code> ή <code>b <where></code>	Βάλτε breakpoint σε μια γραμμή
<code>delete <breakpoint></code>	Διάγραψε ένα breakpoint
<code>enable <breakpoint></code>	Ενεργοποίηση ενός breakpoint
<code>disable <breakpoint></code>	Απενεργοποίηση ενός breakpoint
<code>info breakpoints</code>	Πληροφορίες για τα breakpoints



Βασικές εντολές του GDB

Μετακίνηση μέσα στο πρόγραμμα

step	Προχωρά στην επόμενη εντολή
finish	Τρέχει μέχρι να τερματίσει η συνάρτηση
continue	Συνέχισε την εκτέλεση του προγράμματος



Πως κάνουμε debug ένα πρόγραμμα

Εντοπισμός εντολής που προκαλεί segmentation fault

- Κάνουμε compile το πρόγραμμα όπως έχουμε πει πιο πριν για να μπορέσουμε να χρησιμοποιήσουμε το gdb
- Εκτελούμε στην γραμμή εντολών τα εξής:
 - `gdb -q example3 -tui`
 - `run`
 - `backtrace`

```
example3.cpp
1  #include <stdio.h>
2
3  using namespace std;
4
5  int main(){
6
7      int arr[10];
8
9      for(int i=0;i<10;i++){
10         scanf("%d",arr[i]);
11     }
12
13
14
15
16 }
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
```

child process 27172 In: _GI_IO_vfscanf

Line: ?? PC: 0x7ffff724b181

Reading symbols from /home/students/cs/2016/achara01/Documents/camp2019/example3...done.

(gdb) r

Starting program: /home/students/cs/2016/achara01/Documents/camp2019/./example3

Program received signal SIGSEGV, Segmentation fault.

0x00007ffff724b181 in _GI_IO_vfscanf () from /lib64/libc.so.6

Missing separate debuginfos, use: debuginfo-install glibc-2.17-260.el7_6.4.x86_64 libgcc-4.8.5-36.el7_6.1.x86_64 libstdc++-4.8.5-36.el7_6.1.x86_64

(gdb) backtrace

#0 0x00007ffff724b181 in _GI_IO_vfscanf () from /lib64/libc.so.6

#1 0x00007ffff725a22b in scanf () from /lib64/libc.so.6

#2 0x0000000000400568 in main () at example3.cpp:10

(gdb) □



Πως κάνουμε debug ένα πρόγραμμα

Εντοπισμός λογικού λάθους

- Δεν υπάρχει συγκεκριμένη μεθοδολογία για τον εντοπισμό λογικού λάθους.
- Προσπαθούμε με τις δυνατότητες που μας δίνει ο gdb να βρούμε το λάθος μας.
- Μια απλή προσέγγιση που μπορούμε να κάνουμε:
 - Θέτουμε breakpoints εκεί που πιστεύουμε ότι μπορεί να είναι το λάθος και εκτελούμε το πρόγραμμα μας.
 - Βάζουμε σε κάποιες μεταβλητές watchpoint για παρατηρήσουμε πότε αλλάζουν κατά την διάρκεια που τρέχει το πρόγραμμα μας.
 - Εκτελούμε μια-μια γραμμή κώδικα για να βλέπουμε ακριβώς τι αλλάζει σε κάθε εκτέλεση



References

<https://darkdust.net/files/GDB%20Cheat%20Sheet.pdf>

https://www.tutorialspoint.com/gnu_debugger/gdb_debugging_programs.htm

<https://www.geeksforgeeks.org/gdb-step-by-step-introduction/>