

Λειτουργικά Συστήματα

Πέτρος Αγγελάτος
Διονύσης Ζήνδρος

Άσκηση 1.1

Πηγαίος κώδικας

```
#include "zing.h"

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

Μεταγλώττιση και σύνδεση

```
oslabd10@paxoi:~/exer1/1.1$ gcc -Wall main.c -c -o main.o
oslabd10@paxoi:~/exer1/1.1$ gcc -Wall main.o zing.o -o main
```

Έξοδος εκτέλεσης

```
oslabd10@paxoi:~/exer1/1.1$ ./main
Hello oslabd10!
```

Ερωτήσεις

1. Η επικεφαλίδα είναι ένα αρχείο το οποίο περιέχει δηλώσεις συναρτήσεων χωρίς την υλοποίησή τους. Όταν γίνεται compile ένα αρχείο που χρησιμοποιεί μία συνάρτηση ο compiler χρειάζεται μόνο τη δήλωση της συνάρτησης για να το κάνει. Συνεπώς οι επικεφαλίδες μειώνουν τον χρόνο του compilation.

2. Makefile:

```
main: main.c zing.o
    make compile
    make link

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

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

clean:
    rm -f main.o *~
```

3. Το πρόβλημα αυτό μπορεί να αντιμετωπισθεί φτιάχνοντας ένα αρχείο με όλες τις συναρτήσεις που δε θέλουμε να πειράζουμε με την υλοποίησή τους και ένα αρχείο με τις δηλώσεις από όλες τις συναρτήσεις που δε θέλουμε να πειράζουμε. Στη συνέχεια κάνουμε compile το αρχείο με τις συναρτήσεις και χρησιμοποιούμε το header file για να δουλέψουμε με την συνάρτηση που θέλουμε να κάνουμε αλλαγές.

4. Αυτό που έγινε ήταν να δώσει εντολή στον gcc να γράψει το αποτέλεσμα στο αρχείο foo.c. Το αρχείο foo.c όμως περιέχει τον πηγαίο κώδικα και μετά την εκτέλεση της εντολής θα γίνει overwrite από το αποτέλεσμα της μεταγλώτισης.

Άσκηση 1.2

Πηγαίος Κώδικας

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

const char* FCONC_MESSAGE_USAGE = "Usage: fconc infile1 infile2 ... outfile\n";
const char* FCONC_MESSAGE_FILE_NOT_FOUND = "Unable to open file: %s\n"; // used
both for reading and writing
const char* FCONC_DEFAULT_OUT = "fconc.out";

void doWrite( int fd, const char* buff, int len ) {
    write( fd, buff, len );
}

void copyContents( int outFD, int inFD ) {
    int size;
    char buffer[ 1024 ];

    while ( ( size = read( inFD, buffer, 1024 ) ) ) {
        doWrite( outFD, buffer, size );
    }
}

int parseArgs( int argc, char** argv, char*** inFiles, int* count, char**
outFile ) {
    if ( argc < 3 ) {
        fprintf( stderr, FCONC_MESSAGE_USAGE );
        return 1;
    }

    *inFiles = ( char** )malloc( ( argc - 2 ) * sizeof( char* ) );

    for ( int i = 0; i < argc - 2; i++ ) {
        ( *inFiles )[ i ] = argv[ i + 1 ];
    }
}
```

```

    *outFile = argv[ argc - 1 ];

    return 0;
}

int safeOpen( char* file, int flags ) {
    int fd = open( file, flags, S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH );

    if ( fd == -1 ) {
        fprintf( stderr, FCONC_MESSAGE_FILE_NOT_FOUND, file );
        return -1;
    }
    return fd;
}

int main( int argc, char** argv ) {
    char **inFiles, *outFile;
    int inFD1, inFD2, outFD;

    if ( parseArgs( argc, argv, &inFiles, &outFile ) ) {
        return 2;
    }

    if ( ( outFD = safeOpen( outFile, O_WRONLY | O_CREAT ) ) == -1
        || ( inFD1 = safeOpen( inFile1, O_RDONLY ) ) == -1
        || ( inFD2 = safeOpen( inFile2, O_RDONLY ) ) == -1 ) {
        return 3;
    }

    copyContents( outFD, inFD1 );
    copyContents( outFD, inFD2 );

    close( outFD );
    close( inFD1 );
    close( inFD2 );

    return 0;
}

```

Ερωτήσεις

```

oslabdl0@paxoi:~/exer1/1.2$ strace ./fconc A B
execve("./fconc", [ "./fconc", "A", "B" ], [ /* 16 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) =
0xb782b000
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=62977, ...}) = 0
mmap2(NULL, 62977, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb781b000
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\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) =
0xb76d4000
mmap2(0xb7815000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x141) = 0xb7815000
mmap2(0xb7818000, 10600, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0xb7818000
close(3) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0xb76d3000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb76d36c0, limit:1048575,
seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0,
useable:1}) = 0
mprotect(0xb7815000, 8192, PROT_READ) = 0
mprotect(0xb7849000, 4096, PROT_READ) = 0
munmap(0xb781b000, 62977) = 0
open("fconc.out", O_WRONLY|O_CREAT, 0644) = 3
open("A", O_RDONLY) = 4
open("B", O_RDONLY) = 5
read(4, "Goodbye, \n"..., 1024) = 10
write(3, "Goodbye, \n"..., 10) = 10
read(4, ""..., 1024) = 0
read(5, "and thanks for all the fish! http://9gag.com\n"..., 1024) = 45
write(3, "and thanks for all the fish! http://9gag.com\n"..., 45) = 45
read(5, ""..., 1024) = 0
close(3) = 0
close(4) = 0
close(5) = 0
exit_group(0) = ?
oslabd10@paxoi:~/exer1/1.2$

```