

Contents

1	Επικοινωνία Ανθρώπου-Υπολογιστή (hci):	3
2	Παραδοτέο 1ο / Εισαγωγή:	7
3	Παραδοτέο 2ο : Εγκατάσταση των Arch Linux σε USB	8
4	Παραδοτέο 3ο: Συμμετοχικό περιεχόμενο A1.	11
4.1	Περιγραφή: Python	11
4.2	Περιγραφή: ODR0ID-C4	12
5	Παρακάτω, παραθέτω τα link από το κάθε ένα αρχείο μαζί με τις φωτογραφίες που το συνοδεύουν:	13
6	Παραδοτέο 4ο: Warming Up with CLI Commands.	14
6.0.1	Περιγραφή: Listing των format από το τραγούδι, Downloading το τραγούδι σε .mp4 format & Listing το αρχείο στο directory Downloads μέσω της εφαρμογής Youtube-DL	14
7	Παραδοτέο 5ο: Συμμετοχικό Περιεχόμενο A2.	16
7.1	Περιγραφή: Timelines	17
7.2	Υπολογιστές Τσέπης Timeline	17
7.3	Υψηλού Επιπέδου Γλώσσες Προγραμματισμού Timeline .	18
7.4	Περιγραφή: Slides	18
7.5	Γλώσσες Προγραμματισμού Slide	19
7.6	Υπολογιστές Τσέπης Slide	19
8	Παρακάτω, παραθέτω τα link από το κάθε ένα αρχείο μαζί με τις φωτογραφίες που το συνοδεύουν:	19
9	Παραδοτέο 6ο: Κατασκευάζοντας το βιβλίο.	21
9.1	Βήμα #1: Forking and Cloning the repository . . .	21
9.2	Βήμα #2: Downloading the Suitable Software via Terminal	22
9.3	Βήμα #3: Cloning the Submodules from the Main Repository and Setting them. Part #1	23
9.4	Βήμα #4: Cloning the Submodules from the Main Repository and Setting them. Part #2	24
9.5	Βήμα #5: Creating the Contribution Folder and the Contribution (.md) File	24
9.6	Βήμα #6: Creating the Contribution Filter (contribution.lua) & Editing the Make File (make-latex.sh)	24

9.6.11	26
9.6.22	26
9.7 Βήμα #7: Creating and Merging the (.tex) Files into a Single (.pdf) File	26
9.8 Βήμα #8: Inserting the Contribution Filter Inside the Book	28
9.9 Βήμα #9: Recompiling the Book With the Same Procedure	29
9.10 Βήμα #10: Displaying My Personal Contribution From Inside the PDF Book	29
10 Παραδοτέο 7ο: Συμμετοχικό περιεχόμενο B1.	30
10.1 Guido Van Rossum Biography	33
10.2 Python & Odroid-C4 Case-Study	33
11 Παρακάτω, παραθέτω τα link από το κάθε ένα αρχείο μαζί με τις φωτογραφίες που το συνοδεύουν:	33
11.1 Αρχείο μελέτης περίπτωσης από το αποθετήριο: .md File #Case-Study	33
12 Φωτογραφίες που αξιοποιήθηκαν στην παραπάνω εργασία:	33
12.1 Φωτογραφία για την μελέτη περίπτωσης της Python #4: Python #4	34
13 Πίνακας πνευματικών δικαιωμάτων σχετικά με την αξιοποίηση των πηγών και των φωτογραφιών:	34
14 Παραδοτέο 8ο: Command Line Exercise #2	35
14.1 Βήμα #1: The Creation of the Repository	35
14.2 Βήμα #2: Cloning the Repository Locally	36
14.3 Βήμα #3: Filling out the Script File With Commands	36
14.4 Converting a markdown (.md) file into PDF (.pdf)	37
14.5 Converting a markdown (.md) file into HTML (.html)	38
14.6 Βήμα #4: Checking the Results of the Conversion (PDF) & (HTML)	38
14.7 PDF Results	38
14.8 HTML Results	38
14.9 Βήμα #5: Filling out the README.md File with Useful Information	38
15 Πίνακας των διάφορων link περί εργασίας:	40

1 Επικοινωνία Ανθρώπου-Υπολογιστή (hci):

Ονοματεπώνυμο: Κοκοτός Σπυρίδων Ευτύχιος

AM: 2021098

E-mail: inf2021098@ionio.gr

Github Profile

Ομαδικός Οργανισμός: Terminal-Killers

Προφίλ στο Asciiinema: inf2021098

Αποθετήριο ιστοσελίδας: Pibook Site Repository

Netlify Site των Terminal Killers: Organization Site

Netlify Personal Site Deploy: Site

Αποθετήριο ιστοσελίδας οργανισμού: Pibook Organization Site Repository

Εβδομάδα	Όλα τα παραδοτέα βρίσκονται στην ίδια σελίδα της τελικής αναφοράς με τα προσωπικά στοιχεία σας (Όνομα, AM, github profile) και μαζί με αυτόν εδώ τον πίνακα περιεχομένων	Σύνδεσμος στην εβδομαδιαία παρουσίαση προόδου στις συζητήσεις	Αυτοαξιολόγηση σύμφωνα με τα κριτήρια της αντίστοιχης άσκησης
1	Δημιουργία ομάδας + Φορκ και δημιουργία σελίδας τελικής αναφοράς, προσθήκη πίνακα περιεχομένων, συγγραφή της εισαγωγής, αποστολή της εισαγωγής για σχολιασμό στην συζήτηση και καταγραφή του συνδέσμου συζήτησης δίπλα →	Συζήτηση, Link της ομάδας, Μέλη της ομάδας	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑ (100% Success)
2	Άσκηση γραμμής εντολών (linux install)	Link Συζήτησης	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑ (100% Success)

Εβδομάδα	Όλα τα παραδοτέα βρίσκονται στην ίδια σελίδα της τελικής αναφοράς με τα προσωπικά στοιχεία σας (Όνομα, AM, github profile) και μαζί με αυτόν εδώ τον πίνακα περιεχομένων		
		Σύνδεσμος στην εβδομαδιαία παρουσίαση προόδου στις συζητήσεις	Αυτοαξιολόγηση σύμφωνα με τα κριτήρια της αντίστοιχης άσκησης
3	Συμμετοχικό περιεχόμενο A1	Link Συζήτησης, PR #1, PR #2, Organization Base Site, Link of Work #1 & Link of Work #2	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας <input checked="" type="checkbox"/> (100% Success)
4	Άσκηση γραμμής εντολών (warm up cli)	Link Συζήτησης	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας <input checked="" type="checkbox"/> (100% Success)
5	Συμμετοχικό περιεχόμενο A2	Link Συζήτησης, PR #1, PR #2, Link of Work #1, Link of Work #2, Link of Work #3 & Link of Work #4	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας <input checked="" type="checkbox"/> (100% Success)

Εβδομάδα	Όλα τα παραδοτέα βρίσκονται στην ίδια σελίδα της τελικής αναφοράς με τα προσωπικά στοιχεία σας (Όνομα, AM, github profile) και μαζί με αυτόν εδώ τον πίνακα περιεχομένων	Σύνδεσμος στην εβδομαδιαία παρουσίαση προόδου στις συζητήσεις	Αυτοαξιολόγηση σύμφωνα με τα κριτήρια της αντίστοιχης άσκησης
6	Κατασκευή του βιβλίου A	Link Συζήτησης, Contribution Point, Contribution Photo, .EPUB file, .PDF file, Contribution Filter	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας <input checked="" type="checkbox"/> (100% Success)
7	Συμμετοχικό περιεχόμενο B1	Link Συζήτησης, Link of Case-Study, Link of Biography, Site Link	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας <input checked="" type="checkbox"/> (100% Success)
8	Άσκηση γραμμής εντολών	Link Συζήτησης, Spiritbeat Repository, Converted PDF, Converted HTML & Shell Script	Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας <input checked="" type="checkbox"/> (100% Success)
9	Συμμετοχικό περιεχόμενο B2		

	Όλα τα παραδοτέα βρίσκονται στην ίδια σελίδα της τελικής αναφοράς με τα προσωπικά στοιχεία σας (Όνομα, AM, github profile) και μαζί με αυτόν εδώ τον πίνακα περιεχομένων	Σύνδεσμος στην εβδομαδιαία παρουσίαση προόδου στις συζητήσεις	Αυτοαξιολόγηση σύμφωνα με τα κριτήρια της αντίστοιχης άσκησης
Εβδομάδα			
10	Άσκηση γραμμής εντολών		
11	Κατασκευή του βιβλίου B		
12	Τελική αναφορά*		

2 Παραδοτέο 1ο / Εισαγωγή:

Παρακάτω παραθέτω τους προσωπικούς μου στόχους για το μάθημα “Επικοινωνία Ανθρώπου - Υπολογιστή” :

- i) Η ορθή κατανόηση των κανόνων που θεσπίζουν το μάθημα.
- ii) Η σωστή και εύρυθμη λειτουργία για την φιλοξενία ξένων και προσωπικών αποθετηρίων κώδικα στην πλατφόρμα “Github” (επεξεργασία κώδικα, διόρθωση στοιχείων, merging τα pull requests, δημιουργία αποθετηρίων κα.)
- iii) Εμπλουτισμό γνώσεων με τις διάφορες εγκαταστάσεις ανοιχτού λογισμικού διανομών Linux όπως η διανομή “Arch”. (π.χ. minimal distributions χωρίς το υπάρχων γραφικό περιβάλλον)
- iv) Εκτέλεση άγνωστων εντολών που θα αναζητηθούν μέσα από διάφορα forum (βλέπε: StackOverflow, Forum του Linux Foundation κα.) στην γραμμή εντολών του τερματικού και η περιγραφή των ενεργειών τους.

- ν) Τέλος, να αποκτήσω μέσα από τα παραπάνω γνώσεις προγραμματιστικές οι οποίες θα μου φανούν χρήσιμες για την ακαδημαϊκή και την επαγγελματική μου ζωή.

3 Παραδοτέο 2ο : Εγκατάσταση των Arch Linux σε USB

Μαζί με την εγκατάσταση της διανομής Arch Linux ζητήθηκε και η εκτέλεση των παρακάτω εντολών: `neofetch` και `journalctl -b` για να επιδείξουμε τα βασικά στοιχεία του υπολογιστή μας. Συγκεκριμένα, κατά την εγκατάσταση:

- i) Πραγματοποιήθηκε με επιτυχία η χειροκίνητη εγκατάσταση της διανομής (δηλαδή, χωρίς την εκτέλεση της εντολής `archinstall` η οποία εκκινεί το `script` για την αυτόματη εγκατάσταση του λειτουργικού) σε εξωτερικό USB χωρητικότητας 32GB με `persistent storage` σύμφωνα με το `documentation` των Arch.
- ii) Στην συνέχεια, μόλις το USB έγινε bootable με την διαμόρφωση των μερών σε 3 κομμάτια (πρώτο (1) Linux Filesystem, δεύτερο (2) Boot Files και τρίτο (3) Shared Files) ξεκίνησε η εγκατάσταση ορισμένων πακέτων τα οποία είναι υποχρεωτικά για την εύρυθμη λειτουργία του συστήματος. Μερικά από αυτά είναι: για γραφικά (`xf86-video-intel` , `xf86-video-nouveau`, `xf86-video-ati`) για διαμόρφωση δίσκων: (`gparted`), για σύνδεση στο δίκτυο (`NetworkManager`), για γραφικό περιβάλλον (`xfce4-session`, `xfce4-goodies`) κα. Επιπλέον, εγκαταστάθηκαν και οι `file managers` όπως και οι εντολές `neofetch`, `journalctl` για την καλύτερη διεπαφή ανθρώπου - υπολογιστή.
- iii) Δημιουργήθηκε και ο χρήστης με βασικό όνομα `Spyros` και `hostname` τον προσωπικό μου αριθμό μητρώου (χωρίς τα πρώτα γράμματα, μόνο οι αριθμοί) όπως: `spyros@2021098`, το ίδιο ισχύει και για τον χρήστη `root`: `root@2021098` με την εντολή (`sudo hostname 2021098`). Μετά την δημιουργία των χρηστών, ο χρήστης `spyros` δεν είχε δικαιώματα `sudoer` οπότε έπρεπε να προστεθεί στο αρχείο `sudoers`. Η διαδικασία αυτή επιτεύχθηκε με την παρακάτω εντολή: `usermod -aG sudo spyros`
- iv) Επιπροσθέτως, επιτεύχθηκε και η αλλαγή του `Timezone` στο σωστό μέρος σύμφωνα με την χώρα που βρισκόμαστε.

Συγκεκριμένα, έτρεξα την εντολή `timedatectl list-timezones` με την οποία εμφάνισα την λίστα με τα `timezones` προκειμένου να επιλέξω το σωστό σύμφωνα με την χώρα Ελλάδα. Επέλεξα την: `Europe/Athens` προκειμένου να οριστεί η σωστή ώρα σύμφωνα με την πρωτεύουσα του κράτους μας και όρισα το `timezone` με την εντολή: `timedatectl set-timezone Europe/Athens`.

- ν) Τέλος, εγκατέστησα τον GRUB Boot Loader σε περίπτωση που χρειαστεί να τρέξω παράλληλα δύο λειτουργικά συστήματα. Εφόσον χρειαστεί να τρέξω άλλο λειτουργικό θα πρέπει να κάνω επεξεργασία ορισμένα αρχεία από το GRUB προκειμένου να εμφανίσω δύο (2) ή περισσότερα λειτουργικά συστήματα στο Boot Entry του GRUB (κοινώς το Startup Screen του GRUB)

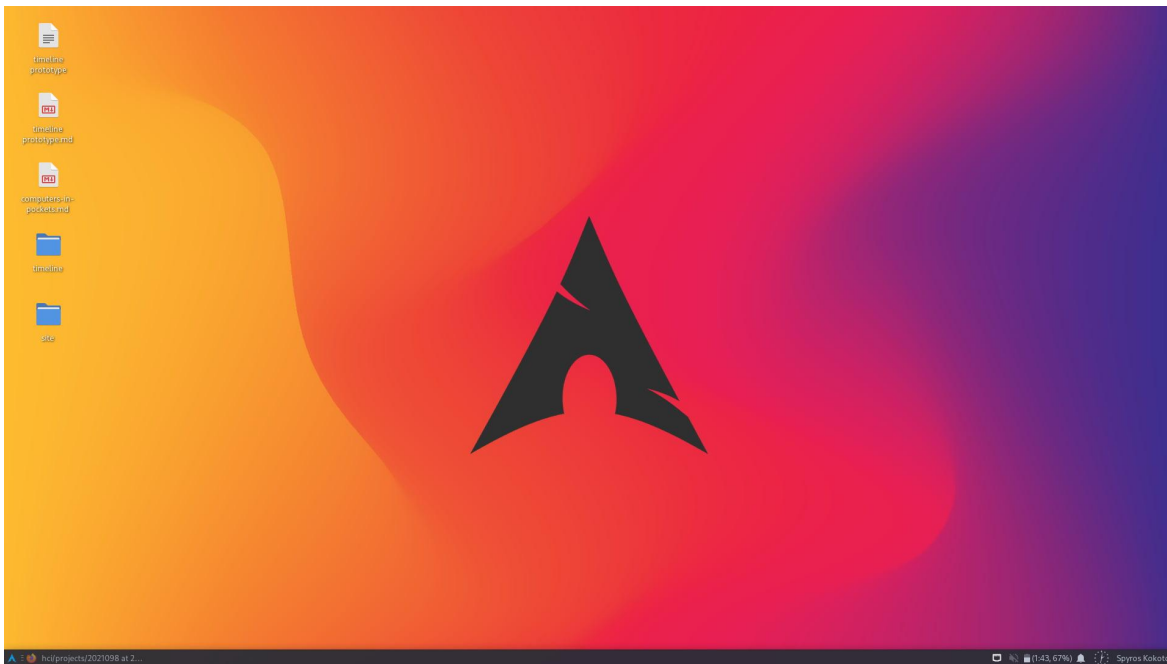


Figure 1: Screenshot_2022-11-08_14-01-35

Προβλήματα:

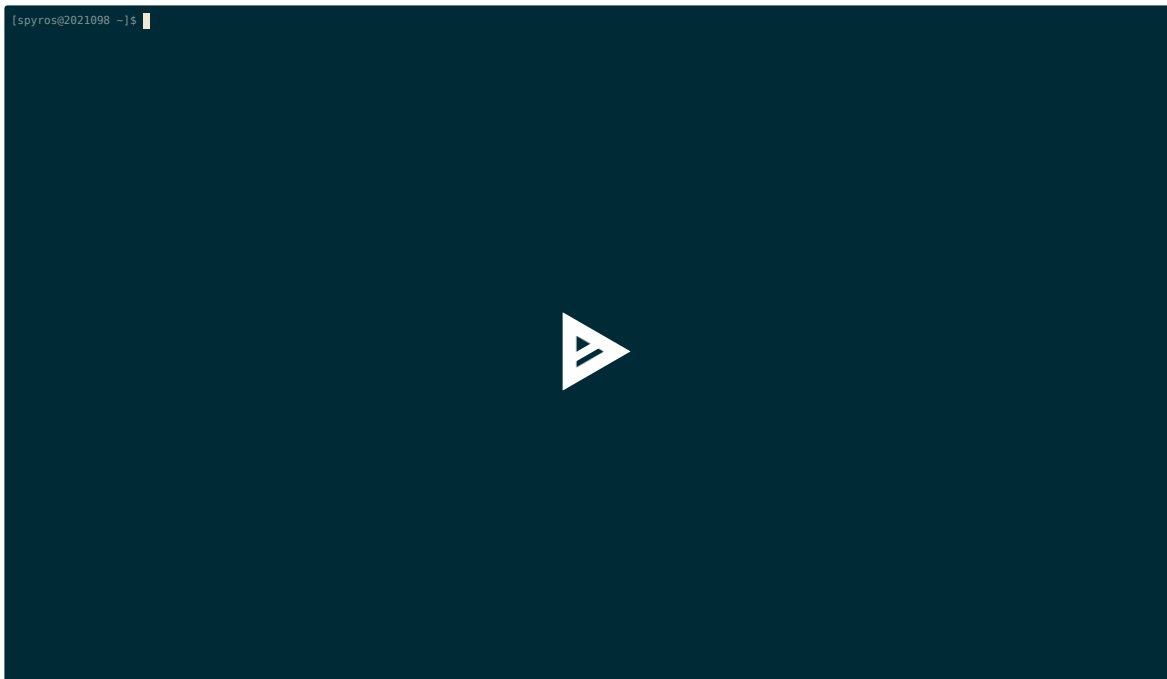
- i) Το `installation guide` που προσφέρει το forum του Arch Linux, ναι μεν είναι πλούσιο σε γνώσεις, αλλά, σε ορισμένα θέματα κάποιος μπορεί να μπερδευτεί πολύ εύκολα. Στην παρούσα φάση χρειάστηκε να αξιοποιήσω μια άλλη πηγή γνώσης (προσφυγή σε tutorial στο YouTube) για να επιλύσω το θέμα με τα `partition` στο USB, πόσα GB παραδείγματος χάριν θα αφιερώσω στο κάθε ένα και για ποιόν λόγο.

- ii) Κατά την διάρκεια της εγκατάστασης ο χρήστης δεν μπορεί να έχει πρόσβαση στο internet μέσω ασύρματου δικτύου και πρέπει αναγκαστικά να αξιοποιήσει το ethernet μέχρι να εγκαταστήσει ο ίδιος τα πακέτα NetworkManager, dialog και wifi-menu για να μπορέσει να συνδεθεί ασύρματα.
- iii) Επιπλέον, ακόμα και μετά την εγκατάσταση και την έναρξη του γραφικού περιβάλλοντος πρέπει να γίνουν αρκετές τροποποιήσεις για να λειτουργήσει σωστά το Wi-Fi. Μια από αυτές είναι να εμφανίσουμε την λίστα των οδηγών δικτύου που είναι εγκατεστημένα στην εκάστοτε διανομή με την εντολή: `ip link show` . Με αυτόν τον τρόπο ο κάθε χρήστης μπορεί να επιλέξει χειροκίνητα τι αντάπτορα (device adaptor) θα χρησιμοποιήσει για να πλοηγηθεί στο διαδίκτυο. Οδηγίες σχετικές με την επίλυση του συγκεκριμένου προβλήματος μπορεί κάποιος να τις βρει στην συγκεκριμένη συζήτηση: Discussion #1069.

Παρατηρήσεις:

- i) Μετά την ολοκλήρωση της εγκατάστασης του λειτουργικού χρησιμοποίησα το terminal προκειμένου να κάνω κάποιες warm-up ασκήσεις με σκοπό να δω αν το συγκεκριμένο λειτουργικό μοιάζει (όσον αφορά τις εντολές) με κάποιο άλλο λειτουργικό που ασχολήθηκα παλαιότερα. Αρχικά παρατήρησα ότι ο τρόπος εγκατάστασης προγραμμάτων γίνεται με την εντολή `pacman -S` όπως ακριβώς και στην διανομή manjaro και έχει αλλάξει γενικά η δομή των εντολών εκτέλεσης στο τερματικό, παραδείγματος χάριν στο ubuntu για να κατεβάσεις κάτι τρέχεις την εντολή `sudo apt-get install -` , ενώ, στα arch διαφοροποιείται ως εξής: `sudo pacman -Syy -` .

Τέλος, παρακάτω παραθέτω το link για την επίδειξη των εντολών `neofetch` & `journalctl -b` μέσω της πλατφόρμας `asciinema`:



4 Παραδοτέο 3ο: Συμμετοχικό περιεχόμενο A1.

Στο συγκεκριμένο παραδοτέο, καλούμαστε να συμβάλλουμε ομαδικά και ατομικά στην ανάπτυξη του ήδη υπάρχοντος site. Σαν αρχηγός του οργανισμού Terminal-Killers δημιουργήσα συγκεκριμένο section για συζήτηση περί επιλογής θεμάτων , προβλημάτων που ίσως προκύψουν στην πορεία και τέλος οδηγίες χρήσης για να καταφέρει το κάθε μέλος ξεχωριστά να συμβάλλει στην ανάπτυξη της ιστοσελίδας.

Προσωπικά, ασχολήθηκα με δύο θέματα τα οποία ανέπτυξα στην ιστοσελίδα τα οποία κατά την γνώμη μου απευθύνονται στις τωρινές καταστάσεις περι επικοινωνίας ανθρώπου - υπολογιστή μέσω υλικών αγαθών. Συγκεκριμένα, αναφέρθηκα στην προγραμματιστική γλώσσα υψηλού επιπέδου Python και στον ολοκαίνουργιο υπολογιστή τσέπης ODR0ID-C4. Αναλυτικότερες πληροφορίες στην επόμενη παράγραφο.

4.1 Περιγραφή: Python

Η Python είναι διερμηνευόμενη (interpreted), γενικού σκοπού (general-purpose) και υψηλού επιπέδου, γλώσσα προγραμματισμού.

Ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο το διαδικαστικό (procedural programming) όσο και το αντικειμενοστρεφές (object-oriented programming) προγραμματιστικό υπόδειγμα (programming paradigm). Είναι δυναμική γλώσσα προγραμματισμού (dynamically typed) και υποστηρίζει συλλογή απορριμμάτων (garbage collection ή GC). Επιλέχθηκε η συγκεκριμένη γλώσσα διότι είναι η πιο πρόσφατη με την οποία δημιουργήσα ένα project για το μάθημα Εφαρμοσμένος Προγραμματισμός με Python του Β' εξαμήνου.

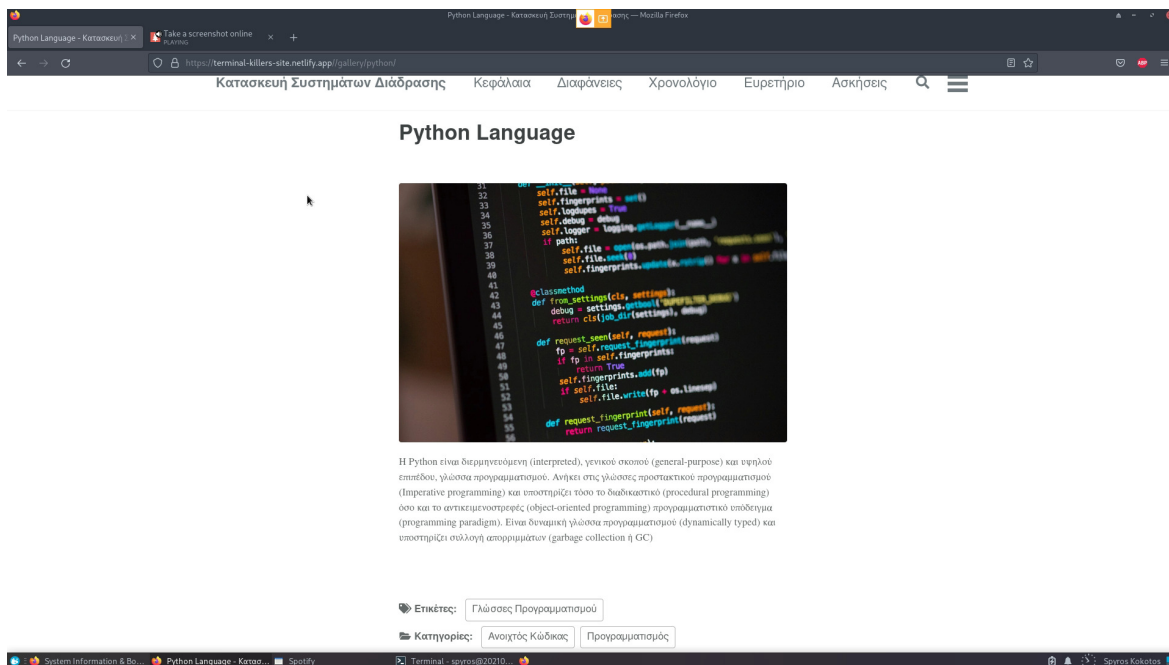


Figure 2: image(7)

4.2 Περιγραφή: ODR0ID-C4

Το ODR0ID-C4 είναι η νέα γενιά μονής μητρικής υπολογιστικής η οποία είναι βασισμένη στην αρχιτεκτονική ARM 64bit υπολογιστών. Ο κύριος επεξεργαστής του ODR0ID-C4 είναι ένας τετραπύρηνος Cortex-A55 με την ολοκαίνουργια ενσωματωμένη κάρτα γραφικών Mali-G31 GPU. Ο A-55 τρέχει σε συχνότητα των 2.0Ghz χωρίς ψύξη αέρος ή νερού, χρησιμοποιώντας μόνο παθητική ψύξη μέσω heat sink κάτι που το καθιστά αθόρυβο σαν υπολογιστή τσέπης.

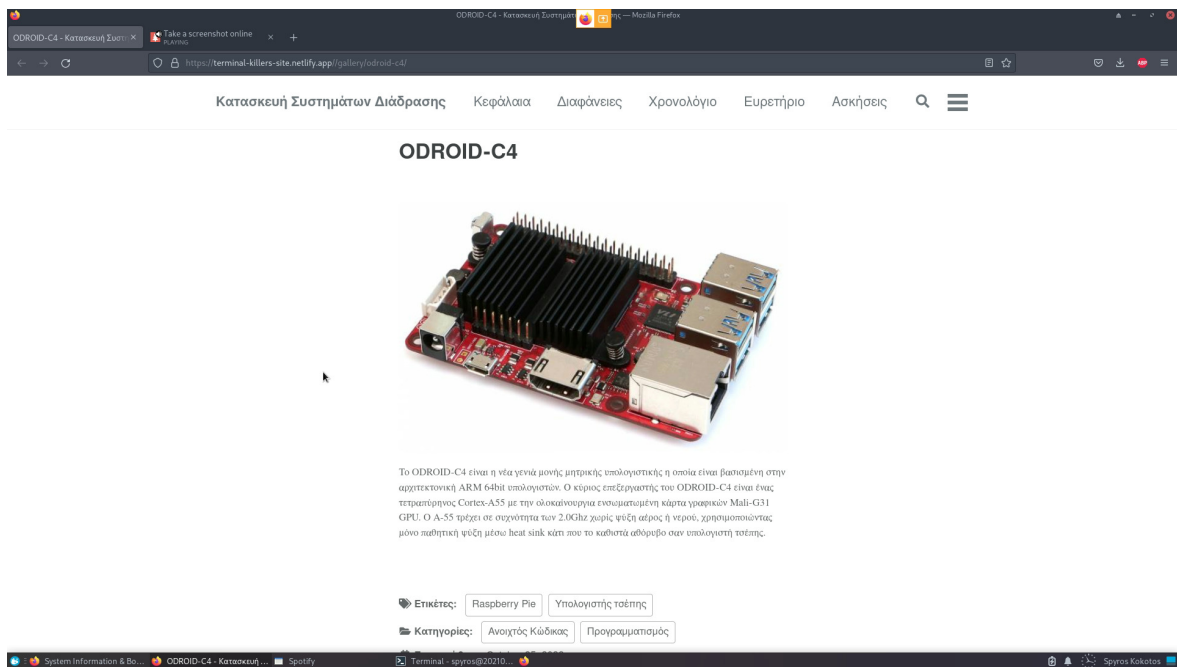


Figure 3: image(8)

5 Παρακάτω, παραθέτω τα link από το κάθε ένα αρχείο μαζί με τις φωτογραφίες που το συνοδεύουν:

- Προσωπικό site από το αποθετήριο μου (test site): Greek-force1821
- Άρθρο για την Python στο προσωπικό αποθετήριο (test site): Python
- Άρθρο για τον ODR0ID-C4 στο προσωπικό αποθετήριο (test site): ODR0ID-C4
- Φωτογραφία της Python σε πραγματική ανάλυση και σε resized ανάλυση στο προσωπικό αποθετήριο (test site)
- Φωτογραφία του ODR0ID-C4 σε πραγματική ανάλυση και σε re-sized ανάλυση στο προσωπικό αποθετήριο (test site)

-
- Το site από το αποθετήριο του οργανισμού: Terminal-Killers
 - Το άρθρο μου για την Python στο αποθετήριο του οργανισμού: Python Organization
 - Το άρθρο μου για τον ODR0ID-C4 στο αποθετήριο του οργανισμού: ODR0ID-C4 Organization

- Φωτογραφία της Python σε πραγματική ανάλυση και σε resized ανάλυση
- Φωτογραφία του ODR0ID-C4 σε πραγματική ανάλυση και σε re-sized ανάλυση

6 Παραδοτέο 4ο: Warming Up with CLI Commands.

Για το τέταρτο (4ο) παραδοτέο της αναφοράς μου, επέλεξα να κάνω την ψυχαγωγικής μορφής άσκηση γραμμής εντολών download mp4 της ενότητας WarmUp. Τα παραδοτέα της άσκησης αυτής, ζητούσαν την αναζήτηση, την λήψη και την αναπαραγωγή του αγαπημένου μας τραγουδιού, χρησιμοποιώντας εργαλεία της γραμμής εντολών. Επέλεξα την συγκεκριμένη άσκηση προκειμένου να γίνει χρήση του τερματικού εντολών και όχι ο απλός και αυτοματοποιημένος τρόπος που προσφέρουν έτοιμα προγράμματα ή αντίστοιχα plugin (βλέπε: flashgot, YTD Video Converter κα.) για τους browser μας.

6.0.1 Περιγραφή: Listing των format από το τραγούδι, Downloading το τραγούδι σε .mp4 format & Listing το αρχείο στο directory Downloads μέσω της εφαρμογής Youtube-DL

Για τις παραπάνω πληροφορίες αξιοποιήθηκαν και ακολουθήθηκαν τα παρακάτω βήματα προκειμένου να ακούσω το αγαπημένο μου τραγούδι.

- 1) Άνοιξα το τερματικό εντολών και μέσω της εντολής curl: `sudo curl -L https://yt-dl.org/downloads/latest/youtube-dl -o /usr/local/bin/youtube-dl` κατέβασα από το official repository το Youtube-DL
- 2) Άνοιξα το YouTube, έγραψα στην γραμμή αναζήτησης το τραγούδι R.E.M. - Shiny Happy People (Official Music Video) και έκανα αντιγραφή τον σύνδεσμο που αντιστοιχεί στο κομμάτι αυτό.
- 3) Στην συνέχεια, άνοιξα ένα παράθυρο του τερματικού εντολής και εκτελώντας την εντολή: `youtube-dl --list-format https://www.youtube.com/watch?v=YYOKMUTTDdA` το τερματικό μου εμφάνισε όλα τα video encodings που υποστηρίζονται από το παραπάνω κομμάτι.

```
[spyros@2021098 Downloads]$ youtube-dl
```

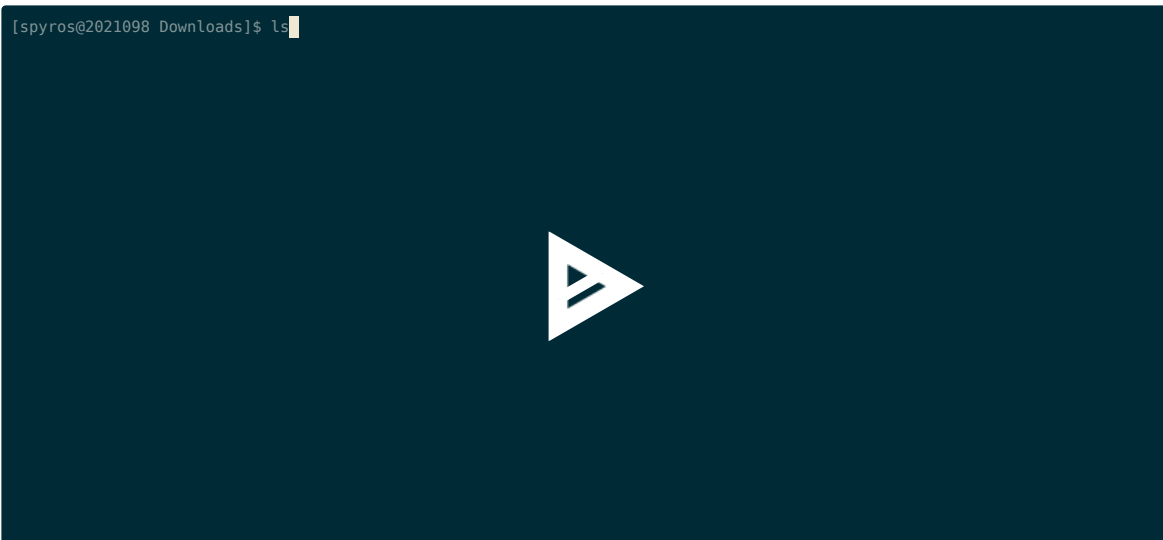


- 4) Παρακάτω, εφόσον τσέκαρα ότι το βίντεο υποστηρίζει .mp4 format έτρεξα την εντολή: `youtube-dl -f mp4 https://www.youtube.com/watch?v=YY0KMUTTDdA` προκειμένου να κατεβάσω το συγκεκριμένο τραγούδι στον υπολογιστή μου. Η παράμετρος `-f` ισούται με το format που έχουμε επιλέξει.

```
[spyros@2021098 Downloads]$ youtube-dl -f mp4 https://www.youtube.com/watch?v=YY0KMUTTDdA
[youtube] YY0KMUTTDdA: Downloading webpage
[download] Destination: R.E.M. - Shiny Happy People (Official Music Video)-YY0KMUTTDdA.mp4
[download] 0.2% of 13.64MiB at 61.30KiB/s ETA 03:47
```



- 5) Τέλος, για να σιγουρευτώ ότι έχω ένα αρχείο .mp4 στον υπολογιστή μου, έτρεξα την εντολή: `cd Downloads/` προκειμένου να αλλάξω directory και στην συνέχεια έτρεξα την εντολή: `ls` για να μου εκτυπώσει το τερματικό τα αρχεία που υπάρχουν στο συγκεκριμένο directory. Στο παρακάτω βίντεο από το asciinema γίνεται αντιληπτό πως το αρχείο είναι 'οσωμένο' locally και δουλεύει με επιτυχία!



Προβλήματα:

Το τερματικό στην εκτέλεση της πρώτης εντολής του 4ου παραδοτέου δεν με άφηνε να ολοκληρώσω την λήψη μέσω curl γιατί παρόλο που είχε γίνει σωστή εγκατάσταση δεν αναγνώριζε την σωστή διαδρομή του πακέτου από το σύστημα. Ευτυχώς, έκανα ένα uninstall και άνοιξα την πόρτα μέσω AUR το οποίο είναι σαν το GitHub, αλλά, εκεί ανεβαίνουν πακέτα από τους χρήστες, άρα θεωρείται ένα user program repository. Εν συνέχεια, κατέβασα τα αρχεία από το αποθετήριο, έκανα αλλαγή στο directory και έκανα compile μέσω του τερματικού με την εντολή: `makepkg -s` προκειμένου να εγκατασταθεί στο σύστημα το curl.

7 Παραδοτέο 5ο: Συμμετοχικό Περιεχόμενο A2.

Στο συγκεκριμένο παραδοτέο, καλούμαστε να συμβάλλουμε ομαδικά και ατομικά για ακόμα μια φορά στην ανάπτυξη του υπάρχοντος site. Σαν αρχηγός του οργανισμού Terminal-Killers έλαβα υπόψιν τα προβλήματα που προέκυψαν στην πορεία τόσο από τους συμφοιτητές μου αλλά και τόσο και από την προσωπική οπτική γωνία και προσπάθησα να τα επιλύσω μέσω των συζητήσεων.

Παρακάτω, ασχολήθηκα με τα προηγούμενα θέματα που ανέπτυξα στην ιστοσελίδα (Python & ODR0ID-C4) και τα ενσωμάτωσα σε δύο διαφάνειες (slides) & σε δύο χρονολόγια (timelines). Αναλυτικότερες λεπτομέρειες στην επόμενη παράγραφο.

7.1 Περιγραφή: Timelines

Στο συγκεκριμένο παραδοτέο, αξιοποίησα τις προηγούμενες μου γνώσεις και δημιούργησα δύο timelines σχετικές με τις Υψηλού επιπέδου γλώσσες προγραμματισμού και αντίστοιχα με τους Υπολογιστές τσέπης. Δημιούργησα δύο αρχεία (.md) με κωδικές ονομασίες: `computers-in-pockets.md` και `programming-languages.md` στο προσωπικό repository της σελίδας μου, έκανα redeploy και τσέκαρα ότι απεικονίζονται σωστά στην ιστοσελίδα, επομένως προχώρησα σε pull request για το submodule `_timeline` στο repo site του οργανισμού. Αφού έγινε deploy τσέκαρα ότι εμφανίστηκαν σωστά τα παραπάνω χρονολόγια στο organization site και προχώρησα στο επόμενο βήμα.

7.2 Υπολογιστές Τσέπης Timeline

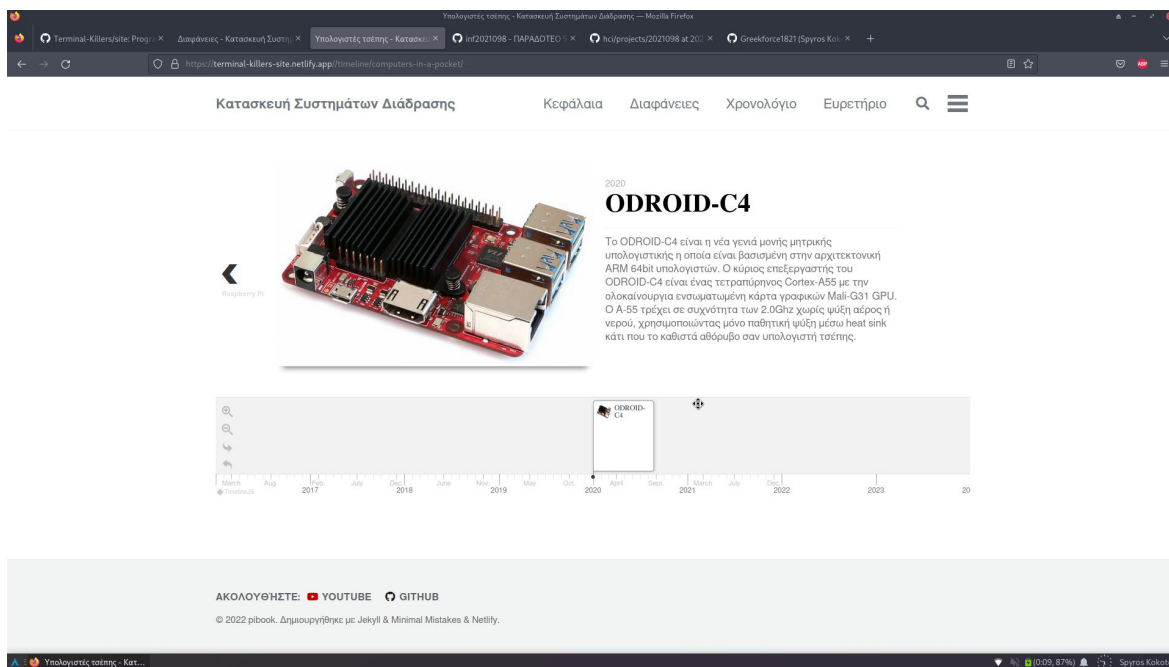


Figure 4: Screenshot_2022-11-07_17-31-50

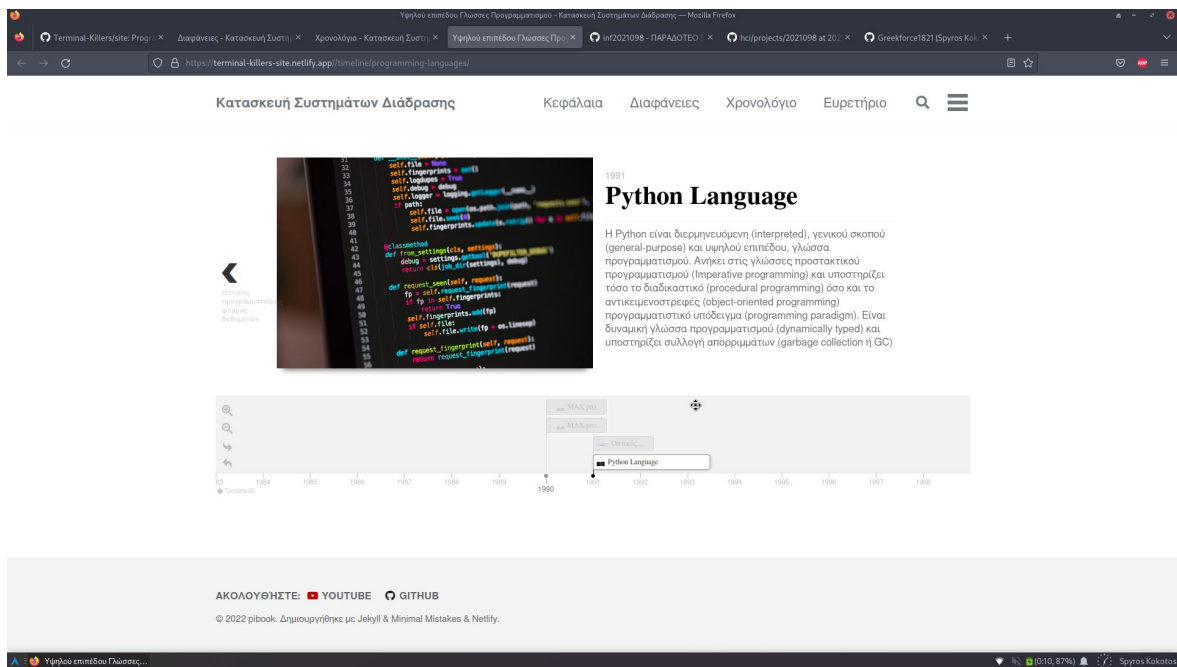


Figure 5: Screenshot_2022-11-07_17-32-09

7.3 Υψηλού Επιπέδου Γλώσσες Προγραμματισμού Timeline

7.4 Περιγραφή: Slides

Στο συγκεκριμένο παραδοτέο, αξιοποίησα τις προηγούμενες μου γνώσεις και δημιούργησα δύο slides σχετικά με τις Υψηλού επιπέδου γλώσσες προγραμματισμού και αντίστοιχα με τους Υπολογιστές τσέπης. Δημιούργησα δύο αρχεία (.md) με κωδικές ονομασίες: computers-in-pockets.md και programming-languages.md στο προσωπικό repository της σελίδας μου, έκανα redeploy και τσέκαρα ότι απεικονίζονται σωστά στην ιστοσελίδα, επομένως προχώρησα σε pull request για το submodule _slides στο repo site του οργανισμού. Αφού έγινε deploy τσέκαρα ότι εμφανίστηκαν σωστά οι παραπ στο organization site και προχώρησα στο επόμενο βήμα.

Στην αναφορά μου στο section των discussion και συγκεκριμένα στο: Discussion #1461 αναφέρω για την επεξήγηση του αστερίσκου(*) . Εφόσον έχω αναλάβει να εκπονήσω αυτήν την εργασία, επέλεξα να ασχοληθώ με μια συγκεκριμένη γλώσσα προγραμματισμού την Python, αν ανοίξω το αποθετήριο των slides παρατηρώ ένα ήδη υπάρχον αρχείο (.md) με κωδική ονομασία:

programming.md. Το συγκεκριμένο αρχείο, παρουσιάζει ήδη μια γκάμα γλωσσών προγραμματισμού, οπότε θεώρησα σωστό εφόσον υπάρχει ήδη κάτι σχετικό με τις γλώσσες προγραμματισμού να μην δημιουργήσω κάτι ξεχωριστό και απλά να ενημερώσω το συγκεκριμένο αρχείο. Για ακριβώς αυτόν τον λόγο δεν δημιούργησα ένα νέο (.md) αρχείο για ένα δεύτερο slide. Παρόλα αυτά, το ενσωμάτωσα με τα υπόλοιπα για να μπορέσω να κάνω contribute στο συγκεκριμένο (.md) αρχείο του site

7.5 Γλώσσες Προγραμματισμού Slide

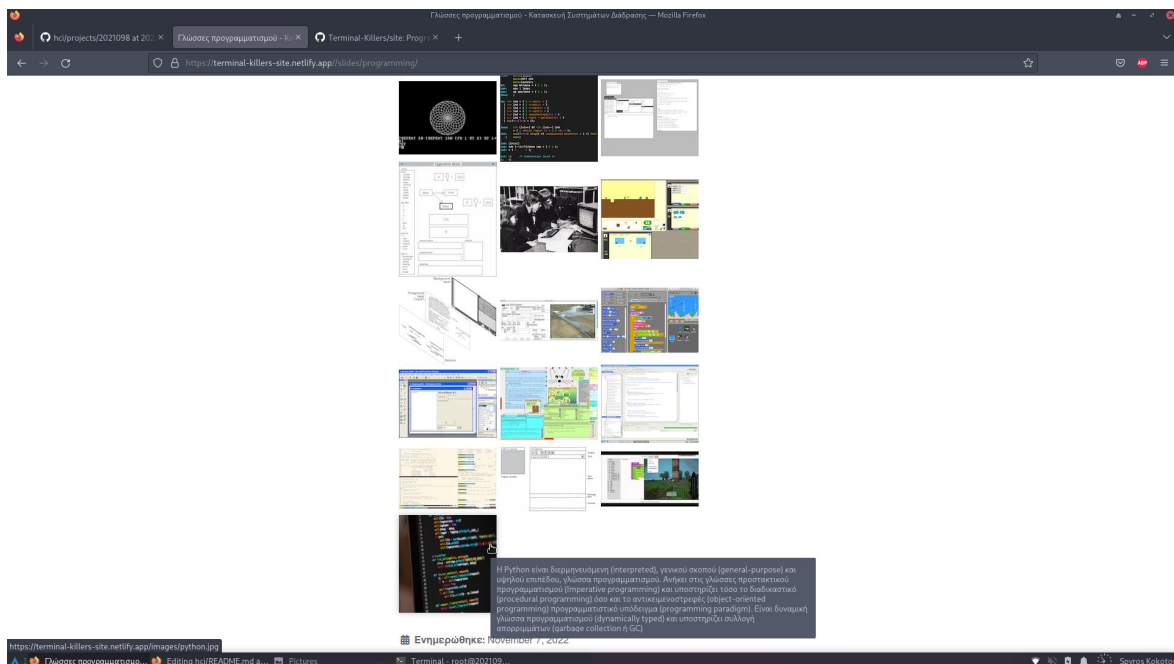


Figure 6: Screenshot_2022-11-07_18-41-57

7.6 Υπολογιστές Τσέπης Slide

8 Παρακάτω, παραθέτω τα link από το κάθε ένα αρχείο μαζί με τις φωτογραφίες που το συνοδεύουν:

- Προσωπικό site από το αποθετήριο μου (test site): [Greek-force1821](#)

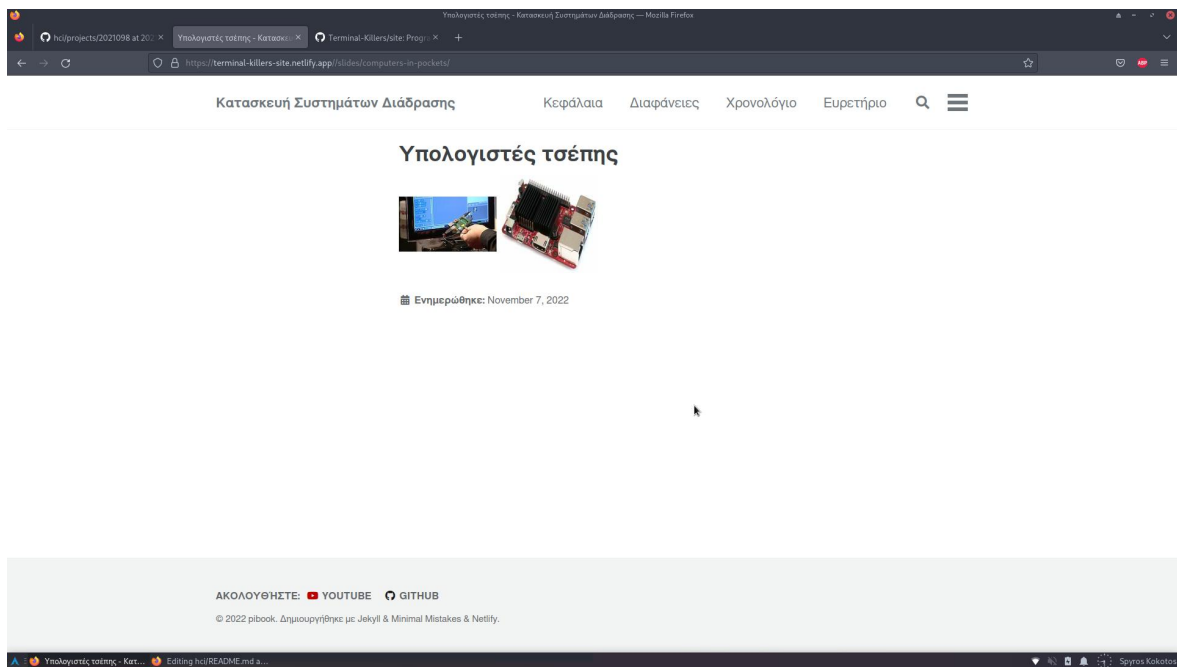


Figure 7: Screenshot_2022-11-07_18-43-32

- Timeline του προσωπικού αποθετηρίου (test site): Timeline Greekforce1821
 - Timeline #1 του προσωπικού αποθετηρίου (test site): Timeline #1 Greekforce1821
 - Timeline #2 του προσωπικού αποθετηρίου (test site): Timeline #2 Greekforce1821
 - Slide του προσωπικού αποθετηρίου (test site): Slide Greekforce1821
 - Slide #1 του προσωπικού αποθετηρίου (test site): Slide #1 Greekforce1821
 - Slide #2 του προσωπικού αποθετηρίου (test site): Slide #2 Greekforce1821
-
- Το site από το αποθετήριο του οργανισμού: Terminal-Killers
 - Timeline του αποθετηρίου του οργανισμού: Timeline Organization
 - Timeline #1 του αποθετηρίου του οργανισμού: Timeline #1 Organization
 - Timeline #2 του αποθετηρίου του οργανισμού: Timeline #2 Organization
 - Slide του αποθετηρίου του οργανισμού: Slide Organization

- Slide #1 του αποθετηρίου του οργανισμού: Slide #1 Organization
- Slide #2 του αποθετηρίου του οργανισμού: Slide #2 Organization

9 Παραδοτέο 6ο: Κατασκευάζοντας το βιβλίο.

Στο συγκεκριμένο παραδοτέο, καλούμαστε να συμβάλλουμε και αντίστοιχα να κατασκευάσουμε το βιβλίο σε ηλεκτρονική μορφή (.pdf) αλλά και σε μορφή (.pub). Το παραδοτέο αυτό, ήταν ατομικό και συγκεκριμένα έπρεπε να δημιουργήσουμε το δικό μας φίλτρο (.lua), το δικό μας contribution, το βιβλίο το οποίο περιείχε το contribution μας και τέλος το βιβλίο σε μορφή (.tex). Περαιτέρω πληροφορίες στην επόμενη παράγραφο.



Figure 8: Screenshot_2022-11-14_16-29-28

9.1 Βήμα #1: Forking and Cloning the repository

Στο συγκεκριμένο βήμα, χρειάστηκε να κάνω fork το αρχικό αποθετήριο: mibook/kallipos στο προφίλ μου: Greekforce1821/kallipos και στην συνέχεια μέσω τερματικού να κάνω clone locally το

αποθετήριο από το προφίλ μου χρησιμοποιώντας την εντολή: (git clone <https://github.com/Greekforce1821/kallipos>) .

```
[spyros@2021098 ~]$ git clone https://github.com/Greekforce1821/kallipos
Cloning into 'kallipos'...
remote: Enumerating objects: 281, done.
Receiving objects: 84% (237/281), 24.25 MiB | 1.88 MiB/s
```



9.2 Βήμα #2: Downloading the Suitable Software via Terminal

Σύμφωνα με τις οδηγίες του μαθήματος guides προκειμένου να εισάγουμε το contribution στο βιβλίο πρέπει να εγκαταστήσουμε στο σύστημά μας διάφορα πακέτα όπως: [pandoc](<https://pandoc.org/>) και [LaTeX](<https://www.latex-project.org/>). Ειδικά, το LaTeX αξιοποιείται από τον χρήστη προκειμένου να κάνει merge όλα τα (.tex) αρχεία σε ένα και στην συνέχεια να γίνει export από (.tex) σε (.pdf) αρχείο.

```
haskell-transformers-base-0.4.6-69 haskell-transformers-compat-0.7.2-1
haskell-type-equality-1.12 haskell-typed-process-0.2.10.1-21
haskell-unicode-collation-0.1.3.3-6 haskell-unicode-data-0.4.0-4
haskell-unicode-transforms-0.4.0.1-36 haskell-uniplatte-1.6.13-160
haskell-unix-compat-0.6-37 haskell-unix-time-0.4.8-9 haskell-unliftio-0.2.23.0-2
haskell-unliftio-core-0.2.0.1-9 haskell-unordered-containers-0.2.19.1-44
haskell-utf8-string-1.0.2-117 haskell-uuid-types-1.0.5-84 haskell-vault-0.3.1.5-123
haskell-vector-0.13.0.0-12 haskell-vector-algorithms-0.9.0.1-18
haskell-vector-stream-0.1.0.0-1 haskell-wai-3.2.3-160 haskell-wai-app-static-3.1.7.4-93
haskell-wai-extra-3.1.13.0-4 haskell-wai-logger-2.4.0-155 haskell-warp-3.3.23-12
haskell-witherable-0.4.2-28 haskell-word8-1.3-22 haskell-x509-1.7.7-24
haskell-x509-store-1.6.9-49 haskell-x509-signature-1.6.7-52 haskell-x509-validation-1.6.12-54
haskell-xml-1.3.14-30 haskell-xml-conduit-1.3.14-4 haskell-xml-types-0.3.8-8
haskell-yaml-0.11.8.0-70 haskell-zip-archive-0.2.2-7 haskell-zlib-0.6.3.0-27 lua-5.4.4-2
lua-lpeg-1.0.2-4 numactl-2.0.16-1 pandoc-2.9.2-29

Total Download Size: 20.63 MiB
Total Installed Size: 505.63 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
haskell-cmdargs-0.10.21-4-x86_64 753.0 B 0.00 B/s --:-- [-----] 0%
Total ( 0/182) 753.0 B 0.00 B/s --:-- [-----] 0%
```

9.3 Βήμα #3: Cloning the Submodules from the Main Repository and Setting them. Part #1

Στο συγκεκριμένο βήμα, όταν έκανα clone το αποθετήριο τοπικά στον υπολογιστή μου, τα submodules δεν είχαν ρυθμιστεί με αποτέλεσμα, να μην μπορώ να τρέξω την εντολή: `./make-latex.sh` το οποίο εκκινεί το script για την δημιουργία των (.tex) αρχείων και την ενοποίησή τους σε (.pdf) αρχείο. Προκειμένου να επιλυθεί το πρόβλημα, έπρεπε να τρέξω μέσω τερματικού τις εντολές: `git submodule update --remote --init` & `git submodule update --remote --merge` οι οποίες έκαναν clone locally τα αρχεία στους ειδικούς φακέλους από το αρχικό αποθετήριο

```
[spyros@2021098 kallipos]$ git submodule update --remote --init
Submodule 'bibliography' (https://github.com/piibook/bibliography.git) registered for path 'bibliography'
Submodule 'extras' (https://github.com/piibook/extras.git) registered for path 'extras'
Submodule 'figures' (https://github.com/piibook/_gallery) registered for path 'figures'
Submodule 'images' (https://github.com/piibook/images) registered for path 'images'
Submodule 'quotes' (https://github.com/piibook/_quotes.git) registered for path 'quotes'
Submodule 'text' (https://github.com/mibook/text.git) registered for path 'text'
Cloning into '/home/spyros/kallipos/bibliography'...
Cloning into '/home/spyros/kallipos/extras'...
Cloning into '/home/spyros/kallipos/figures'...
Cloning into '/home/spyros/kallipos/images'...
```

9.4 Βήμα #4: Cloning the Submodules from the Main Repository and Setting them. Part #2



9.5 Βήμα #5: Creating the Contribution Folder and the Contribution (.md) File

Στο παραπάνω βήμα, εφόσον διάβασα το βιβλίο σε (.pdf) από την βασική ιστοσελίδα του μαθήματος, ήθελα να συνδράμω για την εφαρμογή Skype η οποία αναγράφεται στο βιβλίο στην ενότητα 8 Μοντέλα και στην υποενότητα 8.3 Μέσο Επικοινωνίας. Συγκεκριμένα, δημιούργησα έναν φάκελο με κωδική ονομασία contribution στο αποθετήριο locally και μέσα στον ίδιο δημιούργησα ένα αρχείο skyprech08.md το οποίο περιέχει τις πληροφορίες που θέλω να προσθέσω στο ήδη υπάρχον αρχείο.

9.6 Βήμα #6: Creating the Contribution Filter (contribution.lua) & Editing the Make File (make-latex.sh)

Στο παραπάνω βήμα, δημιούργησα το φίλτρο μου προκειμένου να εισάγω το contribution μου στο βιβλίο και αντίστοιχα, πρόσθεσα στο αρχείο (.sh) το φίλτρο και ορισμένα μηνύματα, μαζί με την δημιουργία directories στο ήδη υπάρχον directory.

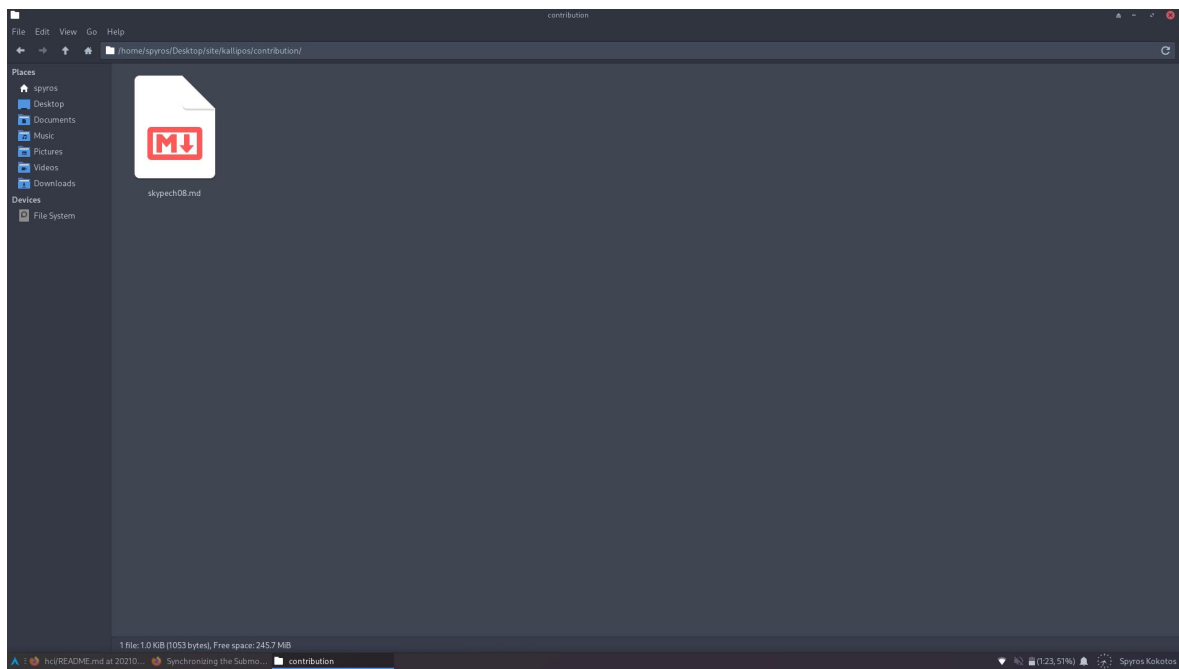


Figure 9: Screenshot_2022-11-14_16-41-36

μηνύματα ούτως ώστε, να υπάρχει μια έξοδος σε περίπτωση που υπάρξει κάποιο πρόβλημα ή αντίστοιχα να γνωρίζω τι πρόοδο έχει κάνει το συγκεκριμένο αρχείο.

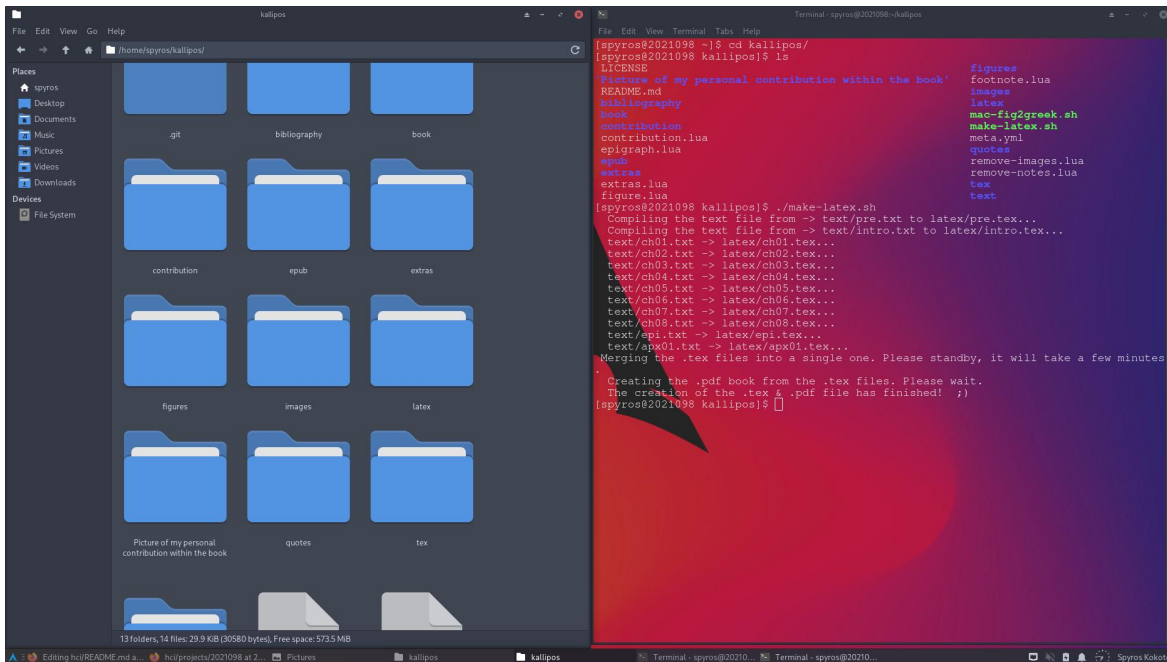


Figure 10: Screenshot_2022-11-17_18-35-47

Προβλήματα:

Όταν εκκίνησα για πρώτη φορά δοκιμαστικά το παραπάνω script παρατήρησα το συγκεκριμένο error στο τερματικό, το οποίο οδηγούσε στην αναζήτηση της γραμματοσειράς MesloLGS NF Regular η οποία δεν υπήρχε εγκατεστημένη στο σύστημα. Μετά από αρκετές ώρες ψάξιμο στο google, κατέληξα στο γεγονός ότι πρέπει να κατεβάσω το αρχείο .ttf της συγκεκριμένης γραμματοσειράς και να το προσθέσω στην παρακάτω διαδρομή: /usr/share/fonts/meslo-lgs-nf/. Αφού πρόσθεσα το αρχείο στην συγκεκριμένη διαδρομή, έτρεξα ξανά το script, αλλά, το τερματικό εμφάνισε ακριβώς το ίδιο πρόβλημα: pandoc: NF: openBinaryFile: does not exists (No such file or directory). Ψάχνοντας λίγο περισσότερο, κατέληξα στο γεγονός ότι έπρεπε να δημιουργήσω έναν κρυφό φάκελο με την ονομασία: .fonts στον χρήστη μου με την διαδρομή: /home/spyros/.fonts/ προκειμένου να επιλυθεί το συγκεκριμένο πρόβλημα.

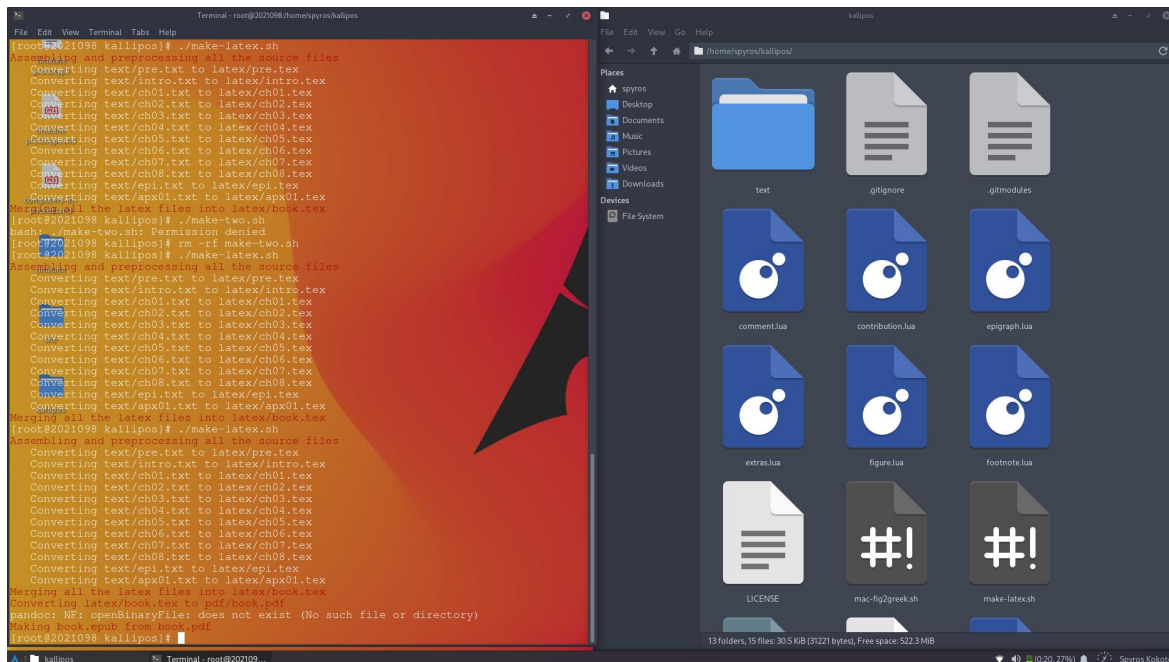


Figure 11: Screenshot_2022-11-13_23-04-39

9.8 Βήμα #8: Inserting the Contribution Filter Inside the Book

Στο συγκεκριμένο βήμα, αφού είχα κατασταλάξει που θα προσθέσω το contribution μου, έπρεπε να ορίσω σε ποίο chapter του βιβλίου θα προστεθεί, επομένως, εφόσον το Skype ανήκει στο Chapter: 06 και συγκεκριμένα στην υποενότητα: 6.0.3 άνοιξα τον φάκελο text και στην συνέχεια το αρχείο ch05.txt έψαξα να βρω που βρίσκεται η αναφορά για το Skype. Μόλις την βρήκα, προκειμένου να ενσωματώσω το φίλτρο μου στο συγκεκριμένο σημείο έπρεπε να προσθέσω την παρακάτω εντολή `{.contribution}` στο αρχείο προκειμένου να εμφανιστεί το contribution μου. Επιπλέον, να αναφέρω πως η δομή της συγκεκριμένης εντολής είναι η εξής: `{.examplefilter}` όπου στο example.md προσθετουμε την ονομασία που δώσαμε στο αρχείο μας και στις στο `{.examplefilter}` αντικαθιστούμε με την ονομασία του φίλτρου που φτιάξαμε.

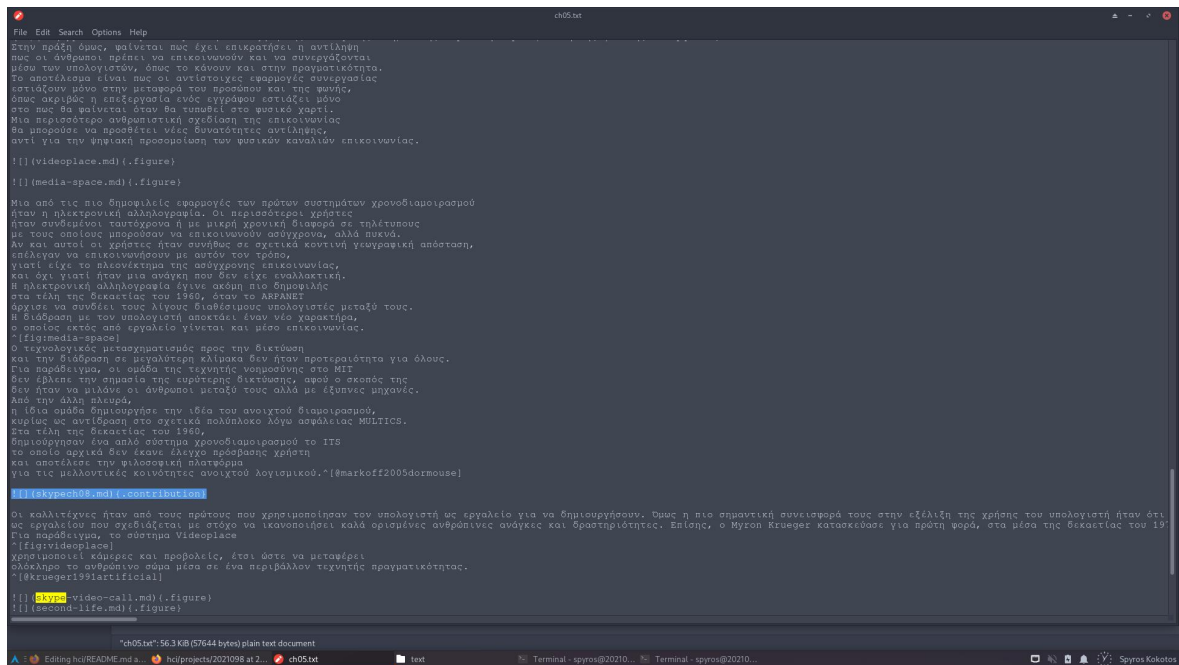


Figure 12: Screenshot_2022-11-17_18-55-08

9.9 Βήμα #9: Recompiling the Book With the Same Procedure

Στο παραπάνω βήμα, εφόσον ολοκλήρωσα τις αλλαγές μου, έπρεπε να ενημερώσω το ήδη υπάρχον βιβλίο, εκτελώντας ξανά την ίδια εντολή `./make-latex.sh` μέσα από το τερματικό προκειμένου να εκκινήσω το scriptάκι `make-latex.sh`. Μόλις ολοκληρώθηκε το `compilation`, προέκυψε το βιβλίο σε μορφή `(.tex)`, σε μορφή `(.pdf)` και στην μορφή `(.erub)`.

9.10 Βήμα #10: Displaying My Personal Contribution From Inside the PDF Book

Μόλις ολοκληρώθηκε και το τελευταίο `compilation` από το τερματικό, άνοιξα το αρχείο `book.pdf`, έτρεξα στην ενότητα 6 και στην συνέχεια στην υποενότητα 6.3 Μέσο Επικοινωνίας προκειμένου να παρατηρήσω την προσωπική μου “πινελιά” στο συγκεκριμένο βιβλίο.

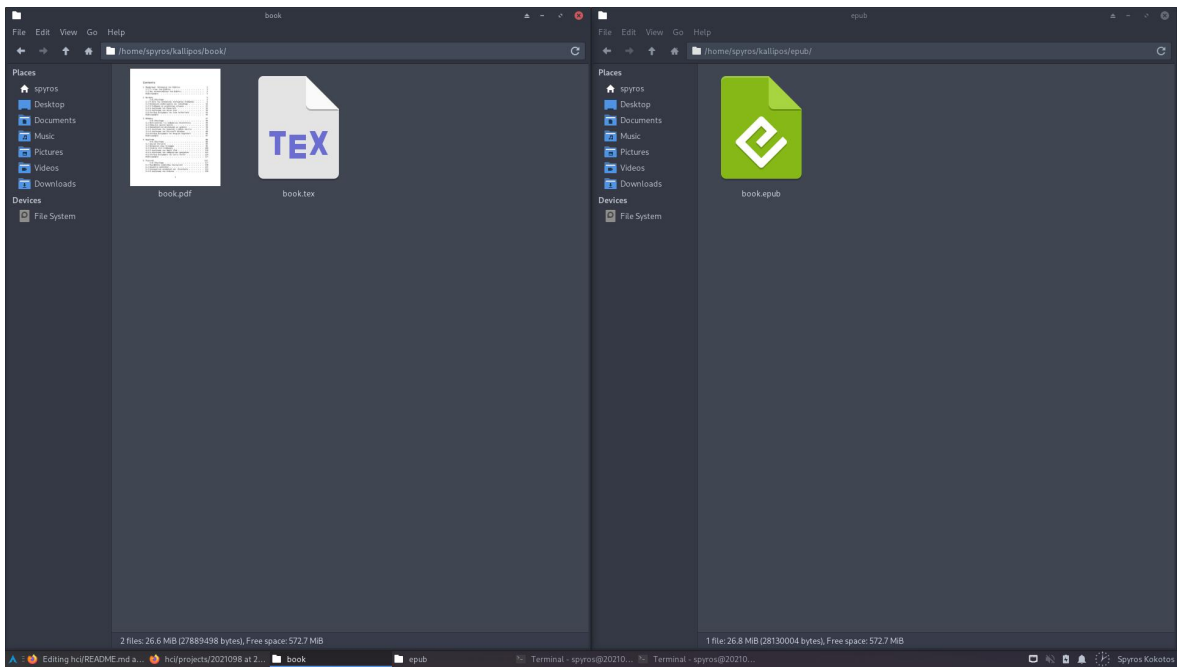


Figure 13: Screenshot_2022-11-17_19-00-12

10 Παραδοτέο 7ο: Συμμετοχικό περιεχόμενο B1.

Στο συγκεκριμένο παραδοτέο, έπρεπε να δημιουργήσουμε μια νέα μελέτη περίπτωσης και μια συγκεκριμένη βιογραφία, σχετικές με τα παραδοτέα A1 και A2 σύμφωνα με τις οδηγίες του μαθήματος και να τα ανεβάσουμε στην ιστοσελίδα μας. Προκειμένου να συνδέσω την βιογραφία με τα προηγούμενα, επέλεξα να ασχοληθώ με τον δημιουργό της υψηλού επιπέδου γλώσσας προγραμματισμού Python, τον Guido Van Rossum και για την συγγραφή της νέας μελέτης περίπτωσης, συγχώνευσα τις πολύτιμες και ιστορικές πληροφορίες της Python σε συνάρτηση με το Odroid-C4 και μια πολύ μικρή αναφορά στο Raspberry Pi.

ήταν συνήθως σε σχετικά κοντινή γεωγραφική απόσταση, επέλεγαν να επικοινωνήσουν με αυτόν τον τρόπο, γιατί είχε το πλεονέκτημα της ασύγχρονης επικοινωνίας, και όχι γιατί ήταν μια ανάγκη που δεν είχε εναλλακτική. Η ηλεκτρονική αλληλογραφία έγινε ακόμη πιο δημοφιλής στα τέλη της δεκαετίας του 1960, όταν το ARPANET άρχισε να συνδέει τους λίγους διαθέσιμους υπολογιστές μεταξύ τους. Η διάδραση με τον υπολογιστή αποκτάει έναν νέο χαρακτήρα, ο οποίος εκτός από εργαλείο γίνεται και μέσο επικοινωνίας.¹⁴² Ο τεχνολογικός μετασχηματισμός προς την δικτύωση και την διάδραση σε μεγαλύτερη κλίμακα δεν ήταν προτεραιότητα για όλους. Για παράδειγμα, οι ομάδα της τεχνητής νοημοσύνης στο MIT δεν έβλεπε την σημασία της ευρύτερης δικτύωσης, αφού ο σκοπός της δεν ήταν να μιλάνε οι άνθρωποι μεταξύ τους αλλά με έξυπνες μηχανές. Από την άλλη πλευρά, η ίδια ομάδα δημιούργησε την ιδέα του ανοιχτού διαμοιρασμού, κυρίως ως αντίδραση στο σχετικά πολύπλοκο λόγω ασφάλειας MULTICS. Στα τέλη της δεκαετίας του 1960, δημιούργησαν ένα απλό σύστημα χρονοδιαμοιρασμού το ITS το οποίο αρχικά δεν έκανε έλεγχο πρόσβασης χρήστη και αποτέλεσε την φιλοσοφική πλατφόρμα για τις μελλοντικές κοινότητες ανοιχτού λογισμικού.¹⁴³

Το Skype είναι μια εφαρμογή VoIP με εκατομμύρια χρήστες από όλον τον κόσμο. Αρχικά ήταν για επικοινωνία από Η/Υ σε Η/Υ. Πλέον προσφέρει κλήσεις σε οποιοδήποτε μέρος του κόσμου, σε οποιοδήποτε δίκτυο τηλεφωνίας, σταθερής και κινητής, με χαμηλές χρεώσεις, Instant Messenger, δυνατότητα αποστολής SMS, αποστολής αρχείων και δυνατότητα βιντεοκλήσης. Διατίθεται σε εκδόσεις για Windows, Mac και Linux και για τις πλατφόρμες φορητών συσκευών Android, και iOS όπως επίσης και στην κονσόλα βιντεοπαιχνιδιών της Microsoft, Xbox One.

Student: Σπυρίδων Κοκοτός (inf2021098)

Οι καλλιτέχνες ήταν από τους πρώτους που χρησιμοποίησαν τον υπολογιστή ως εργαλείο για να δημιουργήσουν. Όμως η πιο σημαντική συνεισφορά τους στην εξέλιξη της χρήσης του υπολογιστή ήταν ότι τον χρησιμοποίησαν ως μέσο για το έργο τους. Για παράδειγμα, η Brenda Laurel με υπόβαθρο στις θεατρικές σπουδές αναγνωρίζει τη δυνατότητα των υπολογιστών να λειτουργούν ως σκηνή όπου οι χρήστες μπορούν να παίζουν ρόλους και να αυτοσχεδιάζουν. Αυτή η αντίληψη έρχεται να συμπληρώσει την αντίληψη του υπολογιστή ως εργαλείου που σχεδιάζεται με

¹⁴² fig:media-space

¹⁴³ Markoff (2005)

Figure 14: Screenshot_2022-11-17_19-07-58

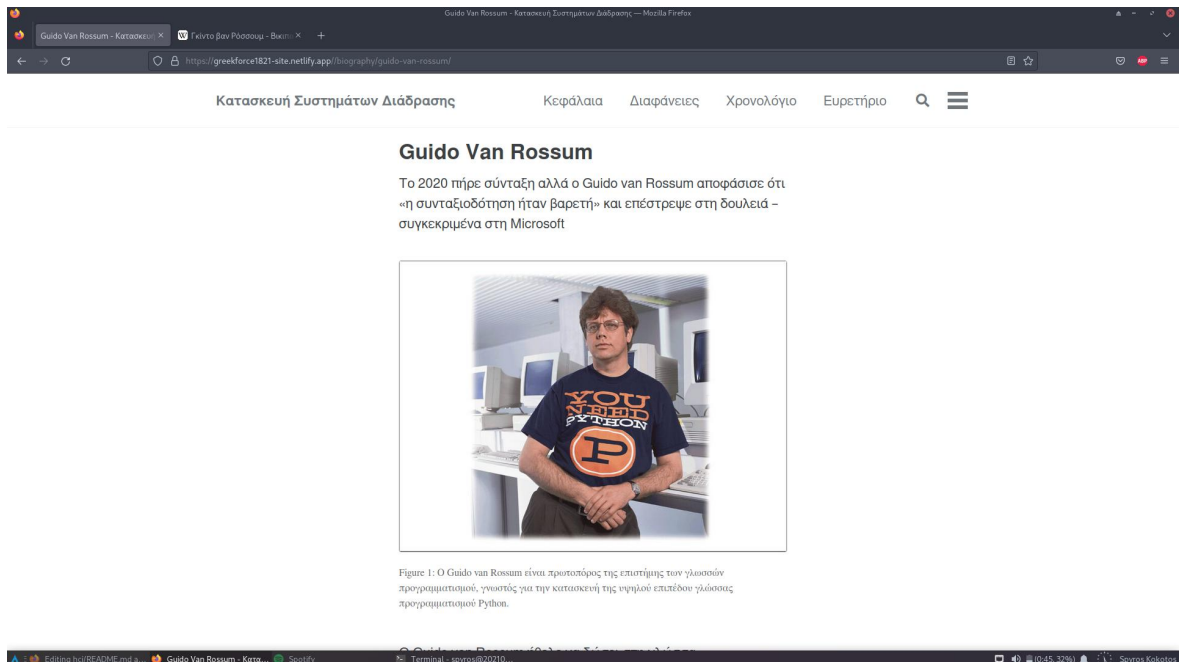


Figure 15: Screenshot_2022-11-19_11-29-28

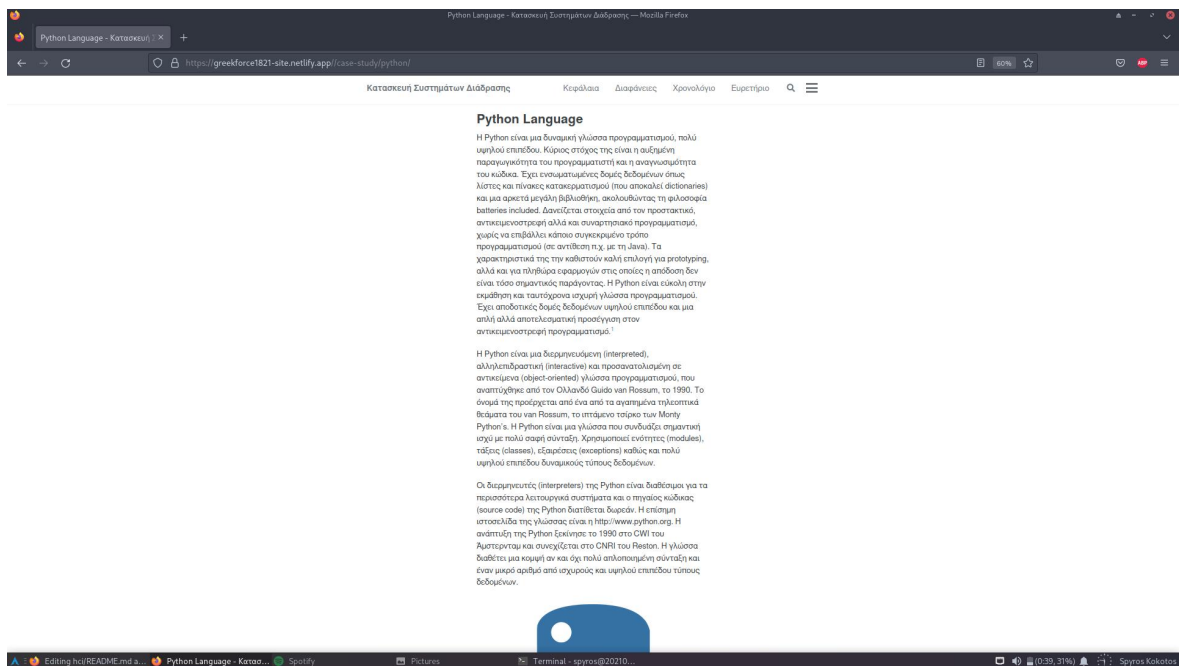


Figure 16: Screenshot_2022-11-19_11-30-49

10.1 Guido Van Rossum Biography

10.2 Python & Odroid-C4 Case-Study

11 Παρακάτω, παραθέτω τα link από το κάθε ένα αρχείο μαζί με τις φωτογραφίες που το συνοδεύουν:

- Προσωπικό site από το αποθετήριο μου: [Greekforce1821](#)
- Biography του προσωπικού αποθετηρίου: [Biography Guido Van Rossum](#)
- Case-Study του προσωπικού αποθετηρίου: [Case-Study Python](#)
- Αρχείο βιογραφίας από το αποθετήριο: [.md File #Biography Thumbnail](#)
- Αρχείο βιογραφίας από το αποθετήριο: [.md File #Biography](#)
- Αρχείο μελέτης περίπτωσης από το αποθετήριο: [.md File #Case-Study Thumbnail](#)
-

11.1 Αρχείο μελέτης περίπτωσης από το αποθετήριο: [.md File #Case-Study](#)

12 Φωτογραφίες που αξιοποιήθηκαν στην παραπάνω εργασία:

- Φωτογραφία thumbnail για την βιογραφία του Guido Van Rossum: [Thumbnail-Guido](#)
- Φωτογραφία για την βιογραφία του Guido Van Rossum: [Van-Rossum Picture](#)
- Φωτογραφία thumbnail για την μελέτη περίπτωσης της Python: [Thumbnail-Python](#)
- Φωτογραφία για την μελέτη περίπτωσης της Python #1: [Python #1](#)
- Φωτογραφία για την μελέτη περίπτωσης της Python #2: [Python #2](#)

- Φωτογραφία για την μελέτη περίπτωσης της Python #3: Python #3

-

12.1 Φωτογραφία για την μελέτη περίπτωσης της Python #4: Python #4

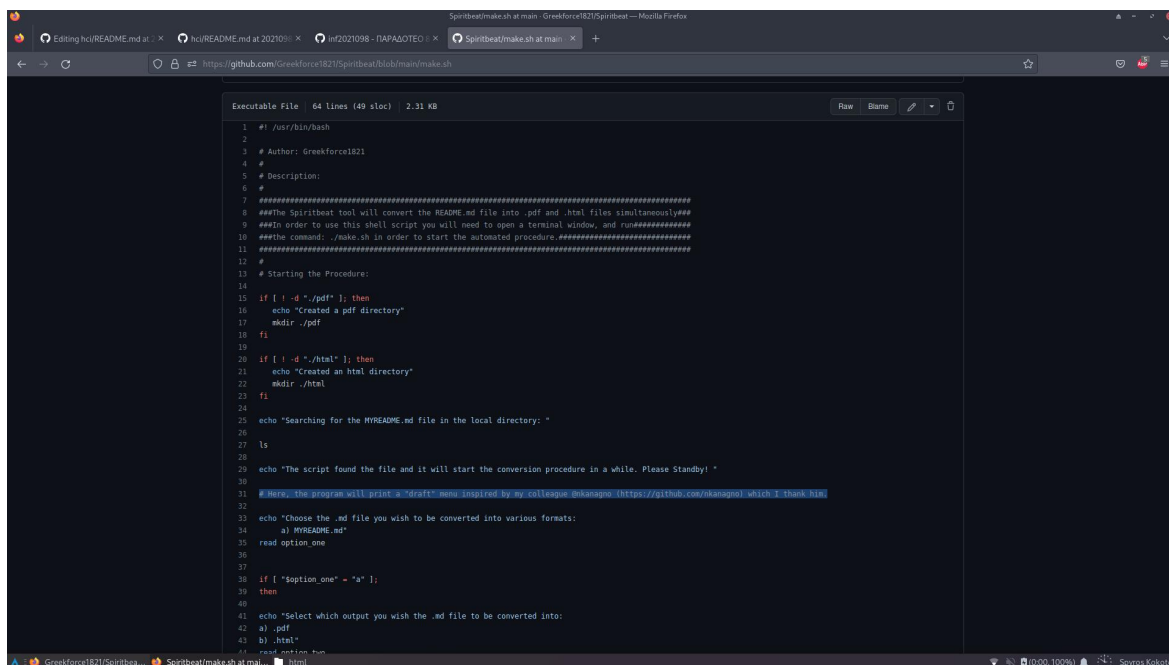
13 Πίνακας πνευματικών δικαιωμάτων σχετικά με την αξιοποίηση των πηγών και των φωτογραφιών:

Φωτογραφίες	Copyright Holders	License	Type of License
Guido Van Rossum #1	Φωτογραφία #1	© Michael Cavotta	CC BY-NC-ND 4.0
Guido Van Rossum #2	Φωτογραφία #2	© Dan Stroud	CC BY-SA 4.0
Guido Van Rossum #3	Φωτογραφία #3	© Unknown	CC BY-NC-ND 4.0
Python Case Study #1	Φωτογραφία #1	© The Python Software Foundation	GNU General Public License
Python Case Study #2	Φωτογραφία #2	© Chris Ried	Unsplash License
Python Case Study #3	Φωτογραφία #3	© Spyros Kokotos	CC BY-NC-ND 4.0
Python Case Study 34	Φωτογραφία #4	© Spyros Kokotos	CC BY-NC-ND 4.0

Άρθρο	Πηγή	Copyright Holder
Βιογραφικό σημείωμα Guido Van Rossum	https://bit.ly/3Xfnym8	Ελληνικό Μουσείο Πληροφορικής

14 Παραδοτέο 8ο: Command Line Exercise #2

Στο συγκεκριμένο παραδοτέο καλούμαστε να δημιουργήσουμε ένα δικό μας shell script το οποίο θα πρέπει να έχει σχέση με τις γνώσεις που αποκτήσαμε στα προηγούμενα παραδοτέα (βλέπε: Παραδοτέο 6 & Παραδοτέο 7). Σύμφωνα με τις οδηγίες του μαθήματος έπρεπε να επιλέξουμε και να υλοποιήσουμε μια οποιαδήποτε εργασία. Στην προκειμένη περίπτωση, επέλεξα να δημιουργήσω ένα script το οποίο θα περιλαμβάνει το pandoc και το texlive (που χρησιμοποίησα στο 6ο παραδοτέο για να κατασκευάσω το βιβλίο) για να μετατρέψω ένα οποιοδήποτε αρχείο Markdown (.md) σε PDF (.pdf) και σε HTML (.html) . Στην συγκεκριμένη εργασία, αξιοποίησα ένα μικρό κομμάτι από το script που έγραψε ο συμφοιτητής: Νίκος Αναγνωστόπουλος τον οποίο ευχαριστώ πολύ για την συμβολή του!



```
Executable File 64 Lines (49 slo) 2.31 KB
1 #!/usr/bin/bash
2
3 # Author: Greekforce1821
4 #
5 # Description:
6 #
7 #####
8 #####The Spiritbeat tool will convert the README.md file into .pdf and .html files simultaneously#####
9 #####In order to use this shell script you will need to open a terminal window, and run#####
10 #####the command: ./make.sh in order to start the automated procedure.#####
11 #####
12 #
13 # Starting the Procedure:
14
15 if [ ! -d ".pdf" ]; then
16     echo "Created a pdf directory"
17     mkdir .pdf
18 fi
19
20 if [ ! -d ".html" ]; then
21     echo "Created an html directory"
22     mkdir .html
23 fi
24
25 echo "Searching for the MYREADME.md file in the local directory: "
26
27 ls
28
29 echo "The script found the file and it will start the conversion procedure in a while. Please Standby! "
30
31 #Here, the program will print a "draft" message inspired by my colleague @kanago (https://github.com/kanagoi/which-I-thank-him)
32
33 echo "Choose the .md file you wish to be converted into various formats:
34 a) MYREADME.md"
35 read option_one
36
37 if [ "$option_one" = "a" ];
38 then
39
40
41 echo "Select which output you wish the .md file to be converted into:
42 a) .pdf
43 b) .html"
44 read option_two
45
```

Figure 17: Screenshot_2022-11-26_14-59-58

14.1 Βήμα #1: The Creation of the Repository

Στο συγκεκριμένο βήμα, δημιούργησα ένα ξεχωριστό αποθετήριο στο προφίλ μου στο Github το οποίο το ονομάσα ως: Spiritbeat. Στην συνέχεια, δημιούργησα ένα README.md αρχείο (κενό για αρχή) και ένα LICENSE.md.

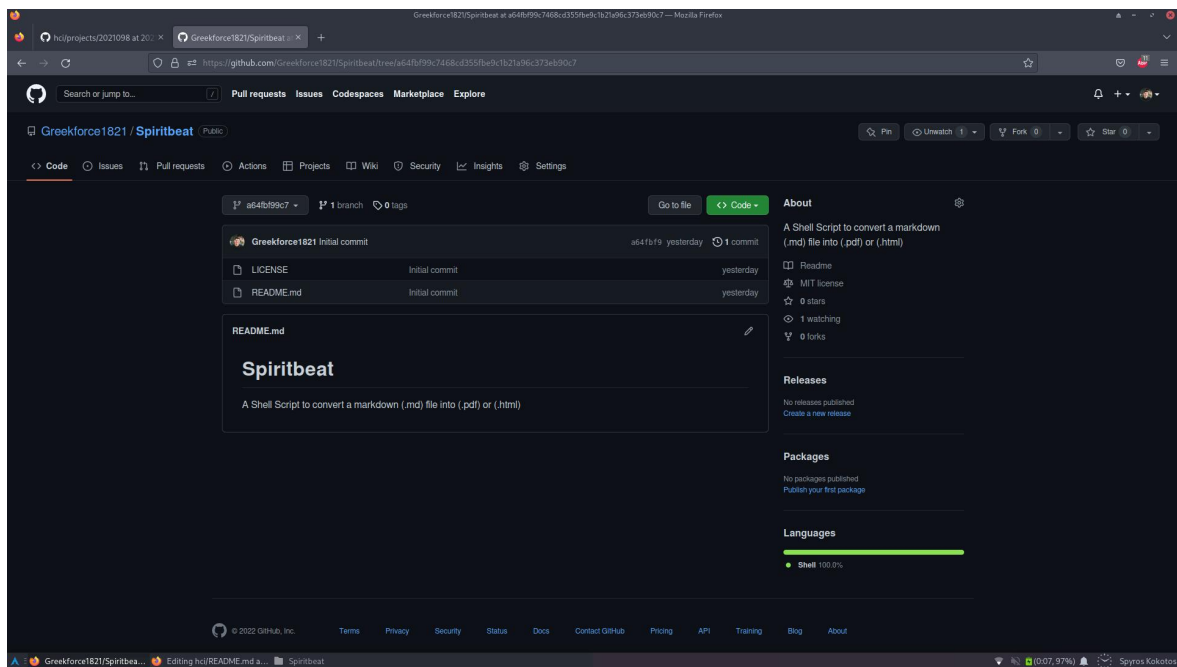


Figure 18: Screenshot_2022-11-26_14-50-13

14.2 Βήμα #2: Cloning the Repository Locally

Στο παραπάνω βήμα, έπρεπε να κάνω clone τοπικά το αποθετήριο με την εντολή: `git clone https://github.com/GreeKforce1821/Spiritbeat` για να μπορέσω να κάνω `git push` μόλις ολοκληρωθεί το script. Στην συνέχεια, δημιούργησα ένα αρχείο `make.sh` στο αποθετήριο τοπικά και ξεκίνησα την υλοποίηση του script.

14.3 Βήμα #3: Filling out the Script File With Commands

Στο συγκεκριμένο βήμα, έκανα τις απαραίτητες αλλαγές, έσβησα, έγραψα, έσβησα και ξαναέγραψα κώδικα στο scriptάκι το οποίο πλέον είναι σε θέση να αναζητήσει, να ελέγξει, να επιλέξει και να μετατρέψει με βάση τις επιλογές που θα δώσει ο χρήστης στο πρόγραμμα. Συγκεκριμένα, το script ψάχνει στο συγκεκριμένο directory όλα τα αρχεία και τα εκτυπώνει στο τερματικό, στην συνέχεια ζητάει από τον χρήστη να επιλέξει το αρχείο (.md) που θέλει να μετατρέψει και στην συνέχεια εμφανίζει μενού με την μετατροπή σε (.pdf) & (.html)

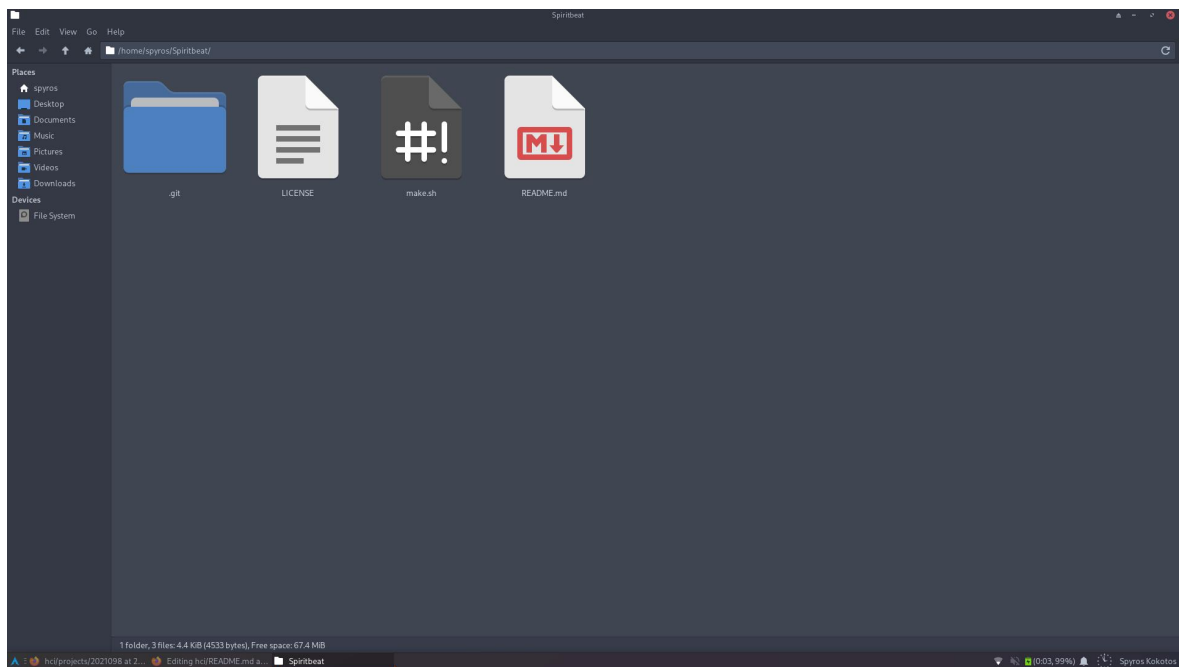


Figure 19: Screenshot_2022-11-26_14-55-01

14.4 Converting a markdown (.md) file into PDF (.pdf)

```
[spyros@2021098 Spiritbeat]$ ls
LICENSE MYREADME.md README.md html make.sh pdf
[spyros@2021098 Spiritbeat]$ ./make.sh
Searching for the MYREADME.md file in the local directory:
LICENSE MYREADME.md README.md html make.sh pdf
The script found the file and it wil start the conversion procedure in a while. Please Standby!
Choose the .md file you wish to be converted into various formats:
  a) MYREADME.md
a
Select which output you wish the .md file to be converted into:
a) .pdf
b) .html
a
Initiating the conversion from .md to .pdf file. Please St
```



14.5 Converting a markdown (.md) file into HTML (.html)

```
[spyros@2021098 Spiritbeat]$ ./make.sh
Searching for the MYREADME.md file in the local directory:
LICENSE MYREADME.md MYREADME.pdf README.md html make.sh pdf
The script found the file and it wil start the conversion procedure in a while. Please Standby!
Choose the .md file you wish to be converted into various formats:
  a) MYREADME.md
a
Select which output you wish the .md file to be converted into:
a) .pdf
b) .html
█
```



14.6 Βήμα #4: Checking the Results of the Conversion (PDF) & (HTML)

Στο συγκεκριμένο βήμα, δημιουργήθηκαν από το script δύο φάκελοι, ένας με ονομασία: pdf και ένας html. Μόλις ολοκληρώθηκε η μετατροπή, δημιουργήθηκε το αρχείο MYREADME.pdf στον φάκελο pdf και αντίστοιχα το αρχείο MYREADME.html στον φάκελο html και τσέκαρα ότι δεν υπήρξαν λάθη κατά την μετατροπή. Παρακάτω, επισυνάπτω ένα κομμάτι από το pdf και ένα από το html.

14.7 PDF Results

14.8 HTML Results

14.9 Βήμα #5: Filling out the README.md File with Useful Information

Στο παραπάνω βήμα, απλά, πρόσθεσα κάτι τελευταίες σημειώσεις στο README.md αρχείο, σχετικά με τι prerequisites πρέπει να έχει ο χρήστης εγκατεστημένα στο σύστημά του προκειμένου να τρέξει το script που δημιουργήσα.

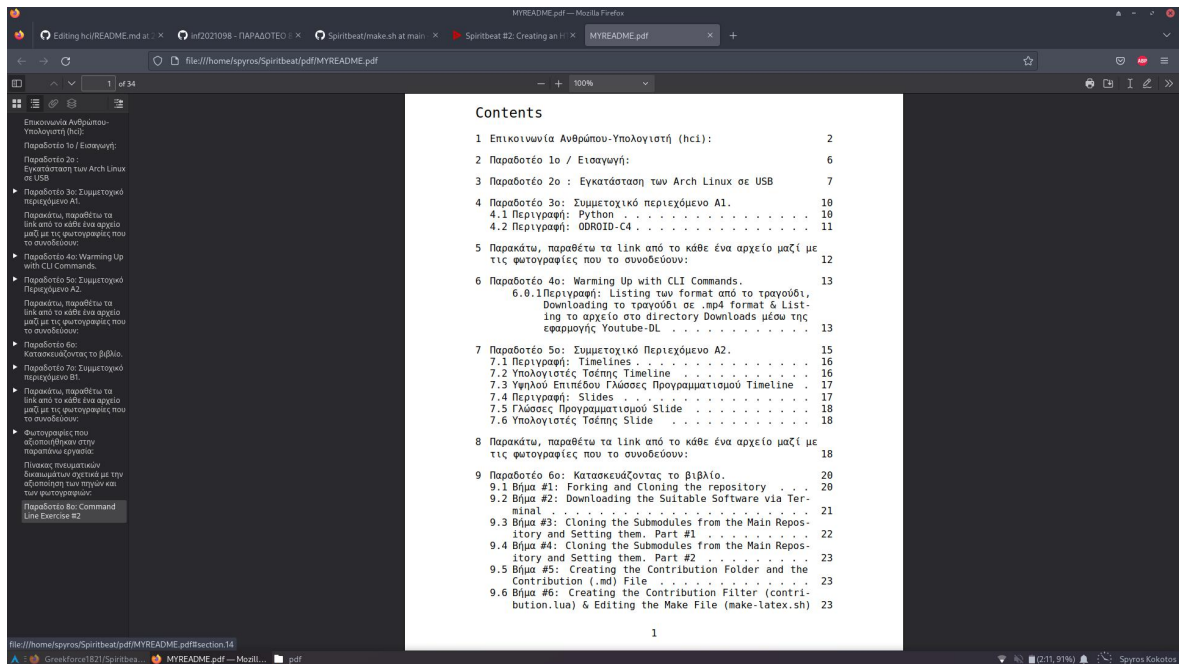


Figure 20: Screenshot_2022-11-26_15-17-58

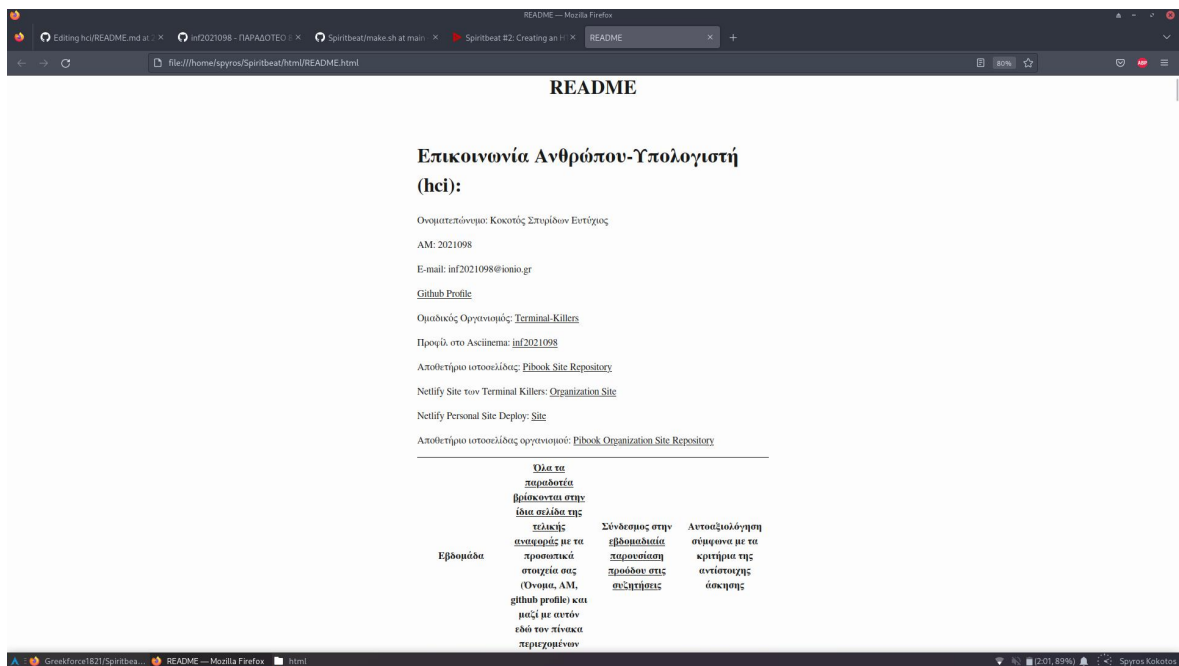


Figure 21: Screenshot_2022-11-26_15-20-13

15 Πίνακας των διάφορων link περί εργασίας:

Links	Platform	Extension
Shell Script	GitHub	.sh
MYREADMEEXAMPLE.md	GitHub	.md
Converted PDF File	GitHub	.pdf
Converted HTML File	GitHub	.html
Guides For Users - How To Use	GitHub	.md
SpiritBeat Shell Script Repository	GitHub	-
PDF Conversion Using Terminal	Asciinema	.cast
HTML Conversion Using Terminal	Asciinema	.cast