



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
6^ο Εξάμηνο: Λειτουργικά Συστήματα 2022-23
Δανάη Σπέντζου (03120237)
Νεκτάριος Μπούμπαλος(03120441)

Άσκηση 1.1

Σύνδεση με αρχείο αντικειμένων

1. Αντιγραφή αρχείων zing.h και zing.o στον κατάλογο εργασίας

```
cp -r /home/oslab/code/zing/zing.h /home/oslab/oslab162/exercise01/  
cp -r /home/oslab/code/zing/zing.o /home/oslab/oslab162/exercise01/
```
2. Δημιουργία αρχείου αντικειμένων main.o για τη συνάρτηση main()

```
#include "zing.h"  
  
int main(){  
    zing();  
    return 0;  
}
```

main.c

```
gcc -Wall -c main.c //compile and create main.o
```

3. Σύνδεση (linking) των δύο αρχείων αντικειμένων

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

```
[oslab162@orion:~/exercise01$ ./zing  
Hello, oslab162
```

Ερωτήσεις:

1. Η επικεφαλίδα περιέχει, συνήθως, δηλώσεις συναρτήσεων, μεταβλητών, σταθερών, μακροεντολών που απαιτούνται από ένα πρόγραμμα. Οι σταθερές μπορεί να αποτελούν καθολικά δεδομένα που είναι κοινά σε όλο το πρόγραμμα, ενώ οι μακροεντολές μπορεί να επιτρέπουν στον προγραμματιστή να προσαρμόσει τη συμπεριφορά στο κώδικα σε διάφορες συνθήκες. Συνολικά, η επικεφαλίδα είναι ένα σημαντικό στοιχείο της διαδικασίας του μεταγλωττισμού και της εκτέλεσης ενός προγράμματος σε πολλές γλώσσες προγραμματισμού.

2. Δημιουργία zing2.c

```
#include<stdio.h>  
#include<unistd.h>  
  
void zing(void){  
  
    printf("Hello, %s \n",getlogin());  
}
```

```
gcc -Wall -c zing2.c. //compile and create zing2.o  
gcc -Wall zing2.o main.o -o zing2 //creates zing2
```

```
[oslab162@orion:~/exercise01$ ./zing2  
Hello, oslab162
```

```

all: zing zing2

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

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

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

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

clean:
    rm -f main.o zing zing2 zing2.o

```

3. Όταν κάνουμε αλλαγές σε μια συνάρτηση, μπορούμε να εκτελέσουμε το `make`. Τότε, μόνο οι απαραίτητες συναρτήσεις θα μεταγλωττιστούν και θα συνδεθούν ξανά, ενώ οι υπόλοιπες θα παραμείνουν αμετάβλητες, εξοικονομώντας χρόνο και ενέργεια στη διαδικασία μεταγλώττισης και εκτέλεσης του προγράμματος.
4. Η εντολή που δόθηκε δεν είναι σωστή. Συγκεκριμένα, η εντολή καλεί τον `gcc` compiler να μεταγλωττίσει το αρχείο `foo.c` και να δημιουργήσει ένα εκτελέσιμο με όνομα `"foo.c"`. Έτσι, το αρχείο `foo.c` έχει αντικατασταθεί από ένα εκτελέσιμο με το ίδιο όνομα, καθιστώντας το αρχικό αρχείο αδύνατο να ανακτηθεί. Σωστή εντολή θα ήταν `gcc -Wall -o foo foo.c`, η οποία θα δημιουργούσε ένα εκτελέσιμο με όνομα `"foo"` από το αρχείο `foo.c`, χωρίς να αντικαθιστά το αρχικό αρχείο.

Άσκηση 1.2

```

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

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

    int fd1, fd2, fd3, oflags, mode;
    char *buff[20];
    memset(buff, 0, sizeof(buff));

    oflags = O_CREAT | O_TRUNC | O_RDWR;
    mode = S_IRUSR | S_IWUSR;

    fd1 = open("A", O_RDONLY);
    fd2 = open("B", O_RDONLY);

    if(fd1==-1){
        printf("A: No such file or directory\n");
        exit(1);
    }
    if(fd2==-1){
        printf("B: No such file or directory\n");
        exit(1);
    }

    if(argv[3])
        fd3 = open("C", oflags, mode);
    else
        fd3 = open("fconc.out", oflags, mode);

    int rdcnt=0, cnt=0;
    while (1){
        rdcnt=read(fd1, buff, sizeof(buff)-1);
        if(rdcnt==-1 || rdcnt==0)
            break;
        while(1){
            cnt=write(fd3, buff, rdcnt);
            if (cnt==-1 || cnt==0 || cnt==rdcnt)
                break;
        }
    }
    rdcnt=0; cnt=0;
    while (1){
        rdcnt=read(fd2, buff, sizeof(buff)-1);
        if(rdcnt==-1 || rdcnt==0)
            break;
        while(1){
            cnt=write(fd3, buff, rdcnt);
            if(cnt==-1 || cnt==0 || cnt==rdcnt)
                break;
        }
    }

    close(fd1); close(fd2); close(fd3);
    return 0;
}

```

Εκτέλεση χωρίς Makefile

```
oslab162@orion:~/exercise02$ ls
fconc.c
oslab162@orion:~/exercise02$ gcc -Wall -c fconc.c
oslab162@orion:~/exercise02$ ls
fconc.c  fconc.o
oslab162@orion:~/exercise02$ gcc -Wall fconc.o -o fconc
oslab162@orion:~/exercise02$ ls
fconc  fconc.c  fconc.o
oslab162@orion:~/exercise02$ ./fconc A
A: No such file or directory
oslab162@orion:~/exercise02$ echo 'Goodbye,' > A
oslab162@orion:~/exercise02$ echo 'and thanks for all the fish!' > B
oslab162@orion:~/exercise02$ ./fconc A B
oslab162@orion:~/exercise02$ cat fconc.out
Goodbye,
and thanks for all the fish!
oslab162@orion:~/exercise02$ ./fconc A B C
oslab162@orion:~/exercise02$ cat C
Goodbye,
and thanks for all the fish! _
```

Εκτέλεση με Makefile

```
all :fconc clean

fconc:fconc.c
    gcc fconc.c -o fconc

A:
    echo 'Goodbye, ' > A

B:
    echo 'and thanks for all the fish!' > B

clean:
    rm -rf C fconc.out
```

```
oslab162@orion:~/exercise02$ make
gcc -Wall fconc.c -o fconc
rm -rf C fconc.out
oslab162@orion:~/exercise02$ ./fconc A B
A: No such file or directory
oslab162@orion:~/exercise02$ make A B
echo 'Goodbye, ' > A
echo 'and thanks for all the fish!' > B
oslab162@orion:~/exercise02$ ./fconc A B
oslab162@orion:~/exercise02$ cat fconc.out
Goodbye,
and thanks for all the fish!
oslab162@orion:~/exercise02$ ./fconc A B C
oslab162@orion:~/exercise02$ cat C
Goodbye,
and thanks for all the fish!
```

Ερώτηση strace:

```
oslab162@orion:~/exercise02$ strace ./fconc A B
execve("./fconc", ["/fconc", "A", "B"], [/* 21 vars */]) = 0
brk(0) = 0x21c7000
access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or
directory)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fbc282cb000
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=29766, ...}) = 0
mmap(NULL, 29766, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fbc282c3000
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\3\0>\0\1\0\0\0P\34\2\0\0\0\0\0"...
, 832)
= 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1738176, ...}) = 0
mmap(NULL, 3844640, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0)
= 0x7fbc27d02000
mprotect(0x7fbc27ea3000, 2097152, PROT_NONE) = 0
mmap(0x7fbc280a3000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a1000) = 0x7fbc280a3000
mmap(0x7fbc280a9000, 14880, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fbc280a9000
close(3) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fbc282c2000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fbc282c1000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7fbc282c0000
arch_prctl(ARCH_SET_FS, 0x7fbc282c1700) = 0
mprotect(0x7fbc280a3000, 16384, PROT_READ) = 0
mprotect(0x7fbc282cd000, 4096, PROT_READ) = 0
munmap(0x7fbc282c3000, 29766) = 0
open("A", O_RDONLY) = 3
open("B", O_RDONLY) = 4
open("fconc.out", O_RDWR|O_CREAT|O_APPEND, 0600) = 5
read(3, "Goodbye, \n", 159) = 10
write(5, "Goodbye, \n\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0\0"...
, 159) = 159
read(3, "", 159) = 0
read(4, "and thanks for all the fish!\n", 159) = 29
write(5, "and thanks for all the fish!\n\0\0\0"...
, 159) = 159
read(4, "", 159) = 0
close(3) = 0
close(4) = 0
close(5) = 0
exit_group(0) = ?
+++ exited with 0 +++
```