



**ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΜΑΚΕΔΟΝΙΑΣ**

Συστήματα Παράλληλης και Κατανεμημένης Επεξεργασίας

Ενότητα: ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ Νο:16 INSPECTOR

Δρ. Μηνάς Δασυγένης

mdasyg@ieee.org

Τμήμα Μηχανικών Πληροφορικής και Τηλεπικοινωνιών

Εργαστήριο Ψηφιακών Συστημάτων και Αρχιτεκτονικής Υπολογιστών

<http://arch.ict.e.uowm.gr/mdasyg>

Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Για εκπαιδευτικό υλικό, όπως εικόνες, που υπόκειται σε άλλου τύπου άδειας χρήσης, η άδεια χρήσης αναφέρεται ρητώς.



Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ψηφιακά Μαθήματα του Πανεπιστημίου Δυτικής Μακεδονίας**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



Περιεχόμενα

1. Σκοπός της άσκησης	4
2. Παραδοτέα	4
3. Ανάλυση σφαλμάτων μνήμης (Οδηγός 1)	4
3.1 Βιβλιογραφία	8
4. Ανάλυση σφαλμάτων Thread (Οδηγός 2)	8
4.1 Βιβλιογραφία	12

1. Σκοπός της άσκησης

- Ανίχνευση σφαλμάτων μνήμης με τη χρήση του εργαλείου της Intel, Intel inspector XE 2013.

Σε αυτήν την άσκηση θα γίνει καθοδήγηση στην χρήση του εργαλείου για την εύρεση και διόρθωση λαθών που οφείλονται στην μνήμη και σε thread.

2. Παραδοτέα

(A) 4 ερωτήσεις

(C) 2 ασκήσεις

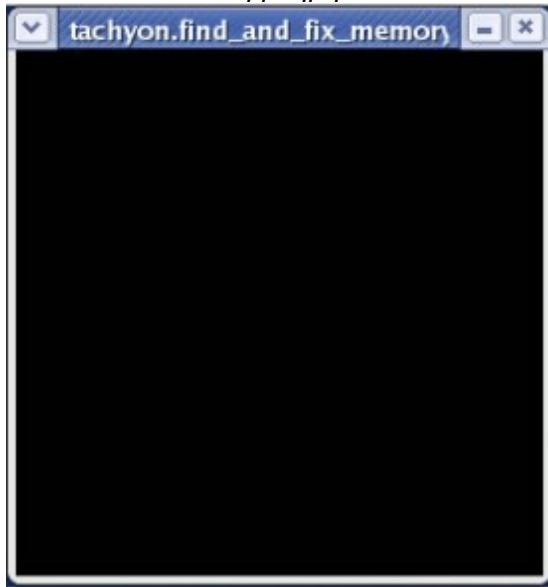
3. Ανάλυση σφαλμάτων μνήμης (Οδηγός 1)

Σε αυτόν τον οδηγό θα χρησιμοποιήσετε το εργαλείο **Intel Inspector XE 2013**, προκειμένου να αναλύσετε έναν κώδικα και να βρείτε σημεία του κώδικα τα οποία δημιουργούν σφάλματα λόγω κακής διαχείρισης της μνήμης.

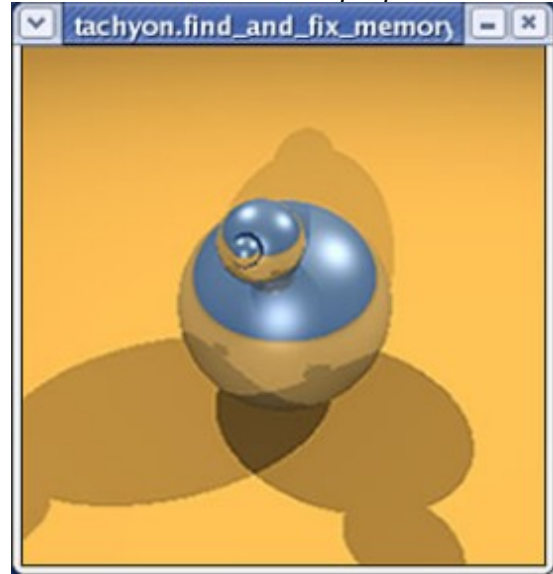
Διαδικασία Εκτέλεσης Άσκησης

1. Για να χρησιμοποιήσετε το εργαλείο **Intel inspector**, θα πρέπει να αρχικοποιήσετε το περιβάλλον, δηλαδή να τεθούν κάποιες μεταβλητές, να τροποποιηθεί η διαδρομή αναζήτησης των εκτελέσιμων εφαρμογών για να προστεθεί η τοποθεσία του inspector.
2. Ανοίξτε ένα παράθυρο τερματικού. Για την αρχικοποίηση δώστε μια από τις παρακάτω εντολές, όπου **<install-dir>** η διαδρομή στην οποία εγκαταστήσατε το inspector:
 - **source <install-dir>/inspxe-vars.sh**
 - **source <install-dir>/inspxe-vars.csh**
3. Ακόμα θέστε την μεταβλητή (*var*) περιβάλλοντος **EDITOR** ή **VISUAL** στον αγαπημένο σας κειμενογράφο (πχ: *nano, vi, gedit, ...*)
 - Εκτελέστε: **export <var>=<editor>**
4. **(C1)** Εκτελέστε **echo \$EDITOR** ή **echo \$VISUAL**, ανάλογα με το ποια μεταβλητή θέσατε στο προηγούμενο βήμα. Δώστε screenshot.
5. Τώρα μπορείτε να ξεκινήσετε την εφαρμογή με: **inspxe-gui**.
6. Σε ένα δεύτερο παράθυρο τερματικού, αντιγράψτε το αρχείο **<install-dir>/samples/<locale>/C++/tachyon_insp_xe.tgz** σε έναν φάκελο στον οποίο έχετε δικαιώματα εγγραφής.
7. Αποσυμπίστε το **tachyon_insp_xe.tgz** (πχ *tar -xvzf <αρχείο>*).
8. Μπείτε στον καινούργιο κατάλογο, **tachyon_insp_xe**, που δημιουργήθηκε από την αποσυμπίηση και εκτελέστε την εντολή καθαρισμού και μεταγλώττισης.
 - Δώστε: **make clean; make**

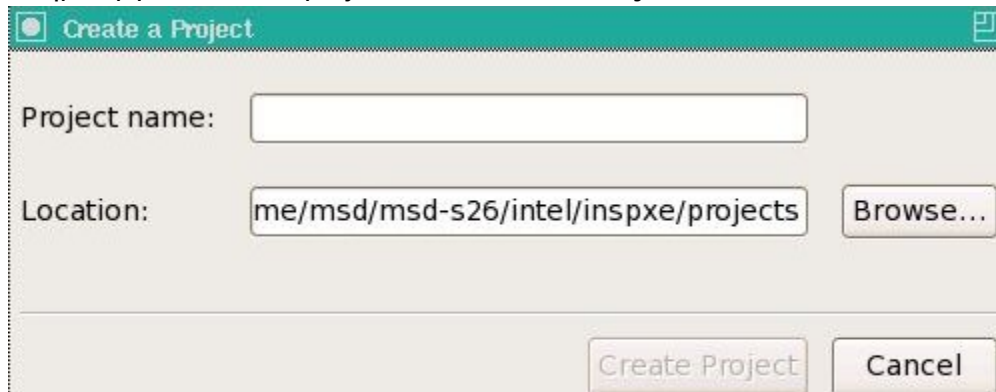
Αρχική εμφάνιση εφαρμογής.
Κενό παράθυρο λόγω προβλημάτων
στη μνήμη



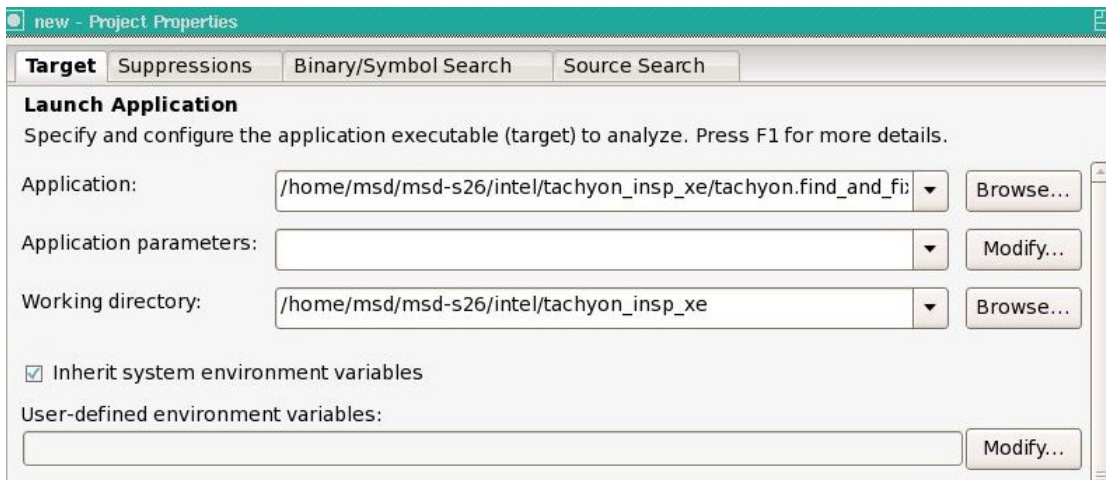
Εμφάνιση, εφόσον
λυθούν κάποια σφάλματα



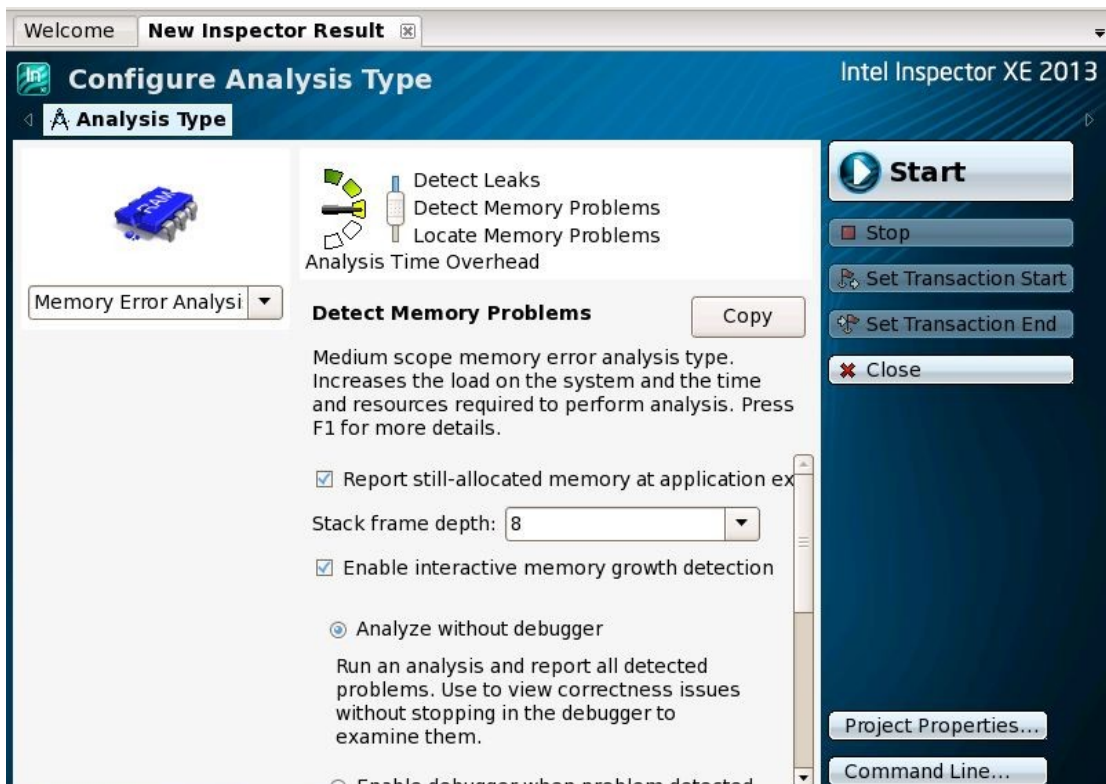
9. Αφού τελειώση επιτυχώς το compile, δώστε στο τερματικό **./tachyon.find_and_fix_memory_errors** για να εκτελέσετε την εφαρμογή.
10. **(A1)** Βεβαιωθείτε ότι εμφανίστηκε το παραπάνω παράθυρο. Δώστε screenshot.
11. Και τώρα αφού έχει γίνει επιτυχώς το compile, πάτε στο Intel inspector και δημιουργείτε ένα νέο project: **File > New > Project...**

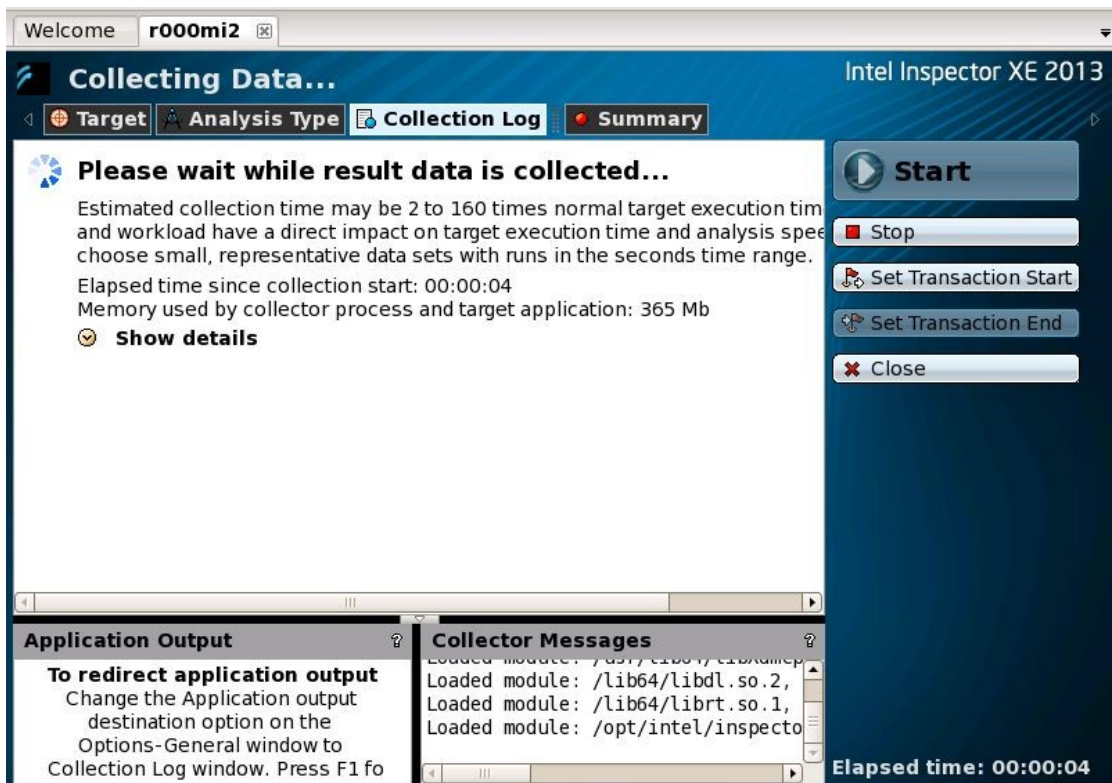


12. Αφού δώσετε ένα όνομα στο Project και πατήσετε **“create project”** στο νέο παράθυρο που εμφανίζεται, φροντίστε στο πλαίσιο **“Application”** να υπάρχει η εφαρμογή **“tachyon.find_and_fix_memory_errors”** και στο **“working directory”** ο κατάλογος εργασίας σας.

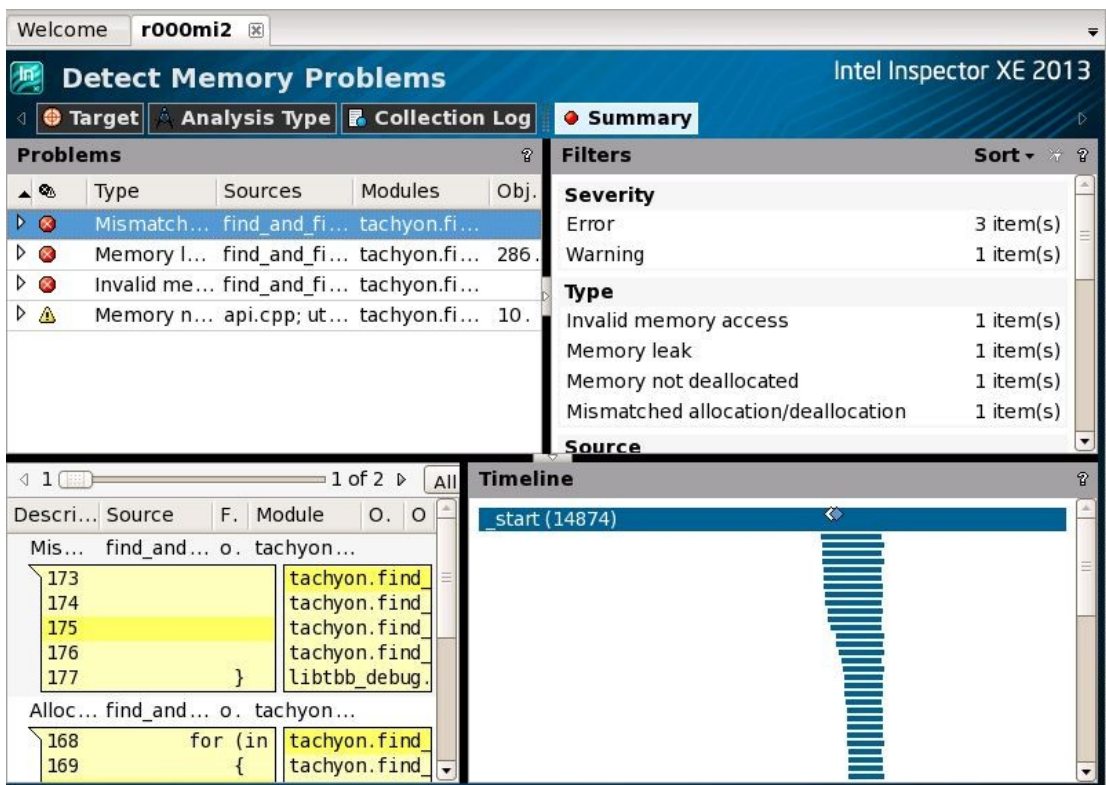


13. **(A2)** Βεβαιωθείτε ότι δώσατε τα σωστά στοιχεία. Δώστε screenshot.
14. Πατήστε OK.
15. Για να εμφανιστεί το παράθυρο ανάλυσης πάτε: **File > New > Analysis...**
16. Στο αναπτυσσόμενο μενού αριστερά, πρέπει να έχετε επιλεγμένη την ανάλυση **“Memory Error Analysis”** και επιλέγετε το **“Detect Memory Problems”**. Τέλος πατήστε **Start**.





17. Αφού ολοκληρωθεί η ανάλυση εμφανίζεται το παρακάτω παράθυρο το οποίο αναφέρει τα προβλήματα τα οποία εντοπίστηκαν.



18. **(A3)** Δώστε το screenshot που δείχνει τα προβλήματα που βρήκατε.
19. Πρέπει να υπάρχει η γραμμή **Mismatched allocation / deallocation**, κάντε διπλό κλικ στη γραμμή.
20. Κάνοντάς δεξί κλικ στην καρτέλα του πηγαίου ή στην καρτέλα του disassembly και μετά Explain Problem, παίρνετε περισσότερες πληροφορίες σχετικά με το πρόβλημα.
21. Κάνοντας διπλό κλικ πάνω στην γραμμή που είναι το λάθος ανοίγει ο editor, όπου μπορείτε να κάνετε αλλαγές.
22. **(C2)** Αφού ψάξετε για το πρόβλημα, μέσω του **“Explain Problem”** ή μέσω διαδικτύου και καταλάβετε που οφείλεται, βρείτε ποιες εντολές δημιουργούν το πρόβλημα και διορθώστε τις. Δώστε screenshot δείχνοντας την αλλαγή που κάνατε.
23. Αφού κάνετε τις αλλαγές σας αποθηκεύστε.
24. Στο τερματικό που κάνατε προηγουμένως compile, βεβαιωθείτε ότι είστε στον κατάλογο **tachyon_insp_xe** και ξανακάνετε **make**.
25. Ξανατρέχετε την ανάλυση πηγαίνοντας **File > New > Memory Error Analysis / Detect Memory Problems** και ελέγξτε ότι το πρόβλημα έχει διορθωθεί.
26. **(A4)** Δώστε το screenshot που δείχνει το παράθυρο με τα εντοπισμένα προβλήματα και ότι διορθώσατε το **Mismatched allocation / deallocation**.
27. Επαναληπτικά μπορείτε να λύσετε όλα τα προβλήματα επιλέγοντας μια άλλη γραμμή στο βήμα 21.

3.1 Βιβλιογραφία

http://software.intel.com/sites/products/documentation/doclib/stdxe/2013/inspectorxe/lin/lin_tutorial_memory_cpp/index.htm

http://software.intel.com/sites/products/documentation/doclib/stdxe/2013/inspectorxe/lin/lin_tutorial_memory_cpp/tutorial_insp_lin.pdf

4. Ανάλυση σφαλμάτων Thread (Οδηγός 2)

Σε αυτόν τον οδηγό θα χρησιμοποιήσετε το εργαλείο **Intel Inspector XE 2013**, προκειμένου να αναλύσετε έναν κώδικα και να βρείτε σημεία του κώδικα τα οποία δημιουργούν σφάλματα λόγω μη σωστής χρήσης των thread.

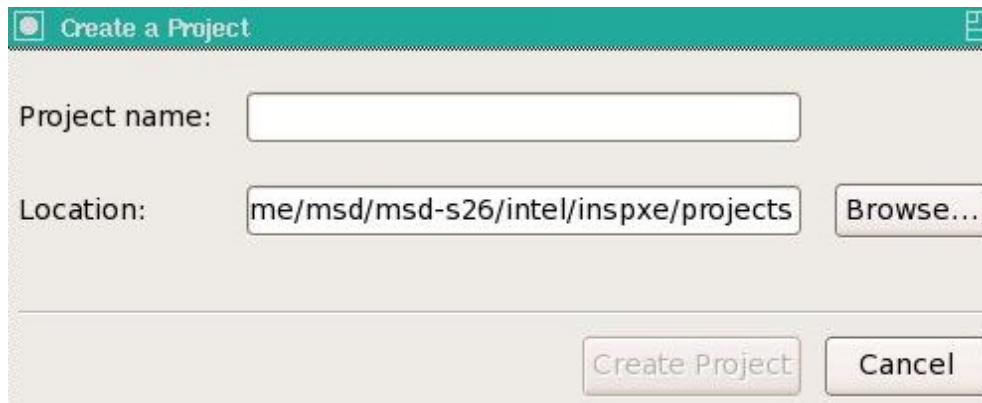
Διαδικασία Εκτέλεσης Άσκησης

1. Για να χρησιμοποιήσετε το εργαλείο **Intel inspector**, θα πρέπει να αρχικοποιήσετε το περιβάλλον, δηλαδή να τεθούν κάποιες μεταβλητές, να τροποποιηθεί η διαδρομή αναζήτησης των εκτελέσιμων εφαρμογών για να προστεθεί η τοποθεσία του inspector.

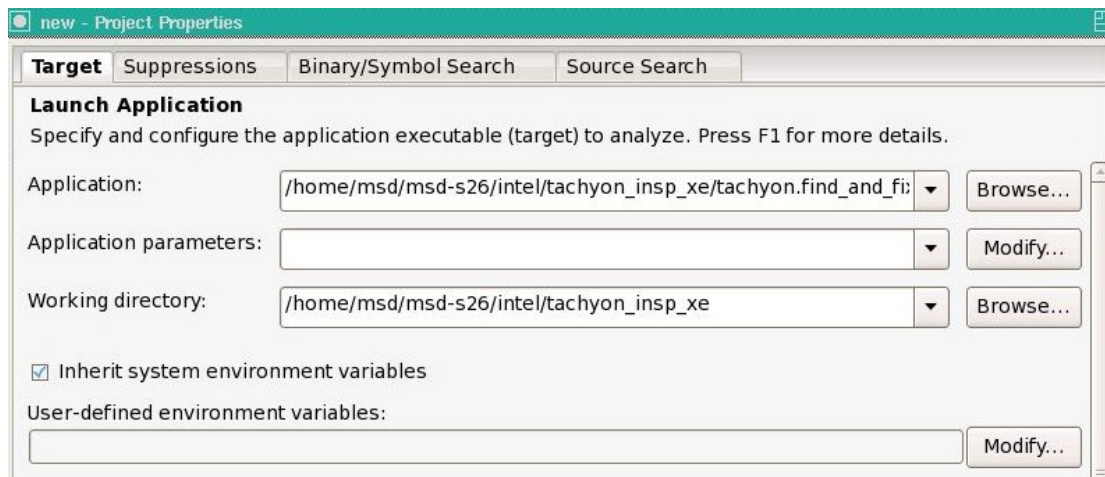
2. Ανοίξτε ένα παράθυρο τερματικού. Για την αρχικοποίηση δώστε μια από τις παρακάτω εντολές, όπου **<install-dir>** η διαδρομή στην οποία εγκαταστήσατε το `inspector`:
 - **source <install-dir>/inspxe-vars.sh**
 - **source <install-dir>/inspxe-vars.csh**
3. Ακόμα θέστε την μεταβλητή (*var*) περιβάλλοντος **EDITOR** ή **VISUAL** στον αγαπημένο σας κειμενογράφο (πχ: *nano, vi, gedit, ...*)
 - Εκτελέστε: **export <var>=<editor>**
4. **(C1)** Εκτελέστε **echo \$EDITOR** ή **echo \$VISUAL**, ανάλογα με το ποια μεταβλητή θέσατε στο προηγούμενο βήμα. Δώστε screenshot.
5. Τώρα μπορείτε να ξεκινήσετε την εφαρμογή με: **inspxe-gui**
6. Σε ένα δεύτερο παράθυρο τερματικού, αντιγράψτε το αρχείο **<install-dir>/samples/<locale>/C++/tachyon_insp_xe.tgz** σε έναν φάκελο στον οποίο έχετε δικαιώματα εγγραφής.
7. Αποσυμπίστε το **tachyon_insp_xe.tgz** (πχ *tar -xvzf <αρχείο>*)
8. Μπείτε στον καινούργιο κατάλογο, **tachyon_insp_xe**, που δημιουργήθηκε από την αποσυμπίηση και εκτελέστε την εντολή καθαρισμού και μεταγλώττισης.
 - Δώστε: `make clean; make`
9. Αφού τελείωση επιτυχώς το `compile`, δώστε στο τερματικό **./tachyon.find_and_fix_threading_errors** για να εκτελέσετε την εφαρμογή.



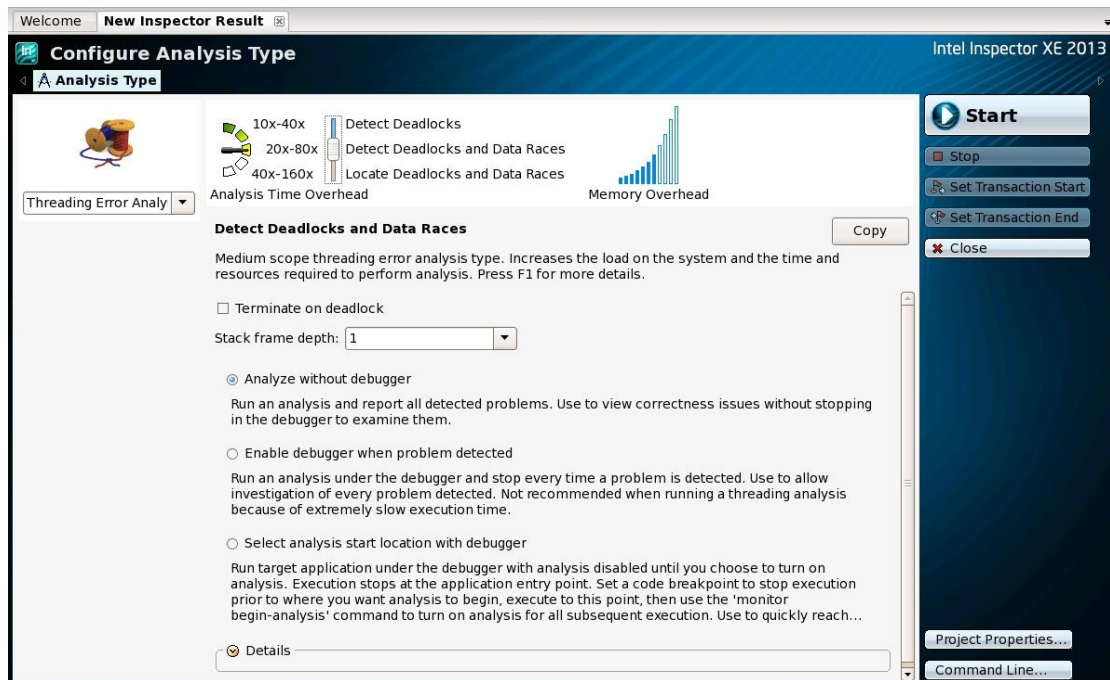
10. **(A1)** Βεβαιωθείτε ότι εμφανίστηκε το παραπάνω παράθυρο. Δώστε screenshot.
11. Και τώρα αφού έχει γίνει επιτυχώς το `compile`, πάτε στο **Intel inspector** και δημιουργείτε ένα νέο project: **File > New > Project...**



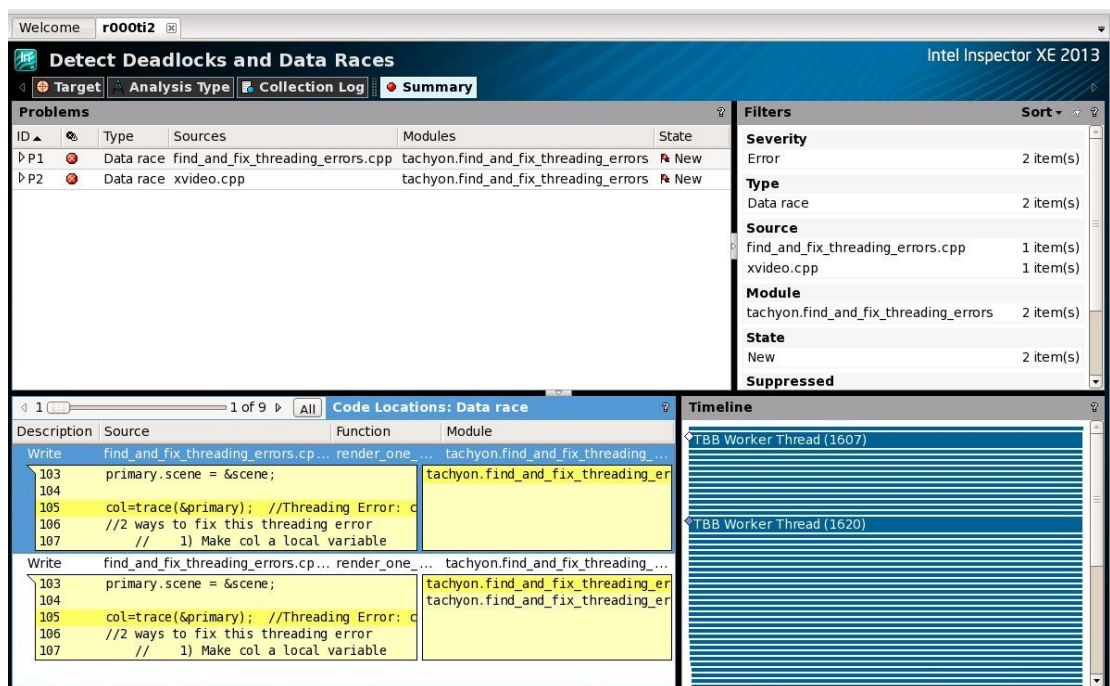
12. Αφού δώσετε ένα όνομα στο Project και πατήσετε **“create project”** στο νέο παράθυρο που εμφανίζεται, φροντίστε στο πλαίσιο **“Application”** να υπάρχει η εφαρμογή **“tachyon.find_and_fix_threading_errors”** και στο **“working directory”** ο κατάλογος εργασίας σας.



13. **(A2)** Βεβαιωθείτε ότι δώσατε τα σωστά στοιχεία. Δώστε screenshot.
14. Πατήστε OK.
15. Για να εμφανιστεί το παράθυρο ανάλυσης πάτε: **File > New > Analysis...**
16. Στο αναπτυσσόμενο μενού αριστερά, πρέπει να έχετε επιλεγμένη την ανάλυση **“Threading Error Analysis”** και επιλέγετε το **“Detect Deadlocks and Data Races”**. Τέλος πατήστε **Start**.



17. Αφού ολοκληρωθεί η ανάλυση εμφανίζεται το παρακάτω παράθυρο το οποίο αναφέρει τα προβλήματα τα οποία εντοπίστηκαν.



18. (A3) Δώστε το screenshot που δείχνει τα προβλήματα που βρήκατε.
19. Κάνουμε διπλό κλικ στο πρώτο πρόβλημα.
20. Κάνοντάς δεξί κλικ στην καρτέλα του **πηγαίου** ή στην καρτέλα του **dissassembly** και μετά **Explain Problem**, παίρνουμε πληροφορίες σχετικά με το πρόβλημα.

21. Κάνοντας διπλό κλικ πάνω στην γραμμή που είναι το λάθος ανοίγει ο editor.
22. **(C2)** Αφού ψάξετε για το πρόβλημα, μέσω του “**Explain Problem**” ή μέσω διαδικτύου και καταλάβετε που οφείλεται, βρείτε σε ποιες γραμμές-εντολές οφείλεται το πρόβλημα και διορθώστε τα. Εξηγήστε ποια αλλαγή κάνατε και δώστε screenshot δείχνοντας την αλλαγή που κάνατε.
23. Αφού κάνετε τις αλλαγές σας αποθηκεύστε.
24. Στο τερματικό που κάνατε προηγουμένως compile, βεβαιωθείτε ότι είστε στον κατάλογο **tachyon_insp_xe** και ξανακάνετε **make**.
25. Ξανατρέχετε την ανάλυση πηγαίνοντας
File> New> Threading Error Analysis / Detect Deadlocks and Data Races
και ελέγξτε ότι το πρόβλημα έχει διορθωθεί.
26. **(A4)** Δώστε το screenshot που δείχνει το παράθυρο με τα εντοπισμένα προβλήματα και ότι διορθώσατε το **πρώτο Data Race**.
27. Επαναληπτικά μπορείτε να λύσετε όλα τα προβλήματα επιλέγοντας μια άλλη γραμμή στο βήμα 21.

4.1 Βιβλιογραφία

http://software.intel.com/sites/products/documentation/doclib/stdxe/2013/inspectorxe/lin/lin_tutorial_threading_cpp/index.htm

http://software.intel.com/sites/products/documentation/doclib/stdxe/2013/inspectorxe/lin/lin_tutorial_threading_cpp/tutorial_insp_lin.pdf