



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ

Λειτουργικά Συστήματα

1η Άσκηση

Ομάδα : oslaba111

Ονοματεπώνυμο : Ιωάννης Παλαιός

A.M : el18184

Ονοματεπώνυμο : Στέφανος Μιχαλάκης

A.M : 03114781

Ασκήσεις :

Σύνδεση με αρχείο αντικειμένων

- 1. Ποιο σκοπό εξυπηρετεί η επικεφαλίδα;

Περιέχει το πρότυπο και την δήλωση της συνάρτησης zing() ενώ στην zing.o περιέχεται η υλοποίηση της. Επίσης μέσω αυτής της επικεφαλίδας επιτυγχάνεται η διεπαφή με άλλα κομμάτια κώδικα όπως γίνεται στην main.c.

- 2. Ζητείται κατάλληλο Makefile για τη δημιουργία του εκτελέσιμου της άσκησης.

Το Makefile ορίστηκε ως εξής :

zing: main.o

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

main.o: main.c

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

- 3. Παράξτε το δικό σας zing2.o, το οποίο θα περιέχει zing() που θα εμφανίζει διαφορετικό αλλά παρόμοιο μήνυμα με τη zing() του zing.o. Συμβουλευτείτε το manual page της getlogin(3). Αλλάξτε το Makefile ώστε να παράγονται δύο εκτελέσιμα, ένα με το zing.o, ένα με το zing2.o, επαναχρησιμοποιώντας το κοινό object file main.o.

Το Makefile ορίστηκε ως εξής:

all: zing zing2

zing: main.o

```

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

```

Έτσι, παράγονται δύο εκτελέσιμα: το zing.o και το zing2.o .

Το zing2.c ορίστηκε ως εξής:

```

#include "zing.h"
#include <unistd.h>
#include <stdio.h>
void zing(void)
{
    printf("Welcome, %s\n", getlogin());
}

```

- 4. Έστω ότι έχετε γράψει το πρόγραμμά σας σε ένα αρχείο που περιέχει 500 συναρτήσεις. Αυτή τη στιγμή κάνετε αλλαγές μόνο σε μία συνάρτηση. Ο κύκλος εργασίας είναι: αλλαγές στον κώδικα, μεταγλώττιση, εκτέλεση, αλλαγές στον κώδικα, κ.ο.κ. Ο χρόνος μεταγλώττισης είναι μεγάλος, γεγονός που σας καθυστερεί. Πώς μπορεί να αντιμετωπισθεί το πρόβλημα αυτό;

Μπορούμε να βαλούμε την συνάρτηση σε ένα header file έτσι ώστε να εφαρμόζουμε τις αλλαγές στο αρχείο υλοποίησης του header file έτσι ώστε κάθε φορά να μεταγλωττίζουμε μόνο την συνάρτηση που χρειαζόμαστε (αυτή που θα είναι στο header file).

- 5. Ο συνεργάτης σας και εσείς δουλεύατε στο πρόγραμμα foo.c όλη την προηγούμενη εβδομάδα. Καθώς κάνατε ένα διάλειμμα και ο συνεργάτης σας δούλεψε στον κώδικα, ακούτε μια απελπισμένη κραυγή. Ρωτάτε τι συνέβει και ο συνεργάτης σας λέει ότι το αρχείο foo.c χάθηκε! Κοιτάτε το history του φλοιού και η τελευταία εντολή ήταν η: gcc -Wall -o foo.c foo.c Τι συνέβη;

Με την εντολή gcc -Wall -o foo.c foo.c λέμε στον compiler gcc να παράξει ένα executable αρχείο με το όνομα foo.c και να διαγράψει το αρχείο foo.c αφού το command είναι το αντίστοιχο με το εξής :

```

$ gcc [options] -o [source files] output files
$ gcc -Wall -o foo.c foo.c

```

Οπότε το output file, δηλαδή εκεί που θα παραχθεί το executable, θα είναι το foo.c οπότε θα διαγραφεί το source file foo.c το οποίο θα περιέχει κώδικα σε γλώσσα c και θα γίνει executable.

Συνένωση δύο αρχείων σε τρίτο

Το source code αυτής της άσκησης είναι :

```

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

```

```

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

int main(int argc, char **argv)
{
    int fd1, fd2;
    fd1=open(argv[1], O_RDONLY);
    fd2=open(argv[2], O_RDONLY);
    if (argc > 4 || argc < 3)
    {
        printf("Usage: ./fconc infile1 infile2 [outfile (default:fconc.out)]\n");
        exit(1);
    }
    if(fd1 == -1 || fd2 == -1)
    {
        if(fd1 == -1)
            printf("%s: No such file or directory\n", argv[1]);
        if(fd2 == -1)
            printf("%s: No such file or directory\n", argv[2]);
        exit(1);
    }
    ssize_t r1, r2;
    char buff1[1024], buff2[1024];
    r1=read(fd1, buff1, sizeof(buff1)-1);
    r2=read(fd2, buff2, sizeof(buff2)-1);
    buff1[r1]='\0';
    buff2[r2]='\0';
    strcat(buff1, buff2);
    close(fd1); close(fd2);
    if(argc==3)
    {
        char *cmp= "fconc.out";
        int flags = O_RDWR|O_CREAT| O_TRUNC ;
        int fd3=open(cmp, flags, 0666); // flag = 0666 for permissions
        write(fd3, buff1, strlen(buff1));
        close(fd3);
    }
    if(argc==4)
    {
        int flags = O_CREAT | O_WRONLY | O_TRUNC;
        int fd3= open(argv[3], flags, 0666);
        write(fd3, buff1, strlen(buff1));
        close(fd3);
    }
    return 0;
}

```

}

- 1. Εκτελέστε ένα παράδειγμα του `fconc` χρησιμοποιώντας την εντολή `strace`. Αντιγράψτε το κομμάτι της εξόδου της `strace` που προκύπτει από τον κώδικα που γράψατε.

Εάν καλέσουμε όπως στο παράδειγμα της εκφώνησης της άσκησης το:
`strace ./fconc A B` θα έχει την εξής έξοδο:

```
execve("./fconc", ["/fconc", "A", "B"], 0x7ffd0cbc5670 /* 64 vars */) = 0
brk(NULL)                               = 0x55b94e6f4000
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=134482, ...}) = 0
mmap(NULL, 134482, PROT_READ, MAP_PRIVATE, 3, 0) =
0x7f0bcd388000
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\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) = 0x7f0bcd386000
mmap(NULL, 4131552, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f0bccd8f000
mprotect(0x7f0bccf76000, 2097152, PROT_NONE) = 0
mmap(0x7f0bcd176000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) =
0x7f0bcd176000
mmap(0x7f0bcd17c000, 15072, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f0bcd17c000
close(3)                                = 0
arch_prctl(ARCH_SET_FS, 0x7f0bcd3874c0) = 0
mprotect(0x7f0bcd176000, 16384, PROT_READ) = 0
mprotect(0x55b94d428000, 4096, PROT_READ) = 0
mprotect(0x7f0bcd3a9000, 4096, PROT_READ) = 0
munmap(0x7f0bcd388000, 134482)          = 0
```

```
openat(AT_FDCWD, "A", O_RDONLY)      = 3
openat(AT_FDCWD, "B", O_RDONLY)      = 4
read(3, "Goodbye,\n", 1023)          = 9
read(4, "and thanks for all the fish!\n", 1023) = 29
close(3)                              = 0
close(4)                              = 0
openat(AT_FDCWD, "fconc.out", O_RDWR|O_CREAT, 0666) = 3
write(3, "Goodbye,\nand thanks for all the "..., 38) = 38
close(3)                              = 0
exit_group(0)                         = ?
+++ exited with 0 +++
```