



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΧΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ  
ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ

Φιλίππου Μιχαήλ el15756

Δημήτριος Ξενίας el15084

1<sup>η</sup> Εργαστηριακή Άσκηση στα Λειτουργικά Συστήματα

## Σύνδεση με αρχείο αντικειμένων

```
oslab01@os-node1: ~/ask1
#include "zing.h"
#include <stdio.h>

int main (int argc, char **argv) {

    zing(" ");
    return 0;
}

"main1.c" 10L, 102C 1,1 All
```

Έπειτα για να κάνουμε την συνένωση των δυο αρχείων (linking) αρχικά χρειάζεται να είμαστε στον ίδιο directory. Έπειτα, αφού έχουμε έτοιμα τα object files με την εντολή `gcc main.o zing.o -o zing` δημιουργείται το εκτελέσιμο `zing`, όπου τρέχει με την εντολή με την εντολή `./zing`.

```
oslab01@os-nodel:~/ask1$ ./zing
Hello, oslab01
oslab01@os-nodel:~/ask1$
```

## Ερωτήσεις

1. Η επικεφαλίδα χρησιμοποιείται για το συντονισμό των δηλώσεων ανάμεσα σε πολλά αρχεία καθώς είναι βολικό αυτές να συγκεντρώνονται σε ένα αρχείο (*header file*). Επίσης ενημερώνουν τον compiler ώστε να περιλαμβάνει άλλα αρχεία όπου καλούνται με την εντολή `include <...>` ή `include "..."`.
2. Για την δημιουργία του `zing2.o` ακολουθούμε την διαδικασία που έχουμε αναφέρει και προηγουμένος. Το καινούργιο `zing2.c` θα είναι το:

```
oslab01@os-node1: ~/ask1
#include <stdio.h>
#include <unistd.h>
#include "zing.h"

void zing (const char *argv) {
    printf("hello voithe! Oi dimiourgoi tis askisis einai oi: %s \n", getlog
in() );
}
```

και η έξοδος που θα δώσει με το linking θα είναι :

```
oslab01@os-node1:~/ask1$ ./zingl
hello voithe! Oi dimiourgoi tis askisis einai oi: oslab01
oslab01@os-node1:~/ask1$
```

Ενώ το νέο Makefile που ικανοποιεί τις προδιαγραφές είναι:

```
oslab01@os-node1: ~/ask1
all: zing zingl
zing : zing.o mainl.o
    gcc -Wall mainl.o zing.o -o zing
zingl: zing2.o mainl.o
    gcc -Wall zing2.o mainl.o -o zingl
mainl.o : mainl.c
    gcc -Wall -c mainl.c
zing2.o: zing2.c
    gcc -Wall -c zing2.c
~
```

3. ..
4. Για να αντιμετωπίσω αυτό το πρόβλημα θα έφτιαχνα ένα Makefile με το οποίο θα μπορούσα να κάνω compile μόνο την συνάρτηση που πείραξα και όχι όλο το project.
5. Ας αναλύσουμε την εντολή `gcc -Wall -o foo.o foo.c` κομμάτι κομμάτι. Με την χρήση του compiler (gcc), η εντολή `-Wall` εμφανίζει τα μηνύματα σφάλματος και warnings του compiler. Έπειτα δημιουργεί το αρχείο `foo.o` (-o) του `foo.c` το οποίο έκανε compile και το αποθηκεύει στο `foo.o`. Άρα δικαιολημένες οι φωνές διότι αντέγραψε το object file

πάνω στο text file και χάσανε όλα τα αρχεία τους (το object file δεν περιέχει πληροφορία για τον χρήστη, και δεν μπορεί να το χρησιμοποιήσει ή να επέμβει).