Λειτουργικά Συστήματα 6ο εξάμηνο,
Ακαδημαϊκή περίοδος 2019-2020
Άσκηση 1: Εισαγωγή στο περιβάλλον προγραμματισμού

1.1 Σύνδεση με αρχείο αντικειμένων

Βήματα:

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

Η αντιγραφή των δυο αυτών αρχείων γίνεται με τις κάτωθι εντολές

```
oslaba30@os-node1:/home/oslab/code/zing$ cp /home/oslab/code/zing/zing.h /home/oslab/oslaba30/ask1/p1.1/
oslaba30@os-node1:/home/oslab/code/zing$ cp /home/oslab/code/zing/zing.o /home/oslab/oslaba30/ask1/p1.1/
```

2. Δημιουργία αρχείου αντικειμένων main.o για τη συνάρτηση main().

oslaba30@os-node1: ~/ask1/p1.1

File Edit View Search Terminal Help

```
#include "zing.h"
int main(int arg, char **argv){
        zing();
        return 0;

}
```

Και τρέχουμε την εντολή :

```
oslaba30@os-node1:~/ask1/p1.1$ gcc -Wall -c main.c
```

και δημιουργείται το αρχείο αντικείμενων main.o

3. Σύνδεση (linking) των δύο αρχείων αντικειμένων

```
oslaba30@os-node1:~/ask1/p1.1$ ./linking
Hello, oslaba30
```

Ερωτήσεις:

1. Ποιο σκοπό εξυπηρετεί η επικεφαλίδα;

Αντί για να κάνω compile την zing.c κάθε φορά που θα κάνω compile την main, δημιουργώ μια επικεφαλίδα για την zing (και για κάθε “zing”). Έτσι, κάθε φορά που εκτελώ την main η zing προϋπάρχει σαν ήδη χτισμένη βιβλιοθήκη και δεν χρειάζεται να γίνεται κάθε φορά compile.

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

Είναι το first κομμάτι του Makefile αυτό που εκτελείται όταν κάνουμε το make, και το οποίο κάνει compile το zing.c και το main.c και το εκτελέσιμο είναι το linking.

oslaba30@os-node1: ~/ask1/p1.1

File Edit View Search Terminal Help

```
first: linking
linking: zing.o main.o
	gcc -o linking zing.o main.o
main.o: main.c
	gcc -Wall -c main.c

second: final

final: zing.o zing2.o main2.o
	gcc -o final zing.o zing2.o main2.o
main2.o: main2.c
	gcc -Wall -c main2.c
zing2.o : zing2.c
	gcc -Wall -c zing2.c
```

3. Παράξτε 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

Zing2.c:

oslaba30@os-node1: ~/ask1/p1.1

File Edit View Search Terminal Help

```
#include <stdio.h>
#include <unistd.h>
void zing2(){
	char *name;
	name = getlogin();
	printf("Geia sou magka %s!\n", name);
}
```

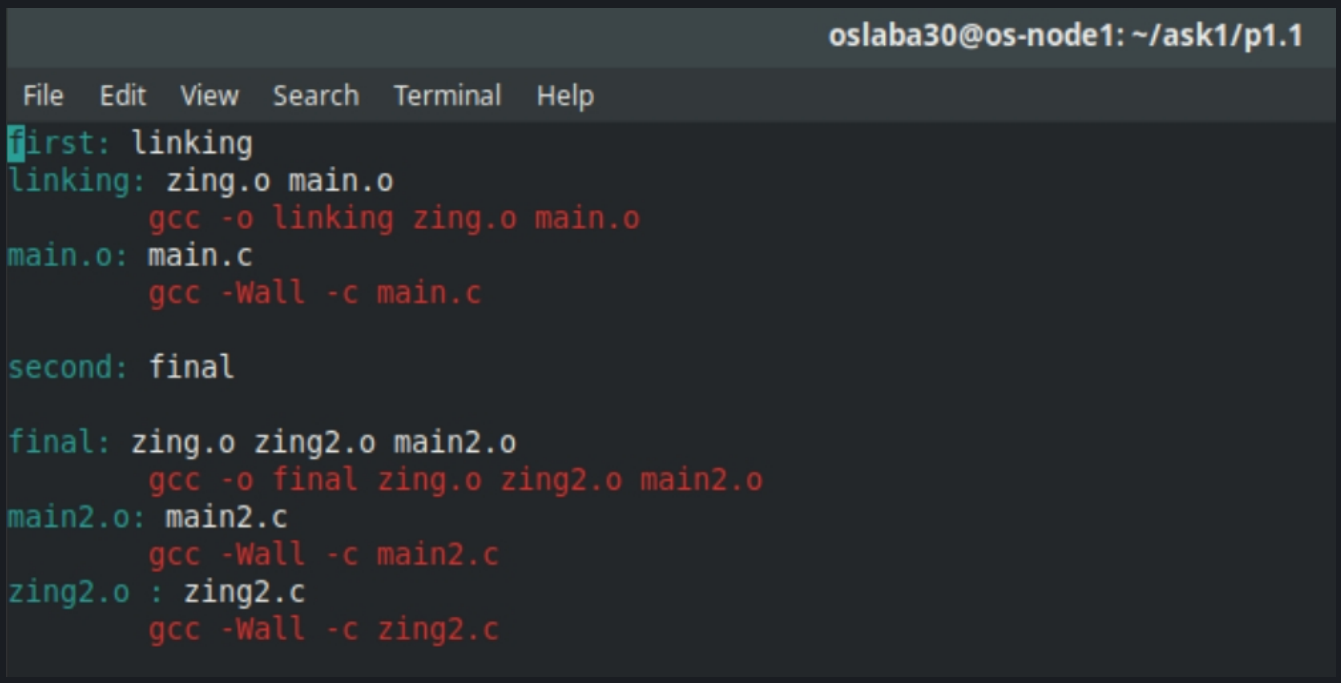
Zing2.h:

oslaba30@os-node1: ~/ask1/p1.1

File Edit View Search Terminal Help

```
#ifndef ZING2_H__
#define ZING2_H__
void zing2();
#endif
~
```

Makefile:


oslaba30@os-node1: ~/ask1/p1.1
File Edit View Search Terminal Help
first: linking
linking: zing.o main.o
gcc -o linking zing.o main.o
main.o: main.c
gcc -Wall -c main.c
second: final
final: zing.o zing2.o main2.o
gcc -o final zing.o zing2.o main2.o
main2.o: main2.c
gcc -Wall -c main2.c
zing2.o : zing2.c
gcc -Wall -c zing2.c

main2.c:

```
oslaba30@os-node1: ~/ask1/p1.1
File  Edit  View  Search  Terminal  Help
#include "zing.h"
#include "zing2.h"

int main(int argc, char **argv){
        zing();
        zing2();
        return 0;
}
```

Αποτέλεσμα final:

```
oslaba30@os-node1:~/ask1/p1.1$ ./final
Hello, oslaba30
Geia sou magka oslaba30!
```

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;

Χρησιμοποιώντας το Makefile και κάνοντας με αυτό τα compile όταν κάνουμε μία αλλαγή σε κάποια συνάρτηση τότε γίνεται compile μόνο αυτή η συνάρτηση και όχι όλο το πρόγραμμα από την αρχή (με όλες τις συναρτήσεις). Με αυτόν τον τρόπο γλυτώνουμε από το κόστος μεταγλώττισης.

5. Ο συνεργάτης σας και εσείς δουλεύατε στο πρόγραμμα foo.c όλη την προηγούμενη εβδομάδα. Καθώς κάνατε ένα διάλειμμα και ο συνεργάτης σας δούλευε 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;

Στην θέση του foo.c που ήταν ο πηγιαίως κώδικας, τώρα αποθηκέυτηκε το εκτελέσιμο αρχείο με όνομα foo.c . Έτσι χάθηκε ο κώδικας μας.

1.2 Συνένωση δύο αρχείων σε τρίτο

Ο κώδικας:

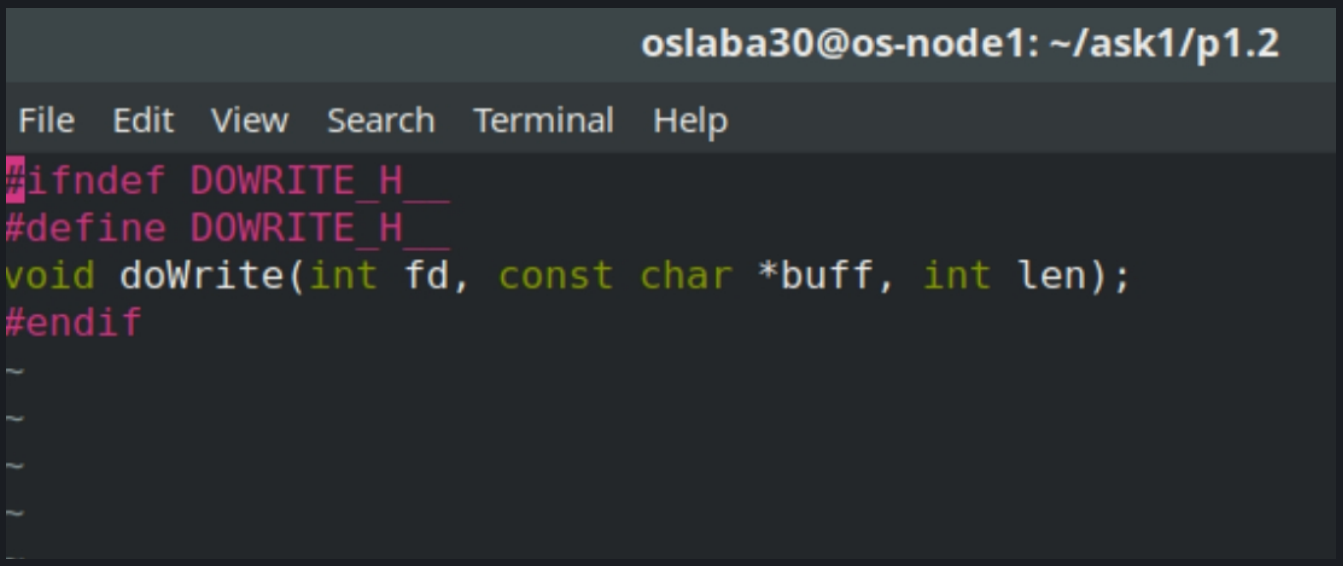
doWrite.c

oslaba30@os-node1: ~/ask1/p1.2

File Edit View Search Terminal Help

```c
#include <unistd.h>
#include <stdio.h>
#include <string.h>
void doWrite(int fd, const char *buff, int len){
        size_t indx=0;
        ssize_t wcnt;
        do{
                wcnt=write(fd, buff + indx , len - indx);
                if (wcnt ==-1)perror("write");
                indx += wcnt;
        } while (indx < len);
}
```

doWrite.h


oslaba30@os-node1: ~/ask1/p1.2
File Edit View Search Terminal Help
#ifndef DOWRITE_H__
#define DOWRITE_H__
void doWrite(int fd, const char *buff, int len);
#endif

write_file.c

oslaba30@os-node1: ~/ask1/p1.2

File Edit View Search Terminal Help

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include "doWrite.h"

void write_file(int fd,const char *infile){// fd ==fd_w
        char buff[1024];
        int i ;
        for(i=0;i<1024;i++){
        buff[i]=(char)0;}
        ssize_t rcnt;
        size_t len ;
        int fd_r=open(infile,O_RDONLY);
        int eof=0;
        while (eof==0){
                rcnt = read(fd_r,buff,sizeof(buff)-1);
                if (rcnt == 0) eof=1;/*end_of_file*/
                if (rcnt == -1)perror("read");/*dont_read_file*/

        }
  len = strlen(buff);
  doWrite(fd ,buff, len );
}
~
```

write_file.h

oslaba30@os-node1: ~/ask1/p1.2

File Edit View Search Terminal Help

```c
#ifndef WRITE_FILE_H__
#define WRITE_FILE_H__
void write_file(int fd, const char *infile);
#endif
~
```

fconc.c

oslaba30@os-node1: ~/ask1/p1.2

File Edit View Search Terminal Help

```c
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include "write_file.h"
int main(int argc, char **argv)
{
        if (argc>4||argc<3 ){
                printf("Usage: ./fconc infile1 infile2 [outfile (default:fconc.out)] \n" );
                return 1;
        }

        if (argc==3 ){
                argv[3]="fconc.out";
       }
        FILE* arxio_1 = fopen (argv[1],"r");
        FILE* arxio_2 = fopen (argv[2],"r");
        if(arxio_1== NULL|| arxio_2== NULL){
                printf("No such file or directory \n");
                return 1;
        }

        if(*argv[1]==*argv[3]||*argv[2]==*argv[3]){
                printf("Infile different from outfile \n");
                return 1;
        }

        int fd_w;

        fd_w=open(argv[3],O_CREAT|O_WRONLY|O_TRUNC, S_IRUSR|S_IWUSR);
        write_file(fd_w , argv[1]);
        write_file(fd_w , argv[2]);
        return 0;
}
```

Makefile

```
oslaba30@os-node1: ~/ask1/p1.2
File  Edit  View  Search  Terminal  Help
fconc: doWrite.o write_file.o fconc.o
        gcc -o fconc doWrite.o write_file.o fconc.o
doWrite.o: doWrite.c
        gcc -Wall -c doWrite.c
write_file.o: write_file.c
        gcc -Wall -c write_file.c
fconc.o: fconc.c
        gcc -Wall -c fconc.c
~
~
~
```

και όταν τρέξουμε μας βγάζει το επιθυμητό αποτέλεσμα:

```
oslaba30@os-node1: ~/ask1/p1.2
File  Edit  View  Search  Terminal  Help
oslaba30@os-node1:~/ask1/p1.2$ ./fconc A
Usage: ./fconc infile1 infile2 [outfile (default:fconc.out)]
oslaba30@os-node1:~/ask1/p1.2$ ./fconc A B
oslaba30@os-node1:~/ask1/p1.2$ ./fconc A B C
oslaba30@os-node1:~/ask1/p1.2$ ./fconc G F C
No such file or directory
oslaba30@os-node1:~/ask1/p1.2$ cat C
Goodbye magka!,
and thanks for all the fish!
oslaba30@os-node1:~/ask1/p1.2$ cat fconc.out
Goodbye magka!,
and thanks for all the fish!
oslaba30@os-node1:~/ask1/p1.2$
```

και οι εντολές strace στις δυο εκτελέσεις:

strace ./fconc A B

```
oslaba30@os-node1:~/ask1/p1.2$ strace ./fconc A B
execve("./fconc", ["./fconc", "A", "B"], [/* 26 vars */]) = 0
brk(0)                                  = 0x1509000
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) = 0x7f26912b0000
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=30952, ...}) = 0
mmap(NULL, 30952, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f26912a8000
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"..., 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) = 0x7f2690ce7000
mprotect(0x7f2690e88000, 2097152, PROT_NONE) = 0
mmap(0x7f2691088000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a1000) = 0x7f2691088000
mmap(0x7f269108e000, 14880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f269108e000
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f26912a7000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f26912a6000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f26912a5000
arch_prctl(ARCH_SET_FS, 0x7f26912a6700) = 0
mprotect(0x7f2691088000, 16384, PROT_READ) = 0
mprotect(0x7f26912b2000, 4096, PROT_READ) = 0
munmap(0x7f26912a8000, 30952)           = 0
brk(0)                                  = 0x1509000
brk(0x152a000)                          = 0x152a000
open("A", O_RDONLY)                     = 3
open("B", O_RDONLY)                     = 4
open("fconc.out", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 5
open("A", O_RDONLY)                     = 6
read(6, "Goodbye magka,\n", 1023)       = 15
read(6, "", 1023)                       = 0
write(5, "Goodbye magka,\n", 15)        = 15
open("B", O_RDONLY)                     = 7
read(7, "thanks for all the fish!1\n", 1023) = 26
read(7, "", 1023)                       = 0
write(5, "thanks for all the fish!1\n", 26) = 26
exit_group(0)                           = ?
+++ exited with 0 +++
oslaba30@os-node1:~/ask1/p1.2$
```

strace./fconc A B C

```
oslaba30@os-node1:~/ask1/p1.2$ strace ./fconc A B C
execve("./fconc", ["./fconc", "A", "B", "C"], [/* 26 vars */]) = 0
brk(0)                                  = 0xb3d000
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) = 0x7fdfc6a2b000
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=30952, ...}) = 0
mmap(NULL, 30952, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fdfc6a23000
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"..., 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) = 0x7fdfc6462000
mprotect(0x7fdfc6603000, 2097152, PROT_NONE) = 0
mmap(0x7fdfc6803000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a1000) = 0x7fdfc6803000
mmap(0x7fdfc6809000, 14880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fdfc6809000
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdfc6a22000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdfc6a21000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fdfc6a20000
arch_prctl(ARCH_SET_FS, 0x7fdfc6a21700) = 0
mprotect(0x7fdfc6803000, 16384, PROT_READ) = 0
mprotect(0x7fdfc6a2d000, 4096, PROT_READ) = 0
munmap(0x7fdfc6a23000, 30952)           = 0
brk(0)                                  = 0xb3d000
brk(0xb5e000)                           = 0xb5e000
open("A", O_RDONLY)                     = 3
open("B", O_RDONLY)                     = 4
open("C", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 5
open("A", O_RDONLY)                     = 6
read(6, "Goodbye magka,\n", 1023)       = 15
read(6, "", 1023)                       = 0
write(5, "Goodbye magka,\n", 15)        = 15
open("B", O_RDONLY)                     = 7
read(7, "thanks for all the fish!1\n", 1023) = 26
read(7, "", 1023)                       = 0
write(5, "thanks for all the fish!1\n", 26) = 26
exit_group(0)                           = ?
+++ exited with 0 +++
oslaba30@os-node1:~/ask1/p1.2$
```