Νικόλαος Γιαννόπουλος – 03122086
Κωνσταντίνος Καργάκος – 03122166

Λειτουργικά Συστήματα Υπολογιστών
Προπαρασκευαστική Εργαστηριακή Άσκηση

1.1 – Εισαγωγή στο shell

8. Η εντολή επιστρέφει το μήνυμα:
`[rmdir: failed to remove 'test_dir': Directory not empty]`
γιατί ο κατάλογος test_dir περιέχει το αρχείο test_file1.

11. Χρησιμοποιώ την εντολή **`[grep -Rn "oslab064" .]`** για να εντοπίσω όλες τις εμφανίσεις του **“`oslab064`”** στον τρέχοντα κατάλογο **`[.]`** και στους υπο-καταλόγους **`[-R]`**, τυπώνοντας τον αριθμό γραμμής κάθε εμφάνισης στο κάθε αρχείο **`[-n]`**.

12. Χρησιμοποιώ την εντολή **`[:%s/oslab064/starship/g]`** για να αντικαταστήσω όλες τις εμφανίσεις του **“`oslab064`”** με τη λέξη **“`starship`”**.

1.2 - Σύνδεση με αρχείο αντικειμένων

1. Η επικεφαλίδα περιέχει το αρχείο zing.h, το οποίο δηλώνει τη συνάρτηση zing(), ώστε να μπορεί να χρησιμοποιηθεί μέσα στο main.c. Γενικά, η επικεφαλίδα μπορεί να δηλώνει συναρτήσεις και τα ορίσματά τους, τύπους δεδομένων, σταθερές κ.α., διευκολύνοντας την επαναχρησιμοποίηση τους από άλλα αρχεία.

2. Δημιουργία αρχείου main.c με χρήση της συνάρτησης zing() και αρχείου Makefile για τη δημιουργία του εκτελέσιμου zing:

```c
#include "zing.h"

int main()
{
        zing();
        return 0;
}
```

```
# Makefile
CC = gcc
CFLAGS = -Wall

zing: main.o zing.o
    $(CC) $(CFLAGS) -o zing main.o zing.o

main.o: main.c zing.h
    $(CC) $(CFLAGS) -c main.c

clean:
    rm -f main.o zing2.o zing zing2
```

3. Το zing2.c υλοποιεί τη συνάρτηση zing() με τη χρήση της getlogin(). Η getlogin() επιστρέφει το όνομα του χρήση που είναι συνδεδεμένος στον server. Στη συνέχεια τυπώνει ένα μήνυμα:

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

void zing(void) {
        const char *user = getlogin();
        printf("Hello %s, how can I help you?\n", user);
}
```

Νέο αρχείο Makefile για τη δημιουργία του εκτελέσιμου zing και zing2:

```
# Makefile
CC = gcc
CFLAGS = -Wall

all: zing zing2

zing: main.o zing.o
    $(CC) $(CFLAGS) -o zing main.o zing.o

zing2: main.o zing2.o
    $(CC) $(CFLAGS) -o zing2 main.o zing2.o

main.o: main.c zing.h
    $(CC) $(CFLAGS) -c main.c

zing2.o: zing2.c zing.h
    $(CC) $(CFLAGS) -c zing2.c

clean:
    rm -f main.o zing2.o zing zing2
```

4. Το πρόβλημα του μεγάλου χρόνου μεταγλώττισης μπορεί να αντιμετωπισθεί αν διασπάσουμε τον κώδικα σε πολλαπλά αρχεία (.c) υλοποιήσεων των συναρτήσεων και παράγουμε χωριστά αρχεία (.o). Έτσι, αν κάνουμε αλλαγές μόνο σε μια συνάρτηση, τότε θα μεταγλωττίσουμε μόνο το αρχείο (.c) με την αλλαγή, ενώ τα υπόλοιπα (.o) δεν χρειάζεται να παραχθούν ξανά. Η μεταγλώττιση γίνεται με το Makefile μόνο στο επεξεργασμένο αρχείο (.c). Με αυτόν τον τρόπο ο χρόνος μεταγλώττισης μειώνεται σημαντικά.

5. Με την εντολή **`[gcc -Wall -o foo.c foo.c]`** ο compiler παίρνει το αρχείο foo.c και παράγει εκτελέσιμο με όνομα επίσης foo.c. Επειδή το όνομα του εκτελέσιμου συμπίπτει με το όνομα του αρχείου του κώδικα, το αρχείο foo.c θα αντικατασταθεί από το εκτελέσιμο αρχείο και ο κώδικας θα χαθεί. Γι' αυτό είναι απαραίτητο να κρατάμε backup των αρχείων, καθώς και να χρησιμοποιούμε Makefile, ώστε να αποφεύγονται τα typos.