ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ

1η Άσκηση: Εισαγωγή στο περιβάλλον προγραμματισμού

ΦΟΙΤΗΤΕΣ: ΚΥΡΙΑΚΟΥ ΔΗΜΗΤΡΗΣ 03117601
ΧΑΤΖΗΧΡΙΣΤΟΦΗ ΧΡΙΣΤΟΣ 03117711
ΟΜΑΔΑ: OSLABC16
ΕΞΑΜΗΝΟ: 6ο

ΜΕΡΟΣ 1

1.1 Σύνδεση με αρχείο αντικειμένων

- Αντιγράφηκαν τα αρχεία zing.h και zing.o στο φάκελο oslabc16 που βρίσκεται στη διεύθυνση /home/oslab/oslabc16

- Δημιουργήθηκε το αρχείο αντικειμένου main.o της συνάρτησης main() με την ακόλουθη εντολή:

```
oslabc16@os-node1:~/Ex1$ gcc -Wall -c main.c
```

Η συνάρτηση main() είναι η εξής:

```c
#include <stdio.h>
#include "zing.h"

int main(int argc, char **argv)
{
        zing();
        return 0;
}
```

- Έγινε σύνδεση των δύο αρχείων αντικειμένων:

```
oslabc16@os-node1:~/Ex1$ gcc -o zing zing.o main.o
```

ΚΥΡΙΑΚΟΥ ΔΗΜΗΤΡΗΣ 03117601
ΧΑΤΖΗΧΡΙΣΤΟΦΗ ΧΡΙΣΤΟΣ 03117711

Ερωτήσεις

1. Η χρήση επικεφαλίδων εξασφαλίζει διαίρεση του προγράμματος σε μικρά κομμάτια κώδικα, πιο εύκολα στη διαχείριση. Ως αποτέλεσμα ο κώδικας γίνεται πιο ευανάγνωστος και η επικεφαλίδα μπορεί να χρησιμοποιηθεί από πολλά διαφορετικά προγράμματα. Επίσης, μειώνεται σημαντικά ο χρόνος μεταγλώττισης αφού, όταν υπάρξουν αλλαγές στον κώδικα, απαιτείται εκ νέου μεταγλώττιση μόνο ενός τμήματος του προγράμματος και όχι ολόκληρου.
2. Με το ακόλουθο make file επιτυγχάνεται δημιουργία του εκτελέσιμου της άσκησης

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

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

3. Για την δημιουργία του αρχείου zing2.o απαιτείται η δημιουργία του ακόλουθου αρχείου zing2.c το οποίο χρησιμιποιεί την επικεφαλίδα zing.h

```
#include <stdio.h>
#include <unistd.h>
#include "zing.h"

void zing(void)
{
        char *name;
        name = getlogin();
        printf("Goodbye, %s\n", name);
}
```

Το νέο Makefile που παράγει δύο εκτελέσιμα έχει την ακόλουθη μορφή

```
all: zing zing2

zing: zing.o main.o
	gcc -o zing zing.o main.o

zing2: zing2.o main.o
	gcc -o zing2 zing2.o main.o

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

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

clean:
	rm main.o zing2.o
```

4. Ο χρόνος μεταγλώττισης μειώνεται δραματικά όταν το πρόγραμμα διαιρείται σε μικρότερα μεταγλωττίσιμα τμήματα, όπως οι επικεφαλίδες. Επομένως συνιστάται η χρήση επικεφαλίδων για κάθε μία από τις 500 συναρτήσεις. Αυτό θα μειώσει το χρόνο μεταγλώττισης αφού τυχόν αλλαγή σε μία από αυτές απαιτεί μεταγλώττιση μόνο εκείνης της επικεφαλίδας και όχι ολόκληρου του προγράμματος.

5. Με τη χρήση της εντολής `gcc -Wall -o foo.c foo.c` το εκτελέσιμο αρχείο κάνει επανεγγραφή του αρχείου που περιέχει τον πηγαίο κώδικα με αποτέλεσμα αυτός να χάνεται και να διατηρείται μόνο το εκτελέσιμο αρχείο. Η σωστή εντολή θα ήταν `gcc -Wall -o foo foo.c`

Ακολουθεί η ολοκληρωμένη διαδικασία μεταγλώττισης και σύνδεσης με τη χρήση του Makefile καθώς και η έξοδος της εκτέλεσης του προγράμματος (πρώτα για το zing.o μόνο του και στη συνέχεια για τον συνδυασμό zing.o και zing2.o)

```
oslabc16@os-node1: ~/Ex1
File  Edit  View  Search  Terminal  Help
oslabc16@os-node1:~/Ex1$ make
gcc -o zing zing.o main.o
oslabc16@os-node1:~/Ex1$ ls
main.c  main.o  Makefile  zing  zing2.c  zing.h  zing.o
oslabc16@os-node1:~/Ex1$ ./zing
Hello, oslabc16
oslabc16@os-node1:~/Ex1$
```

```
oslabc16@os-node1: ~/Ex1
File  Edit  View  Search  Terminal  Help
oslabc16@os-node1:~/Ex1$ make
gcc -Wall -c main.c
gcc -o zing zing.o main.o
gcc -Wall -c zing2.c
gcc -o zing2 zing2.o main.o
oslabc16@os-node1:~/Ex1$ ls
main.c  main.o  Makefile  zing  zing2  zing2.c  zing2.o  zing.h  zing.o
oslabc16@os-node1:~/Ex1$ ./zing
Hello, oslabc16
oslabc16@os-node1:~/Ex1$ ./zing2
Goodbye, oslabc16
oslabc16@os-node1:~/Ex1$
```

ΚΥΡΙΑΚΟΥ ΔΗΜΗΤΡΗΣ 03117601
ΧΑΤΖΗΧΡΙΣΤΟΦΗ ΧΡΙΣΤΟΣ 03117711

ΜΕΡΟΣ 2

1.1 Συνένωση δύο αρχείων σε τρίτο

Για τη συνένωση δύο αρχείων σε ένα τρίτο δημιουργήθηκε η ακόλουθη main και οι συναρτήσεις `void doWrite` και `void write_file` καθώς και το ακόλουθο Makefile

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

        if ((argc > 4)||(argc < 3)){
            printf("Usage ./fconc infile1 infile2 [output (default:fconc.out)]\n");
            return -1;
        }

        if (argc == 3) argv[3] = "fconc.out";

        if(strcmp(argv[1],argv[3] )==0 || strcmp(argv[2],argv[3])==0){
            printf("This action cannot be operated.\n");
            printf("Destination file should be differ from source files.\n");
            return -1;
        }


        char *firstfile = argv[1];
        char *secondfile = argv[2];
        char *destname = argv[3];
        int fd1,fd2,fd3;

        fd1 = open(firstfile, O_RDONLY);
        fd2 = open(secondfile, O_RDONLY);
        fd3 = open(destname, O_WRONLY | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR);

        if((fd1 == -1)||(fd2 == -1)||(fd3 == -1)){
                perror("open");
                exit(1);
        }

        write_file(fd1,fd3,firstfile);
        write_file(fd2,fd3,secondfile);

        close(fd1);
        close(fd2);
        close(fd3);
        return 0;
}
```

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

void doWrite(int fd_dest,const char *buff)
{
        size_t len,idx;
        ssize_t wcnt;

        idx = 0;
        len = strlen(buff);
        do {
                wcnt = write(fd_dest, buff + idx, len - idx);
                if (wcnt == -1){
                        perror("write");
                        exit(1);
                }
                idx += wcnt;

        } while (idx < len);

}

void write_file(int fd,int fd_dest, const char *infile)
{
        ssize_t rcnt;
        char buff[1024];

        for (;;)
        {
                rcnt = read(fd,buff,sizeof(buff)-1);

                if (rcnt == 0)
                        break;
                if ((rcnt == -1)){
                        perror("read");
                        exit(1);
                }
                buff[rcnt] = '\0';
        }

        doWrite(fd_dest,buff);
}
```

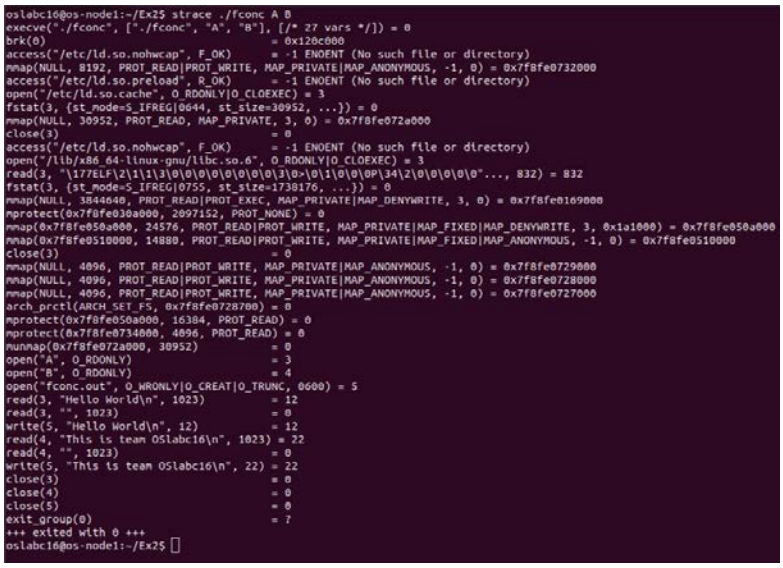
```
fconc: fconc.o
        gcc fconc.o -o fconc

fconc.o: fconc.c
        gcc -Wall -c fconc.c
```

ΚΥΡΙΑΚΟΥ ΔΗΜΗΤΡΗΣ 03117601
ΧΑΤΖΗΧΡΙΣΤΟΦΗ ΧΡΙΣΤΟΣ 03117711

Ερωτήσεις

1. Με τη χρήση της εντολής strace για ένα παράδειγμα του fconc προκύπτουν τα ακόλουθα:



Ακολουθεί η εκτέλεση του προγράμματος με διαφορετικές εισόδους

```
oslabc16@os-node1:~/Ex2$ cat A
goodbye
oslabc16@os-node1:~/Ex2$ cat B
hello
oslabc16@os-node1:~/Ex2$ ls
A  B  fconc  fconc.c  fconc.o  Makefile
oslabc16@os-node1:~/Ex2$ ./fconc A B
oslabc16@os-node1:~/Ex2$ ls
A  B  fconc  fconc.c  fconc.o  fconc.out  Makefile
oslabc16@os-node1:~/Ex2$ cat fconc.out
goodbye
hello
oslabc16@os-node1:~/Ex2$ ./fconc A B C
oslabc16@os-node1:~/Ex2$ ls
A  B  C  fconc  fconc.c  fconc.o  fconc.out  Makefile
oslabc16@os-node1:~/Ex2$ cat C
goodbye
hello
oslabc16@os-node1:~/Ex2$ ./fconc A B C D
Usage ./fconc infile1 infile2 [output (default:fconc.out)
oslabc16@os-node1:~/Ex2$ ./fconc A B A
This action cannot be operated.
Destination file should be differ from source files.
oslabc16@os-node1:~/Ex2$ ./fconc A B B
This action cannot be operated.
Destination file should be differ from source files.
oslabc16@os-node1:~/Ex2$
```