



ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Οδηγός Χρήσης Eclipse IDE for C/C++ Developers

Δημήτρης Ζεϊναλιπούρ

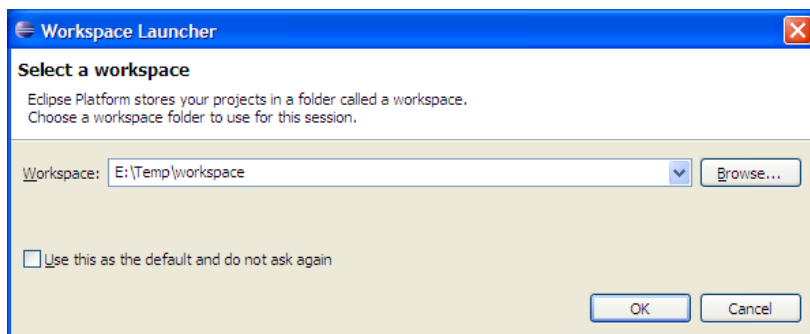
Βήμα 1: Εγκατάσταση eCclipse σε Προσωπικό ΗΥ

Από το δικτυακό χώρο του Eclipse (<http://www.eclipse.org/downloads/>) κατεβάζουμε την έκδοση του Eclipse που είναι επαυξημένη με τα εργαλεία ανάπτυξης για C/C++. Μετά την αποσυμπίεση στον τοπικό μας δίσκο, το **Eclipse** είναι έτοιμο για χρήση.

Σημείωση: Οι πιο κάτω οδηγίες χρήσης έχουν ετοιμαστεί για χρήση του eCclipse σε περιβάλλον **Windows**¹ με χρήση του cygwin. Αντίστοιχες οδηγίες χρήσης μπορούν να ακολουθηθούν και για εγκαταστάσεις οι οποίες έχουν γίνει σε Linux².

Βήμα 2: Εξοικείωση με το περιβάλλον του Eclipse

Στο Eclipse ο χρήστης έχει μία περιοχή στο δίσκο στην οποία «αποθηκεύει» όλα τα projects που αναπτύσσει. Κατά την εκκίνηση το Eclipse μας ζητάει να επιβεβαιώσουμε αυτή την περιοχή (ή, αν υπάρχει οποιοδήποτε πρόβλημα, μας καθοδηγεί να δηλώσουμε νέα).

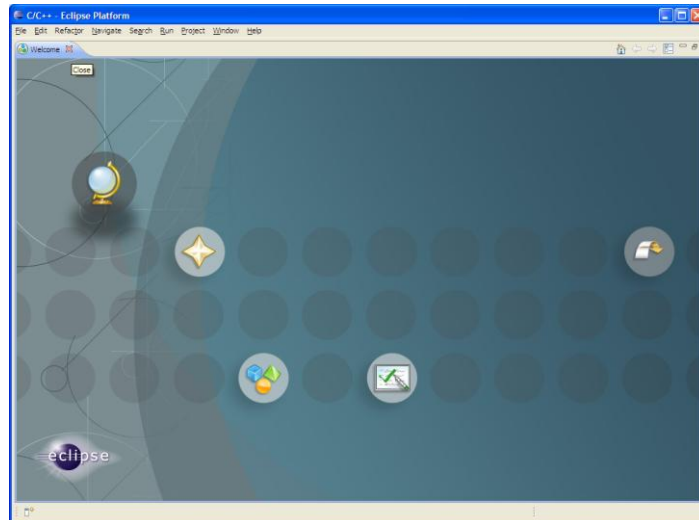


Εικόνα 1

Στη συνέχεια το Eclipse ξεκινά με τη βασική οθόνη που περιέχει οδηγίες χρήσης και tutorials. Στην οθόνη αυτή μπορούμε πάντα να φτάνουμε επιλέγοντας **Help > Welcome**, οπότε προς το παρόν την κλείνουμε (κλείνοντας το αντίστοιχο TAB).

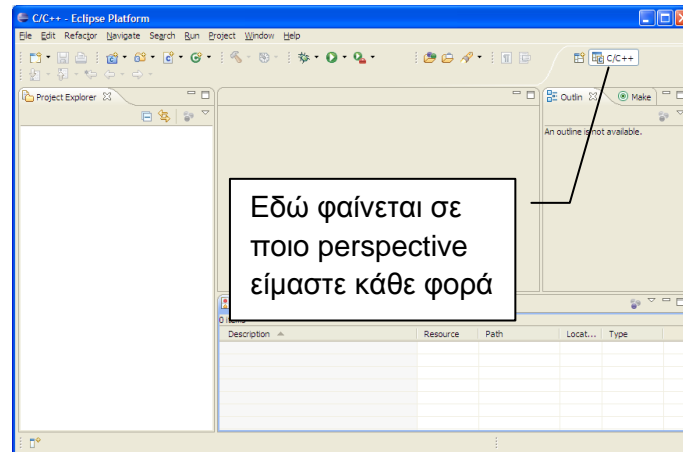
¹ Εάν προτιθήσετε να εγκαταστήσετε το eCclipse σε Windows τότε συνιστάται η εκ' των προτέρων εγκατάσταση του cygwin (<http://www.cygwin.com/>), το οποίο είναι ένα περιβάλλον Linux για Windows

² Προτρέπεται η χρήση του eCclipse στο περιβάλλον Linux.



Εικόνα 2

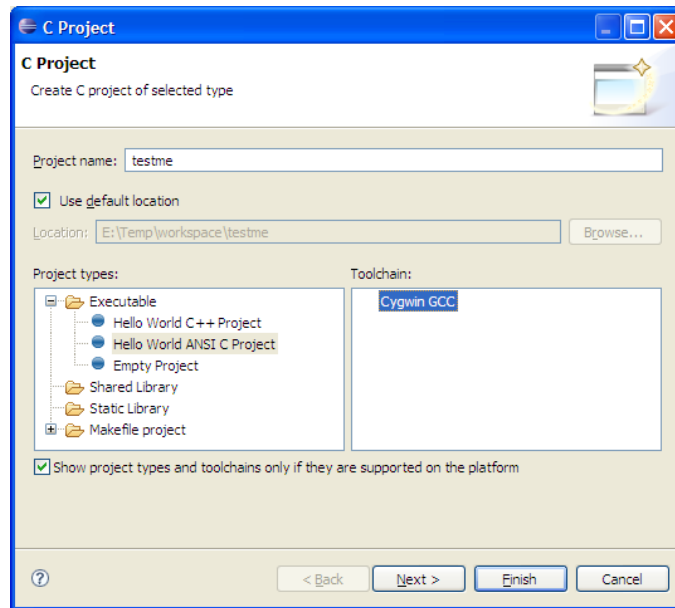
Τώρα εμφανίζεται η βασική οθόνη του Eclipse, με τις μπάρες εργαλείων, τα βοηθητικά Tabs, κλπ. Εδώ αξίζει να σημειωθεί πως η συγκεκριμένη μπάρα εργαλείων του Eclipse εμφανίζεται επειδή είμαστε έτοιμοι να κάνουμε ανάπτυξη σε C/C++ (αφού κατεβάσαμε το Eclipse CDT), και άρα είμαστε, όπως λέγεται στην ορολογία του Eclipse, σε **C/C++ perspective**. Αυτό μπορούμε να το επιβεβαιώσουμε παρατηρώντας την σχετική περιοχή της οθόνης του Eclipse. Στην ανάπτυξη που θα κάνουμε θα χρησιμοποιήσουμε και ένα άλλο perspective, το **Debug perspective**, που θα ενεργοποιήσει διαφορετική μπάρα εργαλείων και θα φανεί στη σχετική περιοχή (περισσότερα στη συνέχεια).



Εικόνα 3

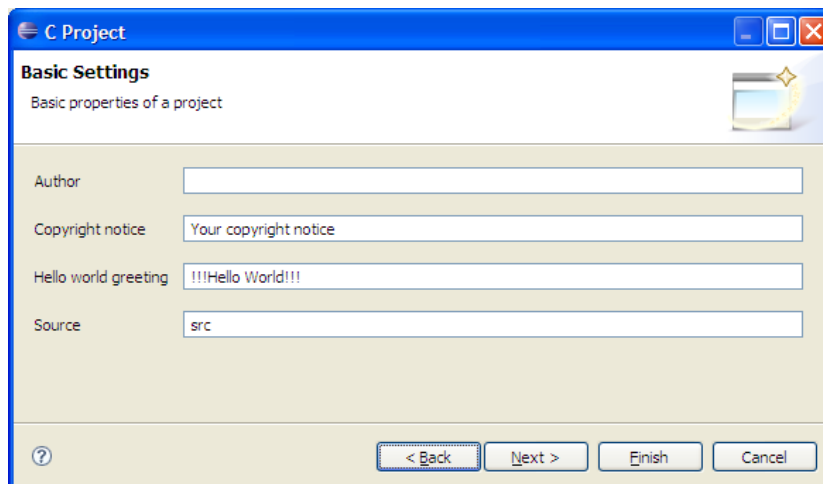
Βήμα 3: Ένα απλό project σε Eclipse

Ξεκινάμε ένα νέο project επιλέγοντας **File > New > C Project** και το Eclipse απαντά ζητώντας ορισμένες πληροφορίες παραπάνω για το project που θέλουμε να δημιουργήσουμε. Εμείς επιλέγουμε το όνομα **testme**, και επιπλέον συσχετίζουμε την ανάπτυξή μας με το toolchain που έχουμε ήδη εγκαταστήσει στα προηγούμενα βήματα και δηλώνουμε πως θέλουμε ένα απλό project μόνο (η διαχείριση πιο σύνθετων projects είναι δυνατή όπως φαίνεται από την αριστερή περιοχή που δείχνει τους δυνατούς τύπους από projects). Αφήνουμε τσεκαρισμένες τις default επιλογές και προχωράμε στο επόμενο βήμα.



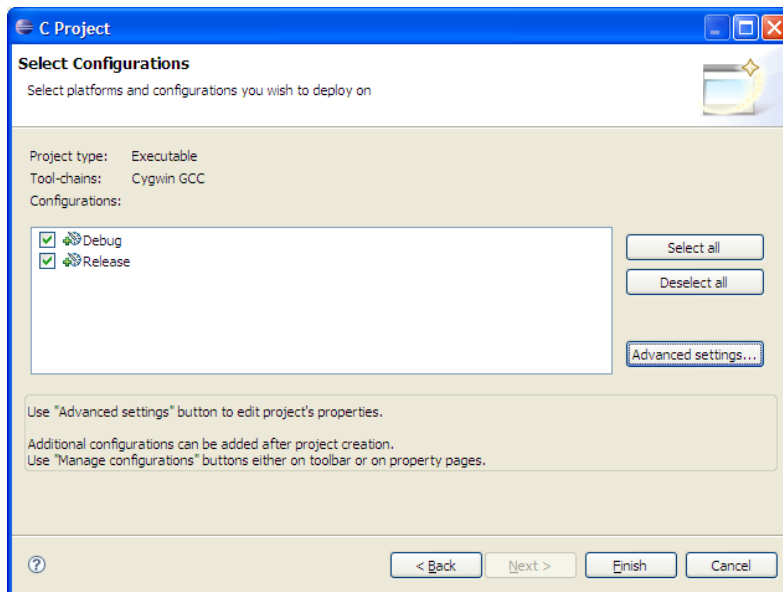
Εικόνα 4

Τώρα, μπορούμε να δώσουμε μερικά στοιχεία παραπάνω για το project μας, αν επιθυμούμε. Συμπληρώνουμε αν θέλουμε και προχωράμε.



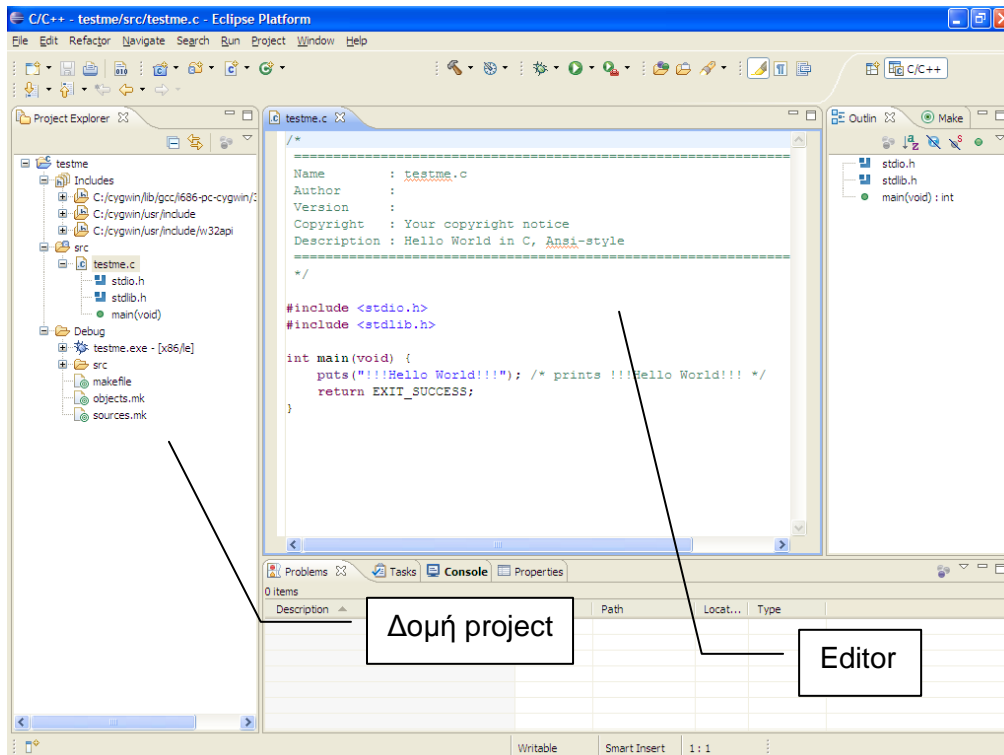
Εικόνα 5

Στη συνέχεια υποδηλώνουμε ότι μας ενδιαφέρει να κατασκευάσουμε και την τελική μορφή εκτελέσιμου κώδικα (Release) αλλά και τη μορφή κώδικα που επιτρέπει εργασίες αποσφαλμάτωσης (Debug). Δεν απαιτείται προς το παρόν ν' ασχοληθούμε με τα Advanced Settings.



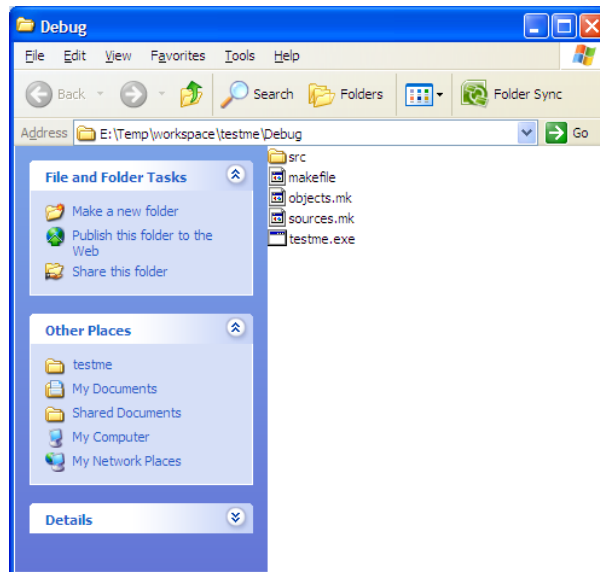
Εικόνα 6

Το Eclipse έχει πλέον παράγει μία βασική δομή για το project μας, η οποία αποτυπώνεται στην αριστερή περιοχή και όπου φαίνονται οι συναρτήσεις του κώδικα, τα αρχεία βιβλιοθήκης που χρησιμοποιούνται, τα βοηθητικά αρχεία που έχουν παραχθεί κλπ. Πατώντας διπλό click πάνω στο αρχείο testme.c που έχει παραχθεί μπορούμε επιπλέον να εμφανίσουμε αυτό το αρχείο σε ένα editor του συστήματος, ο οποίος μάλιστα είναι C sensitive (π.χ. παρατηρήστε το χρώμα στις μεταβλητές ή στα σχόλια).



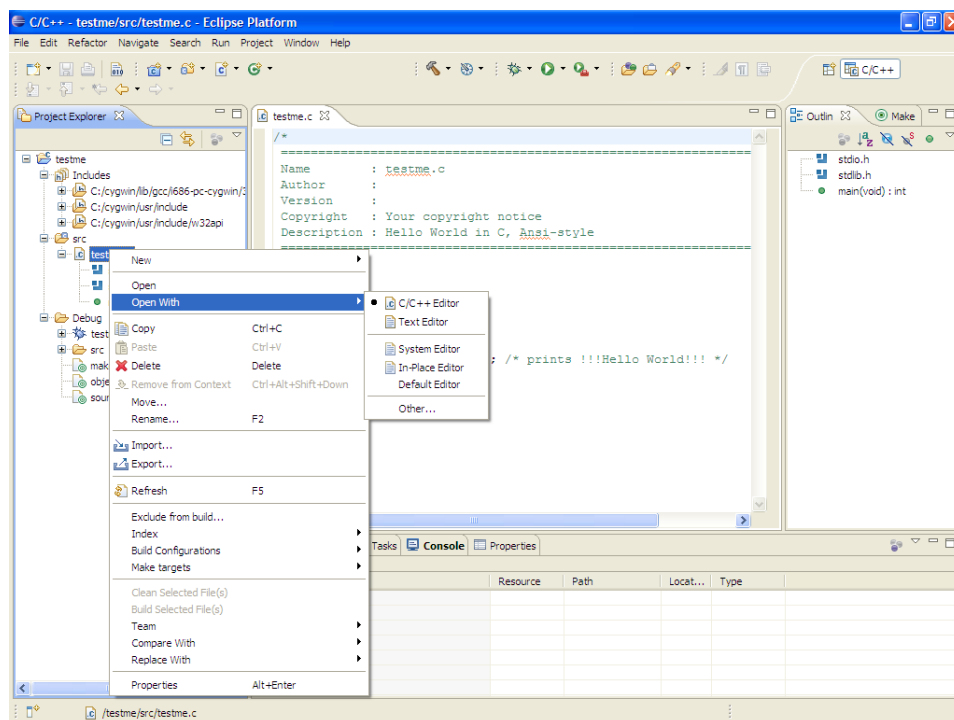
Εικόνα 7

Εδώ αξίζει να σημειώσουμε πως η δομή project αποτυπώνεται αντίστοιχα και στο δίσκο μας, όπως φαίνεται για τον κατάλογο Debug:



Εικόνα 8

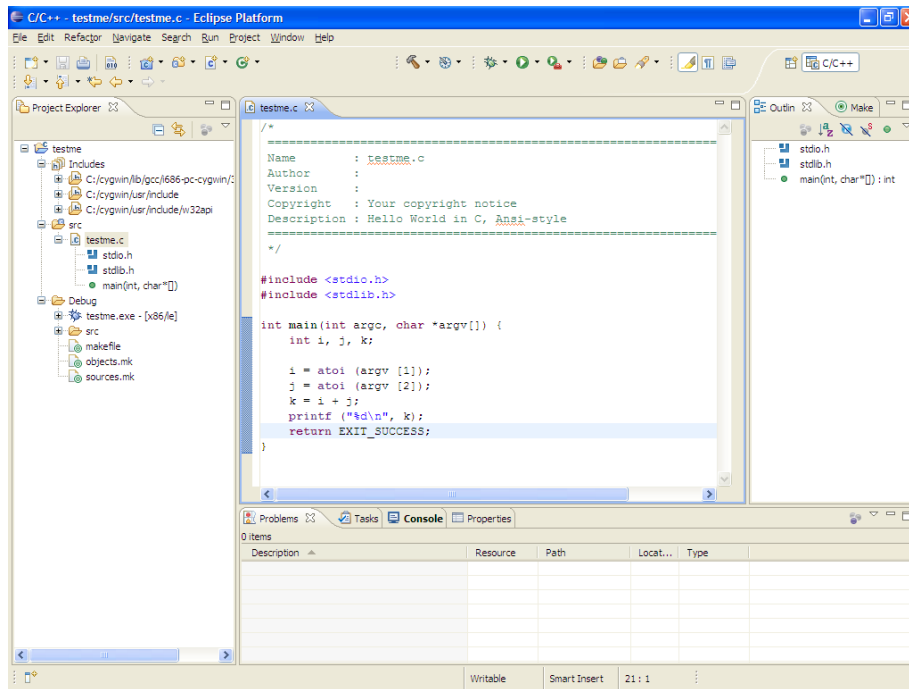
Επιπλέον, μπορούμε να ανοίξουμε κάποιο αρχείο χρησιμοποιώντας οποιονδήποτε από τους διαθέσιμους editors (ως system editor πιθανότατα και στο σύστημά σας νοείται το Notepad).



Εικόνα 9

Μπορούμε να αλλάξουμε τον κώδικά μας πλέον και να ζητήσουμε να αθροίσει δύο ακραίους που διαβάζει από το πληκτρολόγιο (για λόγους επίδειξης, δεν ασχολούμαστε με error checking στην είσοδο). Αρχίστε να πληκτρολογείτε τις αλλαγές από την αρχή του αρχείου προς το τέλος και παρατηρείστε πως όταν ανοίγεται παρένθεση, το σύστημα παράγει αμέσως την αντίστοιχη παρένθεση που κλείνει (το ίδιο ισχύει και για τις αγκύλες). Γενικά, αξίζει τον κόπο να πειραματιστείτε με τον Editor που παρέχει το Eclipse (π.χ. αφού γράψετε τον

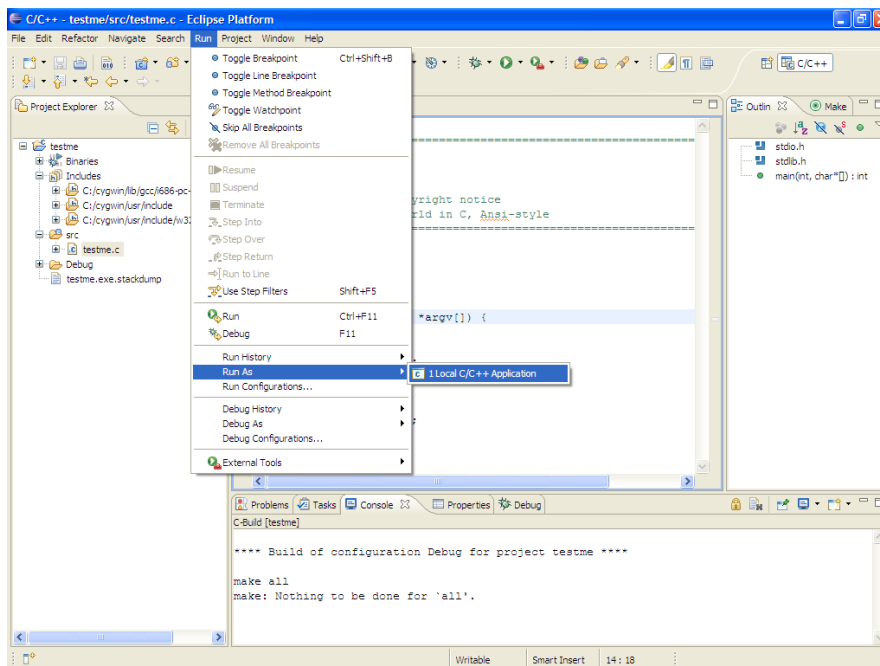
κώδικα, κάντε click πάνω σε μία λέξη **argv** και παρατηρήστε πως σκιάζονται και οι άλλες λέξεις **argv**).



Εικόνα 10

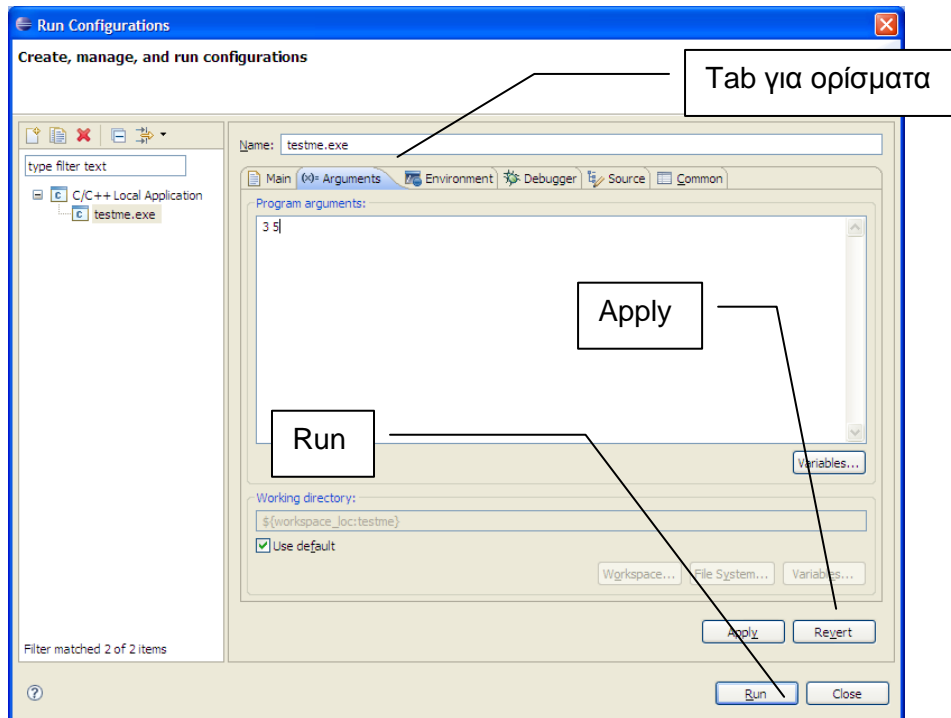
Αφού ολοκληρώσουμε τις αλλαγές και αποθηκεύσουμε το νέο αρχείο, μπορούμε με **Project > Build All** ή με **Ctrl-B** να προκαλέσουμε νέα μεταγλώττιση και παραγωγή εκτελέσιμου κώδικα.

Για να εκτελέσουμε το πρόγραμμά μας πρέπει κατ' αρχήν να υποδείξουμε στο σύστημα ότι επιθυμούμε το αποτέλεσμα της μεταγλώττισης να εκτελεστεί ως τοπική εφαρμογή.



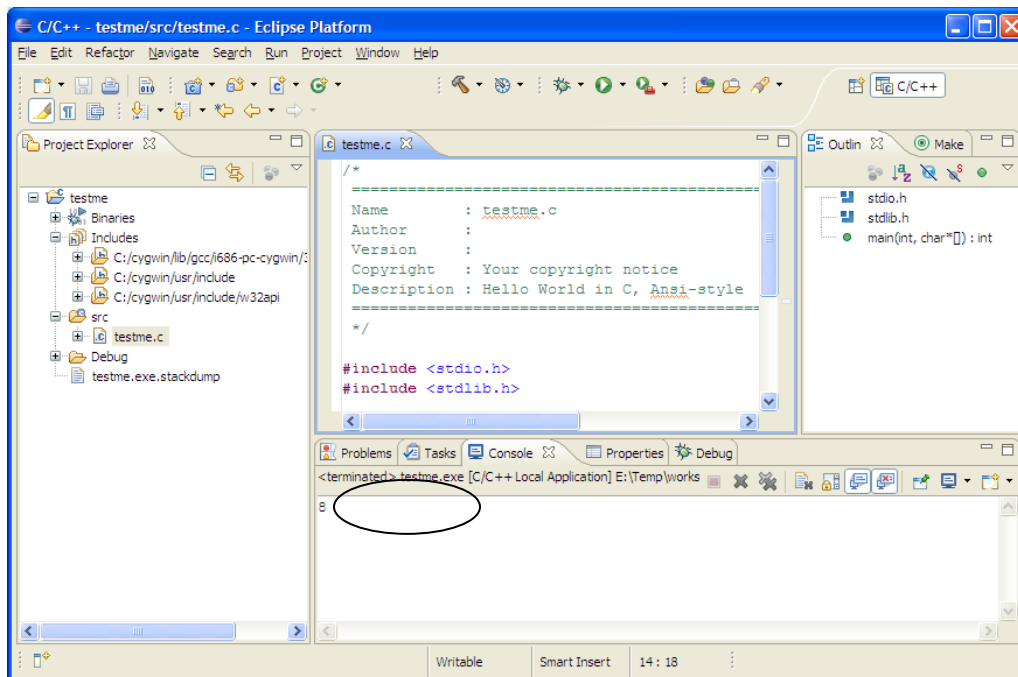
Εικόνα 11

Στη συνέχεια επιλέγουμε **Run > Run Configurations** για να δώσουμε ορίσματα και έστω πως θέλουμε να προσθέσουμε τους ακέραιους 3 και 5:



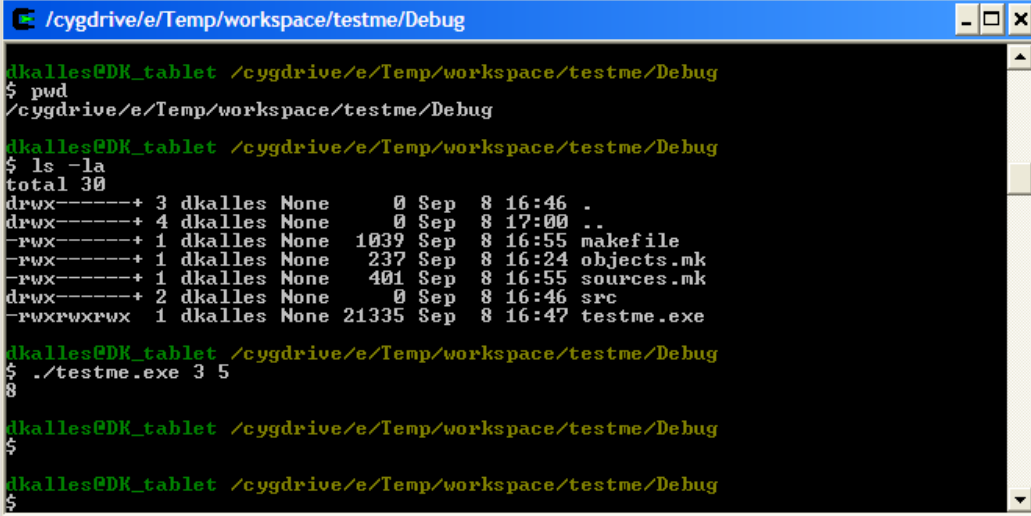
Εικόνα 12

Μετά επιλέγουμε πρώτα Apply (για να χρησιμοποιήσουμε και άλλη φορά αυτά τα ορίσματα, π.χ. κατά την αποσφαλμάτωση) και μετά Run. Το αποτέλεσμα (3+5=8) φαίνεται στην κονσόλα εξόδου.



Εικόνα 13

Εννοείται, πως το ίδιο αποτέλεσμα θα παίρναμε αν πηγαίναμε στον κατάλογο με το εκτελέσιμο αρχείο και δώναμε την εντολή από το shell (παρατηρήστε πως το εκτελέσιμο βρίσκεται μέσα στον κατάλογο Debug):



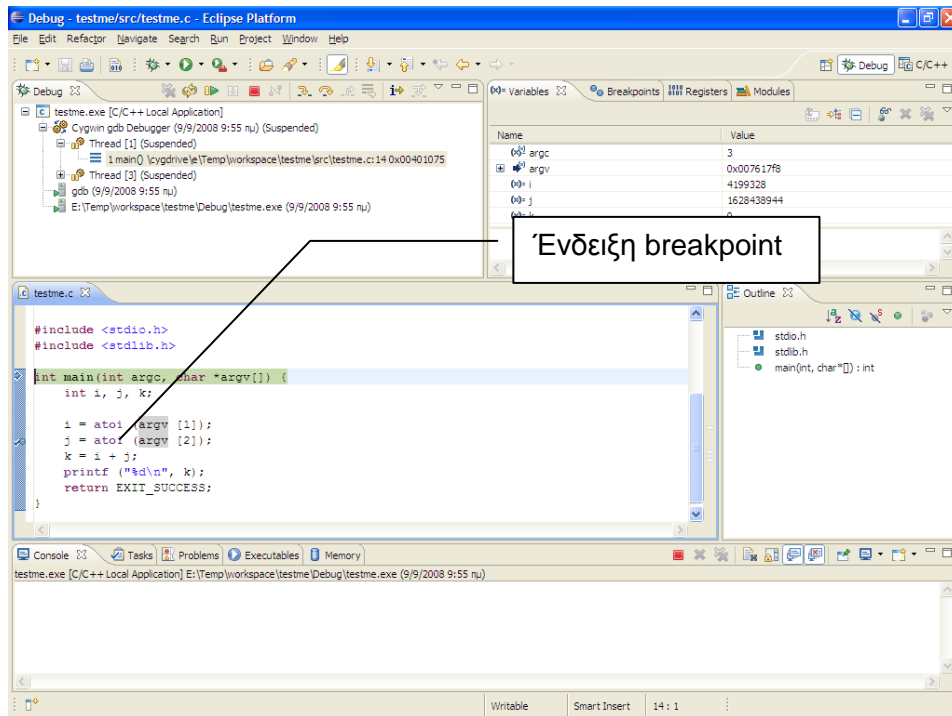
```
dkalles@EDK_tablet /cygdrive/e/Temp/workspace/testme/Debug
$ pwd
/cygdrive/e/Temp/workspace/testme/Debug
dkalles@EDK_tablet /cygdrive/e/Temp/workspace/testme/Debug
$ ls -la
total 30
drwx-----+ 3 dkalles None    0 Sep  8 16:46 .
drwx-----+ 4 dkalles None    0 Sep  8 17:00 ..
-rwx-----+ 1 dkalles None  1039 Sep  8 16:55 makefile
-rwx-----+ 1 dkalles None   237 Sep  8 16:24 objects.mk
-rwx-----+ 1 dkalles None   401 Sep  8 16:55 sources.mk
drwx-----+ 2 dkalles None    0 Sep  8 16:46 src
-rwxrwxrwx  1 dkalles None 21335 Sep  8 16:47 testme.exe
dkalles@EDK_tablet /cygdrive/e/Temp/workspace/testme/Debug
$ ./testme.exe 3 5
8
dkalles@EDK_tablet /cygdrive/e/Temp/workspace/testme/Debug
$
dkalles@EDK_tablet /cygdrive/e/Temp/workspace/testme/Debug
$
```

Εικόνα 14

Βήμα 4: Δουλεύοντας με τον debugger και το Eclipse

Ας υποθέσουμε πως θέλουμε να μελετήσουμε την εκτέλεση του προγράμματός μας βήμα βήμα και να παρατηρούμε τι τιμές έχουν οι μεταβλητές μας. Αντί να γεμίσουμε τον πηγαίο κώδικα με κλήσεις της **printf** μπορούμε να χρησιμοποιήσουμε τον debugger, που μας δίνει ένα περιβάλλον να εξετάζουμε την εκτέλεση και να διερευνούμε τις τιμές μεταβλητών, δεικτών, ή/και τη σειρά κλήσεων συναρτήσεων. Βασικό εργαλείο σε ένα debugger είναι το breakpoint, που είναι ένα σημείο στον κώδικα στο οποίο επιθυμούμε να σταματήσει η εκτέλεση.

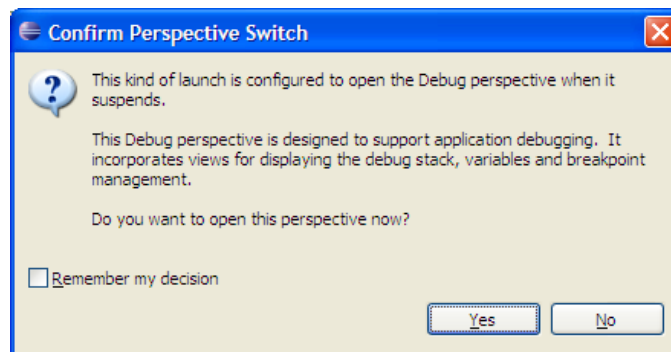
Θα δείξουμε τη λειτουργία ενός breakpoint (και, κατ' επέκταση, του debugger) με ένα παράδειγμα. Από το παράθυρο του editor πάμε στο *αριστερό περιθώριο* της γραμμής που περιέχει την εντολή **j = ...** και πατώντας δεξί click επιλέγουμε στη συνέχεια **Toggle Breakpoint**. Στο σημείο αυτό φαίνεται μία πράσινη μικρή κουκίδα που υποδηλώνει τη δήλωση του breakpoint (ξαναπατώντας **Toggle Breakpoint** η κουκίδα εξαφανίζεται).



Εικόνα 15

Αφού κάνουμε Build (έχοντας κάνει όποιες αλλαγές θέλουμε) ενεργοποιούμε το debugger από το **Run > Debug** (παρατηρείστε πως οι επιλογές ενεργοποίησης του debugger είναι όμοιες με τις επιλογές εκτέλεσης, και αν έχουμε ήδη κάνει μία εκτέλεση το Eclipse θα έχει κρατήσει τα ορίσματα εκτέλεσης ώστε να τα χρησιμοποιήσει και με το debugger).

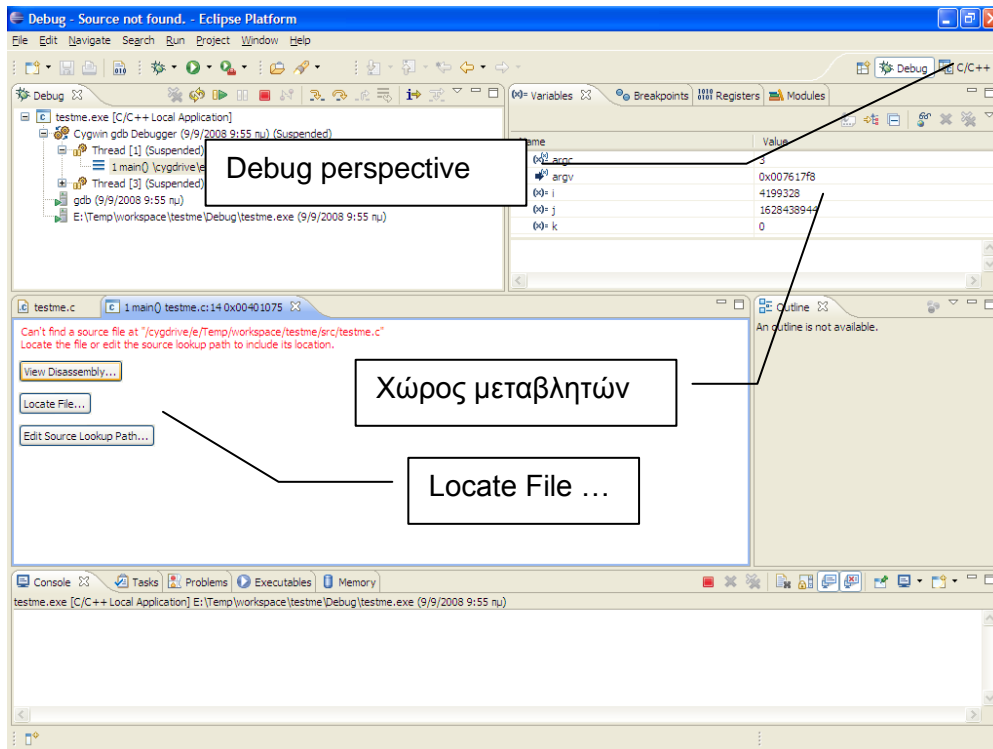
Κατά την ενεργοποίηση του debugger το Eclipse θα μας ζητήσει να επιβεβαιώσουμε πως θα ξεκινήσει λειτουργίες σε Debug perspective (άρα θα αλλάξει η μπάρα εργαλείων).



Εικόνα 16

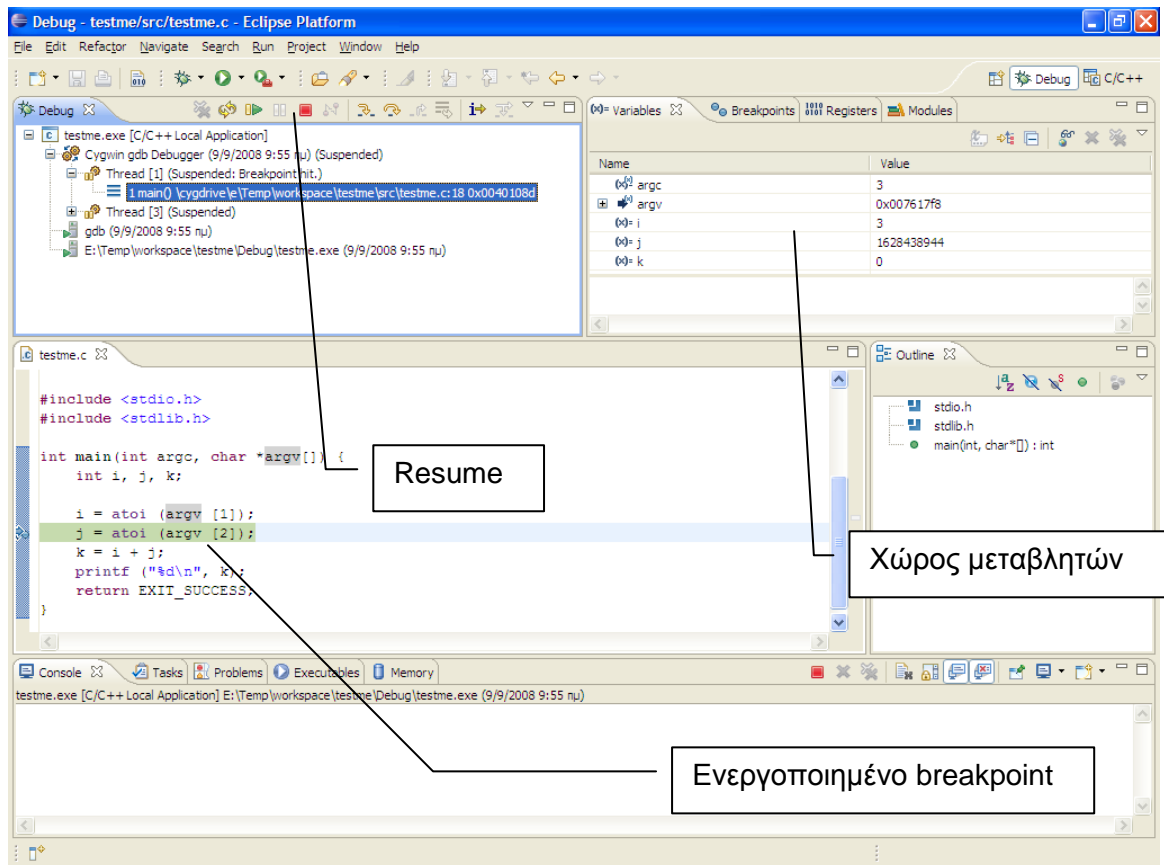
Το Eclipse, στην εγκατάσταση που χρησιμοποιούμε, θα ξεκινήσει το debugger και θ' αναζητήσει το αρχείο πηγαίου κώδικα που θα μας επιδεικνύει παράλληλα με την εκτέλεση. Αυτό πιθανότατα οφείλεται στην unix-like αναπαράσταση των μονοπατιών που παράγεται (αφού το Cygwin προσομοιώνει το Unix) αλλά στα πλαίσια του παρόντος οδηγού δε θ' ασχοληθούμε με τη διόρθωση αυτού του προβλήματος. Πιο εύκολα, θα πλοηγηθούμε μέσω της λειτουργίας **Locate File ...** που μας προτείνεται και να υποδείξουμε το χώρο που βρίσκεται το αρχείο πηγαίου κώδικα. Παρατηρείστε πως ο debugger έχει ξεκινήσει με κανονική εκτέλεση του προγράμματος και η μεταβλητή **argc** έχει την τιμή 3 (για το όνομα

προγράμματος και τις δύο παραμέτρους εισόδου), ενώ η μεταβλητή *i* έχει μία τιμή που σίγουρα δε δόσαμε εμείς ...



Εικόνα 17

Αφού γίνουν τα παραπάνω, μπορούμε να αναζητήσουμε και να εκτελέσουμε τη λειτουργία **Resume**, η οποία θα επαναξεκινήσει την εκτέλεση του προγράμματος, από το σημείο που μόλις είχε σταματήσει. Παρατηρούμε ότι το πρόγραμμα ΔΕΝ εκτελείται μέχρι τέλους, αλλά ΜΟΝΟ μέχρι το breakpoint (συγκεκριμένα, ακριβώς πριν το breakpoint). Μπορούμε να πειστούμε γι' αυτό παρατηρώντας πως η μεταβλητή *i* έχει την αναμενόμενη τιμή, ενώ η μεταβλητή *j* έχει ακόμα την αρχική τιμή που της απέδωσε το σύστημα.



Εικόνα 18

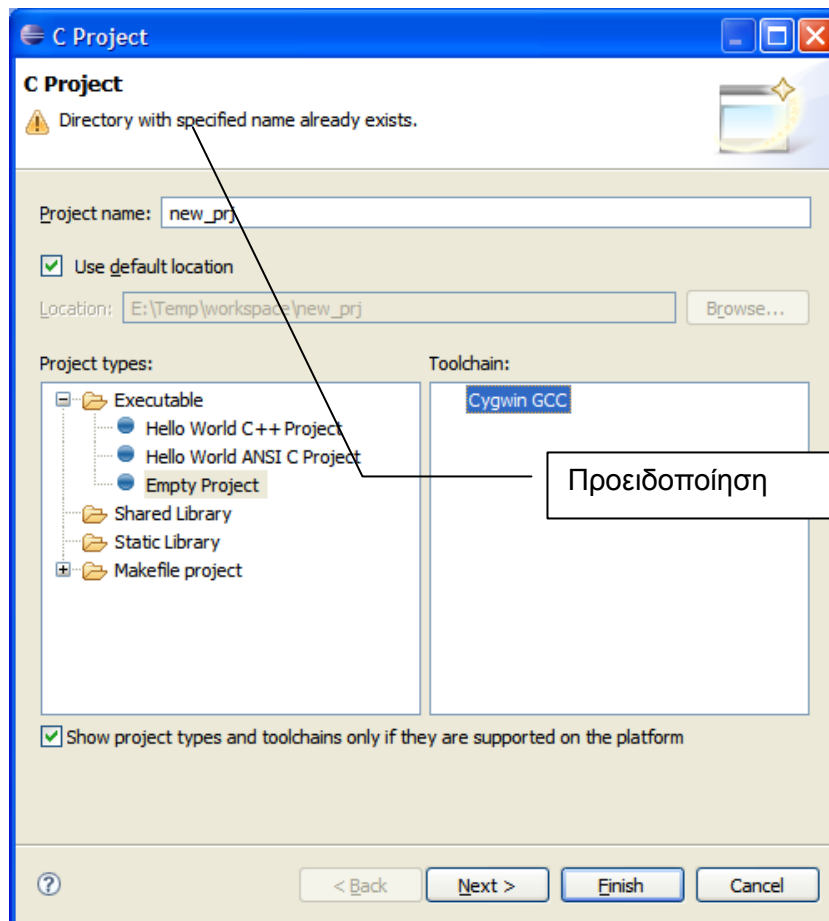
Όταν μέσα στο debugger μπορούμε πάντα να διαχειριστούμε τα breakpoints (αλλά και άλλα παρεμφερούς χρήσης εργαλεία), ν' αλλάξουμε perspective, να ξεκινήσουμε και να δουλέψουμε σ' ένα νέο project, να επανέλθουμε στο debugger κλπ. Η φιλοσοφία ενός IDE είναι η διευκόλυνση του χρήστη στην ανάπτυξη συστημάτων μέσω της οργάνωσης των εργασιών που απαιτούνται για την προγραμματιστική ανάπτυξη με τρόπο που να μπορεί να παρακολουθεί διαφορετικά projects χωρίς ν' απαιτείται να καταναλώνει περιττό χρόνο στη διαχείριση των εργασιών ανάπτυξης.

Βήμα 5: Χρησιμοποιώντας το Eclipse για έργα που έχουμε ήδη αναπτύξει σε άλλο περιβάλλον

Αν έχουμε ήδη ένα αρχείο σε C, το να το φέρουμε μέσα σε περιβάλλον Eclipse και να συνεχίσουμε εκεί την ανάπτυξη είναι απλό.

Κατ'αρχήν μέσα στο workspace του Eclipse φτιάχνουμε ένα νέο κατάλογο, έστω **new_prj**, και μέσα στον κατάλογο αυτό βάζουμε το αρχείο C που έχουμε.

Στη συνέχεια ξεκινάμε ένα νέο project επιλέγοντας **File > New > C Project** αλλά τώρα επιλέγουμε ένα Empty project και ονοματίζουμε το project με το όνομα του καταλόγου που κατασκευάσαμε (παρατηρούμε πως το Eclipse μας προειδοποιεί πως υπάρχει αυτός ο κατάλογος – αυτό είναι φυσικό αφού μόλις το φτιάξαμε, αλλά είναι χρήσιμο για να μην κάνουμε overwrite κατά λάθος κάποιο υπάρχον project).



Εικόνα 19

Στο σημείο αυτό οδηγούμαστε πάλι ακριβώς στο σημείο που ήμασταν όταν ξεκινήσαμε ένα Empty C Project και όσα αναφέραμε εκεί (από την Εικόνα 5 και μετά) ισχύουν πάλι επακριβώς.

Μερικές φορές, βέβαια, το project μας δεν απαρτίζεται μόνο από ένα αρχείο C. Συνήθως σ' αυτές τις περιπτώσεις αυτό που κάνουμε είναι να κατασκευάζουμε μία σειρά από αρχείο .c και .h τα οποία συνδέονται μεταξύ τους, και να περιγράψουμε τις συνδέσεις αυτές μέσω ενός Makefile (χρησιμοποιώντας την εφαρμογή **make**). Και σ' αυτές τις περιπτώσεις, για να φέρουμε ένα τέτοιο project μέσα στο eclipse αρκεί να κάνουμε τα βήματα που περιγράφονται σ' αυτή την παράγραφο (κατασκευή καταλόγου, επικόλληση των αρχείων κώδικα μέσα στον κατάλογο, κατασκευή νέου project με το όνομα του καταλόγου) και το Eclipse θα παράγει τα απαιτούμενα υπόλοιπα βήματα για τη σωστή μεταγλώττιση του όλου project (αλλά θα αγνοήσει το Makefile που θα έχουμε δώσει).

Περισσότερα για το Eclipse

Για περισσότερες πληροφορίες σχετικά με το eCclipse αποταθείτε στον βοηθό εργαστηρίου και στη ηλεκτρονική διεύθυνση <http://www.eclipse.org>.