

ΕΡΓΑΣΤΗΡΙΟ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι, ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2011-2012**Μάθημα: Debugging****Εισαγωγή**

Ο debugger 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 μόνο μέχρι ενός σημείου
- Παρακολουθεί τις τιμές διαφόρων μεταβλητών και εκφράσεων
- Δείχνει τη σειρά συναρτήσεων που κλήθηκαν μέχρι ένα σημείο.

Βασικές λειτουργίες**Breakpoints**

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

Variable Watch

Ο debugger μας δίνει πάντα τη δυνατότητα να παρακολουθήσουμε την τιμή μεταβλητών και εκφράσεων που προσδιορίζουμε.

Step-by-step execution

Η πιο χρήσιμη λειτουργία του debugger είναι η δυνατότητα εκτέλεσης του προγράμματος Π γραμμή προς γραμμή. Αυτό, σε συνδυασμό με την παρακολούθηση μεταβλητών μας επιτρέπει να επιβεβαιώσουμε σε κάθε βήμα ότι το πρόγραμμα λειτουργεί όπως πρέπει (ή να ανακαλύψουμε ότι υπάρχει λάθος).

gdb

Ο debugger που θα χρησιμοποιήσουμε είναι το gdb και το αντίστοιχο Graphical User Interface στο εργαστήριο, το kdbg (ένα άλλο GUI που λειτουργεί με παρόμοιο τρόπο είναι το ddd). Τα παρακάτω βήματα περιγράφουν πως να χρησιμοποιήσετε το gdb στη γραμμή εντολής (command-line interface) και μέσω του kdbg.

Βήμα 1

Κατεβάστε το `debuggindfiles.tgz` από τη σελίδα του εργαστηρίου και αποσυμπίεστε το.

Βήμα 2

Μεταγλωττίστε το `debug0.c` χρησιμοποιώντας την εντολή

`gcc -Wall -g debug0.c -o debug0`

Ο διακόπτης **-g** είναι απαραίτητος. Με αυτόν, ο μεταγλωττιστής αποθηκεύει στο εκτελέσιμο (**debug0**) επιπλέον πληροφορίες (πχ. τον πηγαίο κώδικα) οι οποίες είναι απαραίτητες για να γίνει debugging. Προσωρινώς αγνοείστε τα warnings που εμφανίζονται.

Το **debug0.c** ζητά από τον χρήστη έναν ακέραιο, ανιχνεύει αν αυτός είναι άρτιος ή περιττός κι εκτυπώνει ένα σχετικό μήνυμα. Δοκιμάστε να τρέξετε το εκτελέσιμο με διάφορες εισόδους. Θα δείτε ότι σε κάποιες περιπτώσεις δε βγάζει σωστά αποτελέσματα. ΑΚΟΜΗ ΚΙ ΑΝ ΒΡΕΙΤΕ ΑΜΕΣΩΣ ΤΟ ΛΑΘΟΣ **ΜΗΝ** ΤΟ ΔΙΟΡΘΩΣΕΤΕ. Ο σκοπός είναι να μάθετε πώς να χρησιμοποιείτε το gdb για την εύρεση λαθών.

Βήμα 3

Τώρα είστε έτοιμοι να τρέξετε τον debugger για να ανακαλύψετε πού είναι το λάθος (δυστυχώς ο debugger δε μπορεί να σας πει πώς να το διορθώσετε :)) Αν χρησιμοποιείτε περιβάλλον γραμμής εντολών (command line interface) ακολουθήστε τις μπλε οδηγίες. Αν χρησιμοποιείτε το kdbg ακολουθήστε τις κόκκινες οδηγίες.

[**command-line**] Γράψτε **gdb** και πατείστε Enter.

[**kdbg**] Το kdbg βρίσκεται κάτω απο το μενού Development

Έχουμε μπει στο περιβάλλον του debugger και το επόμενο βήμα είναι να φορτώσουμε το εκτελέσιμο:

[**command-line**] Γράψτε **file debug0** και πατείστε Enter.

[**kdbg**] Επιλέξτε File →Executable και μετά βρείτε και επιλέξτε το debug0 (ΟΧΙ το debug0.c) Το βασικό παράθυρο τώρα περιέχει τον πηγαίο κώδικα και θα εμφανιστεί και ένα παράθυρο με τίτλο Output

Επειδή δεν έχουμε ιδέα πού μπορεί να είναι το λάθος, θα θέσουμε ένα breakpoint στην αρχή του προγράμματος.

[**command-line**] Γράψτε **break main** και πατήστε Enter. Υπάρχουν διάφοροι τρόποι να θέσει κανείς breakpoints. Η γενική σύνταξη είναι

break <σημείο>

όπου το <σημείο> μπορεί να είναι το όνομα μιας συνάρτησης (όπως σε αυτή την περίπτωση), μια συγκεκριμένη γραμμή σε ένα αρχείο (πχ. break debug0.c:28), κ.α.

[**kdbg**] Πηγαίνετε στη γραμμή του κώδικα αμέσως μετά το **int main ()** και κάντε κλικ στο αριστερό μέρος (πιο αριστερά από το +). Θα εμφανιστεί μια κόκκινη κουκκίδα στο σημείο αυτό.

Είμαστε τώρα έτοιμοι να τρέξουμε το πρόγραμμα μέχρι το breakpoint.

[[command-line](#)] Γράψτε **run** και πατείστε Enter. Το πρόγραμμα θα σταματήσει στην αρχή του και το gdb θα σας δείξει τη γραμμή που πρόκειται να εκτελεστεί στο επόμενο βήμα.

[[kdbg](#)] Κάντε κλικ στο εικονίδιο που δείχνει μια σελίδα με ένα βελάκι προς τα κάτω (ή επιλέξτε Run). Το πρόγραμμα θα ξεκινήσει και θα σταματήσει στο breakpoint. Ένας πράσινος δείκτης τίθεται στην αρχή της γραμμής που πρόκειται να εκτελεστεί στο επόμενο βήμα.

Πριν προχωρήσουμε, θέλουμε να επιλέξουμε κάποιες μεταβλητές για παρακολούθηση.

[[command-line](#)] Οποιαδήποτε στιγμή μπορείτε να γράψετε **p *expr*** όπου *expr* είναι μια μεταβλητή ή έκφραση για να δείτε την τιμή της. Το p σημαίνει print.

[[kdbg](#)] Επιλέξτε View→Watched variables. Θα ανοίξει ένα παράθυρο στο οποίο μπορείτε να προσθέσετε τις εκφράσεις ή ονόματα μεταβλητών που επιθυμείτε να παρακολουθήσετε.

Επιπλέον μπορούμε να ελέγχουμε τον τύπο μιας μεταβλητής με την εντολή (στο command-line) **whatis**. Για παράδειγμα, **whatis apotelesma**.

Τώρα είμαστε έτοιμοι να εκτελέσουμε το πρόγραμμα γραμμή-γραμμή:

[[command-line](#)] Γράψτε **s** και πατείστε Enter (το s σημαίνει step). Θα εκτελεστεί η επόμενη γραμμή. Αν θέλετε, χρησιμοποιείτε την εντολή **p** για να δείτε πώς έχουν αλλάξει οι μεταβλητές.

[[kdbg](#)] Πατείστε το κουμπί Step Into το οποίο απεικονίζει δύο αγκύλες με ένα βελάκι να δείχνει ανάμεσα τους.

Υπάρχει μια ιδιαιτερότητα όταν η γραμμή προς εκτέλεση περιέχει κλήση σε κάποια συνάρτηση, όπως για παράδειγμα η γραμμή **apotelesma = isEven(arithmos);**

Έχουμε δύο επιλογές εδώ. Η μία είναι να ζητήσουμε από τον debugger να εκτελέσει ολόκληρη την **isEven(arithmos)** σε ένα βήμα και να συνεχίσει μετά με τη γραμμή που ακολουθεί (if (apotelesma κτλ.). Η άλλη επιλογή είναι να ζητήσουμε να μας δείξει λεπτομερώς, γραμμή-γραμμή και την εκτέλεση της **isEven**, δηλαδή να πάει στην γραμμή **if (num % 2 == 0)** (η πρώτη γραμμή της **isEven**).

Αν επιλέξουμε **s** (στο command-line) ή Step Into (στο kdbg) τότε θα ακολουθηθεί η δεύτερη επιλογή. Για να διαλέξουμε το πρώτο σενάριο, οι αντίστοιχες εντολές είναι **n** (ή next) στο command-line και Step Over (άγκιστρα με ένα βελάκι που περνάει από πάνω) στο kdbg.

Έχετε τώρα όλα τα εργαλεία που χρειάζεστε για να βρείτε το λάθος. Για πιο πολλές πληροφορίες για το gdb απευθυνθείτε στο <http://sourceware.org/gdb/documentation/> και για γρήγορη αναφορά στις εντολές του gdb στο <http://www.yolinux.com/TUTORIALS/GDB-Commands.html>

Δοκιμάστε το gdb στα άλλα δύο αρχεία που βρίσκονταν στο debuggingfiles.tgz