**ΕΘΝΙΚΟ ΜΕΤΣΟΒΕΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΑΝΑΦΟΡΑ ΠΡΟΠΑΡΑΣΚΕΥΑΣΤΙΚΗΣ ΑΣΚΗΣΗΣ**

A black background with a black square

Description automatically generated with medium confidence

**Στοιχεία Ομάδας**

* Αναγνωριστικό: oslab005
* Μέλος 1ο: Πέππας Μιχαήλ – Αθανάσιος, Α.Μ: 03121026
* Μέλος 2ο: Σαουνάτσος Ανδρέας, Α.Μ: 03121197
* Ημερομηνία Παράδοσης Αναφοράς: 03.04.2024
* **Ενότητα 1.1 – Εισαγωγή στο Shell**

Αρχικά, συνδεθήκαμε στο μηχάνημα orion και στη συνέχεια σε ένα εκ των osnode{1,2} με τη χρήση της εντολής student‐pc. Στην παρακάτω εικόνα, επελέγη ο osnode1:  
A computer screen with text on it

Description automatically generated

Ακολούθως, τυπώσαμε τον κατάλογο στον οποίο βρισκόμασταν, καθώς και το περιεχόμενο αυτού – συμπεριλαμβανομένων των κρυφών αρχείων – με χρήση της εντολής ls και δη, της ls – a, η οποία σημαίνει (all) και τυπώνει όλα τα περιεχόμενα, ανεξαρτήτων με τον εάν αρχίζουν με τελεία (.), όπως φαίνεται ακολούθως:  
A black screen with green text

Description automatically generated  
Κατόπιν, δημιουργήσαμε τρία νέα αρχεία με ονόματα test\_file1, test\_file2 και test\_file3, κάνοντας χρήση της εντολής touch 3 φορές διαδοχικά, καθώς και έναν νέο κατάλογο (directory) με όνομα test\_dir, κάνοντας χρήση της εντολής mkdir, όπως φαίνεται στην παρακάτω εικόνα:  
A screen shot of a computer code

Description automatically generated  
  
Ύστερα, μετονομάσαμε το αρχείο test\_file3 και του δώσαμε το νέο όνομα test\_file4, κάνοντας χρήση της εντολής mv και μετά εκτελέσαμε ls, ώστε να τυπώσουμε τα νέα περιεχόμενα του καταλόγου που βρισκόμαστε, όπως βλέπουμε παρακάτω:  
A black screen with white text

Description automatically generated  
Μετά, μεταφέραμε τα αρχεία test\_file1 και test\_file2 στον κατάλογο test\_dir, χρησιμοποιώντας την εντολή mv, διαδοχικά 2 φορές και εκτελέσαμε ls – a, ώστε να επιβεβαιώσουμε τη μετακίνηση:  
Ακολούθως μεταβήκαμε στον κατάλογο test\_dir (με την εντολή cd) και τυπώσαμε τα περιεχόμενά του σε μορφή λίστας και σε αύξουσα χρονολογική σειρά, με την εντολή ls -lrt, όπου το -l προσδίδει long listing format στην εκτύπωση των αρχείων, το -r κάνει reverse τη χρονική σειρά (που είναι και το ζητούμενο), αφού το -t τα τυπώνει από το νεότερο στο παλαιότερο. Το αποτέλεσμα φαίνεται παρακάτω, μία φορά με -rt και μία με -lrt:  
A screenshot of a computer program

Description automatically generated  
Έπειτα, διαγράψαμε το αρχείο test\_file2, με χρήση της εντολής rm, το επιβεβαιώσαμε με ls, και μεταβήκαμε στον γονικό κατάλογο, με την εντολή cd:  
A screenshot of a computer program

Description automatically generated

Κατόπιν, προσπαθήσαμε να διαγράψουμε τον γονικό κατάλογο test\_dir με την εντολή rmdir, αλλά τυπώθηκε το παρακάτω μήνυμα:  
A black and white screen with white text

Description automatically generated  
το οποίο μας επεσήμανε ότι ο κατάλογος δεν είναι άδειος και συνεπώς δεν μπορούμε να τον διαγράψουμε. Έτσι, προσθέσαμε την κατάληξη -r στην εντολή rm, η οποία αφαιρεί τόσο το directory όσο και τα περιεχόμενά του, recursively, επιτυγχάνοντας το επιθυμητό αποτέλεσμα:  
A screen shot of a computer

Description automatically generated  
Μετά, αντιγράψαμε το αρχείο file\_generator.sh από τον κατάλογο /home/oslab/code/shell‐intro στον προσωπικό μας κατάλογο, με χρήση της εντολής cp και εκτελέσαμε ls για την επιβεβαίωση της αντιγραφής και εκτελέσαμε ./file\_generator.sh:  
A screen shot of a computer

Description automatically generated  
Μετά, εντοπίσαμε όλες τις εμφανίσεις του username μας (oslab005) στον τρέχοντα κατάλογο καθώς και σε όλους τους υποκαταλόγους, τυπώνοντας τον αριθμό γραμμής κάθε εμφάνισης της συμβολοσειράς σε κάθε αρχείο, χωρίς να ανοίξουμε και να διαβάσουμε τα περιεχόμενα των αρχείων. Έτσι, χρησιμοποιήσαμε την εντολή grep με την κατάληξη -rn, η οποία εκτυπώνει αναδρομικά όλα τα στοιχεία (-r) και με τον αριθμό γραμμής εμφάνισης (-n), όπως φαίνεται παρακάτω:  
  
A black screen with white text

Description automatically generated  
Τέλος, ανοίξαμε ένα από τα αρχεία όπου εμφανίζεται το username μας (oslab005) πολλαπλές φορές με τη χρήση του επεξεργαστή κειμένου vim και αντικαταστήσαμε όλες τις εμφανίσεις του oslab005 με τη συμβολοσειρά new\_word, χωρίς να επεξεργαστούμε χειροκίνητα κάθε εμφάνιση ξεχωριστά. Για να το επιτύχουμε αυτό, χρησιμοποιήσαμε την εντολή: **%s/oslab005/new\_word/g** και το αποτέλεσμα φαίνεται παρακάτω:  
A black screen with a black background

Description automatically generated

* **Ενότητα 1.2 – Σύνδεση με αρχείο αντικειμένων**

Στην ενότητα αυτή σκοπός μας είναι η δημιουργία ενός εκτελέσιμου αρχείου – με όνομα zing – που θα καλεί τη συνάρτηση zing() που δηλώνεται στο αρχείο zing.h, ενώ έχουμε διαθέσιμο το object file zing.o. Έτσι, ακολουθήσαμε αναλυτικά τα βήματα που υποδείχθηκαν ως εξής:  
  
Βήμα 1:Αντιγράψαμε τα αρχεία zing.h και zing.o από το directory /home/oslab/code/zing στον προσωπικό μας κατάλογο, με χρήση της εντολής cp και έπειτα τα εμφανίσαμε με ls, δύο φορές διαδοχικά, όπως είναι εμφανές παρακάτω:  
A screenshot of a computer code

Description automatically generated  
  
Βήματα 2 και 3: Δημιουργήσαμε το αρχείο αντικειμένων main.o για τη συνάρτηση main που καλείται, εκτελώντας gcc -Wall -c main.c, και κατόπιν συνδέσαμε τα δύο αρχεία αντικειμένων, εκτελώντας gcc main.o zing.o -o zing, και τρέξαμε το πρόγραμμα με ./zing, όπως φαίνεται ακολούθως:  
A screenshot of a computer program

Description automatically generated  
Το αποτέλεσμα ήταν η εκτύπωση της φράσης που βλέπουμε: “Hello, oslab005”.

* **Ερωτήσεις**

1. Ο ρόλος της επικεφαλίδας είναι διττός. Είμαστε σε θέση να γνωρίζουμε αφενός μεν τι παραμέτρους δέχονται και αφετέρου δε τι τιμές επιστρέφουν συναρτήσεις των οποίων η υλοποίηση είναι άγνωστη προς εμάς. Έτσι, αν και δεν μπορούμε να έχουμε πρόσβαση στο σώμα τους, είμαστε σε θέση να τις χρησιμοποιήσουμε τοποθετώντας τα κατάλληλα ορίσματα και αναμένοντας μία προβλέψιμη έξοδο, σε περίπτωση που μας φανούν χρήσιμες.
2. **A computer screen with white text

   Description automatically generated**A screen shot of a computer

   Description automatically generatedΣυντάξαμε το ακόλουθο Makefile, το οποίο χρησιμεύει για την υλοποίηση της άσκησης και διαφαίνεται στην παρακάτω εικόνα:  
   Έτσι, ύστερα από την σύνταξη του ανωτέρω αρχείου, εκτελέσαμε make και κατόπιν ls, ώστε να ελέγχουμε την ύπαρξη όλων των συστατικών στοιχείων του προγράμματος προκειμένου αυτό να τρέξει (και το Makefile). Τέλος, εκτελέσαμε ./zing και, όπως αναμενόταν, εκτυπώθηκε το ίδιο μήνυμα με πριν: “Hello, oslab005”. Η διαδικασία αυτή φαίνεται κάτωθι:
3. A black background with red text

   Description automatically generatedΜε τη χρήση της συνάρτηση getlogin() γράψαμε κατάλληλο κώδικα μέσα στο Makefile, ο οποίος λαμβάνει το όνομα του χρήστη και εμφανίζει το μήνυμα “Hello user oslab005”. Έτσι, ακολουθώντας την ίδια διαδικασία με πριν, ανανεώσαμε το Makefile ώστε με ένα main αρχείο να δημιουργεί δύο διαφορετικά executable files, όπως φαίνεται παρακάτω:

Τελικά, κάνοντας make και τρέχοντας το zing2 παίρνουμε το επιθυμητό αποτέλεσμα, όπως φαίνεται ακολούθως:

A computer screen with white text

Description automatically generated

1. Στην περίπτωση που έχουμε πολλές συναρτήσεις στον κώδικά μας, προκειμένου να αποφευχθεί η επανειλημμένη μεταγλώττιση των συναρτήσεων που δεν αλλάζουμε – μια διαδικασία ιδιαίτερα χρονοβόρα – είναι προτιμότερο να δημιουργούμε ξεχωριστά αρχεία, για ξεχωριστές συναρτήσεις (ή και ομάδες συναρτήσεων) και να τις κάνουμε link και compile μέσω ενός Makefile. Έτσι, θα μεταγλωττίζουμε εκ νέου μόνο τη συνάρτηση στην οποία δουλεύουμε, κι όχι όλες, εξοικονομώντας χρόνο.
2. Η διαδικασία “gcc -Wall -o “ex\_name” “code.o” δημιουργεί εκτελέσιμο αρχείο την code.o στο αρχείο ex\_name (αν δεν υπάρχει το δημιουργεί πρώτα, και αν είναι γραμμένο διαγράφει το περιεχόμενο του). Στη συγκεκριμένη περίπτωση, το όνομα των δύο αρχείων είναι το ίδιο, με αποτέλεσμα, κατά την έναρξη της διαδικασίας, ο compiler να διαγράφει το περιεχόμενο του αρχείου προς εγγραφή και στη συνέχεια να διαβάζει το αρχείο προς ανάγνωση (στο οποίο όμως έχει διαγραφεί το περιεχόμενο) και τελικά να τερματίζει.

**Σ.Η.Μ.Μ.Υ. Ε.Μ.Π.  
Απρίλιος 2024**