ΚΩΤΣΗΣ ΣΤΑΘΗΣ 03115408
ΡΑΓΚΟΥΣΗΣ ΗΛΙΑΣ 03117897

# Λειτουργικά Συστήματα
# 6ο εξάμηνο, Ακαδημαϊκή περίοδος 2019-2020

# Άσκηση 1:
# Εισαγωγή στο περιβάλλον προγραμματισμού

## 1.1 Σύνδεση με αρχείο αντικειμένων

## ΒΗΜΑΤΑ

1) cp /*home/oslab/code/zing/zing.h  /home/oslab/oslaba25/ask1*

*cp /home/oslab/code/zing/zing.o  /home/oslab/oslaba25/ask1*

*ή εφόσον ο φάκελος περιέχει μόνο αυτά τα 2 αρχεία :*

*cp /home/oslab/code/zing/*  /home/oslab/oslaba25/ask1*

*Όπου ask1 είναι νέος φάκελος που φτιάξαμε στ home μας για την άσκηση 1*

*2) gcc -c main.c*

*3)gcc -o zing main.o zing.o*

### ΕΡΩΤΗΣΕΙΣ

1)Η επικεφαλίδα χρησιμοποιείται για τον ορισμό της συνάρτησης ,την περιγραφή των ορισμάτων και του τύπου επιστροφής της .Με το #include <header.h> δηλώνουμε στο πρόγραμμα ότι ο ορισμός της συνάστησης βρίσκεται στο αρχείο .h και δεν δηλώνεται στο ίδιο το πρόγραμμα.

2)

```
CC=gcc
CFLAGS=-I.
DEPS=       zing.h


%.o: %.c $(DEPS)
        $(CC) -c -o $@ $< $(CFLAGS)


main.o:     main.c
            $(CC) -c main.c
zing:   main.o zing.o
        $(CC) -o zing main.o zing.o
```

Οι αρχικές εντολές ορίζουν τα dependencies .o →.c , .h
και τον κανόνα παραγωγής των αρχείων
Ενώ στην συνέχεια ορίζουμε το main.o και zing ποια αρχεία έχουν το κάθε ένα σα depen Dencies και πως παράγονται.

3)

```
CC=gcc
CFLAGS=-I.
DEPS=zing.h zing2.h

%.o: %.c $(DEPS)
	$(CC) -c -o $@ $< $(FLAGS)

all: zing zing2

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

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

zing2: main.o zing2.o
	gcc -c main2.c -o main.o
	gcc -o zing2 main.o zing2.o
```

Ή εναλλακτικά πιο απλά
και διαγράφοντας τα .o files
μόλις τελειώσουμε :

```
all: zing zing2

zing:main.c main.o zing.o zing.h
	gcc -c main.c -o main.o
	gcc -o zing main.o zing.o
	rm -f *.o
zing2:main2.c main.o zing2.o zing2.h
	gcc -c zing2.c -o zing2.o
	gcc -c main2.c -o main.o
	gcc -o zing2 main.o zing2.o
	rm -f *.o
```

4)Θα ορίσουμε σε ένα Makefile τις συναρτήσεις από τις οποίες εξαρτάται το target πρόγραμμα μας σαν dependencies μαζί με τους κανόνες παραγωγής έτσι ώστε όποτε αλλάζουμε μια συνάρτηση τα εκτελέσιμα που εξαρτώνται από αυτή να παράγονται από το Makefile μετά από καθε αλλαγή.Έτσι αποφεύγουμε το χειροκίνητο compilation και linking .

5)Ο συνεργάτης έκανε overwrite το source code αρχείο foo.c με το εκτελέσιμο foo.c που δημιουργήθηκε κατά το compilation ,εφόσον έδωσε στο εκτελέσιμο το ίδιο όνομα foo.c .

## ΕΞΟΔΟΣ ΑΣΚΗΣΗΣ 1

```
oslaba25@os-node1:~/ask1$ make
cc    -c -o main.o main.c
gcc -c main.c -o main.o
gcc -o zing main.o zing.o
gcc -c zing2.c
gcc -c main2.c -o main.o
gcc -o zing2 main.o zing2.o
```

```
oslaba25@os-node1:~/ask1$ ./zing2
what's up , oslaba25
oslaba25@os-node1:~/ask1$ ./zing
Hello, oslaba25
oslaba25@os-node1:~/ask1$
```

## 1.2 Συνένωση δύο αρχείων σε τρίτο

Παρακάτω δίνεται μια πιθανή υλοποίηση σύμφωνα με την οποία:

Αν το 3ο αρχείο υπάρχει ήδη τότε τα περιεχόμενα των 2 πρώτων αρχείων προστίθενται στο 3ο κατα το write.

Αν το 3ο αρχείο δεν υπάρχει δημιουργείται με RW (read write permission) για τον owner και R (read) για το group και others.(644 ή 110 100 100) και κατά το write τα περιεχόμενα του 1ου και 2ου γράφονται στο 3ο.

Αν εισαχθεί οποιοδήποτε από τα αρχεία εισόδου ως έξοδο ο κώδικας θα το αδειάσει και θα

θα το “γεμίσει” με το περιεχόμενο των αρχείων εισόδου με την σειρά που δόθηκαν.

Για να το επιτυχούμε αυτό χρησιμοποιούμε ένα .tmp αρχείο το οποίο και διαγράφεται μετά την λήξη του προγραμμάτος.

Φυσικά μήνυμα , exit έχουμε όποτε γίνεται κάποιο μη προβλεπόμενο error με το άνοιγμα ,γράψιμο και ανάγνωση κάποιου αρχείου .  ΕΞΟΔΟΣ :

```
the-captain@the-pacifier-2: ~/Dropbox/6_0/oslab/ask2
File  Edit  View  Search  Terminal  Help
the-captain@the-pacifier-2:../ask2$ gcc -Wall -Werror -o fconc main.c
the-captain@the-pacifier-2:../ask2$ cat a
Hello guys
the-captain@the-pacifier-2:../ask2$ cat b
we are back
in town so if you wanna hang out
find someone else.
the-captain@the-pacifier-2:../ask2$ ./fconc a b
the-captain@the-pacifier-2:../ask2$ cat fconc.out
Hello guys
we are back
in town so if you wanna hang out
find someone else.
the-captain@the-pacifier-2:../ask2$ cat c
hey this will be different
the-captain@the-pacifier-2:../ask2$ ./fconc a b c
the-captain@the-pacifier-2:../ask2$ cat c
Hello guys
we are back
in town so if you wanna hang out
find someone else.
the-captain@the-pacifier-2:../ask2$ ./fconc a b a
the-captain@the-pacifier-2:../ask2$ cat a
Hello guys
we are back
in town so if you wanna hang out
find someone else.
```

```
the-captain@the-pacifier-2:../ask2$ ./fconc a c c
the-captain@the-pacifier-2:../ask2$ cat c
Hello guys
we are back
in town so if you wanna hang out
find someone else.
Hello guys
we are back
in town so if you wanna hang out
find someone else.
the-captain@the-pacifier-2:../ask2$ ./fconc a b b
the-captain@the-pacifier-2:../ask2$ cat b
Hello guys
we are back
in town so if you wanna hang out
find someone else.
we are back
in town so if you wanna hang out
find someone else.
the-captain@the-pacifier-2:../ask2$
```

### ΕΡΩΤΗΣΕΙΣ

1. Για να απομονώσουμε τις κλήσεις που μας ενδιαφέρουν δίνουμε το -e trace={ system call set } και -o output_file για να στείλουμε την έξοδο σε εξωτερικό αρχείο.

```
st@st-Lenovo-ideapad-330-15IKB:~/Desktop$ strace -e trace=open,read,write -o out.txt ./fconc A B D
st@st-Lenovo-ideapad-330-15IKB:~/Desktop$ cat out.txt
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\260\34\2\0\0\0\0"..., 832) = 832
read(3, "hello g", 7)                   = 7
write(5, "hello g", 7)                  = 7
read(3, "uys\n", 7)                     = 4
write(5, "uys\n", 4)                    = 4
read(3, "", 7)                          = 0
read(4, "we are ", 7)                   = 7
write(5, "we are ", 7)                  = 7
read(4, "back\nin", 7)                  = 7
write(5, "back\nin", 7)                 = 7
read(4, " town s", 7)                   = 7
write(5, " town s", 7)                  = 7
read(4, "o if yo", 7)                   = 7
write(5, "o if yo", 7)                  = 7
read(4, "u wanna", 7)                   = 7
write(5, "u wanna", 7)                  = 7
read(4, " hang o", 7)                   = 7
write(5, " hang o", 7)                  = 7
read(4, "ut\nfind", 7)                  = 7
write(5, "ut\nfind", 7)                 = 7
read(4, " someon", 7)                   = 7
write(5, " someon", 7)                  = 7
read(4, "e else.", 7)                   = 7
write(5, "e else.", 7)                  = 7
read(4, "", 7)                          = 0
+++ exited with 0 +++
```

# Bonus :

Στην πιο απλή περίπτωση μπορούμε να κάνουμε το εξής :

strace strace cat A

οπότε η strace να μας επιστρέψει τις κλήσεις που καλεί η strace καθώς p.x κάνει cat ένα αρχείο στην οθόνη.Στο αντίστοιχο output παρατηρούμε την κλήση ptrace που από το man page της μαθαίνουμε ότι χρησιμοποιείται για monitoring άλλων διεργασιών (tracing ) .

```
execve("/usr/bin/strace", ["strace", "cat", "A"], 0x7ffeb59c2710 /* 62 vars */) = 0
brk(NULL)                               = 0x562b74e28000
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=120369, ...}) = 0
mmap(NULL, 120369, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7fc0ea133000
close(3)                                = 0
```

```
ptrace(PTRACE_SYSCALL, 6947, NULL, SIG_0) = 0
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_TRAPPED, si_pid=6947, si_uid=1000, si_status=
wait4(-1, [{WIFSTOPPED(s) && WSTOPSIG(s) == SIGTRAP | 0x80}], __WALL, NULL) = 6947
rt_sigprocmask(SIG_SETMASK, [HUP INT QUIT PIPE TERM], NULL, 8) = 0
ptrace(PTRACE_GETREGSET, 6947, NT_PRSTATUS, [{iov_base=0x562b7404e660, iov_len=216}]) =
process_vm_readv(6947, [{iov_base="/etc/ld.so.nohwcap\0cannot create"..., iov_len=3966}]
write(2, "access(\"/etc/ld.so.nohwcap\", F_O"..., 33access("/etc/ld.so.nohwcap", F_OK) =
ptrace(PTRACE_SYSCALL, 6947, NULL, SIG_0) = 0
--- SIGCHLD {si_signo=SIGCHLD, si_code=CLD_TRAPPED, si_pid=6947, si_uid=1000, si_status=
rt_sigprocmask(SIG_SETMASK, [], NULL, 8) = 0
wait4(-1, [{WIFSTOPPED(s) && WSTOPSIG(s) == SIGTRAP | 0x80}], __WALL, NULL) = 6947
rt_sigprocmask(SIG_SETMASK, [HUP INT QUIT PIPE TERM], NULL, 8) = 0
ptrace(PTRACE_GETREGSET, 6947, NT_PRSTATUS, [{iov_base=0x562b7404e660, iov_len=216}]) =
write(2, ")      = -1 ENOENT (No such file"..., 47)      = -1 ENOENT (No such file or di
) = 47
```

2) Η αλλαγή οφείλεται στο γεγονός ότι καλείται διαφορετική συνάρτηση κάθε φορά και έτσι θα γίνει και διαφορετικό jump.

4)gdb whoops

break main (add breakpoint for function main)

break whoops (add pending breakline if it finds this function later)

Run

Bt (print stacktrace)

C ( continue)

Όταν συνεχίσει από το breakline της main θα τερματίσει λόγω error στο execution

και μας δείχνει το process id που έχει το πρόβλημα.

# ΚΩΔΙΚΕΣ

## ΑΣΚ1.1

(οι παραπάνω + συνάρτηση
zing2 (char * name){printf(“what’s up, %s “ ,name);}
)

## ΑΣΚ1.2

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

/* second function to implement write */
void write_file(int fd, char *buff, int len) {
        int idx = 0;
        int wcnt;

        while (idx < len) {
                wcnt = write(fd, buff + idx, len - idx);
                if (wcnt == -1) {
                        perror("error in writing");
                        exit(1);
                }
                idx += wcnt;
        }
```

```
}

/* first function implements read and calls write */
void doWrite(int fd1, int fd2, char *buff) {
	int rcnt;
	int total = 0, len = 0;

	for (;;) {

		rcnt = read(fd1, buff, sizeof(buff) - 1);
		total = rcnt + total;
		if (rcnt == 0)
			break;
		if (rcnt == -1) {
			perror("problem in read");
			exit(1);
		}

		buff[rcnt] = '\0';
		len = strlen(buff);
		write_file(fd2, buff, len);
	}
}

int main(int argc, char *argv[]) {
	int check = 0, target;
	char buff[1024];
	char buff2[1024];
	int f1, f2, f3;

	// case1:not enough /more files tha appropriate
	if (argc < 3 || argc > 4) {
		printf("Error:Please pass 2 files for input and no more than 1 file for "
				"output\n");
		exit(1);
	}

 // case 2 :file inputs not exist
	f1 = open(argv[1], O_RDWR, 0777);
	if (f1 == -1) {
		perror("Error:opening file 1 ,maybe it does not exist\n");
		exit(1);
	}
	f2 = open(argv[2], O_RDWR, 0777);
	if (f1 == -1) {
		perror("Error:opening file 2 ,maybe it does not exist\n");
		exit(1);
	}
```

```
	// case 1 b : file 3 is the same as file 1 or file 2


	// creating file 3
	if (argc == 4) {
		if ((strcmp(argv[3], argv[1]) == 0
					|| strcmp(argv[3], argv[2]) == 0)) {
		//printf("output file can not be the same as input file");
		//exit(1);
		f3 = open("tmp", O_RDWR|O_TRUNC|O_CREAT, 0777);
		if(f3 == -1){
			perror("gt re malaka:");
			return 1;
		}
		check = 1;
		if(!(strcmp(argv[3], argv[1]))) target = f1;
		else target = f2;
		}
		else{

			f3 = open(argv[3], O_CREAT | O_RDWR | O_APPEND | O_TRUNC,
0644);
			if (f3 == -1) {
				perror("error opening output file");
				exit(1);
			}
		}
	}
	else if (argc < 4) {
		f3 = open("fconc.out",  O_TRUNC|O_CREAT | O_RDWR | O_APPEND,
0644);
		if (f3 == -1) {
			perror("error opening output file");
			exit(1);
		}
	}

	// read and write file 1 ,file 2 to file 3 and close files
	doWrite(f1, f3, buff);
	doWrite(f2, f3, buff2);
	if(check == 1){
		// lseek einai syscall poy soy dinei thn dynatothta na alla3eis ton pointer toy
arxeioy
		if(lseek(f3, 0, SEEK_SET) < 0) {
			perror("it failed to move the pointer of the tmp file");
			return 1;
		}
		if(lseek(target, 0, SEEK_SET) < 0) {
```

```
                perror("it failed to move the pointer of the target file");
                return 1;
            }

            doWrite(f3, target, buff);
        }
        close(f1);
        close(f2);
        close(f3);
        if(check){
            if(unlink("tmp") == -1){
                perror("Failed to delete tmp file used\
                    for writing in one of the input files\n");
                return 1;
            }
        }
        return 0;
}
```