



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ**

## **ΑΣΚΗΣΗ 1**

**ΟΜΑΔΑ OSLABC34:**

Χριστοδουλόπουλος Δανιήλ                      03114797

Θηβαίος Αναστάσιος                              03114785

## Άσκηση 1:

1) Η επικεφαλίδα περιέχει δηλώσεις πρότυπα συναρτήσεων και global μεταβλητών. Χρησιμοποιεί δηλαδή στο να έχουμε σε κομμάτια κώδικα που είτε βρίσκονται στην βιβλιοθήκη της γλώσσας είτε έχουμε γράψει εμείς (.h αρχεία). Για παράδειγμα όταν γράφουμε `#include <stdio.h>` ο preprocessor της C “αντιγράφει” σ’ εκείνο το σημείο ό,τι περιέχει το αρχείο “stdio.h”, έτσι έχουμε πρόσβαση σε συναρτήσεις όπως `printf`, `scanf` κλπ.

## 2) Makefile

```
main: main.o zing.o
    gcc -o main main.o zing.o

main.o: main.c
    gcc -Wall -c main.c

.PHONY: clean
clean:
    rm main.o
```

## 3) zing2.c

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

void zing(){
    printf("Zing2 %s\n", getlogin());
}
```

## Makefile

```
.PHONY: all clean

all: main zing2

main: main.o zing.o
    gcc -o main main.o zing.o

main.o: main.c zing.h
    gcc -Wall -c main.c

zing2: main.o zing2.o
    gcc -o zing2 main.o zing2.o

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

clean:
    rm main main.o zing2 zing2.o
```

4) Αυτό θα μπορούσε να αντιμετωπισθεί με το να χωρίζουμε τον κώδικα σε μικρότερα αρχεία που το καθένα θα επιτελεί μια συγκεκριμένη δουλεία. Επίσης θα χρειαστούμε ένα Makefile ώστε να μεταγλωτίζουμε όλα αυτά τα αρχεία. Έτσι όταν κάνουμε αλλαγή σε μια μόνο συνάρτηση θα μεταγλωττιστεί ξανά **μόνο** το αρχείο που βρίσκεται αυτή η συνάρτηση και όχι όλος ο κώδικας ξανά και ξανά. Έτσι ο χρόνος μεταγλώττισης θα μειωθεί.

5) Αυτό που συνέβει είναι ότι δόθηκε στον compiler και σαν είσοδο και σαν έξοδο το αρχείο που περιέχει τον πηγαίο κώδικα, έτσι το εκτελέσιμο δημιουργήθηκε “πάνω” στο foo.c με αποτέλεσμα να χαθεί ο πηγαίος κώδικας.

## Άσκηση 2:

### 1) fconc.c

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

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

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

    if(argc < 3){
        printf("Usage: ./fconc infile1 infile2 [outfile (default:fconc.out)]\n");
        return -1;
    }

    int fd;
    mode_t mode = S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH | S_IWOTH;

    if (argc==4) {
        fd = open(argv[3], O_CREAT| O_WRONLY | O_TRUNC, mode);
    }
    else {
        fd = open("fconc.out", O_CREAT| O_WRONLY | O_TRUNC, mode);
    }

    write_file(fd, argv[1]);
    write_file(fd, argv[2]);

    if(close(fd) < 0){
        perror("close");
        exit(1);
    }

    return 0;
}

//continue below
```

```

void write_file(int fd, const char *infile){

    char buff[1024];
    int temp;
    ssize_t ret;

    temp = open(infile, O_RDONLY);

    if (temp < 0){
        perror(infile);
        exit(1);
    }

    while(1){
        ret = read(temp, buff, 1024);

        if ( ret == -1 ) {
            perror("read");
            exit(1);
        }

        doWrite(fd,buff,ret);

        if (ret < 1024) break;

        buff[ret]='\0';
    }

    if(close(temp) < 0 ){
        perror("close");
        exit(1);
    }
}

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

```

## 2) strace ./fconc A B C

```
execve("./fconc", ["/fconc", "A", "B", "C"], [/* 65 vars */]) = 0
brk(NULL) = 0x55ae58137000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ffaf81b1000
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=106243, ...}) = 0
mmap(NULL, 106243, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ffaf8197000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\20\5\2\0\0\0\0"... , 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1856752, ...}) = 0
mmap(NULL, 3959200, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ffaf7bc8000
mprotect(0x7ffaf7d86000, 2093056, PROT_NONE) = 0
mmap(0x7ffaf7f85000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7ffaf7f85000
mmap(0x7ffaf7f8b000, 14752, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ffaf7f8b000
close(3) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ffaf8195000
arch_prctl(ARCH_SET_FS, 0x7ffaf8195700) = 0
mprotect(0x7ffaf7f85000, 16384, PROT_READ) = 0
mprotect(0x55ae56146000, 4096, PROT_READ) = 0
mprotect(0x7ffaf81b4000, 4096, PROT_READ) = 0
munmap(0x7ffaf8197000, 106243) = 0
open("C", O_WRONLY|O_CREAT|O_TRUNC, 0666) = 3
open("A", O_RDONLY) = 4
read(4, "Goodbye,\n", 1024) = 9
write(3, "Goodbye,\n", 9) = 9
close(4) = 0
open("B", O_RDONLY) = 4
read(4, "and thanks for all the fish!\n", 1024) = 29
write(3, "and thanks for all the fish!\n", 29) = 29
close(4) = 0
close(3) = 0
exit_group(0) = ?
+++ exited with 0 +++
```