



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ  
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

## Λειτουργικά Συστήματα

# Άσκηση 1

---

Εισαγωγή στο περιβάλλον  
προγραμματισμού

Εργαστηριακή Ομάδα Β10

Καραμανωλάκης Ιωάννης  
ΑΜ 03111006  
7ο Εξάμηνο

Προβατάς Νικόδημος  
ΑΜ: 03111065  
7ο Εξάμηνο

## 1.1 Συνδεση με αρχείο αντικειμένων

Στην άσκηση αυτή στόχος είναι η δημιουργία ενός εκτελέσιμου που θα καλεί τη συνάρτηση `zing()`. Για το σκοπό αυτό αντιγράφονται αρχικά στον κατάλογο εργασίας το αρχείο `zing.h`, στο οποίο είναι δηλωμένη η συνάρτηση `zing` και το αρχείο αντικειμένων `zing.o`. Αρχικά, γίνεται δημιουργία του αρχείου κώδικα `main.c` το οποίο χρησιμοποιείται για να καλέσει τη συνάρτηση `zing`. Ο κώδικας του `main.c` φαίνεται στη συνέχεια:

```
#include "zing.h"

int main (int argc, char ** argv) {
    zing();
    return 0;
}
```

Για την παραγωγή του εκτελέσιμου αρχείου `zing` που θα καλεί τη συνάρτηση `zing()` αρχικά κάνουμε μεταγλώττιση του κώδικα της `main.c` μέσω της εντολής

```
gcc -Wall -c main.c
```

και στη συνέχεια σύνδεση του αρχείου `main.o` που παράγεται από την παραπάνω εντολή με το αρχείο `zing.o` μέσω της εντολής

```
gcc -Wall -o zing main.o zing.o
```

Έτσι παράγεται το αρχείο `zing`, το οποίο όταν εκτελεστεί καλεί τη συνάρτηση `zing()` και εμφανίζεται το εξής μήνυμα:~

Hello oslab10!

### 1. Ποιο σκοπό εξυπηρετεί η επικεφαλίδα;

Η επικεφαλίδα περιέχει τη δήλωση του τίτλου μίας ή και περισσότερων συνάρτησεων. Αυτό εξυπηρετεί, καθώς μπορεί κάποιο αρχείο να θέλουμε να καλεί κάποια συνάρτηση, η οποία να χρειαστεί να αλλάξει κάποια στιγμή. Έτσι συμπεριλαμβάνοντας την βιβλιοθήκη στον κώδικα του αρχείου που θέλουμε να εκτελεί την συνάρτηση αυτή, έχουμε τη δυνατότητα να περιέχουμε τη συνάρτηση σε κάποιο άλλο αρχείο το οποίο θα συνδέσει αργότερα με τον κώδικα της βασικής συνάρτησης και μπορούμε να το τροποποιήσουμε ανα πάσα στιγμή. Επίσης, μία βιβλιοθήκη μπορεί να φανεί χρήσιμη και όταν κάποιες συναρτήσεις τις χρησιμοποιούμε σε αρκετά αρχεία.

## 2. Ζητείται κατάλληλο Makefile για τη δημιουργία του εκτελέσιμου της άσκησης.

Στη συνέχεια, για τη διευκόλυνση της μεταγλώττισης και σύνδεσης των αρχείων κατασκευάζεται Makefile για τη δημιουργία του εκτελέσιμου, όπως φαίνεται στη συνέχεια:

```
all: out

out: main.o zing.o
    gcc -Wall -o zing main.o zing.o

main: main.c
    gcc -Wall -c main.c
```

Με τη δημιουργία του αρχείου αυτού καλώντας την εντολή `make` εντοπίζεται το αρχείο Makefile και εκτελούνται οι εντολές που υπάρχουν σε αυτό με αποτέλεσμα την παραγωγή του εκτελέσιμου αρχείου. Στην συνέχεια κάθε επόμενη φορά που εκτελείται η εντολή `make`, εκτελούνται οι παραπάνω εντολές εάν έχει υπάρξει κάποια μεταβολή στα αρχεία αναφέρονται στις ετικέτες.

## 3. Γράψτε το δικό σας `zing2.o`, το οποίο θα περιέχει `zing()` που θα εμφανίζει διαφορετικό αλλά παρόμοιο μήνυμα με τη `zing()` του `zing.o`. Συμβουλευτείτε το `manual page` της `getlogin(3)`. Αλλάξτε το Makefile ώστε να παράγονται δύο εκτελέσιμα, ένα με το `zing.o`, ένα με το `zing2.o`, επαναχρησιμοποιώντας το κοινό object file `main.o`.

Αρχικά, κατασκευάζουμε το αρχείο `zing2.c`, το οποίο περιέχει ένα παρόμοιο μήνυμα με αυτό που θα εμφανίζει η `zing`. Ο κώδικας του αρχείου αυτού φαίνεται στη συνέχεια.

```
#include <stdio.h>
#include <unistd.h>

void zing (void) {
    printf("Hello! This is %s!\n", getlogin());
}
```

Η μεταγλώττιση του παραπάνω κώδικα γίνεται με την εντολή

```
gcc -Wall -c zing2.c
```

και στη συνέχεια σύνδεση του αρχείου `zing2.o` που παράγεται από την παραπάνω εντολή με το αρχείο `main.o` μέσω της εντολής

```
gcc -Wall -o zing2 main.o zing2.o
```

Έτσι παράγεται το αρχείο zing, το οποίο όταν εκτελεστεί καλεί τη συνάρτηση zing() και εμφανίζεται το εξής μήνυμα:~

Hello! This is oslab10!

Τροποποιούμε και το Makefile ώστε να συμπεριλαμβάνει όλα τα παραπάνω αρχεία και γίνεται:

```
all: out2 out

out2: main.o zing2.o
    gcc -Wall -o zing2 main.o zing2.o

out: main.o zing.o
    gcc -Wall -o zing main.o zing.o

zing2: zing2.c
    gcc -Wall -c zing2.c

main: main.c
    gcc -Wall -c main.c
```

**4. Έστω ότι έχετε γράψει το πρόγραμμά σας σε ένα αρχείο που περιέχει 500 συναρτήσεις. Αυτή τη στιγμή κάνετε αλλαγές μόνο σε μία συνάρτηση. Ο κύκλος εργασίας είναι: αλλαγές στον κώδικα, μεταγλώττιση, εκτέλεση, αλλαγές στον κώδικα, κ.ο.κ. Ο χρόνος μεταγλώττισης είναι μεγάλος, γεγονός που σας καθυστερεί. Πώς μπορεί να αντιμετωπισθεί το πρόβλημα αυτό;**

Το πρόβλημα αυτό μπορεί να αντιμετωπισθεί με τη χρήση ενός Makefile και σπάσιμο του αρχείου σε μικρότερα, που κάθε συνάρτηση θα περιέχεται σε ένα αρχείο και το κύριο μέρος του κώδικα σε ένα άλλο. Χρησιμοποιώντας αυτό ανάλογα με τις ετικέτες και τα ορίσματα που τους έχουμε βάλει στο Makefile, ελέγχει ποιο ή ποια αρχεία έχουν τροποποιηθεί από την τελευταία μεταγλώττιση και εκτελεί μόνο τις εντολές που αφορούν τα τροποποιημένα αρχεία. Έτσι στην παραπάνω περίπτωση μεταγλωττίζεται μόνο το αρχείο κώδικα που αφορά τη συγκεκριμένη συνάρτηση και γίνεται και η εντολή που αφορά το link των αρχείων. Με τα παραπάνω μειώνεται αρκετά ο χρόνος μεταγλώττισης καθώς μεταγλωττίζονται μόνο όσα κομμάτια είναι απαραίτητα.

5. Ο συνεργάτης σας και εσείς δουλεύατε στο πρόγραμμα `foo.c` όλη την προηγούμενη εβδομάδα. Καθώς κάνατε ένα διάλειμμα και ο συνεργάτης σας δούλεψε στον κώδικα, ακούτε μια απελπισμένη κραυγή. Ρωτάτε τι συνέβει και ο συνεργάτης σας λέει ότι το αρχείο `foo.c` χάθηκε! Κοιτάτε το `history` του φλοιού και η τελευταία εντολή ήταν η:

```
gcc -Wall -o foo.c foo.c
```

Τι συνέβη;

Με την εκτέλεση της παραπάνω εντολής, ο χρήστης προσπαθώντας να παράξει το εκτελέσιμο, το αποθήκευσε στο αρχείο `foo.c`, δηλαδή το έγραψε πάνω στο αρχείο κώδικα. Αυτό έχει ως αποτέλεσμα να χαθεί το αρχείο κώδικα `foo.c` και να μην είναι πλέον προσβάσιμο.

## 1.2 Συνένωση δύο αρχείων σε τρίτο

Στην άσκηση αυτή πρέπει να υλοποιηθεί το πρόγραμμα `fconc` το οποίο θα συνενώνει δύο αρχεία σε ένα τρίτο. Το αρχείο εξόδου της συνάρτησης θα είναι είτε το τρίτο όρισμα που δίνεται στην `fconc` είτε το αρχείο `fconc.out` στην περίπτωση που η συνάρτηση έχει μόνο δύο ορίσματα. Για την υλοποίηση της παραπάνω χρησιμοποιούνται διάφορες σύναρτήσεις που εκτελούν κλήσεις συστήματος για το άνοιγμα, το κλείσιμο, το διάβασμα και την εγγραφή σε αρχείων. Η βασική συνάρτηση που περιλαμβάνει το πρόγραμμα είναι η `write_file`, που αναλαμβάνει την αντιγραφή ενός αρχείου σε ένα άλλο που ξέρει το αναγνωριστικό του. Για τη διαδικασία της εγγραφής στο αρχείο χρησιμοποιείται η συνάρτηση `doWrite` η οποία αντιγράφει ότι διαβάστηκε στον πίνακα `buff` στο αρχείο εξόδου. Ο κώδικας της `fconc` φαίνεται στη συνέχεια.

```
#include <stdio.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include <unistd.h>

void doWrite(int fd, const char* buff, int len);
void write_file(int fd, const char * infile);

int main (int argc, char ** argv){
    int fd;
    int oflags = O_CREAT | O_WRONLY | O_TRUNC;
    int mode = S_IRUSR | S_IWUSR;
    char * filename;
    if ((argc == 1) || (argc == 2))
    {
        printf("Usage:      ./fconf      infile1      infile2
[outfile (default:fconc.out)]\n");
```

```

        exit(1);
    }
    else if (argc==3)
    {
        filename = "fconc.out";
    }
    else filename = argv[3];

    fd = open(filename,oflags,mode);
    if (fd ==-1){
        perror("open");
        exit(1);
    }

    write_file(fd, argv[1]);
    write_file(fd,argv[2]);
    close(fd);
    return 0;
}

void doWrite(int fd, const char* buff, int len){
    int idx = 0;
    int wcnt;
    do {
        wcnt = write(fd,buff+idx, len - idx);
        if (wcnt == -1) {
            perror("write");
            exit(1);
        }
        idx += wcnt;
    } while (idx<len);
}

void write_file(int fd, const char * infile){
    int fread = open(infile, O_RDONLY);
    if (fread==-1){
        perror("open");
        exit(1);
    }
    char buff[16];
    ssize_t rcnt;
    for (;;) {
        rcnt = read(fread,buff,sizeof(buff));
        if (rcnt ==0) break;
        if (rcnt == -1) {
            perror("read");
            exit(1);
        }
    }
}

```

```
        doWrite(fd , buff, rcnt);
    }
    close(fread);
}
```

Εάν έχουμε δύο αρχεία A και B και εκτελέσουμε

```
cat ./A
```

βλέπουμε ότι το περιεχόμενο του αρχείου A είναι

```
This is file A
This is file A
This is file A
This is file A
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A
This is file A
This is file A
This is file A
```

ενώ με εκτέλεση της εντολής

```
cat ./B
```

βλέπουμε ότι το περιεχόμενο του αρχείου B θα είναι:

```
this is B
this is B
this is B
this is B
this is B to concat with file A
this is B to concat with file A
this is B to concat with file A
this is B to concat with file A
```

```
this is B to concat with file A
this is B to concat with file A
this is B to concat with file A
```

```
this is B
```

```
this is B
```

```
this is B
```

Αν στην συνέχεια καλέσουμε το πρόγραμμα fconc με την εντολή

```
./fconc A B C
```

τα αρχεία A και B θα αντιγραφούν στο αρχείο C και έτσι εκτελώντας την εντολή cat των linux βλέπουμε ότι το αρχείο C είναι η συνένωση των δύο αρχείων, δηλαδή περιέχει το κείμενο:

```
This is file A
This is file A
This is file A
This is file A
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A to concat with file B
This is file A
This is file A
This is file A
This is file A
this is B
this is B
this is B
this is B
this is B to concat with file A
this is B to concat with file A
this is B to concat with file A
this is B to concat with file A
```



```
this is B to concat with file A
this is B to concat with file A
this is B to concat with file A
```

```
this is B
```

```
this is B
```

```
this is B
```

Σημειώνεται ότι εάν είχαμε τρέξει την εντολή `fconc` χωρίς το τρίτο όρισμα `C`, τότε το παραπάνω αρχείο θα εγγραφόταν στο default αρχείο εξόδου `fconc.out`. Εάν τρέξουμε το πρόγραμμα `fconc` με χρήση του `strace`, δηλαδή γράφοντας:

```
strace ./fconc A B C
```

λαμβάνουμε την ακόλουθη έξοδο:

```
execve("./fconc", ["/fconc", "A", "B", "C"], [/* 18 vars */) = 0
brk(0) = 0x804a000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or
directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -
1, 0) = 0xb777e000 access("/etc/ld.so.preload", R_OK) = -1
ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY) = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=67766, ...}) = 0
mmap2(NULL, 67766, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb776d000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or
directory)
open("/lib/i686/cmov/libc.so.6", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\260\1\1\0004\0
\0\0\34"... , 512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1331684, ...}) = 0
mmap2(NULL, 1337704, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE,
3, 0) = 0xb7626000
mmap2(0xb7767000, 12288, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x141) = 0xb7767000
mmap2(0xb776a000, 10600, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0xb776a000
close(3) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -
1, 0) = 0xb7625000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb76256c0,
limit:1048575, seg_32bit:1, contents:0, read_exec_only:0,
limit_in_pages:1, seg_not_present:0, useable:1}) = 0
mprotect(0xb7767000, 8192, PROT_READ) = 0
mprotect(0xb779c000, 4096, PROT_READ) = 0
munmap(0xb776d000, 67766) = 0
open("C", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 3
```

```

open("A", O_RDONLY)           = 4
read(4, "This ia file A\nT"..., 16) = 16
write(3, "This ia file A\nT"..., 16) = 16
read(4, "his ia file A\nTh"..., 16) = 16
write(3, "his ia file A\nTh"..., 16) = 16
read(4, "is ia file A\nThi"..., 16) = 16
write(3, "is ia file A\nThi"..., 16) = 16
read(4, "s ia file A\nThis"..., 16) = 16
write(3, "s ia file A\nThis"..., 16) = 16
read(4, " ia file A to co"..., 16) = 16
write(3, " ia file A to co"..., 16) = 16
read(4, "ncat with file B"..., 16) = 16
write(3, "ncat with file B"..., 16) = 16
read(4, "\nThis ia file A "..., 16) = 16
write(3, "\nThis ia file A "..., 16) = 16
read(4, "to concat with f"..., 16) = 16
write(3, "to concat with f"..., 16) = 16
read(4, "ile B\nThis ia fi"..., 16) = 16
write(3, "ile B\nThis ia fi"..., 16) = 16
read(4, "le A to concat w"..., 16) = 16
write(3, "le A to concat w"..., 16) = 16
read(4, "ith file B\nThis "..., 16) = 16
write(3, "ith file B\nThis "..., 16) = 16
read(4, "ia file A to con"..., 16) = 16
write(3, "ia file A to con"..., 16) = 16
read(4, "cat with file B\n"..., 16) = 16
write(3, "cat with file B\n"..., 16) = 16
read(4, "This ia file A t"..., 16) = 16
write(3, "This ia file A t"..., 16) = 16
read(4, "o concat with fi"..., 16) = 16
write(3, "o concat with fi"..., 16) = 16
read(4, "le B\nThis ia fil"..., 16) = 16
write(3, "le B\nThis ia fil"..., 16) = 16
read(4, "e A to concat wi"..., 16) = 16
write(3, "e A to concat wi"..., 16) = 16
read(4, "th file B\nThis i"..., 16) = 16
write(3, "th file B\nThis i"..., 16) = 16
read(4, "a file A to conc"..., 16) = 16
write(3, "a file A to conc"..., 16) = 16
read(4, "at with file B\nT"..., 16) = 16
write(3, "at with file B\nT"..., 16) = 16
read(4, "his ia file A to"..., 16) = 16
write(3, "his ia file A to"..., 16) = 16
read(4, " concat with fil"..., 16) = 16
write(3, " concat with fil"..., 16) = 16
read(4, "e B\nThis ia file"..., 16) = 16
write(3, "e B\nThis ia file"..., 16) = 16
read(4, " A to concat wit"..., 16) = 16
write(3, " A to concat wit"..., 16) = 16
read(4, "h file B\nThis ia"..., 16) = 16
write(3, "h file B\nThis ia"..., 16) = 16
read(4, " file A to conca"..., 16) = 16
write(3, " file A to conca"..., 16) = 16

```

```

read(4, "t with file B\nTh"..., 16)    = 16
write(3, "t with file B\nTh"..., 16)    = 16
read(4, "is ia file A to "..., 16)      = 16
write(3, "is ia file A to "..., 16)      = 16
read(4, "concat with file"..., 16)      = 16
write(3, "concat with file"..., 16)      = 16
read(4, " B\nThis ia file "..., 16)     = 16
write(3, " B\nThis ia file "..., 16)     = 16
read(4, "A to concat with"..., 16)      = 16
write(3, "A to concat with"..., 16)      = 16
read(4, " file B\nThis ia "..., 16)     = 16
write(3, " file B\nThis ia "..., 16)     = 16
read(4, "file A to concat"..., 16)      = 16
write(3, "file A to concat"..., 16)      = 16
read(4, " with file B\nThi"..., 16)     = 16
write(3, " with file B\nThi"..., 16)     = 16
read(4, "s ia file A to c"..., 16)      = 16
write(3, "s ia file A to c"..., 16)      = 16
read(4, "oncat with file "..., 16)      = 16
write(3, "oncat with file "..., 16)      = 16
read(4, "B\nThis ia file A"..., 16)     = 16
write(3, "B\nThis ia file A"..., 16)     = 16
read(4, "\nThis ia file A\n"..., 16)    = 16
write(3, "\nThis ia file A\n"..., 16)    = 16
read(4, "This ia file A\nT"..., 16)     = 16
write(3, "This ia file A\nT"..., 16)     = 16
read(4, "his ia file A\n"..., 16)       = 14
write(3, "his ia file A\n"..., 14)       = 14
read(4, ""..., 16)                      = 0
close(4)                                = 0
open("B", O_RDONLY)                     = 4
read(4, "this is B\nthis i"..., 16)     = 16
write(3, "this is B\nthis i"..., 16)     = 16
read(4, "s B\nthis is B\nth"..., 16)    = 16
write(3, "s B\nthis is B\nth"..., 16)    = 16
read(4, "is is B\nthis is "..., 16)     = 16
write(3, "is is B\nthis is "..., 16)     = 16
read(4, "B to concat with"..., 16)      = 16
write(3, "B to concat with"..., 16)      = 16
read(4, " file A\nthis is "..., 16)     = 16
write(3, " file A\nthis is "..., 16)     = 16
read(4, "B to concat with"..., 16)      = 16
write(3, "B to concat with"..., 16)      = 16
read(4, " file A\nthis is "..., 16)     = 16
write(3, " file A\nthis is "..., 16)     = 16
read(4, "B to concat with"..., 16)      = 16
write(3, "B to concat with"..., 16)      = 16
read(4, " file A\nthis is "..., 16)     = 16
write(3, " file A\nthis is "..., 16)     = 16
read(4, "B to concat with"..., 16)      = 16
write(3, "B to concat with"..., 16)      = 16
read(4, " file A\nthis is "..., 16)     = 16
write(3, " file A\nthis is "..., 16)     = 16

```

```

read(4, "B to concat with"..., 16)      = 16
write(3, "B to concat with"..., 16)      = 16
read(4, " file A\nthis is "..., 16)      = 16
write(3, " file A\nthis is "..., 16)      = 16
read(4, "B to concat with"..., 16)      = 16
write(3, "B to concat with"..., 16)      = 16
read(4, " file A\nthis is "..., 16)      = 16
write(3, " file A\nthis is "..., 16)      = 16
read(4, "B to concat with"..., 16)      = 16
write(3, "B to concat with"..., 16)      = 16
read(4, " file A\n\nthis is"..., 16)      = 16
write(3, " file A\n\nthis is"..., 16)      = 16
read(4, " B\n\nthis is B\n\n"..., 16)    = 16
write(3, " B\n\nthis is B\n\n"..., 16)    = 16
read(4, "his is B\n"..., 16)             = 9
write(3, "his is B\n"..., 9)              = 9
read(4, ""..., 16)                       = 0
close(4)                                 = 0
close(3)                                 = 0
exit_group(0)                            = ?

```

Παρατηρώντας την πορεία εξέλιξης του προγράμματος βλέπουμε ότι η κλήση συστήματος `open` επιστρέφει το αναγνωριστικό του αρχείου, το οποίο το λαμβάνει η `read` για να διαβάσει. Η τελευταία μας λέει πόσους χαρακτήρες διάβασε και ο αριθμός αυτό εισάγεται στην `write` για να ξέρει πόσους χαρακτήρες πρέπει να γράψει στο αρχείο εξόδου. Από τα παραπάνω συμπεραίνουμε ότι η λειτουργία του προγράμματος είναι αυτή που αναμέναμε.