Λειτουργικά Συστήματα Υπολογιστών

1η Εργαστηριακή Αναφορά

Θοδωρής Παπαρρηγόπουλος el18040
Ορφέας Φιλιππόπουλος el18082

Άσκηση 1

Πηγαίος Κώδικας

main.c:

```
#include “zing.h”
int main(int argc, char **argv) {
   zing();
   return 0;
}
```

Ερωτήσεις

1) Γενικά, η χρήση επικεφαλίδας χρησιμοποιείται προκειμένου να δηλώσουμε συναρτήσεις χωρίς την υλοποίηση τους. Οι επικεφαλίδες μας εξυπηρετούν για πολλούς σκοπούς. Αρχικά, όταν γίνεται compile ένα αρχείο που χρησιμοποιεί μία συνάρτηση, ο compiler χρειάζεται μόνο το declaration της συνάρτησης. Έτσι, μειώνεται ο χρόνος το compile. Επιπλέον, σε μεγάλα project, το να έχεις χωρισμένο τα αρχεία σε declaration και implementation βολεύει από άποψη ότι εύκολα ανατρέχεις να δεις τις διαθέσιμες συναρτήσεις σε κάθε αρχείο χωρίς να χρειάζεται να περνάς και τις υλοποιήσεις τις κάθε συνάρτησης.

2) Makefile:

```
all: main mainy

main: main.o zing.o
	gcc -o main main.o zing.o
mainy: main.o zing2.o
	gcc -o mainy main.o zing2.o

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

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

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

3) Κάνουμε implement την zing() στο zing2.c ως
#include "zing2.h"

```
void zing(){
   char *s;
   if( ( s = getlogin() ) == NULL ) {
      printf( "cannot find login name\n" );
   } else {
      printf( "Hello,your login name is %s\n", s );
   }
}
```

4) Γενικά είναι good practice να σπάμε τα αρχεία μας σε υπο αρχεία. Στη συγκεκριμένη περίπτωση, μπορούμε να κάνουμε ένα νέο αρχείο που θα βάλουμε την συνάρτηση που δουλεύουμε, έτσι στο compile δεν θα ανιχνεύει αλλαγή στις υπόλοιπες 499 συναρτήσεις του άλλου αρχείο. Συνεπώς το compile θα γίνεται πολύ πιο γρήγορα σε σχέση με πριν.

5) Αυτό που έγινε ήταν ότι έδωσε για εκτελέσιμο αρχείο το αρχείο που είχε τον πηγαίο κώδικα. Έτσι, ο κώδικας που έγραφε έγινε overwrite από το αποτέλεσμα του compile (το binary file). Για πειραματικούς σκοπούς έφτιαξα ένα απλό αρχείο test.c

```
#include <stdio.h>
int main() {
        printf(“Hello world\n”);
        return 0;
}
```

Τρέξαμε gcc -Wall -o test.c test.c
Και μετά χρησιμοποιώντας το νέο test.c τρέξαμε ./test.c και έτρεξε κανονικά τυπώνοντας Hello world !

Ωστόσο, τρέχοντας την ίδια εντολή σε native linux environment με gcc (Ubuntu 7.5.0-3ubuntu1~18.04) 7.5.0 μου πέταξε το εξής error ο compiler:
gcc: fatal error: input file ‘test.c’ is the same as output file
compilation terminated.
Συνεπώς, ο συνεργάτης καλό είναι να δουλεύει locally πρώτα.

Άσκηση 2

Πηγαίος Κώδικας
Έχουμε κάνει 2 διαφορετικές υλοποιήσεις.
1)

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define defaultOutput "fconc.out"
#define defaultOutput2 "fconc_2.out"
#define defaultOutput3 "fconc_3.out"

int main(int argc, char **argv) {
   FILE *inputFile1, *inputFile2;
   FILE *outputFile;

   /* Check if number of arguments is correct, if not show message to standard error */
   if (argc != 3 && argc != 4) {
      fprintf(stderr, "Usage: ./fconc infile1 infile2 [outfile (default:fconc.out)]\n");
      return 0;
   }

   /* Open the 2 input files */
   inputFile1 = fopen(argv[1],"r");
   if (inputFile1 == NULL) {
         fprintf(stderr, "%s: No such file or directory\n", argv[1]);
         exit(EXIT_FAILURE);
   }

   inputFile2 = fopen(argv[2], "r");
   if (inputFile2 == NULL) {
         fclose(inputFile1);
         fprintf(stderr, "%s: No such file or directory\n", argv[2]);
         exit(EXIT_FAILURE);
   }

   /* Open the output file */
   if (argc == 3) {
      if ( !((strcmp(argv[1], defaultOutput) == 0) | (strcmp(argv[2], defaultOutput) == 0) )){
         outputFile = fopen(defaultOutput, "w+");
         printf("Output file is at: %s\n", defaultOutput);
      }
      else if(! ((strcmp(argv[1], defaultOutput2) == 0) | (strcmp(argv[2], defaultOutput2) == 0) ) ) {
         outputFile = fopen(defaultOutput2, "w+");
         printf("Output file is at: %s\n", defaultOutput2);
      }
      else {
         outputFile = fopen(defaultOutput3, "w+");
         printf("Output file is at: %s\n", defaultOutput3 );
      }
   }
```

```
   else {    /* argc == 4 */
      if (! ((strcmp(argv[1], argv[3]) == 0) | (strcmp(argv[2], argv[3]) == 0) )) {
         /* Print to argv[3] */
         outputFile = fopen(argv[3], "w+");
         printf("Output file is: %s\n", argv[3]);
      }
      else {
         /* Output file is the same with input -> It will be overwritten */
         if ( !((strcmp(argv[1], defaultOutput) == 0) | (strcmp(argv[2], defaultOutput) == 0) )){
            outputFile = fopen(defaultOutput, "w+");
            printf("Output file is at: %s\n", defaultOutput);
         }
         else if(! ((strcmp(argv[1], defaultOutput2) == 0) | (strcmp(argv[2], defaultOutput2) == 0) ) )
{
            outputFile = fopen(defaultOutput2, "w+");
            printf("Output file is at: %s\n", defaultOutput2);
         }
         else {
            outputFile = fopen(defaultOutput3, "w+");
            printf("Output file is at: %s\n", defaultOutput3 );
         }
      }
   }

   /* Read from files character by character and immediately
      write to the output file (char by char)*/
   char ch;
   while((ch = fgetc(inputFile1)) != EOF) {
      fputc(ch,outputFile);
   }

   while((ch = fgetc(inputFile2)) != EOF) {
      fputc(ch,outputFile);
   }

   fclose(inputFile1);
   fclose(inputFile2);
   fclose(outputFile);

   return 0;
}
```

2)

==========

----------game.c(main)----------

```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include "write_file.h"
//i#include "write_fun.h"
#include <unistd.h>
```

```
#include <stdbool.h>

struct all{

        int fd;
        char* copy;
        bool boo;
};


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

    if(argc>4){
        fprintf(stderr, "Too many arguments\n");
        return 0;
    }

    struct all array[argc-2];


    printf("Nubmer of arguments: %d \n", argc);
    if(argc > 4 || argc < 3){
        fprintf(stderr, "You need to only add 2 or 3 arguments\n");

        return 0;
    }

    int oflags = O_CREAT | O_WRONLY | O_TRUNC;

    int mode = S_IRUSR | S_IWUSR;

    int fd3;
    int i;

    for(i = 0; i < argc-2; i++){
        if(*argv[i+1] == *argv[argc-1]){
                array[i].boo = true;
                printf("Waring: %s is already passed as an argument and it will be overwritten, are you
sure?\n", argv[i]);
                printf("Press enter to continue or any other button to do not");

                char c;

                scanf("%c",&c);

                if(c == 10){
                        continue;
                }
               else return 0;

        }
    }
```

```
 FILE* files[argc-2];

 for(i = 0; i<2; i++){
    if(array[i].boo == true){
          printf("we go a true one\n");

          array[i].fd = open(argv[i+1], O_RDONLY);

          if(array[i].fd == -1){
          printf("%s: No such file or directory \n", argv[i+1]);
          exit(1);
          }


          files[i] = fdopen(array[i].fd, "r");

          fseek(files[i], 0L, SEEK_END);
          long int res = ftell(files[i]); // it affects fd1 too
          //fseek(fp, 0L, SEEK_SET); it doesnt affect fd1
          fclose(files[i]);

          array[i].fd = open(argv[i+1], O_RDONLY);
          printf("%d\n", array[i].fd);
          array[i].copy = (char *)calloc(res+1, sizeof(char));

          ssize_t rcnt1 = 0;

          for(;;){
                printf("hallo there\n");
                rcnt1 = read(array[i].fd, array[i].copy, res);
                if(rcnt1 == 0){
                      break;
                }

                if(rcnt1 == -1){
                      perror("read");
                      exit(1);
                 }

          }

    }
 }

if(argc == 4) fd3 = open(argv[argc-1], oflags, mode);
else fd3 = open("fconc.out", oflags, mode);

printf("%d\n", fd3);

for(i = 1; i<argc-1; i++){
    write_file(fd3, argv[i], array[i-1].fd, array[i-1].copy, array[i-1].boo);
```

```
   }



   for(i = 1; i<argc-2; i++){
       free(array[i].copy);
   }

   return 0;
}
```

----------end of main----------

----------WRITE_FILE----------

----------write_file.h----------

```
#ifndef WRITE_FILE_H
#define WRITE_FILE_H
#include "write_fun.h"
#include <unistd.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>

void write_file(int fd, const char *infile, int fd_inn,  char* buff, bool boo);

#endif
```

----------write_file.c----------

```
#include "write_file.h"


void  write_file(int fd, const char *infile, int fd_inn, char* buff, bool boo){

       if(boo == true){
             doWrite(fd, buff, strlen(buff));

       }
       else{

             int fd1;
             fd1 = open(infile, O_RDONLY);
```

```
        if(fd1 == -1){
                printf("%s: No such file or directory \n", infile);
                exit(1);
        }

        FILE * fp = fdopen(fd1, "r");

        fseek(fp, 0L, SEEK_END);
        long int res = ftell(fp); // it affects fd1 too
//      fseek(fp, 0L, SEEK_SET); it doesnt affect fd1

        fd1 = open(infile, O_RDONLY);

        char * con;
        con = (char *)calloc(res+1, sizeof(char));



        ssize_t rcnt1 = 0;

        for(;;){

                rcnt1 = read(fd1, con, res);
                if(rcnt1 == 0){
                        break;
                }

                if(rcnt1 == -1){
                        perror("read");
                        exit(1);
                }

        }

        //con[res-1] = 0; without \n between the files

        doWrite(fd, con, strlen(con));

        if(close(fd1) == -1){
                printf("error closing file %s", infile);
                exit(1);
        }

        free(con);
        fclose(fp);
    }
}
```

----------WRITE_FUN----------

----------write_fun.h----------

```
#ifndef WRITE_FUN_H
#define WRITE_FUN_H

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

void  doWrite(int fd, char *  buff, int len);

#endif
```

----------write_fun.c----------

```
#include "write_fun.h"


void  doWrite(int fd, char *  buff, int len){
	int idx = 0;
	idx = 0;
	int wcnt;
	do {
	   wcnt = write(fd, buff + idx, len - idx);
	   if(wcnt == -1){
	      perror("write");
	      exit(1);
	   }
	   else{
	   idx += wcnt;
	   }
	} while (idx < len);

}
```

==========END_OF_CODE==========

Ερωτήσεις

- Στον πρώτο πηγαίο κώδικα αντιστοιχεί :

```
thodpap@thodpap:~/Documents/Σχολή/University/ComputerFlow/OperationalSystems/Labab/2nd Exercise(Theo)$ strace ./fconc A B C
execve("./fconc", ["./fconc", "A", "B", "C"], 0x7ffd4aa176d8 /* 70 vars */) = 0
brk(NULL)                               = 0x56529b325000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=209156, ...}) = 0
mmap(NULL, 209156, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f2cf5455000
close(3)                                = 0
```

```
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/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\0\20\35\2\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=2030928, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f2cf5453000
mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f2cf4e6f000
mprotect(0x7f2cf5056000, 2097152, PROT_NONE) = 0
mmap(0x7f2cf5256000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) =
0x7f2cf5256000
mmap(0x7f2cf525c000, 15072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7f2cf525c000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f2cf54544c0) = 0
mprotect(0x7f2cf5256000, 16384, PROT_READ) = 0
mprotect(0x56529b296000, 4096, PROT_READ) = 0
mprotect(0x7f2cf5489000, 4096, PROT_READ) = 0
munmap(0x7f2cf5455000, 209156)          = 0
brk(NULL)                               = 0x56529b325000
brk(0x56529b346000)                     = 0x56529b346000
openat(AT_FDCWD, "A", O_RDONLY)         = 3
openat(AT_FDCWD, "B", O_RDONLY)         = 4
openat(AT_FDCWD, "C", O_RDWR|O_CREAT|O_TRUNC, 0666) = 5
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 1), ...}) = 0
write(1, "Output file is: C\n", 18Output file is: C
)     = 18
fstat(3, {st_mode=S_IFREG|0644, st_size=17, ...}) = 0
read(3, "This is a test1.\n", 4096)     = 17
fstat(5, {st_mode=S_IFREG|0644, st_size=0, ...}) = 0
read(3, "", 4096)                       = 0
fstat(4, {st_mode=S_IFREG|0644, st_size=25, ...}) = 0
read(4, "This is the second test.\n", 4096) = 25
read(4, "", 4096)                       = 0
close(3)                                = 0
close(4)                                = 0
write(5, "This is a test1.\nThis is the sec"..., 42) = 42
close(5)                                = 0
exit_group(0)                           = ?
```

- Στον δεύτερο πηγαίο κώδικα αντιστοιχεί :

```
thodpap@thodpap:~/Documents/Σχολή/Ροή Υ/Λειτουργικά Συστήματα Υπολογιστών/Εργαστήριο/1η Σειρά/2η Άσκηση(ορφεας)$
strace ./fconc A B C
execve("./fconc", ["./fconc", "A", "B", "C"], 0x7ffd98d6f8d8 /* 72 vars */) = 0
brk(NULL)                               = 0x55ffeb7b8000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=209156, ...}) = 0
mmap(NULL, 209156, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fbca7025000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/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\0\20\35\2\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=2030928, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fbca7023000
mmap(NULL, 4131552, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7fbca6a3f000
mprotect(0x7fbca6c26000, 2097152, PROT_NONE) = 0
mmap(0x7fbca6e26000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000)
= 0x7fbca6e26000
mmap(0x7fbca6e2c000, 15072, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0x7fbca6e2c000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7fbca70244c0) = 0
mprotect(0x7fbca6e26000, 16384, PROT_READ) = 0
mprotect(0x55ffe9ccd000, 4096, PROT_READ) = 0
mprotect(0x7fbca7059000, 4096, PROT_READ) = 0
munmap(0x7fbca7025000, 209156)          = 0
openat(AT_FDCWD, "C", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 3
openat(AT_FDCWD, "A", O_RDONLY)         = 4
fcntl(4, F_GETFL)                       = 0x8000 (flags O_RDONLY|O_LARGEFILE)
```

```
brk(NULL)                               = 0x55ffeb7b8000
brk(0x55ffeb7d9000)                     = 0x55ffeb7d9000
fstat(4, {st_mode=S_IFREG|0644, st_size=17, ...}) = 0
fstat(4, {st_mode=S_IFREG|0644, st_size=17, ...}) = 0
lseek(4, 0, SEEK_SET)                   = 0
read(4, "This is a test1.\n", 17)       = 17
openat(AT_FDCWD, "A", O_RDONLY)         = 5
read(5, "This is a test1.\n", 17)       = 17
read(5, "", 17)                         = 0
write(3, "This is a test1.\n", 17)      = 17
close(5)                                = 0
close(4)                                = 0
openat(AT_FDCWD, "B", O_RDONLY)         = 4
fcntl(4, F_GETFL)                       = 0x8000 (flags O_RDONLY|O_LARGEFILE)
fstat(4, {st_mode=S_IFREG|0644, st_size=25, ...}) = 0
fstat(4, {st_mode=S_IFREG|0644, st_size=25, ...}) = 0
lseek(4, 0, SEEK_SET)                   = 0
read(4, "This is the second test.\n", 25) = 25
openat(AT_FDCWD, "B", O_RDONLY)         = 5
read(5, "This is the second test.\n", 25) = 25
read(5, "", 25)                         = 0
write(3, "This is the second test.\n", 25) = 25
close(5)                                = 0
close(4)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```