**Λειτουργικά Συστήματα**

**Άσκηση1: Εισαγωγή στο περιβάλλον προγραμματισμού**

**1.1 Σύνδεση με αρχείο αντικειμένων**

Αρχικά δημιουργήσαμε τη συνάρτηση main() (αρχείο main.c) μέσα από την οποία καλέσαμε τη συνάρτηση zing(). Για την κλήση της zing μέσω της main χρειάστηκε compile της main και σύνδεση των δύο αρχείων με τις πιο κάτω εντολές:

gcc -Wall -c main.c

gcc main.o zing.o -o zing

Καλώντας το εκτελέσιμο zing η έξοδος είναι: Hello, oslaba05

Ο κώδικας που χρησιμοποιήσαμε φαίνεται πιο κάτω:

#include "zing.h"

int main() { zing();

return 0;

}

**Απαντήσεις**:

**1.** Στη main() χρησιμοποιήσαμε την επικεφαλίδα **#include <zing.h>**. Στην επικεφαλίδα αυτή δηλώνεται το πρότυπο της zing(). Δηλαδή τι τύπο ορισμάτων δέχεται και τι τύπος είναι τα δεδομένα που επιστρέφει.

**2.** Το αρχείο Makefile περιείχε τα εξής:

zing: zing.o main.o

gcc zing.o main.o -o zing

main.o: main.c

gcc -Wall -c main.c

**3.** Δημιουργήσαμε τη zing2.

Καλώντας το εκτελέσιμο zing2 η έξοδος είναι: I am useroslaba05

O κώδικας φαίνεται πιο κάτω:

#include <unistd.h>

#include <stdio.h>

#include <stdlib.h>

int zing()

{

int k;

char \*p;

p = getlogin(); //returns a pointer to a string //containing the name of the user //logged in on the controlling //terminal of the process, or a NULL //pointer if this information cannot be //determined.

if (p==NULL) //checks if error occurred

{

perror("Error getting username");

return 1; }

k = write(1, "I am user ",10);

if (k==-1) //error indicates that a later //request may succeed

{

perror("Error in write attemp 1");

//print a system error message

return 1; }

k = write(1, p, 10);

if (k==-1)

{

perror("Error in write attemp 2");

return 1; }

k = write(1,"\n",1);

if (k==-1)

{

perror("Error in write attemp 3");

return 1; }

return 0;

}

Επίσης αλλάξαμε το Makefile σε:

all: zing zing2

zing2: zing2.o main.o

gcc zing2.o main.o -o zing2

zing: zing.o main.o

gcc zing.o main.o -o zing

zing2.o: zing2.c

gcc -Wall -c zing2.c

main.o: main.c

gcc -Wall -c main.c

clean:

rm zing zing2

**4.**Ο τρόπος για να μειωθεί ο χρόνος μεταγλώττισης είναι ο εξής: γράφουμε την κάθε συνάρτηση και τη main() που τις καλεί, σε ξεχωριστά αρχεία. Κατόπιν γράφουμε τους κανόνες μεταγλώττισης και σύνδεσης(linking) των αντικειμένων σε ένα Makefile. Έτσι εκτελώντας την εντολή make μεταγλωττίζονται μόνο οι συναρτήσεις που έχει αλλάξει το περιεχόμενό τους.

**5.** Εκτελώντας την εντολή gcc -Wall -o foo.c foo.c λέμε στον compiler να μεταγλωττίσει το αρχείο foo.c και το εκτελέσιμο που θα παραχθεί να το ονομάσει foo.c, το οποίο αντικαθιστά το προηγούμενο που περιέχει τον κώδικα του προγράμματος .