

# Λειτουργικά Συστήματα Υπολογιστών

## Αναφορά στην 1η Εργαστηριακή Άσκηση

Αλέξανδρος Σκούρας, 03120105

Ιωάννης Τσαντήλας, 03120883

Εξάμηνο: Εαρινό 2022-23

### 1.1 Σύνδεση με αρχείο αντικειμένων

#### 1.1.1 Πηγαίος Κώδικας

Για να προσπελάσουμε τα αρχεία zing.h και zing.o ακολουθούμε τα εξής βήματα:

- Χρησιμοποιούμε την εντολή `cd ../code/zing` για να μεταβούμε από τον προσωπικό μας κατάλογο(`/home/oslab/oslab16`) στο directory στο οποίο βρίσκονται τα ζητούμενα αρχεία(`/home/oslab/code/zing`).

```
oslab16@orion:~$ cd ../code/zing
```

- Κάνοντας χρήση της εντολής `cp` αντιγράφουμε τα αρχεία zing.h και zing.o στον προσωπικό μας κατάλογο ως εξής:

```
oslab16@orion:/home/oslab/code/zing$ cp zing.h /home/oslab/oslab16
oslab16@orion:/home/oslab/code/zing$ cp zing.o /home/oslab/oslab16
oslab16@orion:/home/oslab/code/zing$
```

Στην συνέχεια γράφουμε την main στο αρχείο main.c από την οποία απλά καλούμε την συνάρτηση zing. Ο κώδικας της main:

```
#include "zing.h"
#include <stdio.h>

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

#### 1.1.2 Διαδικασία μεταγλώττισης και σύνδεσης

Με τις παρακάτω εντολές αρχικά παράγουμε το object file της main (**main.o**) και στη συνέχεια συνενώνουμε τα δύο αρχεία αντικειμένων (**main.o** και **zing.o**) στο τελικό εκτελέσιμο zing:

```
oslab16@orion:~$ gcc -Wall -c main.c
oslab16@orion:~$ gcc main.o zing.o -o zing
```

### 1.1.3 Έξοδο εκτέλεσης

Τρέχουμε το τελικό εκτελέσιμο και παίρνουμε το παρακάτω μήνυμα στην οθόνη μας:

```
oslab16@orion:~$ ./zing
Hello, oslab16
```

### 1.1.4 Ερωτήσεις

- Ερώτημα 1

Μας επιτρέπει να καλέσουμε την συνάρτηση `zing()` στο αρχείο `main.c`, ενώ ο κώδικας της συνάρτησης `zing` βρίσκεται στο αρχείο `zing.c`.

- Ερώτημα 2

Για να δημιουργήσουμε το Makefile γράφουμε στην γραμμή εντολών:

```
oslab16@orion:~$ vim Makefile
```

Εντός του αρχείου, γράφουμε:

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

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

- Ερώτημα 3

Η `getlogin()` επιστρέφει έναν pointer σε string με περιεχόμενο το username μας. Αλλάζουμε λίγο την `zing()`, σε ένα νέο αρχείο `zing2.c`, όπου θα γράφει:

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

void zing(void){
    char *login_name = getlogin();
    printf("Hello dream team %s!\n",login_name);
}
```

Ακολουθούμε την ίδια διαδικασία μεταγλώττισης όπως προηγουμένως και παράγουμε το εκτελέσιμο **zing2** με έξοδο:

```
oslab16@orion:~$ ./zing2
Hello dream team oslab16!
```

Ενώ, αλλάζουμε το Makefile σε:

```
all: zing zing2

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

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

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

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

- Ερώτημα 4:

Αφού κάνουμε αλλαγές μόνο σε μία συνάρτηση, μία λύση θα ήταν να την απομονώσουμε σε ένα ξεχωριστό αρχείο π.χ. **my\_function.c** και να κρατήσουμε όλες τις υπόλοιπες στο άλλο αρχείο, π.χ. **functions.c**.

Επίσης, φτιάχνουμε ένα Makefile, το οποίο θα μεταγλωττίζει ξεχωριστά τα δύο αρχεία και θα τα κάνει link μεταξύ τους. Στην εκτέλεση της εντολής *make*, η μεταγλώττιση θα γίνει μόνο στα αρχεία που θα αλλάχουν, δηλαδή μόνο στο **my\_function.c**, εξοικονομώντας μας χρόνο.

- Ερώτημα 5:

Ο gcc φτιάχνει το εκτελέσιμο του foo.c και το γράφει στο foo.c, με αποτέλεσμα, το περιεχόμενο του αρχικού foo.c να χάνεται. Ουσιαστικά, ο κώδικας που υπήρχε στο foo.c γίνεται overwrite από το binary αρχείο του εκτελέσιμου του.

## **1.2 Συνένωση δύο αρχείων σε τρίτο**

### **1.2.1 Πηγαίος κώδικας**

```

#include <sys/types.h> //for ssize_t, size_t
#include <sys/stat.h> //for mode
#include <fcntl.h> //for oflags, mode, open, close
#include <stdio.h>
#include <stdlib.h> //for size_t
#include <unistd.h> //for read, write, close
#include <string.h>

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

void write_file(int fd, const char *infile){
    int fd_in=open(infile, O_RDONLY);
    if(fd_in==-1){
        perror(infile);
        exit(1);
    }
    char buff[1024];
    ssize_t rcnt; //ssize_t is a "wider" int, same as size_t, but
signed, to represent "-1"
    while(1){
        rcnt = read(fd_in, buff, sizeof(buff)-1);
        if(rcnt == 0) break;
        if(rcnt == -1){
            perror("Error reading input file");
            exit(1);
        }
        doWrite(fd, buff, rcnt);
    }
    close(fd_in);
}

int main(int argc, char ** argv){
    if(argc<3 || argc>4){
        fprintf(stderr, "Usage: %s infile1 infile2 [outfile (default:fconc.out)]\n", argv[0]);
        exit (1);
    }

    if(argc == 3 && (strcmp(argv[2], "fconc.out") == 0 || strcmp(argv[1], "fconc.out") == 0) ){
        fprintf(stderr, "Reading and writing on the same file\n");
    }
}

```

```
if(argc == 4 && (strcmp(argv[2], argv[3]) == 0 || strcmp(argv[1], argv[3]) == 0 ) ){
    fprintf(stderr, "Reading and writing on the same file\n");
    exit(1);
}

int fd_out, oflags, mode;

oflags = O_CREAT | O_WRONLY | O_TRUNC; //Create one if it doesn't exist, only write, erase contents if they exist
mode = S_IRUSR | S_IWUSR;             //user is able to read or write

if(argc==4) fd_out=open(argv[3], oflags, mode);
else fd_out=open("fconc.out", oflags, mode);

if(fd_out == -1){
    perror("Error opening output file");
    exit(1);
}

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

close(fd_out);
return 0;
```

### 1.2.2 Ερωτήσεις

Εκτελώντας `strace ./fconc B B`, όπου το B περιέχει το μήνυμα “dream team!!!!”:

```
oslab16@orion:~$ strace ./fconc B B
execve("./fconc", ["/fconc", "B", "B"], [/* 19 vars */]) = 0
brk(0) = 0x15d0000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or d
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or d
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=29766, ...}) = 0
mmap(NULL, 29766, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8ec3109000
close(3) = 0
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or d
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\3\0>\0\1\0\0\0P\34\2\0\0\0\0\
fstat(3, {st_mode=S_IFREG|0755, st_size=1738176, ...}) = 0
mmap(NULL, 3844640, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3,
mprotect(0x7f8ec2ce9000, 2097152, PROT_NONE) = 0
mmap(0x7f8ec2ee9000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXE
mmap(0x7f8ec2eef000, 14880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXE
close(3) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
arch_prctl(ARCH_SET_FS, 0x7f8ec3107700) = 0
mprotect(0x7f8ec2ee9000, 16384, PROT_READ) = 0
mprotect(0x7f8ec3113000, 4096, PROT_READ) = 0
munmap(0x7f8ec3109000, 29766) = 0
open("fconc.out", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 3
open("B", O_RDONLY) = 4
read(4, "dream team!!!!\n", 1023) = 16
write(3, "dream team!!!!\n", 16) = 16
read(4, "", 1023) = 0
close(4) = 0
open("B", O_RDONLY) = 4
read(4, "dream team!!!!\n", 1023) = 16
write(3, "dream team!!!!\n", 16) = 16
read(4, "", 1023) = 0
close(4) = 0
close(3) = 0
exit_group(0) = ?
+++ exited with 0 +++
```

## 1.3 Προαιρετικές ερωτήσεις

### Ερώτημα 3

Αλλάζουμε μόνο την main:

```
int main(int argc, char ** argv){
    if(argc<3){
        fprintf(stderr, "Usage: %s infile1 infile2 [outfile (default:fconc.out)]\n", argv[0]);
        exit (1);
    }

    if((argc == 3) && ((strcmp(argv[1], "fconc2.out") == 0) || (strcmp(argv[2], "fconc2.out") == 0))) {
        fprintf(stderr, "Reading and writing on the same file\n");
        exit(1);
    }

    for(int i=1; i<argc-1; i++){
        if((argc > 3) && (strcmp(argv[i], argv[argc-1]) == 0) ) {
            fprintf(stderr, "Reading and writing on the same file\n");
            exit(1);
        }
    }
    int fd_out, oflags, mode;
    oflags = O_CREAT | O_WRONLY | O_TRUNC; //Create one if it doesn't exist, only write, erase contents if they exist
    mode = S_IRUSR | S_IWUSR;           //user is able to read or write

    if(argc==3) fd_out=open("fconc2.out", oflags, mode);
    else fd_out=open(argv[argc-1], oflags, mode);

    for(int i=1; i<argc-1; i++) write_file(fd_out, argv[i]);

    close(fd_out);
    return 0;
}
```