

## Λειτουργικά Συστήματα

### Άσκηση1: Εισαγωγή στο περιβάλλον προγραμματισμού

#### 1.1 Σύνδεση με αρχείο αντικειμένων

Αρχικά δημιουργήσαμε τη συνάρτηση `main()` (αρχείο `main.c`) μέσα από την οποία καλέσαμε τη συνάρτηση `zing()`. Για την κλήση της `zing` μέσω της `main` χρειάστηκε `compile` της `main` και σύνδεση των δύο αρχείων με τις πιο κάτω εντολές:

```
gcc -Wall -c main.c
```

```
gcc main.o zing.o -o zing
```

Καλώντας το εκτελέσιμο `zing` η έξοδος είναι: `Hello, oslaba05`

Ο κώδικας που χρησιμοποιήσαμε φαίνεται πιο κάτω:

```
#include "zing.h"

int main() { zing();

            return 0;

            }
```

#### Απαντήσεις:

1. Στη `main()` χρησιμοποιήσαμε την επικεφαλίδα **`#include <zing.h>`**. Στην επικεφαλίδα αυτή δηλώνεται το πρότυπο της `zing()`. Δηλαδή τι τύπο ορισμάτων δέχεται και τι τύπος είναι τα δεδομένα που επιστρέφει.

**2. Το αρχείο Makefile περιείχε τα εξής:**

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

**3. Δημιουργήσαμε τη zing2.**

Καλώντας το εκτελέσιμο zing2 η έξοδος είναι: I am useroslaba05

Ο κώδικας φαίνεται πιο κάτω:

```
#include <unistd.h>  
#include <stdio.h>  
#include <stdlib.h>  
int zing()  
{  
    int k;  
    char *p;  
    p = getlogin(); //returns a pointer to a string  
                   //containing the name of the user  
                   //logged in on the controlling  
                   //terminal of the process, or a NULL  
                   //pointer if this information cannot be  
                   //determined.  
    if (p==NULL) //checks if error occurred  
    {  
        perror("Error getting username");
```

```
return 1;    }
```

```
k = write(1, "I am user ",10);
```

```
if (k==-1)    //error indicates that a later  
              //request may succeed
```

```
{
```

```
    perror("Error in write attemp 1");
```

```
    //print a system error message
```

```
    return 1;    }
```

```
k = write(1, p, 10);
```

```
if (k==-1)
```

```
{
```

```
    perror("Error in write attemp 2");
```

```
    return 1;    }
```

```
k = write(1,"\n",1);
```

```
if (k==-1)
```

```
{
```

```
    perror("Error in write attemp 3");
```

```
    return 1;    }
```

```
return 0;
```

```
}
```

Επίσης αλλάξαμε το Makefile σε:

```
zing2: zing2.o main.o zing.o
    gcc zing2.o main.o -o zing2 g
    cc zing.o main.o -o zing
zing2.o: zing2.c
    gcc -Wall -c zing2.c
main.o: main.c
    gcc -Wall -c main.c
```

**4.**Ο τρόπος για να μειωθεί ο χρόνος μεταγλώττισης είναι ο εξής: γράφουμε την κάθε συνάρτηση και τη main() που τις καλεί, σε ξεχωριστά αρχεία. Κατόπιν γράφουμε τους κανόνες μεταγλώττισης και σύνδεσης(linking) των αντικειμένων σε ένα Makefile. Έτσι εκτελώντας την εντολή make μεταγλωττίζονται μόνο οι συναρτήσεις που έχει αλλάξει το περιεχόμενό τους.

**5.** Εκτελώντας την εντολή gcc -Wall -o foo.c foo.c λέμε στον compiler να μεταγλωττίσει το αρχείο foo.c και το εκτελέσιμο που θα παραχθεί να το ονομάσει foo.c, το οποίο αντικαθιστά το προηγούμενο που περιέχει τον κώδικα του προγράμματος .

**Άσκηση 1.2 – Συνένωση δύο αρχείων σε τρίτο**

1. Σε αυτό το μέρος της άσκησης καλούμαστε να δημιουργήσουμε ένα εκτελέσιμο αρχείο με όνομα `fconc` το οποίο θα παίρνει τα περιεχόμενα 2 αρχείων και θα τα συνενώνει σε ένα τρίτο αρχείο, ενώ το τρίτο όρισμα θα είναι το όνομα του αρχείου εξόδου το οποίο είναι προαιρετικό με προεπιλεγμένο όνομα το `fconc.out`. Στον φάκελο της αναφοράς συμπεριλαμβάνω το αρχείο κώδικα για καλύτερη και ευκολότερη ανάγνωση του.

Πιο κάτω παρατίθεται ο πηγαίος κώδικας για το αρχείο `fconc.c`

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
```

```
#include <fcntl.h>
```

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

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

```
#include <unistd.h>
```

```
void doWrite(int fd, const char *buff, int len){
```

```
    size_t idx = 0;
```

```
    ssize_t wcnt;
```

```
    do {
```

```
        wcnt = write(fd, buff + idx, len - idx);
```

```
        if (wcnt == -1){ /* error */
```

```
            perror("write");
```

```
        exit(1);
    }

    idx += wcnt;
} while (idx < len);

return;
}

void write_file(int fdw, int ftr){
    char buff[1024];
    ssize_t rcnt = read(ftr,buff,sizeof(buff));
    for (;;) {
        if (rcnt == 0) /* end-of-file */
            return;

        if (rcnt == -1) { /* error */
            perror("read");
            exit(1);
        }

        doWrite(fdw,buff,rcnt);
        rcnt = read(ftr,buff,sizeof(buff));
    }
}
```

```
int main(int argc, char **argv)
{
    //Check inputs
    if ( ( argc < 3 ) || (argc > 4) ) {
        write(1, "Usage: ./fconc infile1 infile2 [outfile
(default:fconc.out)]\n",61) ;
        return(1) ;
    }

    //open read A
    int fda,fdb,fdc;

    fda = open(argv[1], O_RDONLY); // flag = read only
    if (fda == -1){
        perror("open");
        return(1);
    }

    //open read B
    fdb = open(argv[2], O_RDONLY); // flag = read only
    if (fdb == -1){
```

```
perror("open");

return(1);

}

char *fout;

if ( argc == 4 )

    fout = argv[3] ;

else

    fout = "fconc.out" ;


//open write C

int oflags, mode;

oflags = O_CREAT|O_WRONLY|O_TRUNC;

// flag O_CREAT => create the file if it does not already exist

// flag O_WRONLY => open for writing only

// flag O_TRUNC => truncate existing file to zero length (if file already
exists)

mode = S_IRUSR|S_IWUSR;

// mode : the permissions to be set on the file , only if open() creates it

// mode S_IRWXU => permissions read , write , execute/search by owner

// mode S_IRWXG => permissions read , write , execute/search by group

// mode S_IRWXO => permissions read , write , execute/search by others
```



```
fdc = open(fout, oflags, mode);

if (fdc == -1){

perror("error open");

return(1);

}

write_file(fdc,fda);

write_file(fdc,fdb);

close(fda);

close(fdb);

close(fdc);

return (0);

}
```

**Απαντήσεις:**

2. Εκτελέσαμε ένα παράδειγμα της fconc χρησιμοποιώντας την εντολή strace και πήραμε το κάτωθεν αποτέλεσμα. Στον φάκελο της αναφοράς συμπεριλαμβάνω txt αρχείο για καλύτερη ανάγνωση του αποτελέσματος.

```
oslab05@os-node2:~/lab1$ strace ./fconc A B C
execve("./fconc", [ "./fconc", "A", "B", "C" ], [/* 26 vars */]) = 0
brk(0)                               = 0x1362000
```

```
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) = 0x7fc7b1cab000
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=30868, ...}) = 0
mmap(NULL, 30868, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fc7b1ca3000
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\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) = 0x7fc7b16e2000
mprotect(0x7fc7b1883000, 2097152, PROT_NONE) = 0
mmap(0x7fc7b1a83000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1a1000) =
0x7fc7b1a83000
mmap(0x7fc7b1a89000, 14880, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7fc7b1a89000
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fc7b1ca2000
mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fc7b1ca1000
mmap(NULL, 4096, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7fc7b1ca0000
```

```
arch_prctl(ARCH_SET_FS, 0x7fc7b1ca1700) = 0
mprotect(0x7fc7b1a83000, 16384, PROT_READ) = 0
mprotect(0x7fc7b1cad000, 4096, PROT_READ) = 0
munmap(0x7fc7b1ca3000, 30868) = 0
open("A", O_RDONLY) = 3
open("B", O_RDONLY) = 4
open("C", O_WRONLY|O_CREAT|O_TRUNC, 0600) = 5
read(3, "Hello World\n", 1024) = 12
write(5, "Hello World\n", 12) = 12
read(3, "", 1024) = 0
read(4, "and Universe\n", 1024) = 13
write(5, "and Universe\n", 13) = 13
read(4, "", 1024) = 0
close(3) = 0
close(4) = 0
close(5) = 0
exit_group(0) = ?
+++ exited with 0 +++
```