



ΠΡΟΠΑΡΑΣΚΕΥΑΣΤΙΚΗ ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΝΑΦΟΡΑ

Μέλη Ομάδας:

- ΙΩΑΝΝΗΣ ΠΟΛΥΧΡΟΝΟΠΟΥΛΟΣ Α.Μ: 03121089
- ΣΟΦΙΑ ΣΑΒΒΑ Α.Μ: 03121189

Εισαγωγή στο shell

1. Αρχικά, συνδεόμαστε στο μηχάνημα orion με την παρακάτω εντολή και παραθέτουμε τον κατάλληλο κωδικό:
`ssh oslab019@orion.cslab.ece.ntua.gr`
2. Τυπώνουμε τον κατάλογο στον οποίο βρισκόμαστε και τυπώνουμε το περιεχόμενό του, συμπεριλαμβανομένων των κρυφών αρχείων με τις εντολές `pwd` (print working directory) και `ls -a` (list all files, including the hidden ones) αντίστοιχα.
3. Δημιουργούμε τα επιθυμητά αρχεία και κατάλογο με την ακόλουθη εντολή:
`touch test_file1 test_file2 test_file3 && mkdir test_dir`. Η εντολή `touch` δημιουργεί τα αρχεία `test_file1`, `test_file2`, `test_file3` και η εντολή `mkdir` δημιουργεί τον κατάλογο `test_dir`. Πραγματοποιούμε τις δύο αυτές εντολές σε μια γραμμή του terminal με τον λογικό τελεστή `&&` (AND).
4. Μπορούμε να μετονομάσουμε το αρχείο `test_file3` με την εντολή:
`mv test_file3 my_testfile3`
Move test_file3 to the same directory but with the name
my_testfile3
5. Μετακινούμε τα αρχεία `test_file1` και `test_file2` στον επιθυμητό κατάλογο με την εντολή `mv test_file1 test_file2 test_dir`
6. Μετακινούμαστε στο κατάλογο `test_dir` με την εντολή `cd test_dir` (change directory) και τυπώνουμε τα περιεχόμενα του σε μορφή λίστας (`-l` option) σε αύξουσα χρονολογική σειρά (`-t` option) με την εντολή `ls -lt`
7. Διαγράφουμε το αρχείο `test_file2` με την εντολή `rm test_file2` (remove the file test_file2) και στη συνέχεια επιστρέφουμε στο γονικό κατάλογο με την εντολή `cd ..` (`..` is a special notation that refers to the parent directory of the current/working directory)

8. Εκτελούμε την εντολή `rmdir test_dir` και λαμβάνουμε το ακόλουθο error message
`rmdir: failed to remove 'test_dir': Directory not empty`
Συνεπώς, δεν μπορούμε να διαγράψουμε τον κατάλογο όταν περιέχει είτε αρχεία είτε καταλόγους με την εντολή `rmdir`. Για την λύση αυτού του προβλήματος μπορούμε είτε να αδειάσουμε τον κατάλογο `test_dir` είτε να χρησιμοποιήσουμε την εντολή:
`rm -r test_dir (-r: recursive)`
9. Έχοντας επιστρέψει στον προσωπικό μας κατάλογο (δηλαδή το current directory είναι το ~) αντιγράφουμε το αρχείο `file_generator.sh` με την εντολή:
`cp /home/oslab/code/shell-intro/file_generator.sh .`
10. Εκτελούμε το εκτελέσιμο bash script αρχείο τρέχοντας `./file_generator.sh`. Το script δημιούργησε ένα κατάλογο με όνομα `test` που περιέχει 10 text files. (output-1-10.txt)
11. Θα χρησιμοποιήσουμε την εντολή `grep -r -n -H "$USER" .`
Επεξήγηση:
-r : Αναδρομική αναζήτηση υποκαταλόγων.
-H : Εκτύπωσε το όνομα αρχείου για κάθε αντιστοίχιση.
-n : Εκτύπωσε την αντίστοιχη γραμμή που έγινε η αντιστοίχιση για κάθε αρχείο.
"\$USER": Χρησιμοποιώ τη μεταβλητή περιβάλλοντος `$USER` όπου είναι αποθηκευμένο το username μου (oslab019).
.: Ψάχνουμε αναδρομικά από το current directory.
12. Ανοίγουμε το αρχείο `output-8.txt` με τον vim editor: `vim output-8.txt`, στη συνέχεια μεταφερόμαστε στο command mode πατώντας άνω κάτω τελεία (:) και εκτελούμε την ακόλουθη εντολή:
`%s/oslab019/new_oslab019/g`
 - **%s**: Καθορίζει ότι η αντικατάσταση πρέπει να εκτελεστεί σε ολόκληρο το αρχείο.
 - **/oslab019/**: Καθορίζει την ακολουθία χαρακτήρων που θέλετε να αντικαταστήσετε.
 - **/new_oslab019/**: Καθορίζει το κείμενο αντικατάστασης που επέλεξα.
 - **g**: Stands for "global" και λέει στο Vim να αντικαταστήσει όλες τις εμφανίσεις `oslab019` σε ολόκληρο το αρχείο. Εάν παραλείψουμε το `/g`, το Vim θα αντικαταστήσει μόνο την πρώτη εμφάνιση σε κάθε γραμμή.

Σύνδεση με αρχείο αντικειμένων

Βήματα:

1. Αντιγράφουμε τα αρχεία `zing.h` και `zing.o` στον προσωπικό μας κατάλογο:
 - `cd /home/oslab/code/zing`
 - `cp zing.h zing.o ~`

2. Έπειτα φτιάχνουμε, με τη χρήση του vim, το αρχείο main.c

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

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

3. Μεταγλωττίζουμε το αρχείο σε αρχείο αντικειμένων main.o:

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

Τέλος συνδέουμε τα 2 αρχεία αντικειμένων με την εντολή:

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

Ερωτήσεις

1^η Ερώτηση

Η επικεφαλίδα σε ένα πρόγραμμα στη γλώσσα C είναι ένα αρχείο που περιέχει δηλώσεις συναρτήσεων, προτύπων και άλλων δεδομένων που μπορούν να χρησιμοποιηθούν στο κύριο πρόγραμμα (*main()*). Στη συγκεκριμένη περίπτωση, το header αρχείο *zing.h* περιέχει την συνάρτηση *zing()*, την οποία καλούμε στην συνάρτηση *main()*. Με την επικεφαλίδα, επιτρέπουμε τον διαχωρισμό της συνάρτησης *zing()*, δίνοντας τη δυνατότητα στον προγραμματιστή να την επεξεργαστεί ανεξάρτητα από τα προγράμματα που την χρησιμοποιούν. Συνεπώς, η επικεφαλίδα καθιστά τον πηγαίο κώδικα πιο σύντομο και κατανοητό.

2^η Ερώτηση

Αρχικά, δημιουργούμε το εκτελέσιμο αρχείο δεδομένου πως έχουμε το *zing.o* και έχουμε δημιουργήσει το *main.o*. Αν δεν έχουμε δημιουργήσει το *main.o*, τότε πηγαίνουμε στο παρακάτω target, όπου δημιουργούμε το object file *main.o*.

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

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

3^η Ερώτηση

Αρχικά, φτιάχνουμε ένα αρχείο `zing2.c` και το μετατρέπουμε σε object file για να παράξουμε το `zing2.o`.

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

void zing(){
    char *name = getlogin(3); # Retrieves the username
    printf("Hello %s\n",name);
}
```

Στη συνέχεια, προσθέτουμε τους εξής κανόνες στο Makefile, ώστε να παράγονται 2 εκτελέσιμα, ένα με το `zing.o` (`zing`) και ένα με το `zing2.o` (`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
```

4^η Ερώτηση

Σπαταλάμε χρόνο με το να μεταγλωττίζουμε, ξανά και ξανά όλες τις συναρτήσεις ενώ αλλάζουμε μόνο μία. Επομένως, μπορούμε να ορίσουμε τη συνάρτηση σε ένα ξεχωριστό header file το οποίο μετά θα κάνουμε `include` στον κώδικά μας έτσι ώστε να μπορούμε να επεξεργαζόμαστε και να μεταγλωττίζουμε τη συνάρτηση ξεχωριστά από τον υπόλοιπο κώδικα μειώνοντας τον χρόνο μεταγλώττισης.

5^η Ερώτηση

Ο συνεργάτης μας κατά την κλήση της εντολής `gcc` προσθέτοντας το `-o foo.c` ονόμασε το εκτελέσιμο αρχείο `foo.c`, το ίδιο όνομα με το αρχείο που περιέχει το source code επομένως το αρχείο αυτό αντικαταστάθηκε από το εκτελέσιμο.