Λειτουργικά Συστήματα – Άσκηση 1η

ΑΓΓΕΛΟΣ ΣΤΑΗΣ 03117435, ΣΩΚΡΑΤΗΣ ΠΟΥΤΑΣ 03117054 (Ομάδα: oslabb08)

Άσκηση 1.1 – Σύνδεση με Αρχείο Αντικειμένων

Κώδικας main.c

```
#include <stdio.h>
#include "zing.h"
int main(int argc, char ** argv)
{
        zing();
        return 0;
}
```

Διαδικασία Μεταγλώττισης και Σύνδεσης

```
gcc -Wall -c main.c
gcc main.o zing.o -o exec1
```

Έξοδος Εκτέλεσης

```
Hello, oslabb08
```

Ερωτήσεις

1. Οι επικεφαλίδες περιέχουν τις δηλώσεις συναρτήσεων και ορισμό global μεταβλητών που είναι κοινοί σε πολλαπλά αρχεία ενός προγράμματος.

2. **Κώδικας MakeFile**

```
exec1 :  main.o zing.o
         gcc main.o zing.o -o exec1
main.o : main.c
         gcc -Wall -c main.c
```

Εκτέλεση MakeFile: `make -f Makefile` **ή** `make`

3. **Κώδικας zing2.c**

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

Κώδικας Νέου MakeFile

```
all : exec1 exec2
exec1 : main.o zing.o
        gcc -o exec1 main.o zing.o
exec2 : main.o zing2.o
        gcc -o exec2 main.o zing2.o
main.o : main.c
         gcc -c main.c
zing2.o : zing2.c
          gcc -c zing2.c
```

4. Μπορεί ο κώδικας της συνάρτησης αλλά και άλλων επιμέρους συναρτήσεων να μεταφερθεί σε ξεχωριστό αρχείο (πχ fun.c) και να συμπεριληφθεί στο κυρίως πρόγραμμα χρησιμοποιώντας #include “fun.h”. Με αυτό το τρόπο θα γίνει ξεχωριστά η μεταγλώττιση αυτής της συνάρτησης και δε θα χρειάζεται κάθε φορά μεταγλώττιση του υπόλοιπου προγράμματος στο οποίο δε κάνουμε αλλαγές. Στη συνέχεια θα γίνει η σύνδεση όλων των αρχείων για τη δημιουργία του τελικού εκτελέσιμου αρχείου.

5. Ουσιαστικά με τη τελευταία εντολή δημιουργήσαμε ένα εκτελέσιμο αρχείο foo.c το οποίο αντικατέστησε το αντίστοιχο αρχικό αρχείο πηγαίου κώδικα αφού δώσαμε ως όρισμα ονόματος για το εκτελέσιμο που θα παράγει ο μεταγλωττιστής το foo.c.

Άσκηση 1.2 - Συνένωση δύο αρχείων σε τρίτο

Κώδικας

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

void doWrite(int fd, const char * buff,size_t len) //len was suggested to
be of type int
{
        size_t idx = 0;
        ssize_t wcount;

        do{
                wcount = write(fd, buff + idx, len - idx);
                if(wcount == -1)
                {
                        perror("write");
                        exit(1);
                }

                idx += wcount;

        } while(idx < len);
}

void write_file(int fd, const char * infile)
{
        int read_fd;
        read_fd = open(infile, O_RDONLY);

        if( read_fd == -1)
        {
                perror("open");
                exit(1);
        }
```

```
        char buff[1024];
        size_t rcount;

        rcount = read(read_fd, buff, 1023);

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

        while(rcount > 0)
        { //not End Of File
                doWrite(fd, buff, rcount);
                rcount = read(read_fd, buff, 1023);
        }

        if(rcount == -1) // check if exited while due to a read error
        {
                perror("read");
                exit(1);
        }

        close(read_fd);
}

int main(int argc, char** argv)
{
        if(argc < 3 || argc > 4)
        { //error
                printf("Impropper call.\n");
                printf("Usage: ./exec input_file_1 input_file_2 ");
                printf("output_file[default: fconc.out]\n");
                return 0;
        }

        int fd_out;

        if(argc == 3)
        { // use default output
                fd_out = open("fconc.out", O_CREAT|O_WRONLY|O_TRUNC,
S_IRUSR | S_IWUSR | S_IRGRP | S_IROTH);
        }
        else if(argc ==4)
        { // use argv[3] for output

            if( strcmp(argv[1], argv[3]) == 0 || strcmp(argv[2], argv[3])
== 0 )
              {
                 printf("error: output file needs to be different from
input files\n");
                 return 0;
           }
                fd_out = open(argv[3], O_WRONLY|O_TRUNC);
        }
```

```
        write_file(fd_out, argv[1]);
        write_file(fd_out, argv[2]);

        close(fd_out);

        return 0;
}
```

Ερωτήσεις

1. Εκτελώντας την εντολή strace ./fconc A B C παίρνουμε την έξοδο:

```
oslabb08@os-node1:~/askisi_1/section_1_2$ strace ./fconc A B  C
execve("./fconc", ["./fconc", "A", "B", "C"], [/* 18 vars */]) = 0
brk(0)                                  = 0xcca000
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) = 0x7f5122273000
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) = 0x7f512226b000
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) = 0x7f5121caa000
mprotect(0x7f5121e4b000, 2097152, PROT_NONE) = 0
mmap(0x7f512204b000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a1000) = 0x7f512204b000
mmap(0x7f5122051000, 14880, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f5122051000
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f512226a000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f5122269000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f5122268000
arch_prctl(ARCH_SET_FS, 0x7f5122269700) = 0
mprotect(0x7f512204b000, 16384, PROT_READ) = 0
mprotect(0x7f5122275000, 4096, PROT_READ) = 0
munmap(0x7f512226b000, 30952)           = 0
open("C", O_WRONLY|O_TRUNC)             = 3
open("A", O_RDONLY)                     = 4
read(4, "Goodbye,\n", 1023)             = 9
write(3, "Goodbye,\n", 9)               = 9
read(4, "", 1023)                       = 0
close(4)                                = 0
open("B", O_RDONLY)                     = 4
read(4, "and thanks for the fish.\n", 1023) = 25
write(3, "and thanks for the fish.\n", 25) = 25
read(4, "", 1023)                       = 0
close(4)                                = 0
close(3)                                = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```