**Λειτουργικά Συστήματα**

**Άσκηση 1**

Περράκης Γιώργος 03113511

**Άσκηση 1.1**

main.c

#include “zing.h”

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

{

zing();

return 0;

}

**Ερώτηση 1:**

H επικεφαλίδα μας δίνει τη δυνατότητα να εκτελούμε μέσω της main πολλές διαφορετικές συναρτήσεις ανάλογα με το εκτελέσιμο αρχείο που επιλέγουμε να τρέξουμε. Όλες αυτές οι διαφορετικές συναρτήσεις συνδέονται με την main μέσω της επικεφαλίδας, η οποία κάθε φορά μπορεί να αναφέρεται σε διαφορετική συνάρτηση.

**Ερώτηση 2:**

**Makefile**

zing: zing.o main.o

gcc zing.o main.o –o zing

main.o: main.c

gcc –Wall –c main.c

**Ερώτηση 3:**

zing2.c

#include <stdio.h>

#include <unistd.h>

void zing(void)

{

char \*username=getlogin();

printf(“What’s up %s!\n”, username);

}

Compiling main.c

gcc –Wall –c main.c

linking main.o with zing.o και δημιουργία αρχείου εκτελέσιμου zing

gcc main.o zing.o –o zing

Αντίστοιχα για zing2

gcc –Wall –c zing2.c

gcc main.o zing2.o –o zing2

**Makefile**

all: zing zing2

zing: main.o zing.o

gcc –o zing.o main.o zing

zing2: main.o zing2.o

gcc –o zing2.o main.o zing2

main.o: main.c

gcc –Wall –c main.c

**Ερώτηση 4:**

Για την επίλυση του συγκεκριμένου προβλήματος θα φτιάξω ένα ξεχωριστό αρχείο και θα βάλω μέσα μόνο την συνάρτηση που κάνω αλλαγές τώρα. Στην συνέχεια, με την βοήθεια του Makefile θα μεταγλωττίζω μόνο το συγκεκριμένο αρχείο, καθώς και αρχεία που εξαρτώνται από αυτό (όπως η main) και έτσι θα αποφεύγω τις περιττές μεταγλωττίσεις των υπόλοιπων συναρτήσεων.

**Ερώτηση 5:**

Αυτό που συνέβη είναι ότι το αρχείο foo.c μετατράπηκε σε εκτελέσιμο αρχείο και ως εκ τούτου είναι αρχείο που μπορώ να το τρέξω και όχι να το ανοίξω για να διαβάσω και αλλάξω τον πηγαίο κώδικα.

**Άσκηση 1.2**

Πηγαίος κώδικας:

fconc1.c

#include <unistd.h>

#include <stdio.h>

#include "header.h"

#include <fcntl.h>

#include <unistd.h>

int main(int argc, char \*argv[])

{

int fd,oflags,mode;

oflags = O\_CREAT | )\_WRONLY | O\_TRUNC;

mode = S\_IRUSR | S\_IWUSR;

if(argc == 1 | argc == 2)

{

printf("Usage: ./fconc infile1,infile2 [outfile (default:fconc.out)]");

}

else if(argc == 3)

{

fd = open("fconc.out", oflags, mode);

if(fd == -1)

{

perror("open");

return 2;

}

function(fd,argv[1]);

function(fd,argv[2]);

}

else if(argc == 4)

{

fd = open(argv[3], oflags, mode);

if(fd == -1)

{

perror("open");

return 3;

}

function(fd,argv[1]);

function(fd,argv[2]);

close(fd);

}

return 0;

}

doWrite1.c

#include <stdio.h>

#include "header.h"

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#include <sys/stat.h>

#include <stdlib.h>

#include <sys/types.h>

#include <errno.h>

int function(int output\_fd, char \*source\_file)

{

char buff[1024];

ssize\_t rcnt, wcnt; //number of bytes returned by read() and write()

size\_t len, idx;

int input\_fd; //input file descriptors

int fd, oflags, mode;

oflags = O\_WRONLY | O\_CREAT;

mode = S\_IRUSR | S\_IWUSR;

input\_fd = open(source\_file, O\_RDONLY);

if(input\_fd == -1)

{

perror("open");

return 2;

}

for(;;)

{

rcnt = read(input\_fd, buff, 1024);

if(rcnt == 0)

return 0;

if(rcnt == -1)

{

perror("read");

return 4;

}

idx = 0;

do

{

wcnt = write(output\_fd, buff + idx, rcnt - idx);

if(wcnt == -1)

{

perror("write");

return 5;

}

idx += wcnt;

} while ( idx < len);

}

close(input\_fd);

return (EXIT\_SUCCESS);

}

Υπάρχει και το header.h

int function(int output\_fd, char \*source\_file);

**Eρώτηση**

execve("./fconc1", ["./fconc1", "A", "B", "C"], [/\* 18 vars \*/]) = 0

open("C", O\_WRONLY|O\_CREAT|O\_TRUNC, 0600) = 3

open("A", O\_RDONLY) = 4

read(4, "Goodbye,you \n", 1023) = 13

write(3, "Goodbye,you \n", 13) = 13

read(4, "", 1023) = 0

open("B", O\_RDONLY) = 5

read(5, "and thanks for all the fish!\n", 1023) = 29

write(3, "and thanks for all the fish!\n", 29) = 29

read(5, "", 1023) = 0

close(3) = 0

exit\_group(0) = ?