

## Σύντομος οδηγός *gdb*

### Εισαγωγή

Ο debugger (αποσφαλματωτής) είναι ένα πρόγραμμα το οποίο διευκολύνει την εύρεση λογικών λαθών σε ένα άλλο πρόγραμμα Π. Τυπικές λειτουργίες ενός debugger είναι:

- Εκτελεί το πρόγραμμα Π γραμμή-γραμμή, ή μέχρι ενός σημείου
- Παρακολουθεί τις τιμές διαφόρων μεταβλητών και εκφράσεων
- Δείχνει τη σειρά συναρτήσεων που κλήθηκαν μέχρι ένα σημείο.

Ο debugger που θα χρησιμοποιήσουμε στο μάθημα είναι το *gdb*. Μπορούμε να τρέξουμε το *gdb* είτε στη γραμμή εντολών είτε μέσω του γραφικού περιβάλλοντος *kdbg*. Στο μάθημα σας δείξαμε τον πρώτο τρόπο.

### Πώς να κάνετε *compile*

Για να μπορέσετε να χρησιμοποιήσετε *gdb* για την αποσφαλμάτωση ενός προγράμματος θα πρέπει να κάνετε το πρόγραμμα *compile* με την επιλογή *-g*. Αυτή λέει στον compiler να αποθηκεύσει στο εκτελέσιμο κάποιες επιπλέον πληροφορίες που θα χρειαστεί το *gdb*.

### Πώς να φορτώσετε το πρόγραμμά σας στο *gdb*

Υπάρχουν δύο τρόποι να φορτώσετε το πρόγραμμά σας στο *gdb*:

1. Κατά την εκκίνηση του *gdb*, γράφοντας στο τερματικό *gdb* ακολουθούμενο από το όνομα του εκτελέσιμου.
2. Μέσα από το περιβάλλον του *gdb* γράφοντας την εντολή του *gdb* **file** (ή **f**) ακολουθούμενη από το όνομα του εκτελέσιμου.

Εφόσον φορτωθεί ένα εκτελέσιμο στο *gdb* μπορείτε να το τρέξετε με την εντολή του *gdb* **run** (ή **r**) ακολουθούμενη από το όνομα του εκτελέσιμου.

### Πώς να θέσετε ένα *breakpoint*

Το *breakpoint* είναι ένα σημείο του προγράμματος στο οποίο επιλέγουμε να σταματήσει (προσωρινά) η εκτέλεση, συνήθως για να μελετήσουμε τις τιμές μεταβλητών σε αυτό το σημείο. Συντά, επιλέγουμε να θέσουμε ένα *breakpoint* στην αρχή του προγράμματος (αμέσως μετά τη γραμμή `int main()`), στην αρχή μιας συνάρτησης ή σε συγκεκριμένη γραμμή.

Για να θέσουμε ένα *breakpoint* στην αρχή μιας συνάρτησης γράφουμε **break** (ή **b**) ακολουθούμενο από το όνομα της συνάρτησης. Για παράδειγμα, **break main** θέτει ένα *breakpoint* στην αρχή της `main`. Μπορούμε να θέσουμε *breakpoint* στην αρχή οποιασδήποτε συνάρτησης έχουμε γράψει στο πρόγραμμά μας.

Για να θέσουμε ένα *breakpoint* σε μια συγκεκριμένη γραμμή γράφουμε **break** ακολουθούμενο από τον αριθμό της γραμμής. Για παράδειγμα, **break 22** θέτει ένα *breakpoint* στη γραμμή 22 (η εκτέλεση θα διακοπεί ακριβώς μετά τη γραμμή 21).

## Πώς να εκτυπώσετε την τιμή μιας έκφρασης

Χρησιμοποιήστε την εντολή του gdb `print` (ή `p`) ακολουθούμενη από την έκφραση την τιμή της οποίας επιθυμείτε να δείτε. Για παράδειγμα,

```
print table[i] εκτυπώνει τα περιεχόμενα στη θέση i ενός πίνακα table
```

```
print price εκτυπώνει την τιμή της μεταβλητής price
```

```
print len*win υπολογίζει και μετά εκτυπώνει το γινόμενο len*win
```

Μπορείτε να εκτυπώσετε την τιμή μιας μεταβλητής στη δυαδική της μορφή γράφοντας `print/t` (ή `p/t`) και μετά το όνομα της μεταβλητής. Για τη δεκαεξαδική μορφή γράφετε αντίστοιχα `print/x` (ή `p/x`).

## Πώς να συνεχίσετε την εκτέλεση

Εφόσον η εκτέλεση έχει σταματήσει σε κάποιο σημείο, χρησιμοποιήστε την εντολή του gdb `next` (ή `n`) για να εκτελέσετε την επόμενη εντολή του προγράμματος.

Η εντολή του gdb `step` (ή `s`) είναι παρόμοια με τη `next` με την εξής σημαντική διαφορά: Αν η επόμενη εντολή του προγράμματος που εκτελείται μέσω gdb είναι κλήση προς μια συνάρτηση, τότε γράφοντας `next` θα εκτελεστεί σε ένα βήμα η συνάρτηση και θα πάμε άμεσα στην εντολή του προγράμματος που ακολουθεί μετά την κλήση στη συνάρτηση. Αν όμως γράψουμε `step` τότε το gdb θα "μπει" μέσα στη συνάρτηση και θα εκτελεί μία-μία της εντολές που βρίσκονται μέσα σε αυτή.

Τέλος, η εντολή του gdb `continue` (ή `c`) εκτελεί το πρόγραμμα χωρίς διακοπές από το σημείο που είχαμε σταματήσει μέχρι το επόμενο breakpoint ή μέχρι να τερματίσει (αν δεν υπάρχει μεταγενέστερο breakpoint)

## Πώς να βρείτε τη σειρά κλήσεων συναρτήσεων μέχρι ένα σημείο (backtrace)

Μπορείτε από οποιοδήποτε σημείο του προγράμματος να δείτε τη σειρά κλήσεων συναρτήσεων μέχρι εκείνο το σημείο. Αυτό γίνεται με την εντολή `backtrace` (ή `bt`) η οποία εμφανίζει μια αριθμημένη λίστα των stack frames μέχρι το τρέχον σημείο, ξεκινώντας από το τελευταίο που δημιουργήθηκε, μέχρι και το πρώτο (της `main`). Κάθε γραμμή της λίστας ξεκινά από έναν ακέραιο που χαρακτηρίζει το κάθε frame, τον οποίο μπορείτε να χρησιμοποιήσετε σε συνδυασμό με την εντολή `frame` (ή `f`) για να μεταβείτε σε αυτό, για παράδειγμα `f 3` για να μεταβείτε στο frame #3

Η εντολή `backtrace` είναι ιδιαίτερος χρήσιμη αν το πρόγραμμά σας έχει τερματίσει πρόωρα (και άδοξα) λόγω `segmentation fault`.

## Πώς να τερματίσετε το gdb

Γράψτε την εντολή `quit` (ή `q`). Αν δεν έχει ολοκληρωθεί η εκτέλεση του προγράμματος που ελέγχετε, θα σας ζητήσει επιβεβαίωση.

## Περισσότερες πληροφορίες.

Το επίσημο `documentation` του gdb βρίσκεται [εδώ](#), αλλά μπορείτε να βρείτε πολλές πληροφορίες κι από μια απλή αναζήτηση για `gdb tutorial`. Επίσης, μπορείτε μέσα από το περιβάλλον του gdb να γράψετε `help` είτε μόνο του είτε ακολουθούμενο από οποιαδήποτε εντολή του gdb για την οποία χρειάζεστε βοήθεια.

Χρήστες `macOS` μπορούν να χρησιμοποιήσουν τον debugger `lldb` ο οποίος είναι παρόμοιος. Για την αντιστοιχία εντολών ανάμεσα σε `gdb` σε `lldb` δείτε [εδώ](#). Οι πιο ουσιαστικές διαφορές είναι ότι για τον ορισμό ενός breakpoint γράφουμε `break` (όχι `b`) και για την μετάβαση σε ένα frame γράφουμε `frame select` αριθμός (όχι `f` αριθμός)