# Επικοινωνία Ανθρώπου-Υπολογιστή (hci):

Ονοματεπώνυμο: Κοκοτός Σπυρίδων Ευτύχιος

ΑΜ: 2021098

E-mail: inf2021098@ionio.gr

[Github Profile](https://github.com/Greekforce1821)

Ομαδικός Οργανισμός: [Terminal-Killers](https://github.com/Terminal-Killers)

Προφίλ στο Asciinema: [inf2021098](https://asciinema.org/~inf2021098)

Αποθετήριο ιστοσελίδας: [Pibook Site Repository](https://github.com/Greekforce1821/site)

Netlify Site των Terminal Killers: [Organization Site](https://terminal-killers-site.netlify.app)

Netlify Personal Site Deploy: [Site](https://greekforce1821-site.netlify.app/)

Αποθετήριο ιστοσελίδας οργανισμού: [Pibook Organization Site Repository](https://github.com/Terminal-Killers/site)

| Εβδομάδα | [Όλα τα παραδοτέα βρίσκονται στην ίδια σελίδα της τελικής αναφοράς](https://courses-ionio.github.io/help/deliverables/) με τα προσωπικά στοιχεία σας (Όνομα, ΑΜ, github profile) και μαζί με αυτόν εδώ τον πίνακα περιεχομένων | Σύνδεσμος στην [εβδομαδιαία παρουσίαση προόδου στις συζητήσεις](https://github.com/courses-ionio/help/discussions/categories/show-and-tell) | Αυτοαξιολόγηση σύμφωνα με τα κριτήρια της αντίστοιχης άσκησης |
| --- | --- | --- | --- |
| 1 | [Δημιουργία ομάδας](https://github.com/courses-ionio/hci/discussions/1794) + [Φορκ και δημιουργία σελίδας τελικής αναφοράς](https://courses-ionio.github.io/help/guide/), [προσθήκη πίνακα περιεχομένων](https://raw.githubusercontent.com/courses-ionio/hci/master/README.md), [συγγραφή της εισαγωγής](https://courses-ionio.github.io/help/intro/), αποστολή της εισαγωγής [για σχολιασμό στην συζήτηση](https://github.com/courses-ionio/help/discussions/categories/show-and-tell) και καταγραφή του συνδέσμου συζήτησης δίπλα –> | [Συζήτηση](https://github.com/courses-ionio/help/discussions/830), [Link της ομάδας](https://github.com/Terminal-Killers), [Μέλη της ομάδας](https://github.com/courses-ionio/hci/discussions/1794#discussioncomment-3815476) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 2 | Άσκηση γραμμής εντολών (linux install) | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1021) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 3 | Συμμετοχικό περιεχόμενο A1 | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1185), [PR #1](https://github.com/Terminal-Killers/_gallery/pull/1), [PR #2](https://github.com/Terminal-Killers/images/pull/1), [Organization Base Site](https://terminal-killers-site.netlify.app/), [Link of Work #1](https://terminal-killers-site.netlify.app//gallery/python/) & [Link of Work #2](https://terminal-killers-site.netlify.app//gallery/odroid-c4/) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 4 | Άσκηση γραμμής εντολών (warm up cli) | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1290) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 5 | Συμμετοχικό περιεχόμενο A2 | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1461), [PR #1](https://github.com/Terminal-Killers/site/pull/1), [PR #2](https://github.com/Terminal-Killers/site/pull/2), [Link of Work #1](https://terminal-killers-site.netlify.app//slides/computers-in-pockets/), [Link of Work #2](https://terminal-killers-site.netlify.app//slides/programming/), [Link of Work #3](https://terminal-killers-site.netlify.app//timeline/computers-in-a-pocket/) & [Link of Work #4](https://terminal-killers-site.netlify.app//timeline/programming-languages/) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 6 | Κατασκευή του βιβλίου Α | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1586), [Contribution Point](https://github.com/Greekforce1821/kallipos/blob/master/contribution/skypech08.md), [Contribution Photo](https://github.com/Greekforce1821/kallipos/blob/master/Picture%20of%20my%20personal%20contribution%20within%20the%20book/Picture-of-my-contribution.jpg), [.EPUB file](https://github.com/Greekforce1821/kallipos/blob/master/epub/book.epub), [.PDF file](https://github.com/Greekforce1821/kallipos/blob/master/book/book.pdf), [Contribution Filter](https://github.com/Greekforce1821/kallipos/blob/master/contribution.lua) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 7 | Συμμετοχικό περιεχόμενο B1 | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1675), [Link of Case-Study](https://greekforce1821-site.netlify.app///case-study/python/) & [Site Link](https://greekforce1821-site.netlify.app/) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 8 | Άσκηση γραμμής εντολών | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1760), [Spiritbeat Repository](https://github.com/Greekforce1821/Spiritbeat), [Converted PDF](https://github.com/Greekforce1821/Spiritbeat/blob/main/pdf/MYREADME.pdf), [Converted HTML](https://github.com/Greekforce1821/Spiritbeat/blob/main/html/README.html), [Shell Script](https://github.com/Greekforce1821/Spiritbeat/blob/main/make.sh), [Converted ODT](https://github.com/Greekforce1821/Spiritbeat/blob/main/odt/MYREADME.odt), [Converted TXT](https://github.com/Greekforce1821/Spiritbeat/blob/main/txt/MYREADME.txt), [Converted IPYNB](https://github.com/Greekforce1821/Spiritbeat/blob/main/ipynb/MYREPORT.ipynb) & [Converted DOCX](https://github.com/Greekforce1821/Spiritbeat/blob/main/docx/MYREADME.docx) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 9 | Συμμετοχικό περιεχόμενο B2 | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1817), [Link of Biography](https://greekforce1821-site.netlify.app//biography/guido-van-rossum/) & [Site Link](https://greekforce1821-site.netlify.app/) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 10 | Άσκηση γραμμής εντολών | [Link Συζήτησης](https://github.com/courses-ionio/help/discussions/1880) & [Asciinema Link](https://asciinema.org/a/544020) | Επιτυχής ολοκλήρωση του παραδοτέου εντός προθεσμίας ☑️ (100% Success) |
| 11 | Κατασκευή του βιβλίου Β |  |  |
| 12 | Τελική αναφορά\* |  |  |

# Παραδοτέο 1ο / Εισαγωγή:

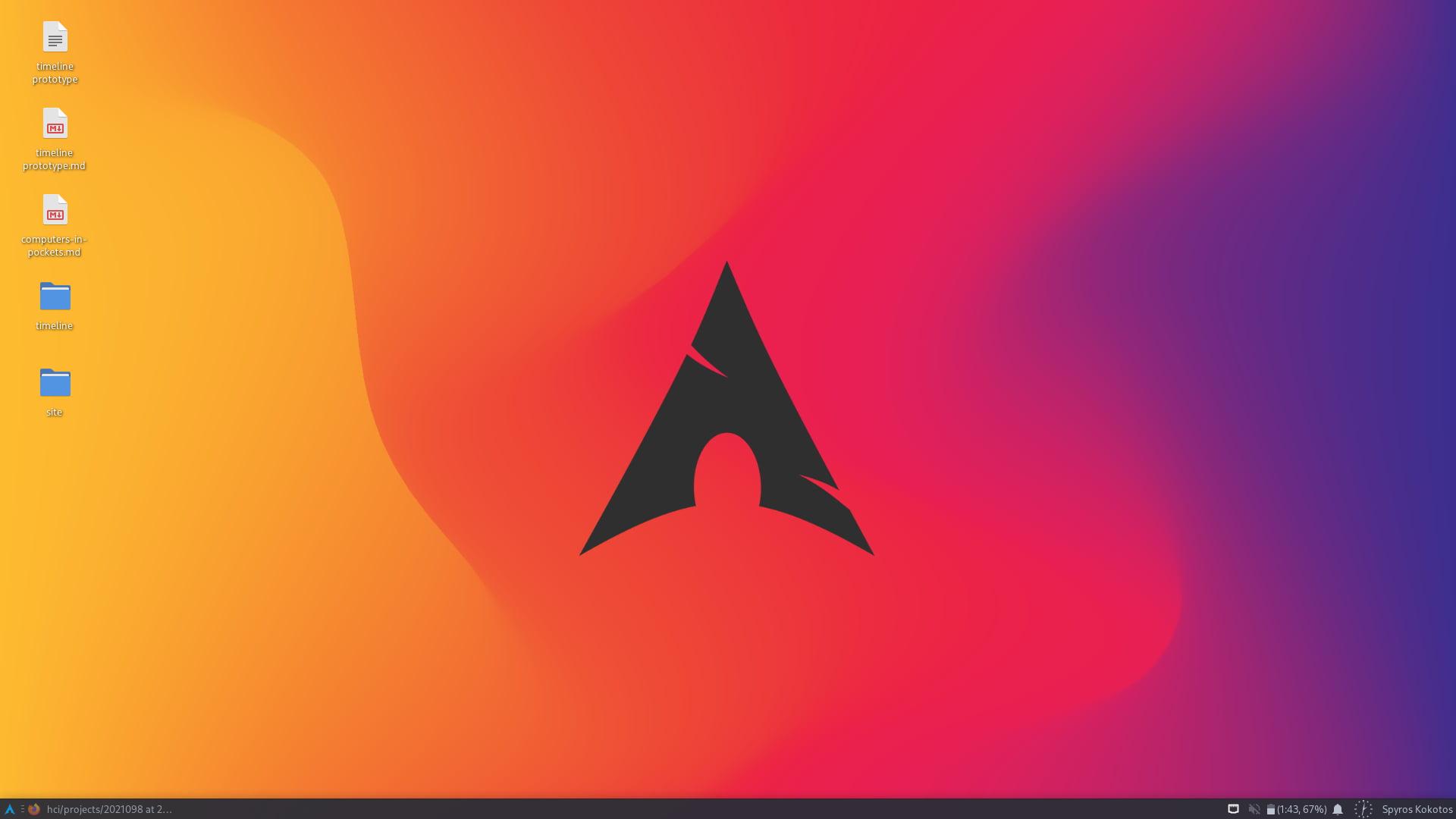
Παρακάτω παραθέτω τους προσωπικούς μου στόχους για το μάθημα “Επικοινωνία Ανθρώπου - Υπολογιστή” :

1. Η ορθή κατανόηση των κανόνων που θεσπίζουν το μάθημα.
2. 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.)
3. Εμπλουτισμό γνώσεων με τις διάφορες εγκαταστάσεις ανοιχτού λογισμικού διανομών Linux όπως η διανομή “Arch”. (π.χ. minimal distributions χωρίς το υπάρχων γραφικό περιβάλλον)
4. 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.
5. Τέλος, να αποκτήσω μέσα από τα παραπάνω γνώσεις προγραμματιστικές οι οποίες θα μου φανούν χρήσιμες για την ακαδημαϊκή και την επαγγελματική μου ζωή.

# Παραδοτέο 2ο : Εγκατάσταση των Arch Linux σε USBimage

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:

1. Πραγματοποιηθήκε 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](https://wiki.archlinux.org/title/installation_guide).
2. 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.
3. 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. H διαδικασία αυτή επιτεύχθει με την παρακάτω εντολή: usermod -aG sudo spyros
4. Επιπροσθέτως, επιτεύχθει και η αλλαγή του Timezone στο σωστό μέρος σύμφωνα με την χώρα που βρισκόμαστε. Συγκεγκριμένα, 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.
5. Τέλος, εγκατέστησα τον [GRUB](https://wiki.archlinux.org/title/GRUB) Βoot 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)



Screenshot\_2022-11-08\_14-01-35

Προβλήματα:

1. Tο 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](https://youtu.be/SqA72or_-GE)) για να επιλύσω το θέμα με τα partition στο USB, πόσα GB παραδείγματος χάριν θα αφιερώσω στο κάθε ένα και για ποιόν λόγο.
2. 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.
3. 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](https://github.com/courses-ionio/help/discussions/1069#discussioncomment-3887459).

Παρατηρήσεις:

1. 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](https://manjaro.org/) και έχει αλλάξει γενικά η δομή των εντολών εκτέλεσης στο τερματικό, παραδείγματος χάριν στο [ubuntu](https://ubuntu.com/) 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:

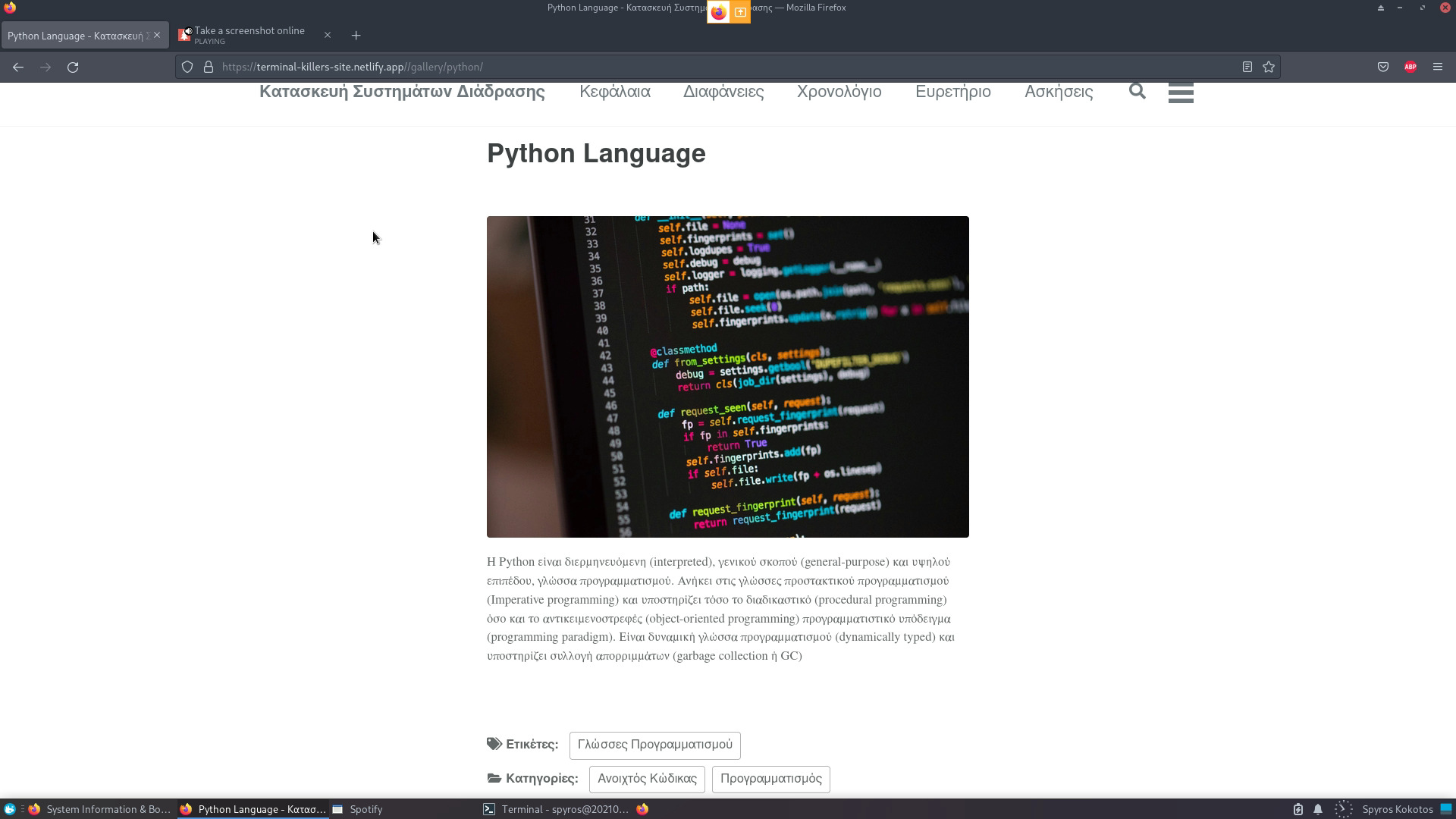
# Παραδοτέο 3ο: Συμμετοχικό περιεχόμενο Α1.

Στο συγκεκριμένο παραδοτέο, καλούμαστε να συμβάλλουμε ομαδικά και ατομικά στην ανάπτυξη του ήδη υπάρχοντος [site](https://pibook.epidro.me/). Σαν αρχηγός του οργανισμού [Terminal-Killers](https://github.com/Terminal-Killers) δημιούργησα συγκεκριμένο section για συζήτηση περί [επιλογής θεμάτων](https://github.com/Terminal-Killers/Terminal-Killers/discussions/11) , [προβλημάτων που ίσως προκύψουν στην πορεία](https://github.com/Terminal-Killers/Terminal-Killers/discussions/14) και τέλος [οδηγίες χρήσης](https://github.com/Terminal-Killers/Terminal-Killers/discussions/12) 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](https://www.python.org/) και στον ολοκαίνουργιο υπολογιστή τσέπης [ODROID-C4](https://www.hardkernel.com/shop/odroid-c4/). Αναλυτικότερες πληροφορίες στην επόμενη παράγραφο.

## Περιγραφή: Python

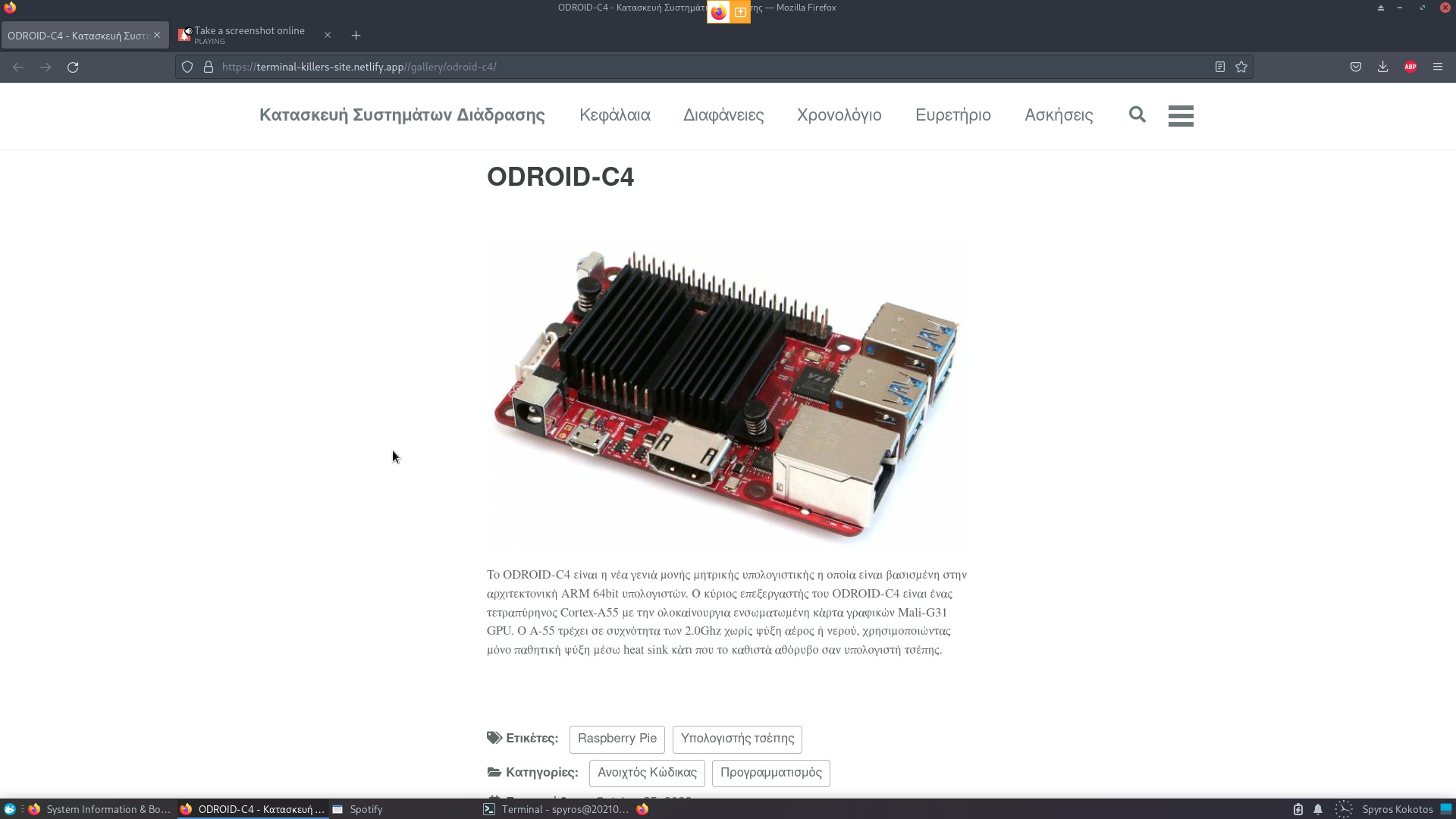
Η Python είναι διερμηνευόμενη (interpreted), γενικού σκοπού (general-purpose) και υψηλού επιπέδου, γλώσσα προγραμματισμού. Ανήκει στις γλώσσες προστακτικού προγραμματισμού (Imperative programming) και υποστηρίζει τόσο το διαδικαστικό (procedural programming) όσο και το αντικειμενοστρεφές (object-oriented 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 δημιούργησα ένα [project](https://github.com/ThePythonologists/Eudoxus-Api) για το μάθημα Εφαρμοσμένος Προγραμματισμός με Python του Β’ εξαμήνου.



image(7)

## Περιγραφή: ODROID-C4

Το ODROID-C4 είναι η νέα γενιά μονής μητρικής υπολογιστικής η οποία είναι βασισμένη στην αρχιτεκτονική ARM 64bit υπολογιστών. Ο κύριος επεξεργαστής του ODROID-C4 είναι ένας τετραπύρηνος Cortex-A55 με την ολοκαίνουργια ενσωματωμένη κάρτα γραφικών Mali-G31 GPU. O 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.



image(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): [Greekforce1821](https://greekforce1821-site.netlify.app/)
* Άρθρο για την Python στο προσωπικό αποθετήριο (test site): [Python](https://greekforce1821-site.netlify.app/gallery/python/)
* Άρθρο για τον ODROID-C4 στο προσωπικό αποθετήριο (test site): [ODROID-C4](https://greekforce1821-site.netlify.app/gallery/odroid-c4/)
* Φωτογραφία της [Python](https://github.com/Greekforce1821/images/blob/master/python.jpg) σε πραγματική ανάλυση και σε [resized ανάλυση](https://github.com/Greekforce1821/images/blob/master/python-thumb.jpg) στο προσωπικό αποθετήριο (test site)
* Φωτογραφία του [ODROID-C4](https://github.com/Greekforce1821/images/blob/master/odroid_c4.jpg) σε πραγματική ανάλυση και σε [resized ανάλυση](https://github.com/Greekforce1821/images/blob/master/odroid_c4-thumb.jpg) στο προσωπικό αποθετήριο (test site)
* Το site από το αποθετήριο του οργανισμού: [Terminal-Killers](https://terminal-killers-site.netlify.app/)
* Το άρθρο μου για την Python στο αποθετήριο του οργανισμού: [Python Organization](https://terminal-killers-site.netlify.app//gallery/python/)
* Το άρθρο μου για τον ODROID-C4 στο αποθετήριο του οργανισμού: [ODROID-C4 Organization](https://terminal-killers-site.netlify.app//gallery/odroid-c4/)
* Φωτογραφία της [Python](https://github.com/Terminal-Killers/images/blob/master/python.jpg) σε πραγματική ανάλυση και σε [resized ανάλυση](https://github.com/Terminal-Killers/images/blob/master/python-thumb.jpg)
* Φωτογραφία του [ODROID-C4](https://github.com/Terminal-Killers/images/blob/master/odroid_c4.jpg) σε πραγματική ανάλυση και σε [resized ανάλυση](https://github.com/Terminal-Killers/images/blob/master/odroid_c4-thumb.jpg)

# Παραδοτέο 4o: 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](https://github.com/epidrome/dokey#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](https://flashgot.net/), [YTD Video Converter](http://www.ytddownloader.com/) κα.) για τους browser μας.

### Περιγραφή: Listing των format από το τραγούδι, Downloading το τραγούδι σε .mp4 format & Listing το αρχείο στο directory Downloads μέσω της εφαρμογής [Youtube-DL](https://github.com/ytdl-org/youtube-dl)

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

1. Άνοιξα το τερματικό εντολών και μέσω της εντολής [curl](https://curl.se/): 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](https://github.com/ytdl-org/youtube-dl)
2. Άνοιξα το [YouTube](https://youtube.com), 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](https://www.youtube.com/watch?v=YYOKMUTTDdA).
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.
4. Παρακάτω, εφόσον τσέκαρα ότι το βίντεο υποστηρίζει .mp4 format έτρεξα την εντολή: youtube-dl -f mp4 https://www.youtube.com/watch?v=YYOKMUTTDdA προκειμένου να κατεβάσω το συγκεκριμένο τραγούδι στον υπολογιστή μου. Η παράμετρος -f ισούται με το format που έχουμε επιλέξει.
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 ‘σωσμένο’ 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](https://aur.archlinux.org/) το οποίο είναι σαν το GitHub, αλλά, εκεί ανεβαίνουν πακέτα από τους χρήστες, άρα θεωρείται ένα user program repository. Eν συνέχεια, κατέβασα τα αρχεία από το [αποθετήριο](https://aur.archlinux.org/packages/curl-http3-msquic), έκανα αλλαγή στο directory και έκανα compile μέσω του τερματικού με την εντολή: makepkg -s προκειμένου να εγκατασταθεί στο σύστημα το curl.

# Παραδοτέο 5ο: Συμμετοχικό Περιεχόμενο Α2.

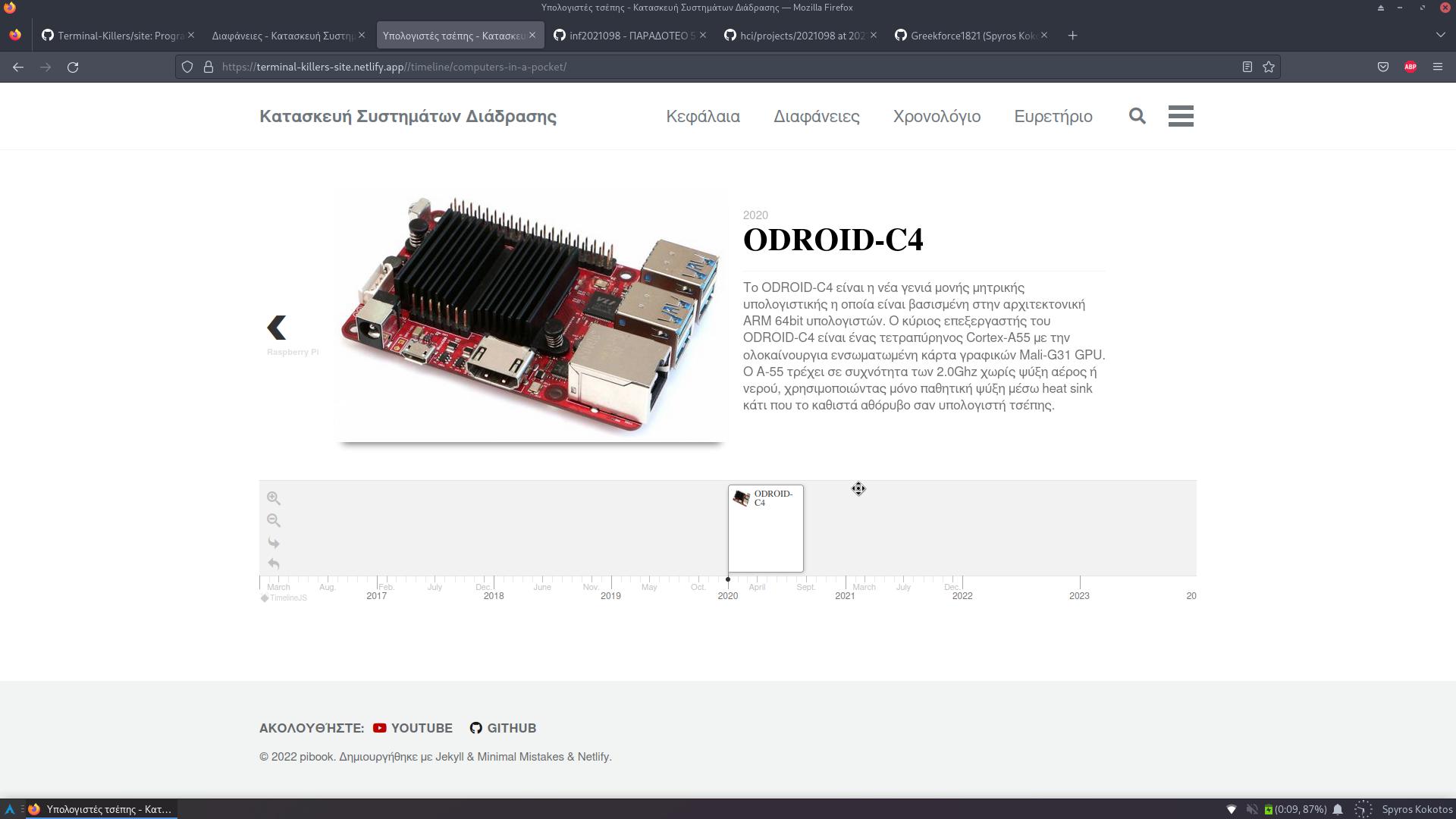
Στο συγκεκριμένο παραδοτέο, καλούμαστε να συμβάλλουμε ομαδικά και ατομικά για ακόμα μια φορά στην ανάπτυξη του υπάρχοντος [site](https://pibook.epidrome.me). Σαν αρχηγός του οργανισμού [Terminal-Killers](https://github.com/Terminal-Killers) έλαβα υπόψιν τα [προβλήματα που προέκυψαν στην πορεία](https://github.com/Terminal-Killers/Terminal-Killers/discussions/14) 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 & ODROID-C4) και τα ενσωμάτωσα σε δύο διαφάνειες (slides) & σε δύο χρονολόγια (timelines). Αναλυτικότερες λεπτομέρειες στην επόμενη παράγραφο.

## Περιγραφή: Timelines

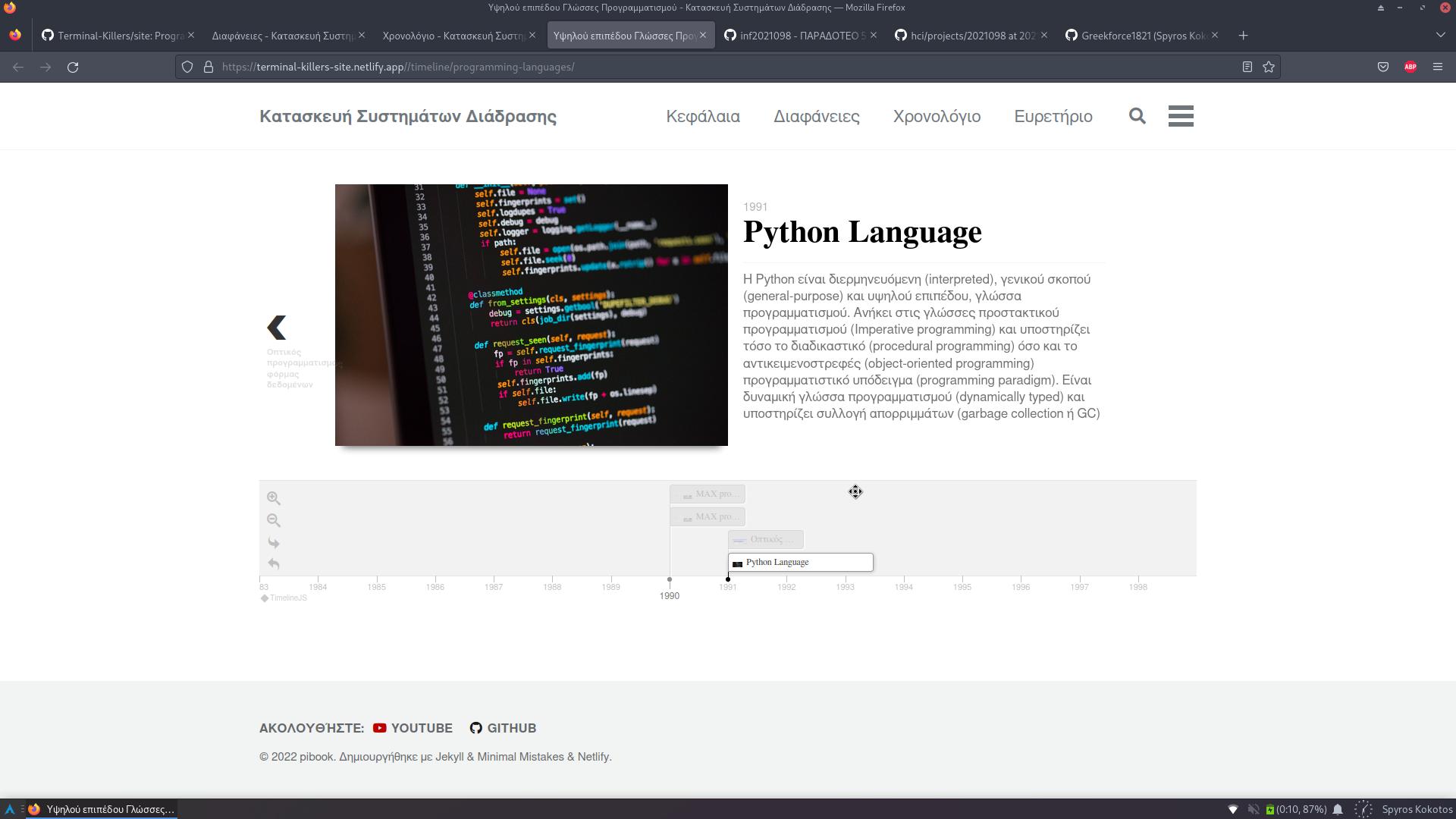
Στο συγκεκριμένο παραδοτέο, αξιοποίησα τις προηγούμενες μου γνώσεις και δημιούργησα δύο timelines σχετικές με τις Υψηλού επιπέδου γλώσσσες προγραμματισμού και αντίστοιχα με τους Υπολογιστές τσέπης. Δημιούργησα δύο αρχεία (.md) με κωδικές ονομασίες: computers-in-pockets.md και programming-languages.md 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](https://github.com/Terminal-Killers/site). Αφού έγινε deploy τσέκαρα ότι εμφανίστηκαν σωστά τα παραπάνω χρονολόγια στο [organization site](https://terminal-killers-site.netlify.app/timeline) και προχώρησα στο επόμενο βήμα.

## Υπολογιστές Τσέπης Timeline



Screenshot\_2022-11-07\_17-31-50

## Υψηλού Επιπέδου Γλώσσες Προγραμματισμού Timeline



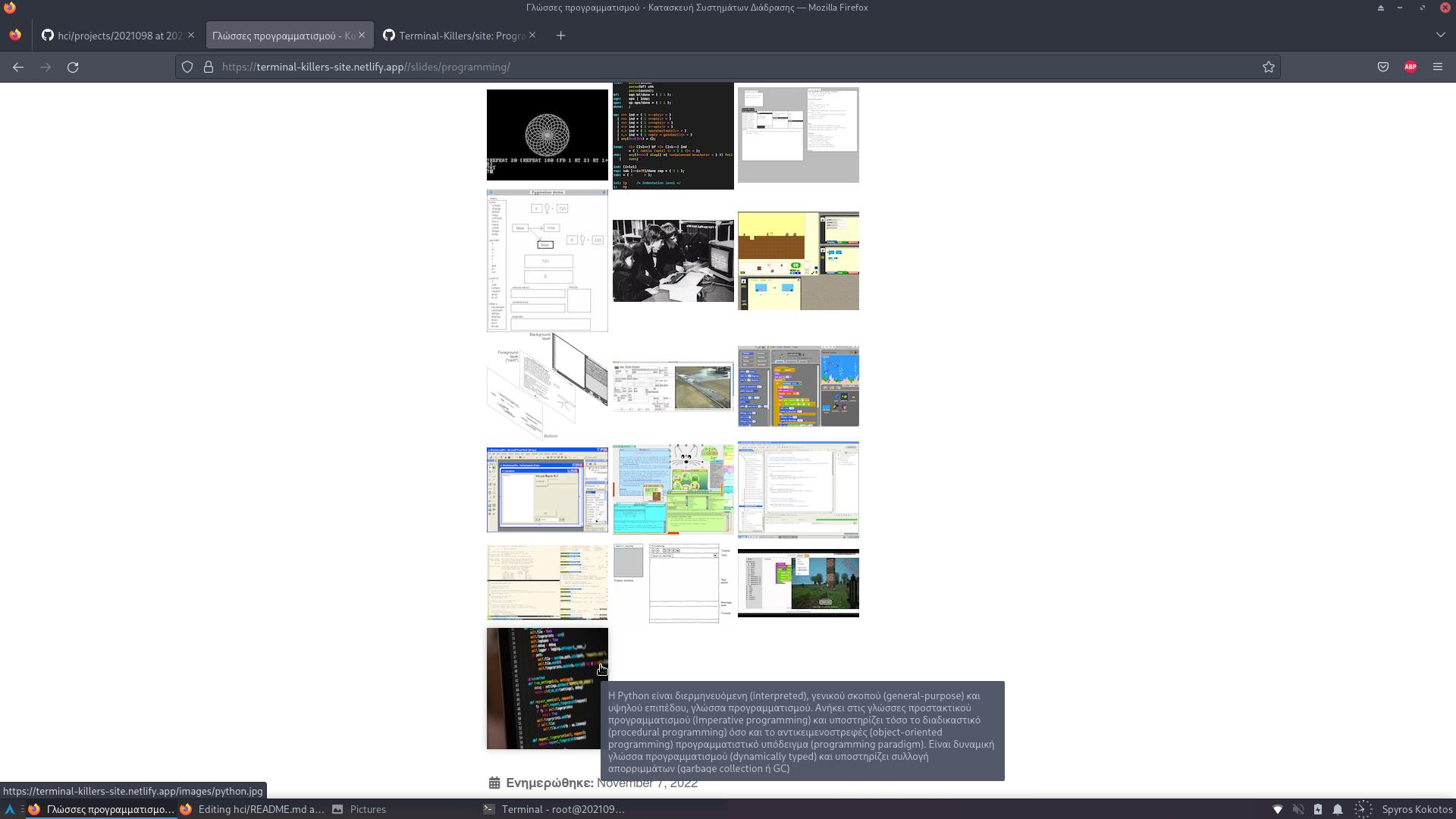
Screenshot\_2022-11-07\_17-32-09

## Περιγραφή: Slides

Στο συγκεκριμένο παραδοτέο, αξιοποίησα τις προηγούμενες μου γνώσεις και δημιούργησα δύο slides σχετικά με τις Υψηλού επιπέδου γλώσσσες 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](https://github.com/Terminal-Killers/site). Αφού έγινε deploy τσέκαρα ότι εμφανίστηκαν σωστά οι παραπ στο [organization site](https://terminal-killers-site.netlify.app/slides) και προχώρησα στο επόμενο βήμα.

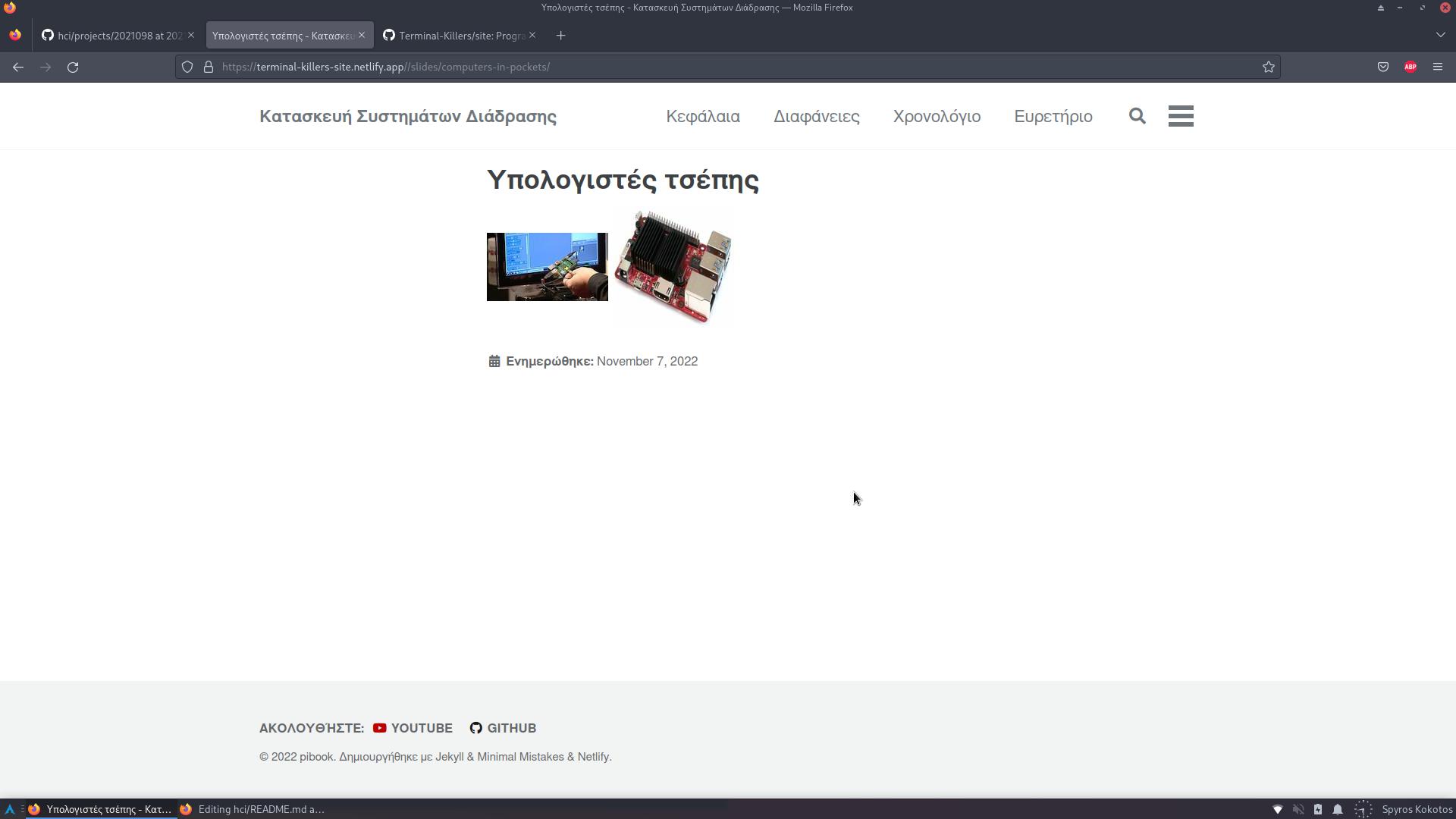
Στην αναφορά μου στο section των discussion και συγκεκριμένα στο: [Discussion #1461](https://github.com/courses-ionio/help/discussions/1461) αναφέρω για την επεξήγηση του αστερίσκου(\*) . Εφόσον έχω αναλάβει να εκπονήσω αυτήν την εργασία, επέλεξα να ασχοληθώ με μια συγκεκριμένη γλώσσα προγραμματισμού 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

## Γλώσσες Προγραμματισμού Slide



Screenshot\_2022-11-07\_18-41-57

## Υπολογιστές Τσέπης Slide



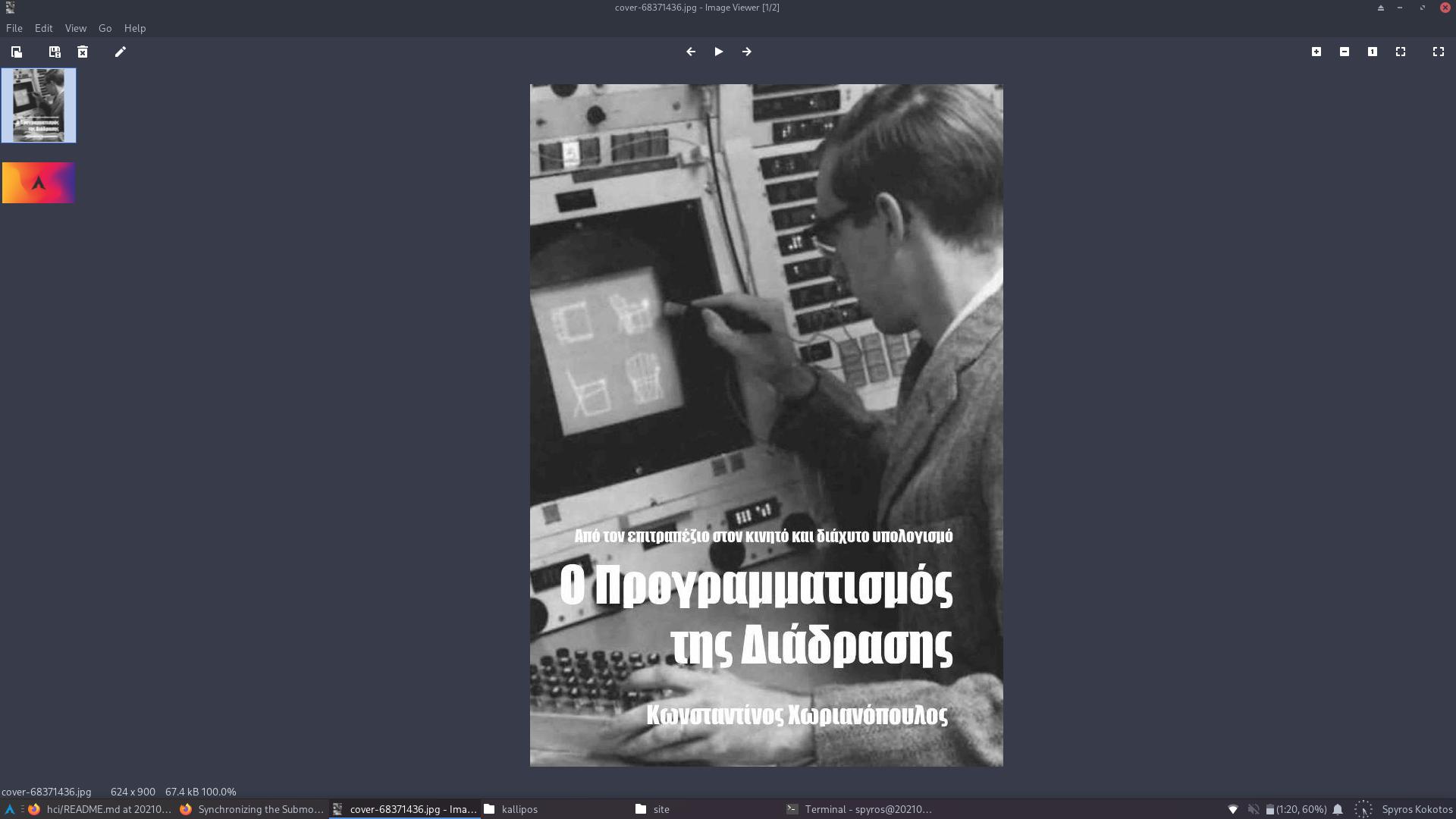
Screenshot\_2022-11-07\_18-43-32

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

* Προσωπικό site από το αποθετήριο μου (test site): [Greekforce1821](https://greekforce1821-site.netlify.app/)
* Timeline του προσωπικού αποθετηρίου (test site): [Timeline Greekforce1821](https://greekforce1821-site.netlify.app/timeline)
* Timeline #1 του προσωπικού αποθετηρίου (test site): [TImeline #1 Greekforce1821](https://greekforce1821-site.netlify.app/timeline/programming-languages/)
* Timeline #2 του προσωπικού αποθετηρίου (test site): [Timeline #2 Greekforce1821](https://greekforce1821-site.netlify.app/timeline/computers-in-a-pocket/)
* Slide του προσωπικού αποθετηρίου (test site): [Slide Greekforce1821](https://greekforce1821-site.netlify.app/slides)
* Slide #1 του προσωπικού αποθετηρίου (test site): [Slide #1 Greekforce1821](https://greekforce1821-site.netlify.app/slides/programming/)
* Slide #2 του προσωπικού αποθετηρίου (test site): [Slide #2 Greekforce1821](https://greekforce1821-site.netlify.app/slides/computers-in-pockets/)
* Το site από το αποθετήριο του οργανισμού: [Terminal-Killers](https://terminal-killers-site.netlify.app)
* Timeline του αποθετηρίου του οργανισμού: [Timeline Organization](https://terminal-killers-site.netlify.app/timeline)
* Timeline #1 του αποθετηρίου του οργανισμού: [Timeline #1 Organization](https://terminal-killers-site.netlify.app/timeline/programming-languages/)
* Timeline #2 του αποθετηρίου του οργανισμού: [Timeline #2 Organization](https://terminal-killers-site.netlify.app/timeline/computers-in-a-pocket/)
* Slide του αποθετηρίου του οργανισμού: [Slide Organization](https://terminal-killers-site.netlify.app/slides)
* Slide #1 του αποθετηρίου του οργανισμού: [Slide #1 Organization](https://terminal-killers-site.netlify.app/slides/programming/)
* Slide #2 του αποθετηρίου του οργανισμού: [Slide #2 Οrganization](https://terminal-killers-site.netlify.app/slides/computers-in-pockets/)

# Παραδοτέο 6ο: Κατασκευάζοντας το βιβλίο.

Στο συγκεκριμένο παραδοτέο, καλούμαστε να συμβάλλουμε και αντίστοιχα να κατασκευάσουμε το [βιβλίο](https://github.com/mibook/kallipos) σε ηλεκτρονική μορφή (.pdf) αλλά και σε μορφή (.pub). Το παραδοτέο αυτό, ήταν ατομικό και συγκεκριμένα έπρεπε να δημιουργήσουμε το δικό μας [φίλτρο](https://github.com/Greekforce1821/kallipos/blob/master/contribution.lua) (.lua), το δικό μας [contribution](https://github.com/Greekforce1821/kallipos/blob/master/contribution/skypech08.md), το [βιβλίο](https://github.com/Greekforce1821/kallipos/blob/master/book/book.pdf) το οποίο περιείχε το contribution μας και τέλος το βιβλίο σε [μορφή](https://github.com/Greekforce1821/kallipos/blob/master/book/book.tex) (.tex). Περαιτέρω πληροφορίες στην επόμενη παράγραφο.



Screenshot\_2022-11-14\_16-29-28

## Βήμα #1: Forking and Cloning the repository

Στο συγκεκριμένο βήμα, χρειάστηκε να κάνω fork το αρχικό αποθετήριο: [mibook/kallipos](https://github.com/mibook/kallipos) στο προφίλ μου: [Greekforce1821/kallipos](https://github.com/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) .

## Bήμα #2: Downloading the Suitable Software via Terminal

Σύμφωνα με τις οδηγίες του μαθήματος [guides](https://github.com/courses-ionio/hci/discussions/1832) 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 προκειμένουν να κάνει merge όλα τα (.tex) αρχεία σε ένα και στην συνέχεια να γίνει export από (.tex) σε (.pdf) αρχείο.

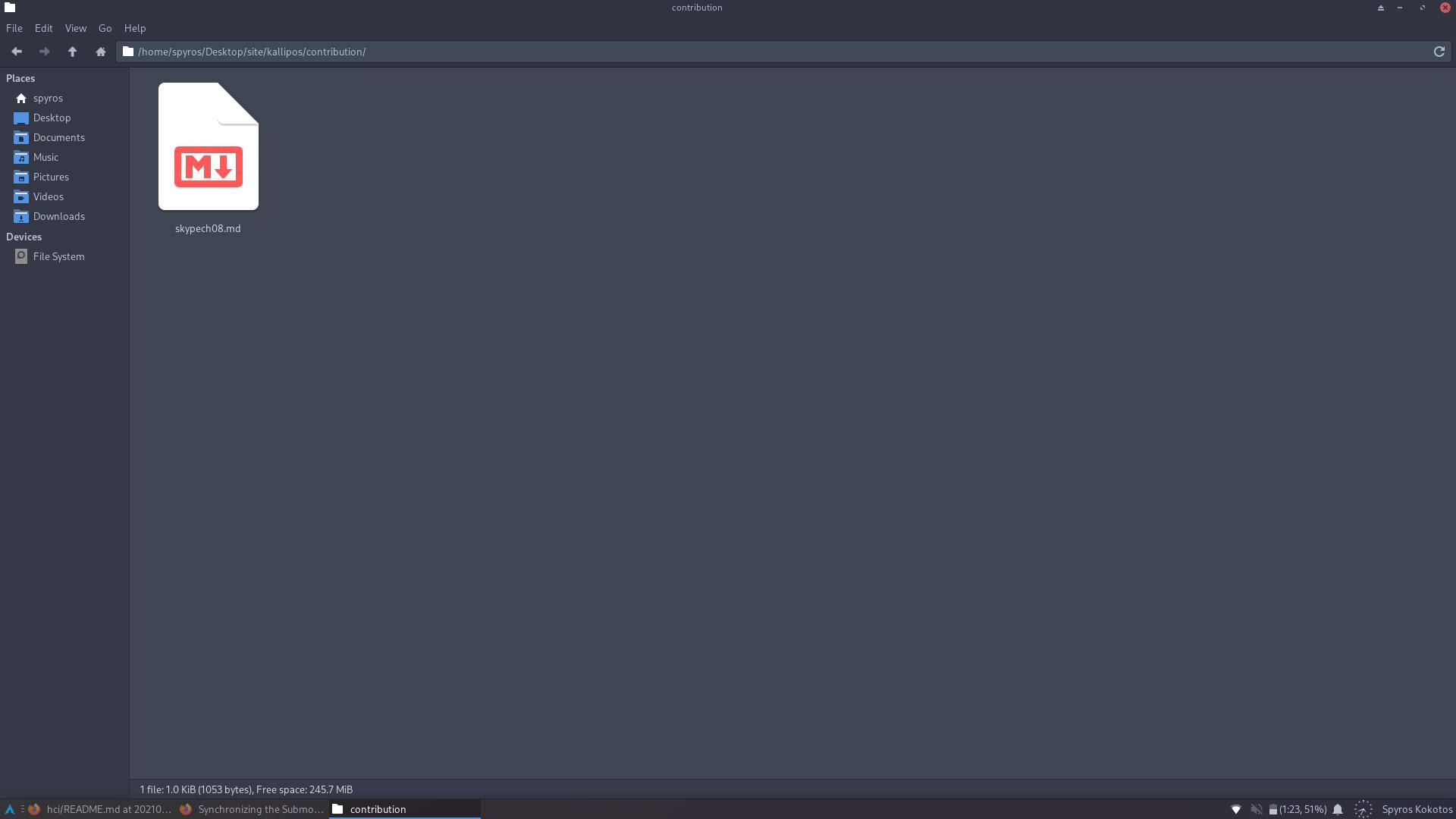
## Bήμα #3: Cloning the Submodules from the Main Repository and Setting them. Part #1

Στο συγκεκριμένο βήμα, όταν έκανα clone το αποθετήριο τοπικά στον υπολογιστή μου, τα submodules δεν είχα ρυθμίστεί 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

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

## 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](https://pibook.epidro.me/), 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 skypech08.md το οποίο περιέχει τις πληροφορίες που θέλω να προσθέσω στο ήδη υπάρχον αρχείο.



Screenshot\_2022-11-14\_16-41-36

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

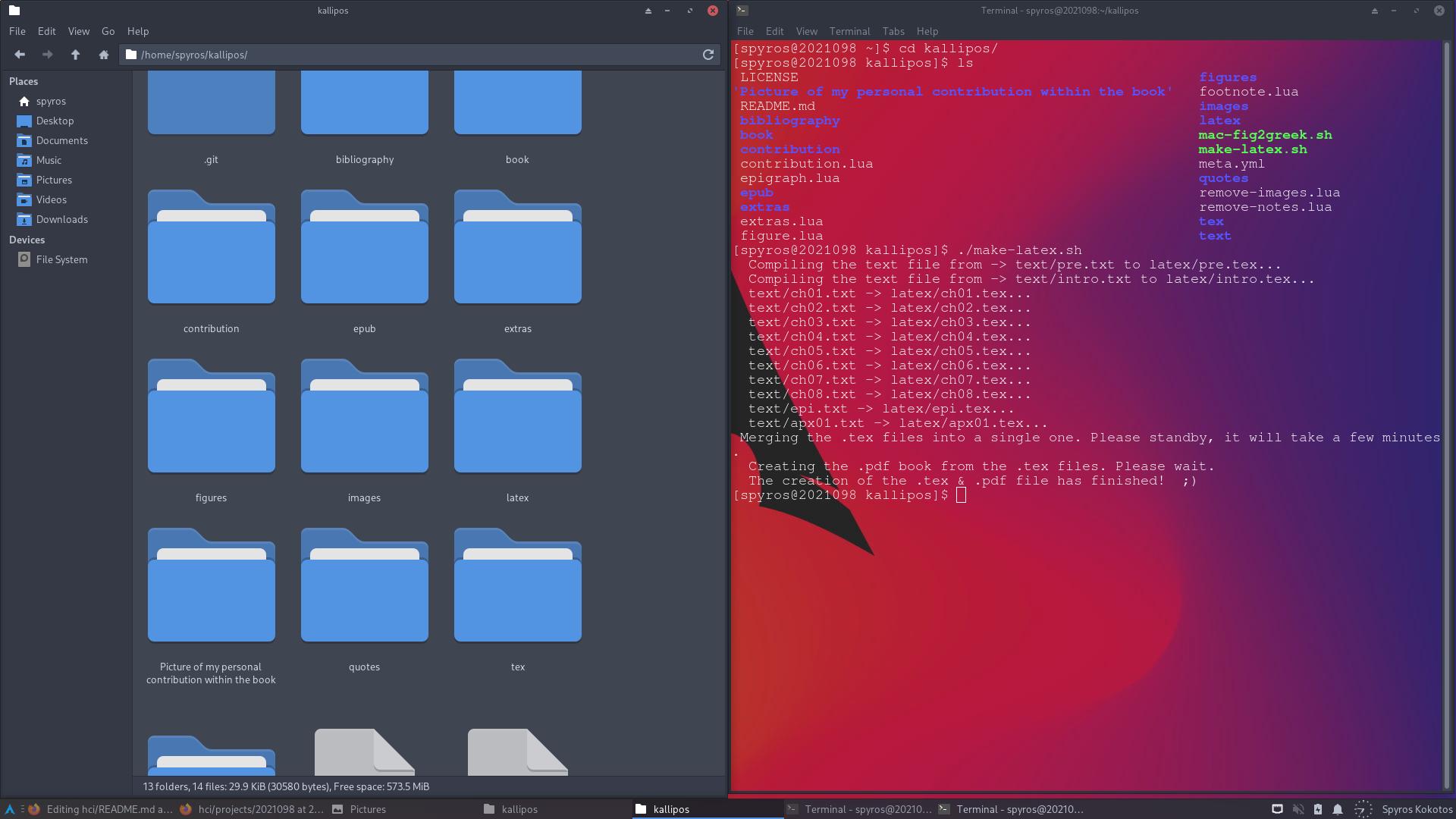
Στο παραπάνω βήμα, δημιούργησα το [φίλτρο μου](https://github.com/Greekforce1821/kallipos/blob/master/contribution.lua) προκειμένου να εισάγω το contribution μου στο βιβλίο και αντίστοιχα, πρόσθεσα στο αρχείο (.sh) το φίλτρο και ορισμένα μηνύματα, μαζί με την δημιουργία directories στο ήδη υπάρχον directory.

### 1 Screenshot_2022-11-17_18-27-44

### 2 Screenshot_2022-11-17_18-28-58

## Βήμα #7: Creating and Merging the (.tex) Files into a Single (.pdf) File

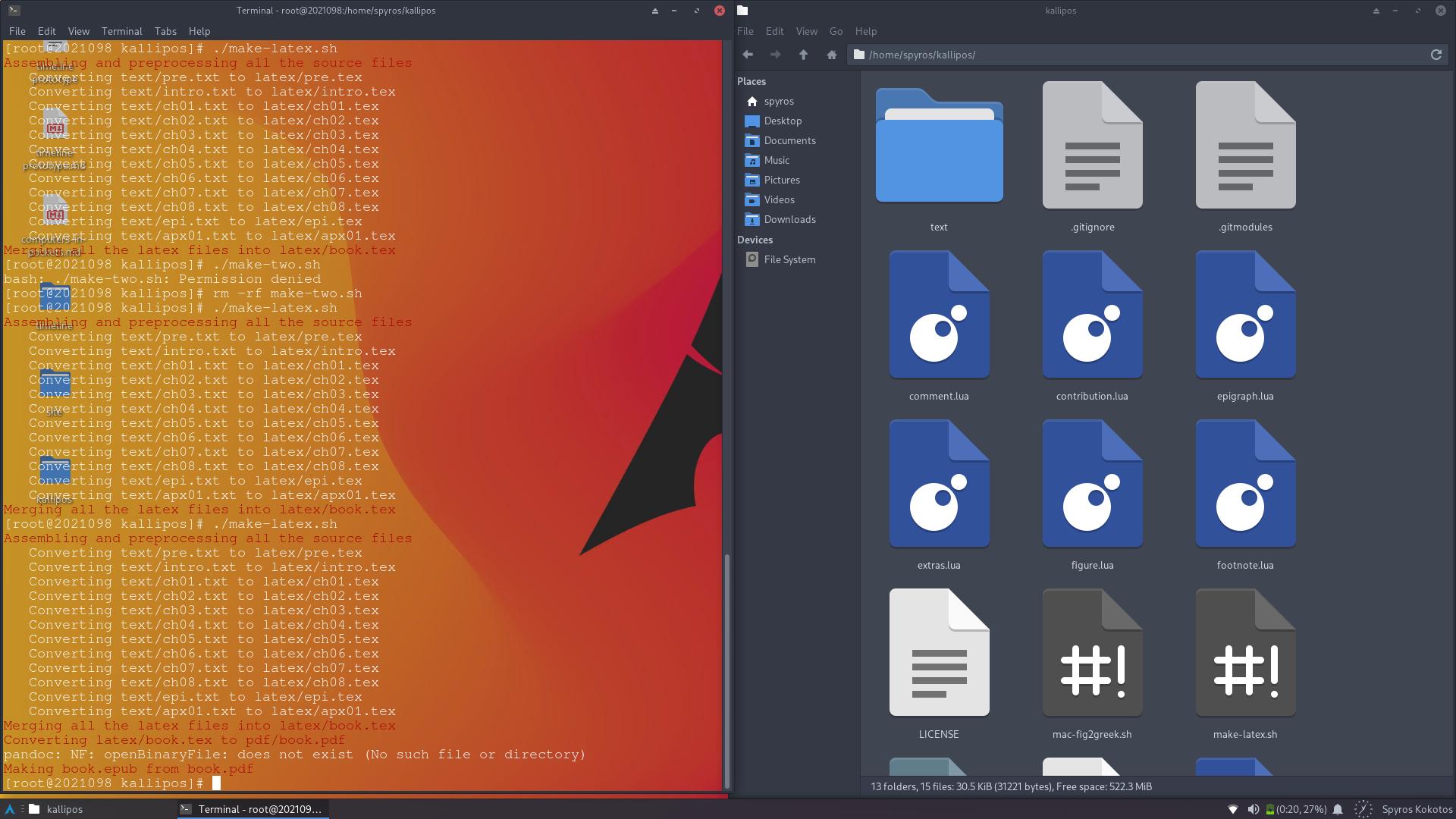
Στο παραπάνω βήμα, εκκίνησα το scriptάκι make-latex.sh το οποίο ξεκινά την διαδικασία “συναρμολόγησης” των αρχείων (.tex) σε ένα μεμονομένο και αντίστοιχα η μετατροπή του σε αρχείο (.pdf). Στο αρχείο (.sh) έχω προσθέσει με την εντολή echo διάφορα μηνύματα ούτως ώστε, να υπάρχει μια έξοδος σε περίπτωση που υπάρξει κάποιο πρόβλημα ή αντίστοιχα να γνωρίζω τι πρόοδο έχει κάνει το συγκεκριμένο αρχείο.



Screenshot\_2022-11-17\_18-35-47

Προβλήματα:

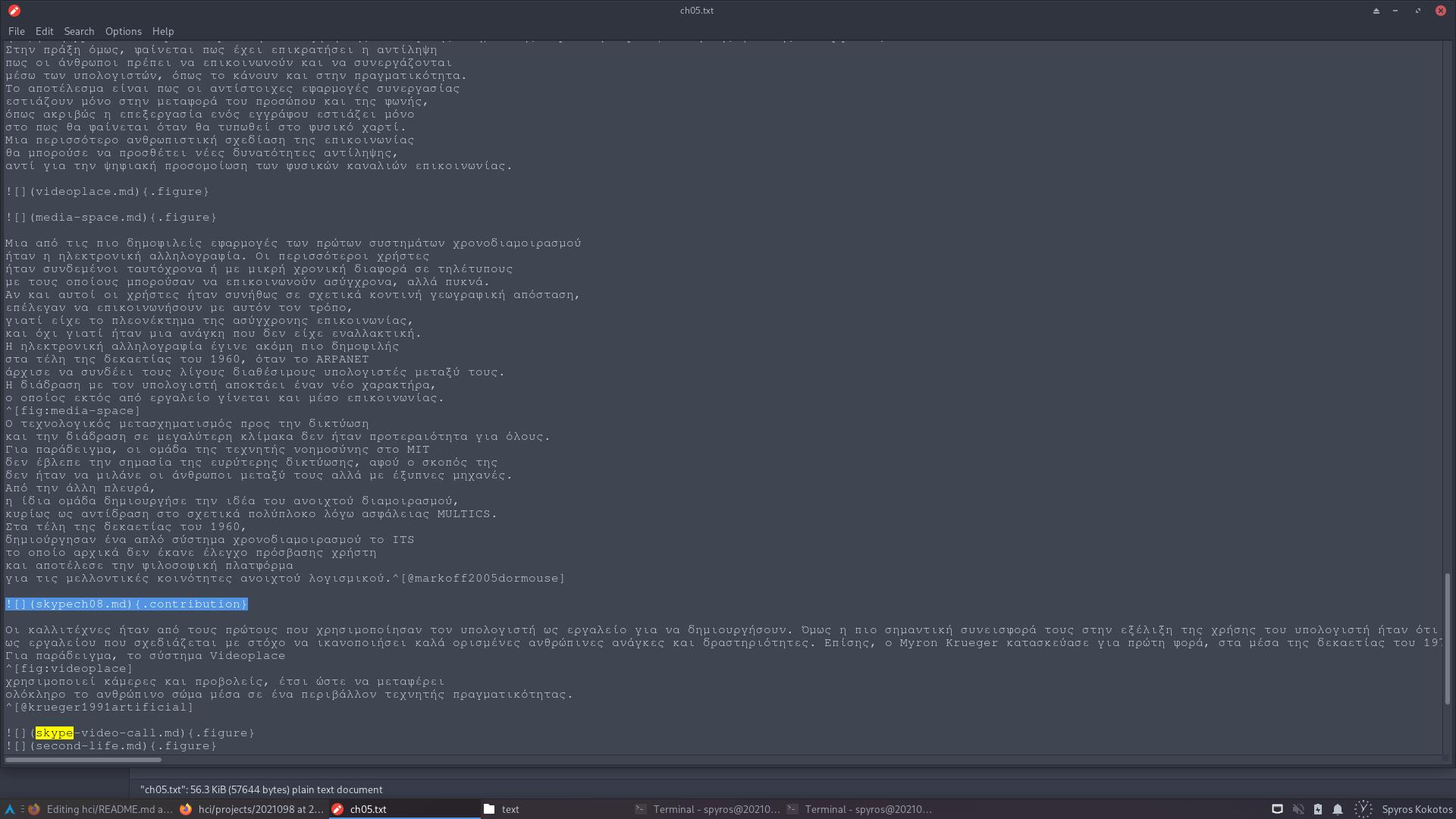
Όταν εκκίνησα για πρώτη φορά δοκιμαστικά το παραπάνω scriptάκι παρατήρησα το συγκεκριμένο error στο τερματικό, το οποίο οδηγούσε στην αναζήτηση της γραμματοσειράς MesloLGS NF Regular η οποία δεν υπήρχε εγκατεστημένη στο σύστημα. Μετά από αρκετές ώρες ψάξιμο στο google, κατέληξα στο γεγονός ότι πρέπει να κατεβάσω το αρχείο .ttf της συγκεκριμένης γραμματοσειράς και να το προσθέσω στην παρακάτω διαδρομή: /usr/share/fonts/meslolgs-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 exits (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.



Screenshot\_2022-11-13\_23-04-39

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

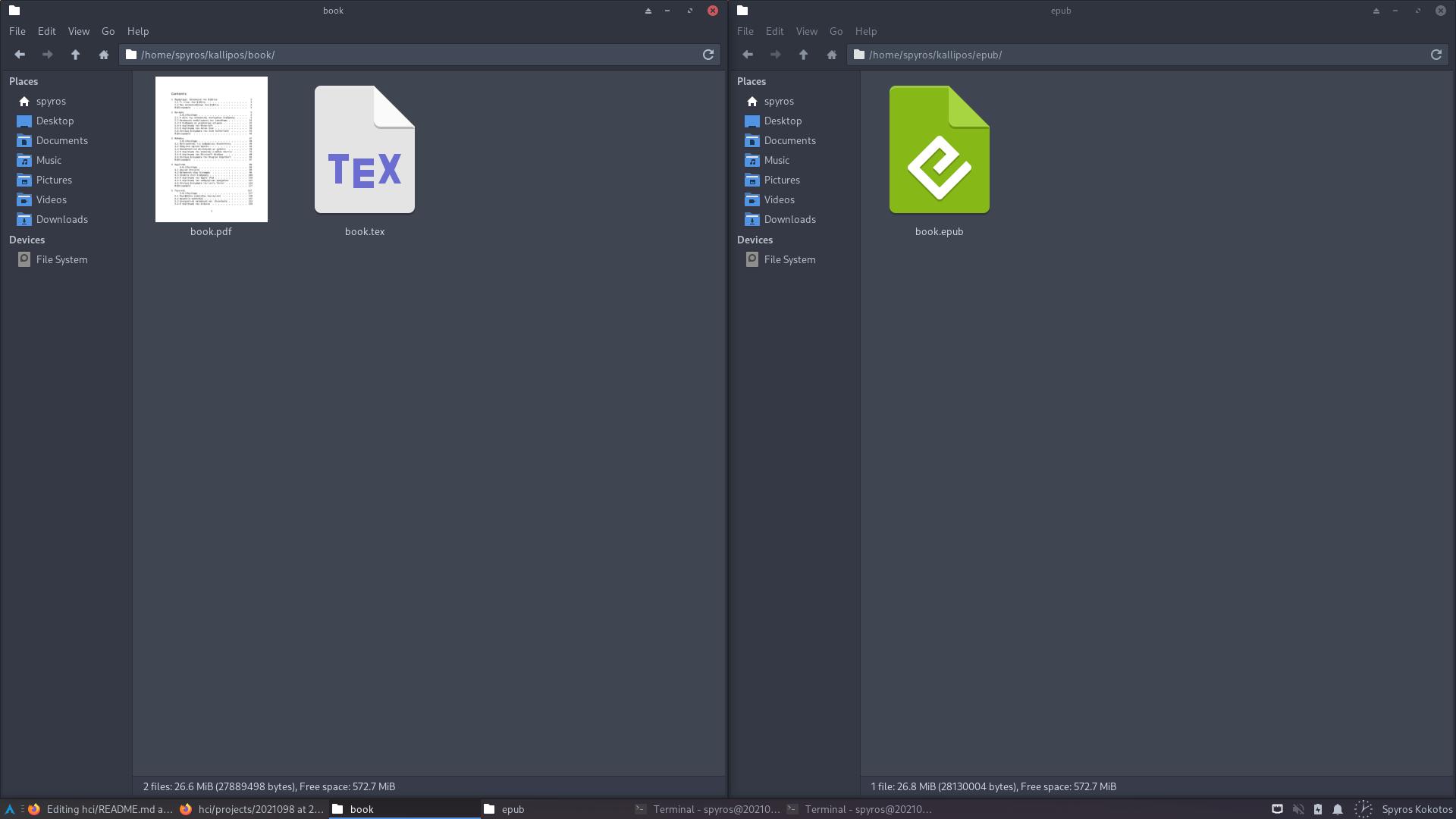
Στο συγκεκριμένο βήμα, αφού είχα κατασταλάξει που θα προσθέσω το contribution μου, έπρεπε να ορίσω σε ποιό chapter του βιβλίου θα προστεθεί, επομένως, εφόσον το Skype ανήκει στo Chapter: 06 και συγκεκριμένα στην υποενότητα: 6.0.3 άνοιξα τον φάκελο text και στην συνέχεια το αρχείο ch05.txt έψαξα να βρω που βρίσκεται η αναφορά για το Skype. Μόλις την βρήκα, προκειμένου να ενσωματώσω το φίλτρο μου στο συγκεκριμένο σημέιο έπρεπε να προσθέσω την παρακάτω εντολή ![](skypech08.md){.contribution} στο αρχείο προκειμένου να εμφανιστεί το contribution μου. Επιπλέον, να αναφέρω πως η δομή της συγκεκριμένης εντολής είναι η εξής: ![](example.md){.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.



Screenshot\_2022-11-17\_18-55-08

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

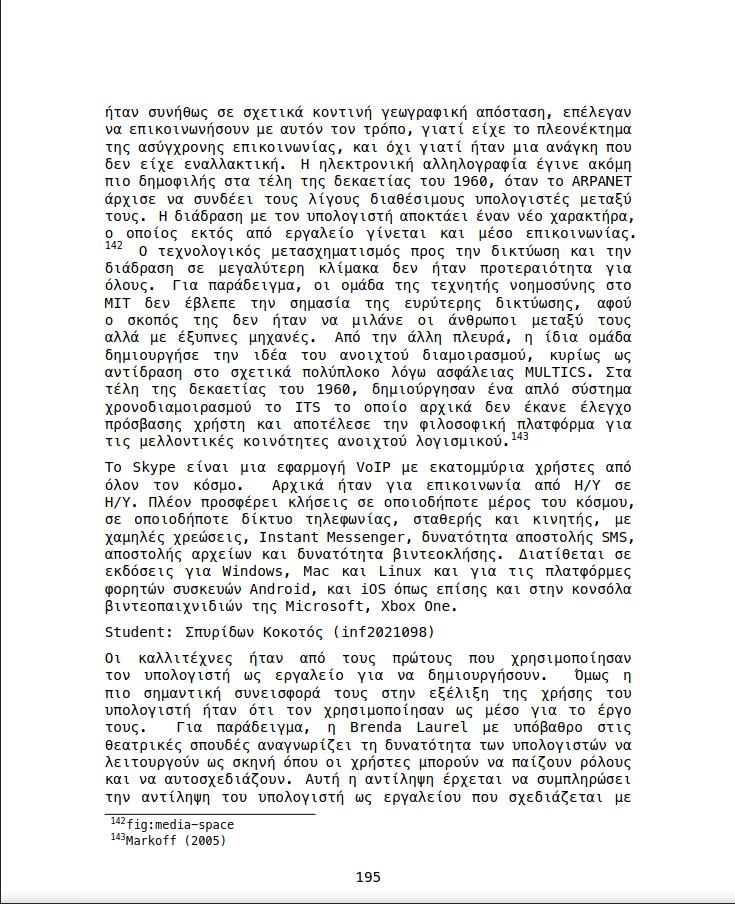
Στο παραπάνω βήμα, εφόσον ολοκλήρωσα τις αλλαγές μου, έπρεπε να ενημερώσω το ήδη υπάρχον βιβλίο, εκτελώντας ξανά την ίδια εντολή ./make-latex.sh μέσα από το τερματικό προκειμένου να εκκινήσω το scriptάκι make-latex.sh. Μόλις ολοκληρώθηκε το compilation, προέκυψε το βιβλίο σε μορφή (.tex), σε μορφή (.pdf)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



Screenshot\_2022-11-17\_19-00-12

## 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.

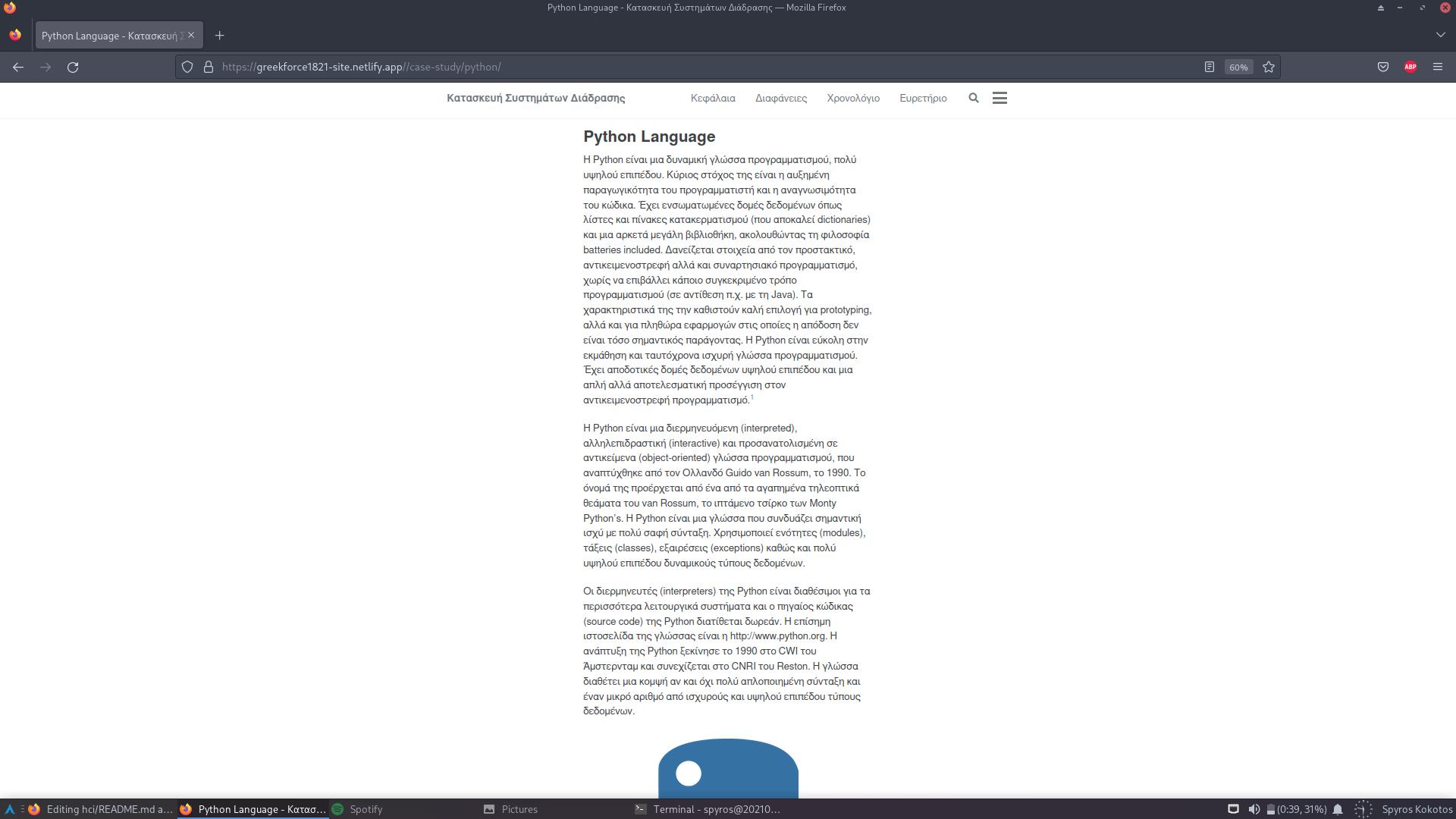


Screenshot\_2022-11-17\_19-07-58

# Παραδοτέο 7ο: Συμμετοχικό περιεχόμενο Β1.

Στο συγκεκριμένο παραδοτέο, έπρεπε να δημιουργήσουμε μια νέα μελέτη περίπτωσης σχετική με τα παραδοτέα [Α1](https://github.com/courses-ionio/help/discussions/1185) και [Α2](https://github.com/courses-ionio/help/discussions/1461) σύμφωνα με τις [οδηγίες](https://courses-ionio.github.io/help/social/) του μαθήματος και να τα ανεβάσουμε στην [ιστοσελίδα](https://greekforce1821-site.netlify.app/) μας. Συγκεκριμένα, ασχολήθηκα με την μελέτη περίπτωσης: Η γλώσσα προγραμματισμού Python και η χρήση της από τον υπολογιστή τσέπης Odroid-c4. Παρακάτω επισυνάπτω την φωτογραφία από το case-study.

## Python & Odroid-C4 Case-Study



Screenshot\_2022-11-19\_11-30-49

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

* Προσωπικό site από το αποθετήριο μου: [Greekforce1821](https://greekforce1821-site.netlify.app/)
* Case-Study του προσωπικού αποθετηρίου: [Case-Study Python](https://greekforce1821-site.netlify.app//case-study/python/)
* Αρχείο μελέτης περίπτωσης από το αποθετήριο: [.md File #Case-Study Thumbnail](https://github.com/Greekforce1821/site/blob/master/_case-study/python.md)

## Αρχείο μελέτης περίπτωσης από το αποθετήριο: [.md File #Case-Study](https://github.com/Greekforce1821/site/blob/master/_includes/cs-python.md)

# Φωτογραφίες που αξιοποιήθηκαν στην παραπάνω εργασία:

* Φωτογραφία thumbnail για την μελέτη περίπτωσης της Python: [Thumbnail-Python](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/python-thumb.jpg)
* Φωτογραφία για την μελέτη περίπτωσης της Python #1: [Python #1](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/python-logo.jpg)
* Φωτογραφία για την μελέτη περίπτωσης της Python #2: [Python #2](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/guido-headshot.jpg)
* Φωτογραφία για την μελέτη περίπτωσης της Python #3: [Python #3](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/python-statistics-2022.jpg)

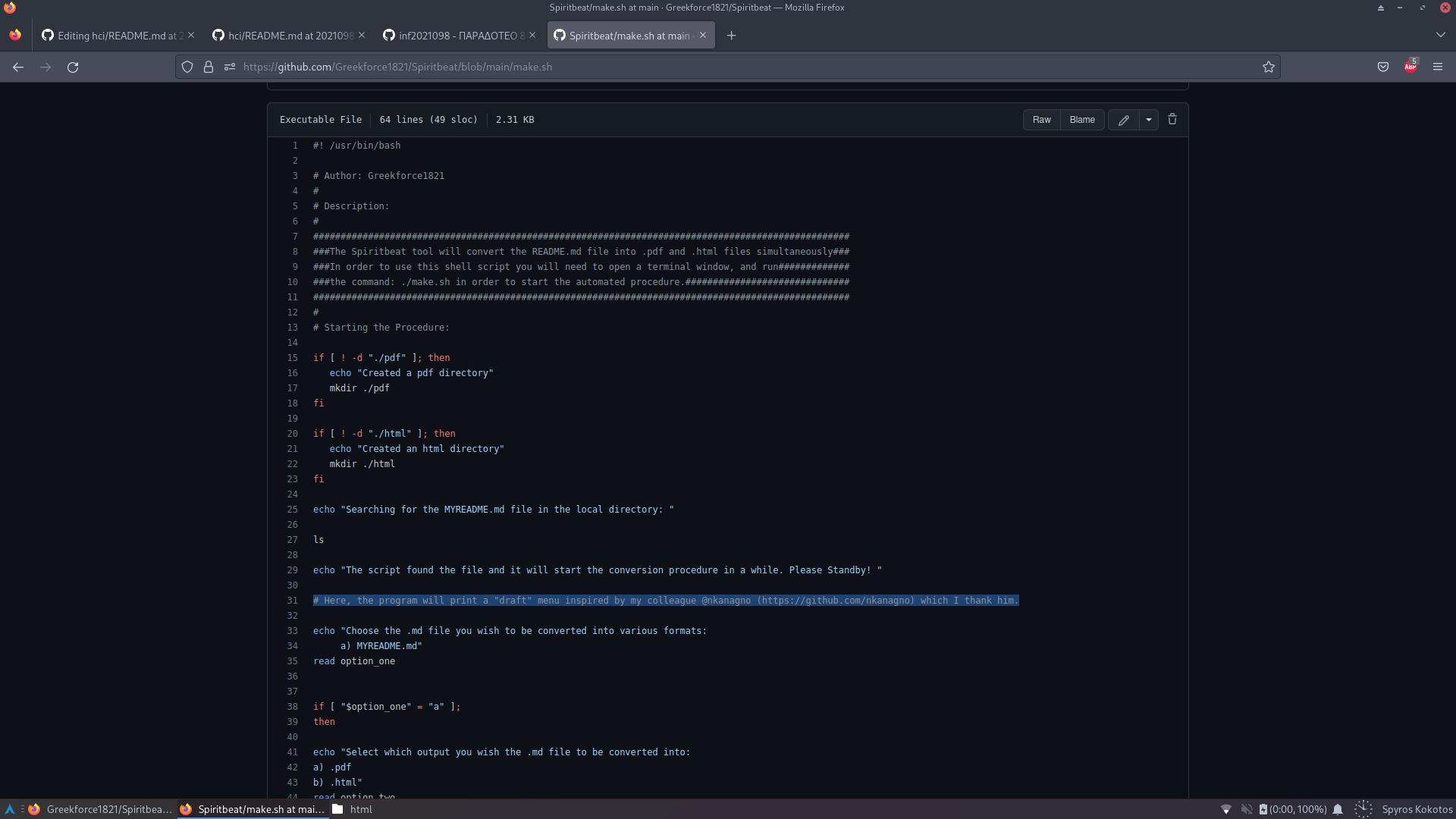
## Φωτογραφία για την μελέτη περίπτωσης της Python #4: [Python #4](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/programming-languages-statistics-2022.jpg)

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

| Φωτογραφίες | Copyright Holders | License | Type of License |
| --- | --- | --- | --- |
| [Python Case Study #1](https://greekforce1821-site.netlify.app/case-study/python/) | [Φωτογραφία #1](https://upload.wikimedia.org/wikipedia/commons/archive/c/c3/20220730085403%21Python-logo-notext.svg) | © The Python Software Foundation | [GNU General Public License](https://www.gnu.org/licenses/gpl-3.0.html) |
| Python Case Study #2 | [Φωτογραφία #2](https://images.unsplash.com/photo-1515879218367-8466d910aaa4?ixlib=rb-4.0.3&ixid=MnwxMjA3fDB8MHxwaG90by1wYWdlfHx8fGVufDB8fHx8&auto=format&fit=crop&w=1169&q=80) | © Chris Ried | [Unsplash License](https://unsplash.com/license) |
| Python Case Study #3 | [Φωτογραφία #3](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/python-statistics-2022.jpg) | © Spyros Kokotos | [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) |
| Python Case Study 34 | [Φωτογραφία #4](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/programming-languages-statistics-2022.jpg) | © Spyros Kokotos | [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) |

# Παραδοτέο 8ο: Command Line Exercise #2

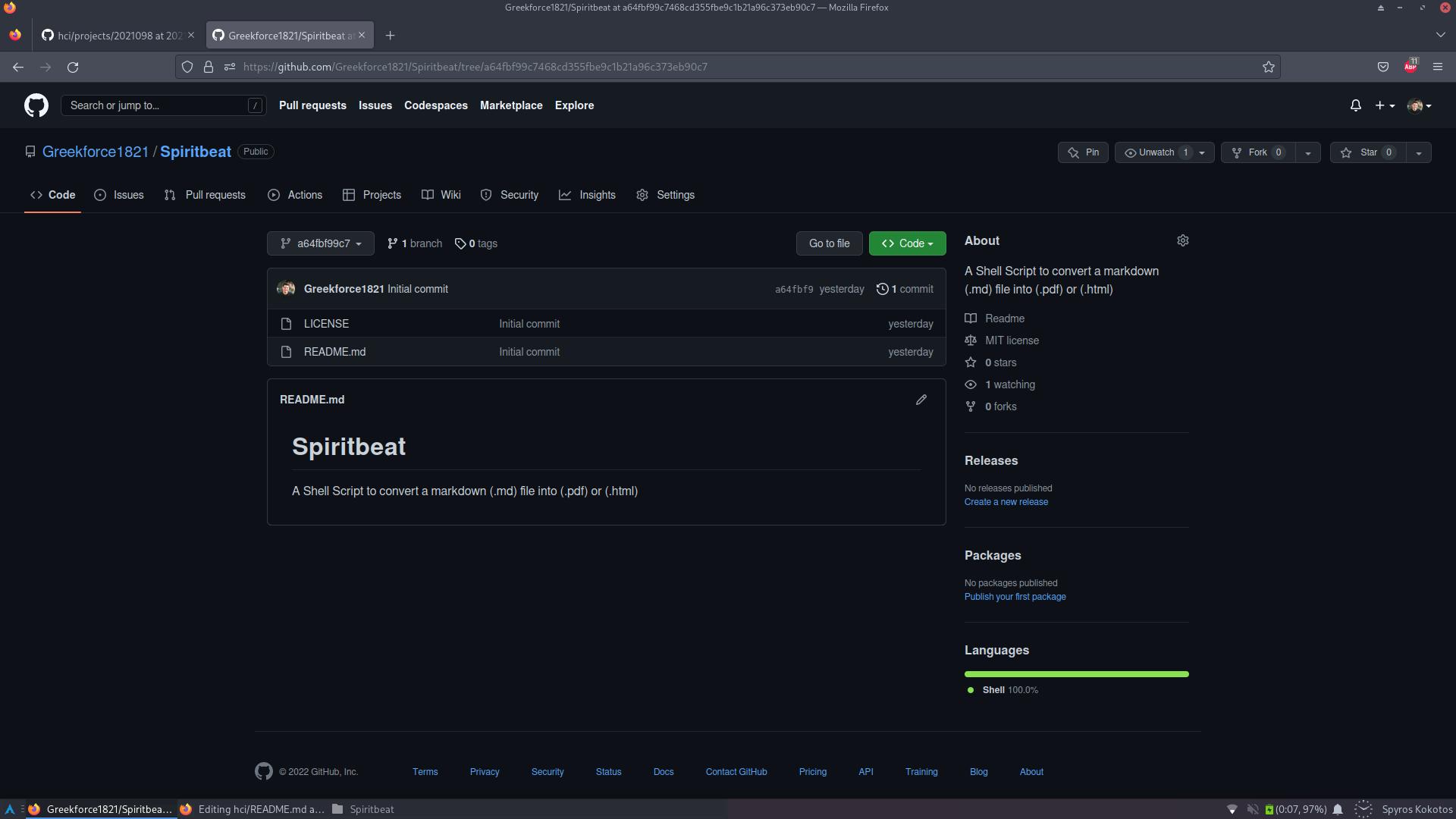
Στο συγκεκριμένο παραδοτέο καλούμαστε να δημιουργήσουμε ένα δικό μας shell script το οποίο θα πρέπει να έχει σχέση με τις γνώσεις που αποκτήσαμε στα προηγούμενα παραδοτέα (βλέπε: [Παραδοτέο 6](https://github.com/courses-ionio/help/discussions/1586) & [Παραδοτέο 7](https://github.com/courses-ionio/help/discussions/1675) ). Σύμφωνα με τις [οδηγίες](https://github.com/epidrome/dokey#hci) 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](https://github.com/nkanagno/paradoteo-8/blob/main/file-conversion.sh) τον οποίο ευχαριστώ πολύ για την συμβολή του!



Screenshot\_2022-11-26\_14-59-58

## Βήμα #1: The Creation of the Repository

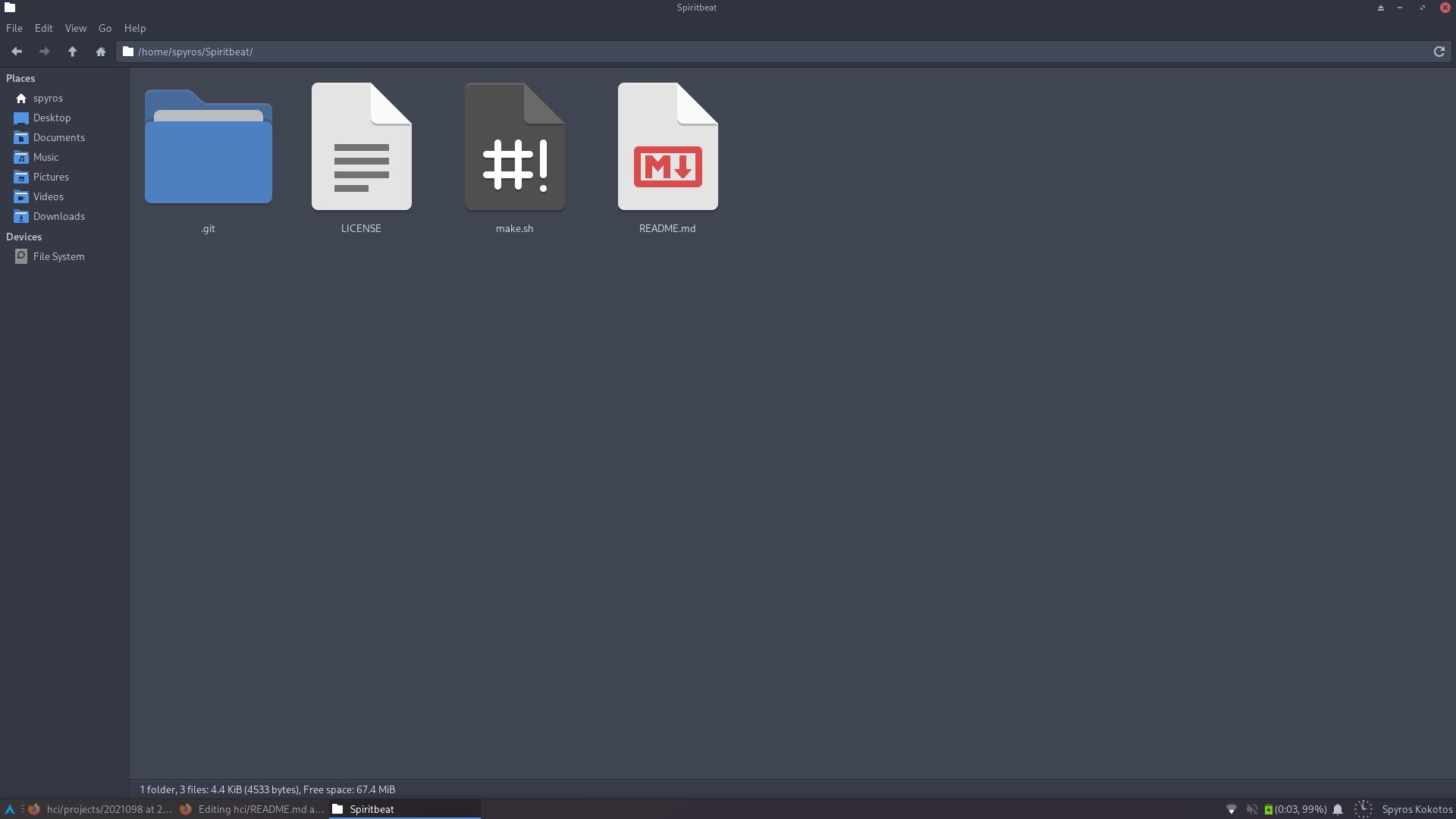
Στο συγκεκριμένο βήμα, δημιούργησα ένα [ξεχωριστό αποθετήριο](https://github.com/Greekforce1821/Spiritbeat) στο προφίλ μου στο Github το οποίο το ονομάσα ως: Spiritbeat. Στην συνέχεια, δημιούργησα ένα README.md αρχείο (κενό για αρχή) και ένα LICENSE.md.



Screenshot\_2022-11-26\_14-50-13

## Bήμα #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.



Screenshot\_2022-11-26\_14-55-01

## 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 τερμάτικο, 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)

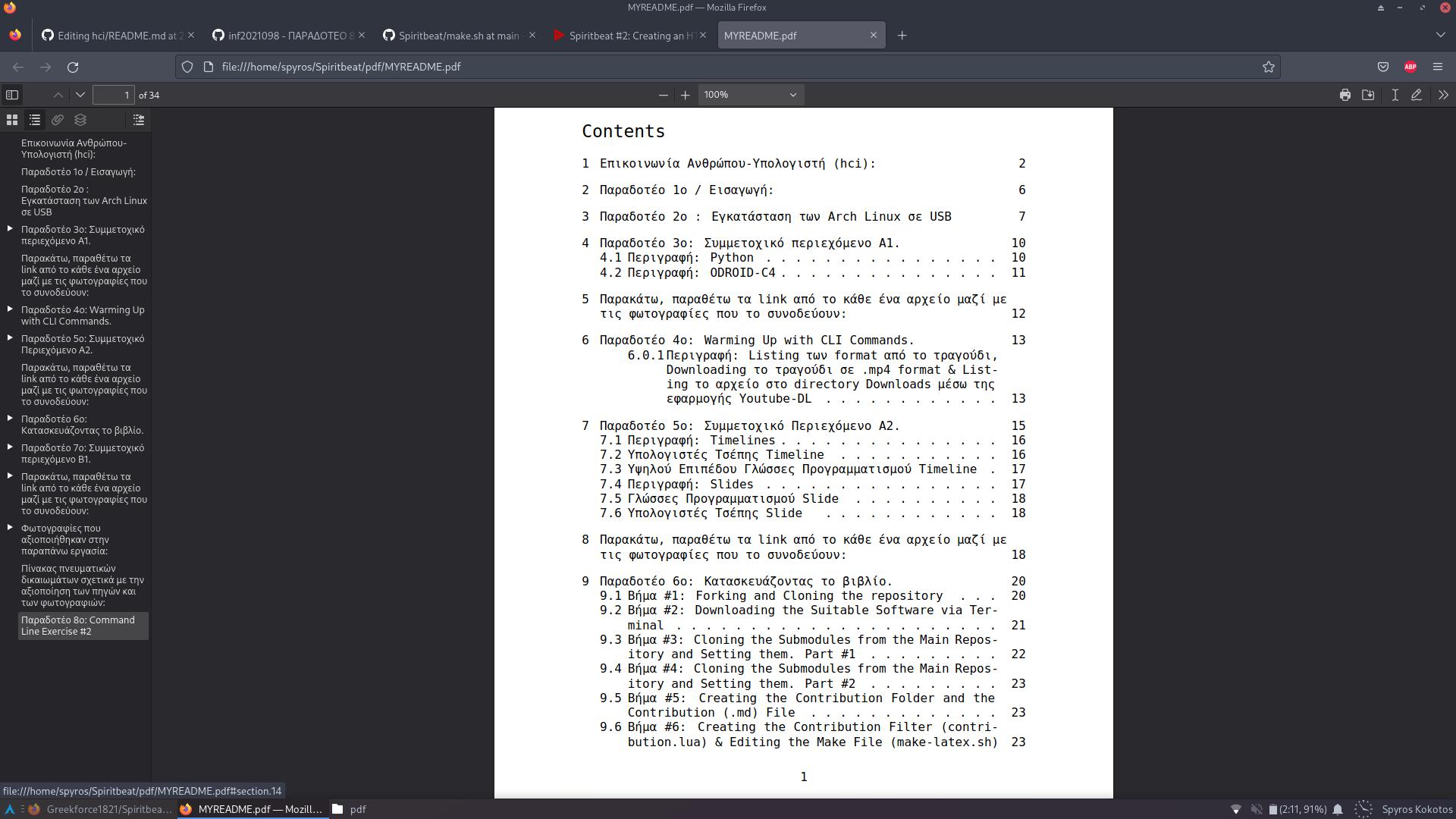
## Converting a markdown (.md) file into PDF (.pdf)

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

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

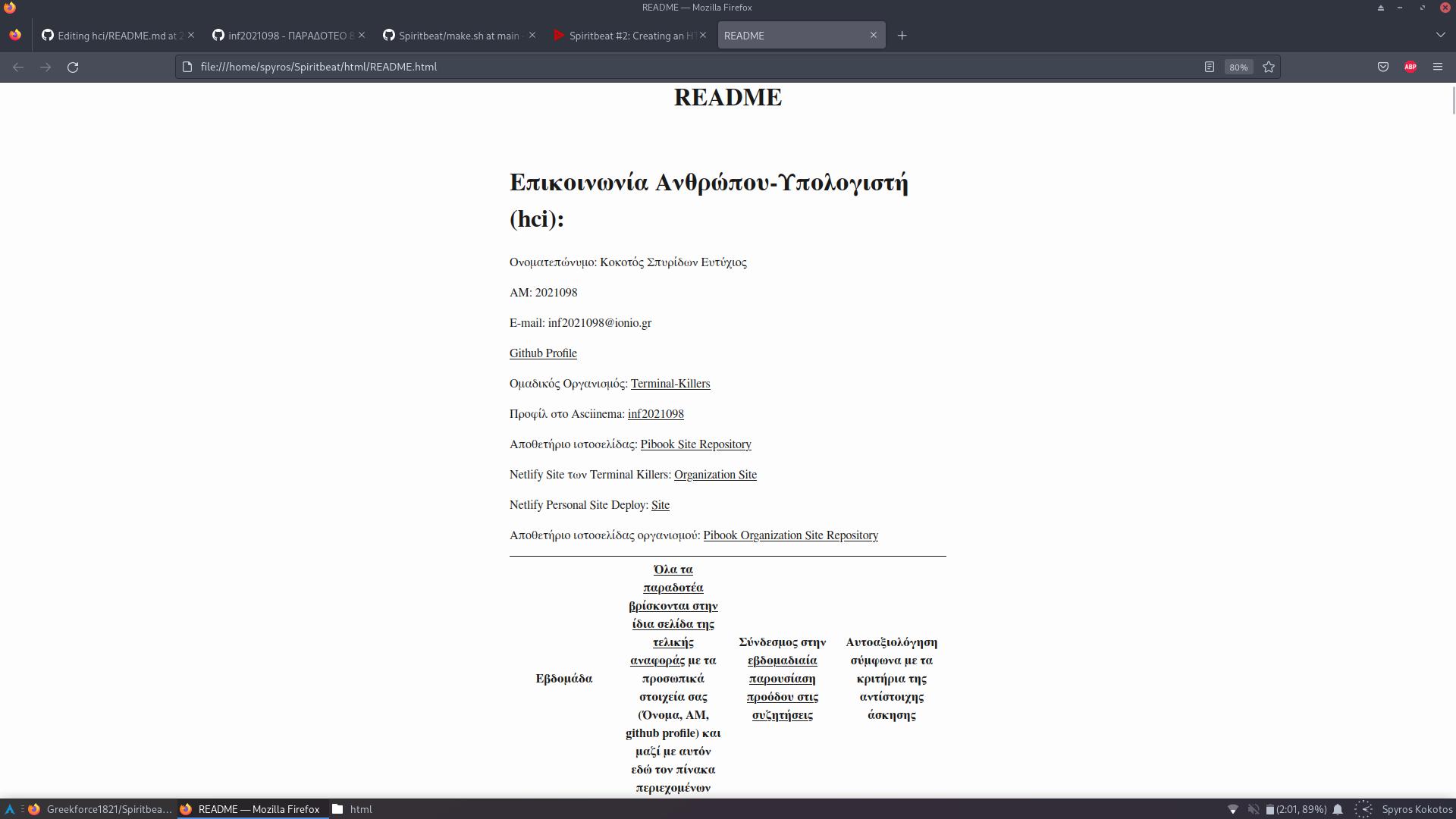
Στο συγκεκριμένο βήμα, δημιουργήθηκαν από το script δύο φάκελοι, ένας με ονομασία: pdf και ένας html. Μόλις ολοκληρώθηκε η μετατροπή, δημιουργήθηκε το αρχείο MYREADME.pdf στον φάκελο pdf και αντίστοιχα το αρχείο MYREADME.html στον φάκελο html και τσέκαρα ότι δεν υπήρξαν λάθη κατά την μετατροπή. Παρακάτω,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

## PDF Results



Screenshot\_2022-11-26\_15-17-58

## HTML Results



Screenshot\_2022-11-26\_15-20-13

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

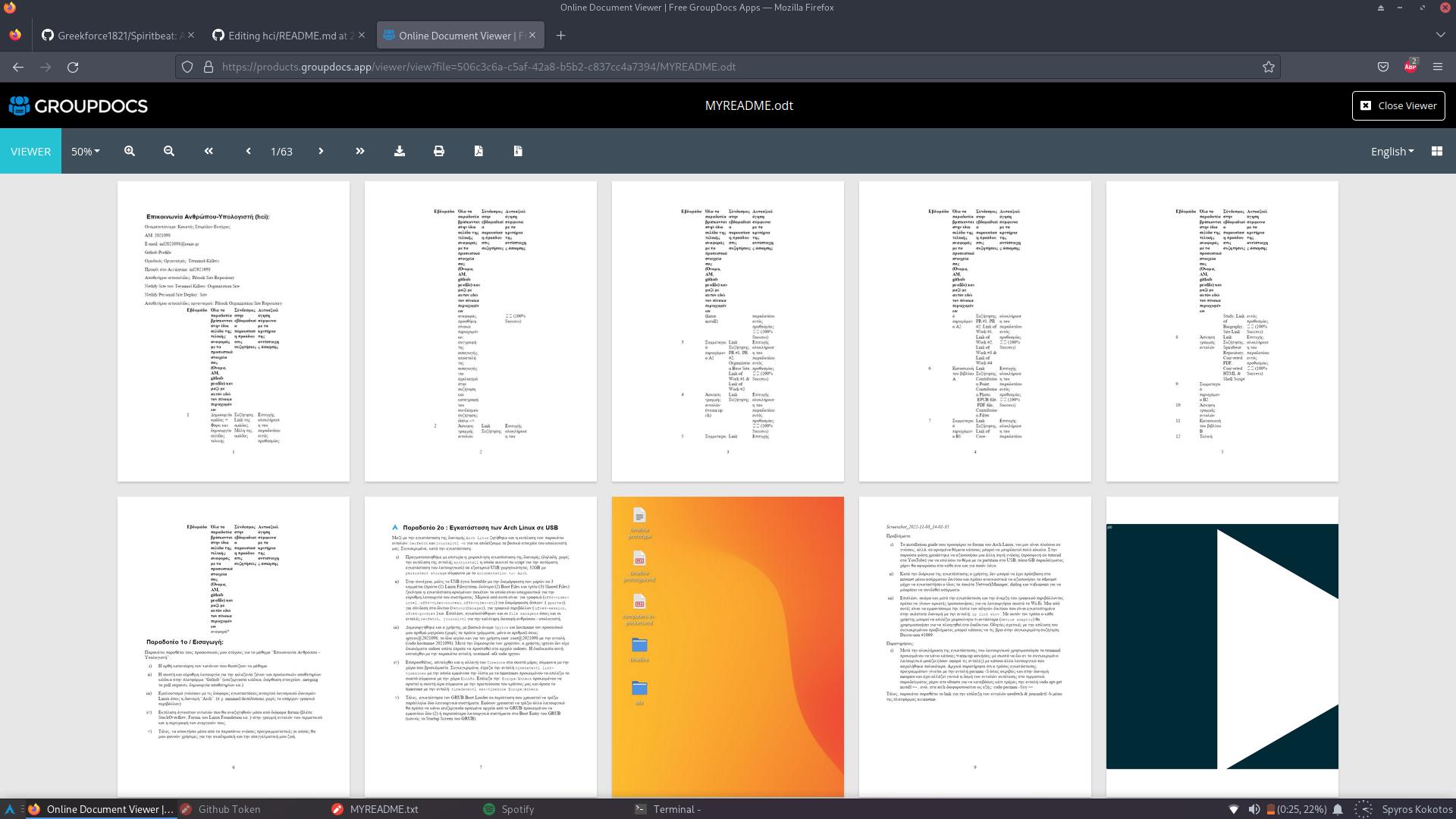
Στο παραπάνω βήμα, απλά, πρόσθεσα κάτι τελευταίες σημειώσεις στο README.md αρχείο, σχετικά με τι prerequisites πρέπει να έχει ο χρήστης εγκατεστημένα στο σύστημά του προκειμένου να τρέξει το script που δημιούργησα.

## Update: Added Support For (.odt), (.txt), (.ipynb) & (.docx)

Στο παραπάνω πρόγραμμα, υπήρχαν περιθώρια βελτίωσης τα οποία τα αξιοποίησα προσθέτοντας υποστήριξη για μετατροπή αρχείων markdown (.md) σε αρχεία όπως: (.odt), (.txt), (.ipynb) και (.docx). Παρακάτω παραθέτω την μετατροπή του (.md) αρχείο μου (προσωπική αναφορά εργασίας για το HCI) στις παραπάνω μορφές αρχείων.

## Converting a markdown (.md) file into ODT (.odt)

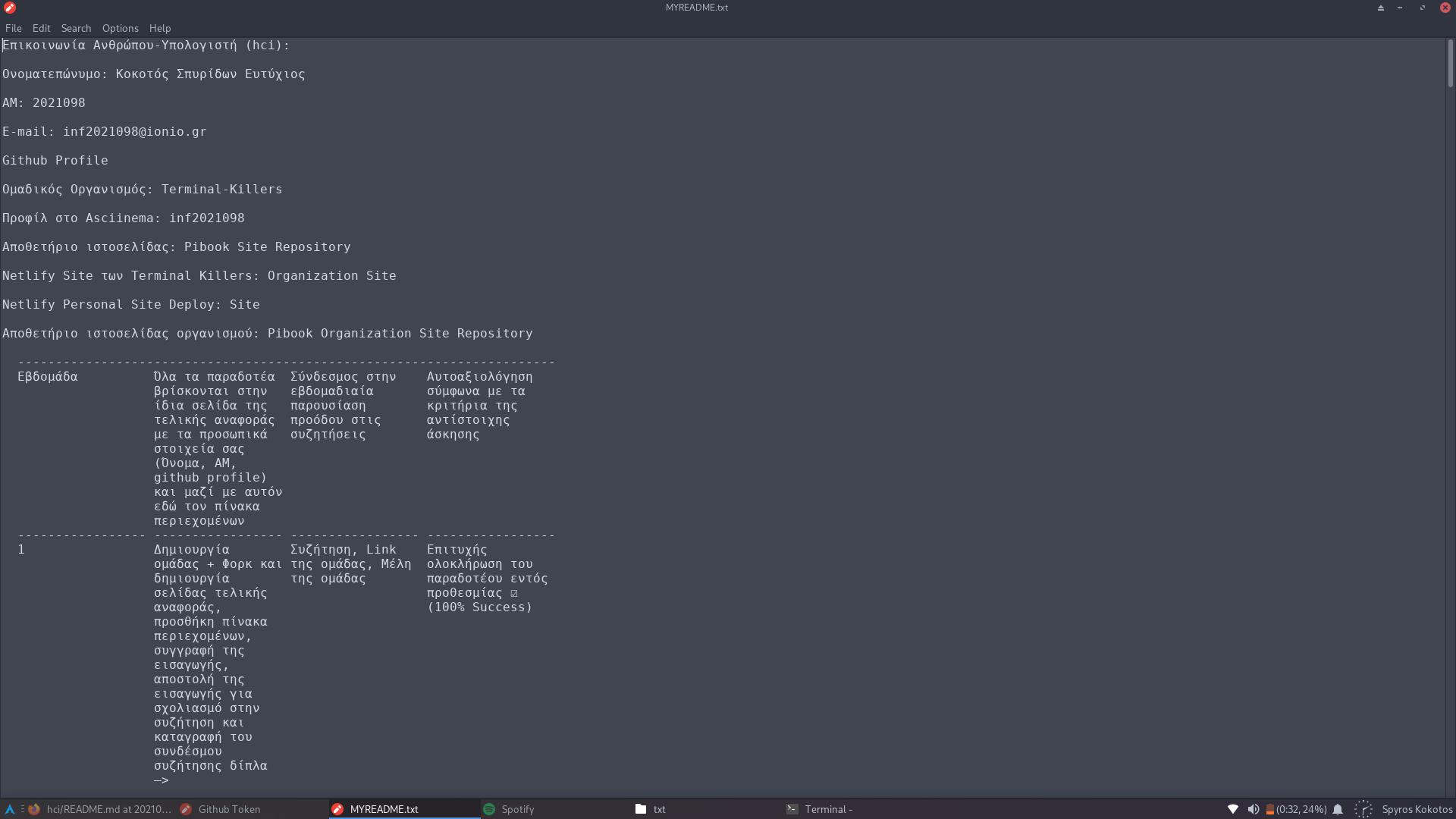
## ODT (.odt) Results:



Screenshot\_2022-11-28\_02-13-21

## Converting a markdown (.md) file into TXT (.txt)

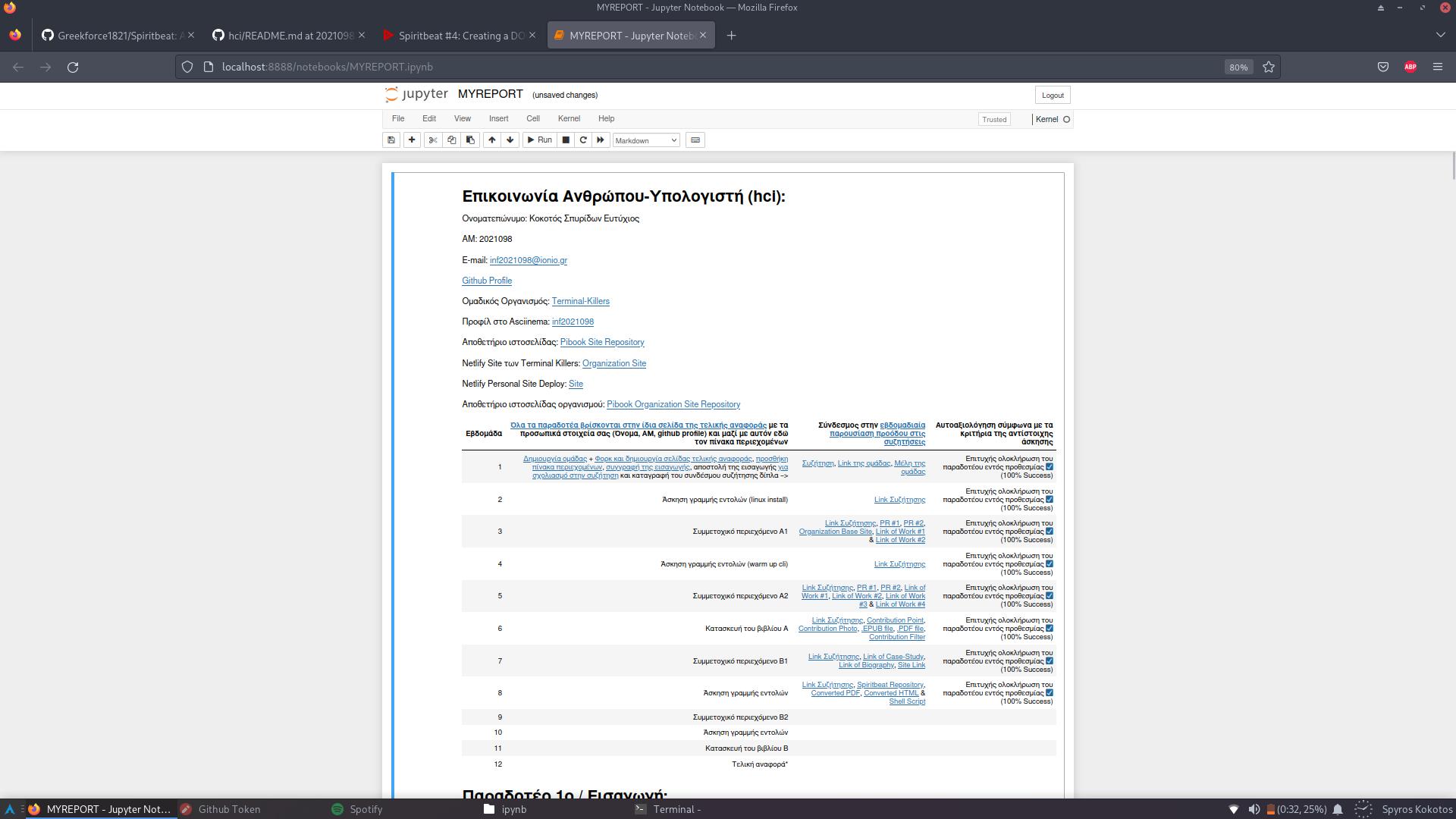
## TXT (.txt) Results:



Screenshot\_2022-11-28\_02-10-33

## Converting a markdown (.md) file into IPYNB (.ipynb)

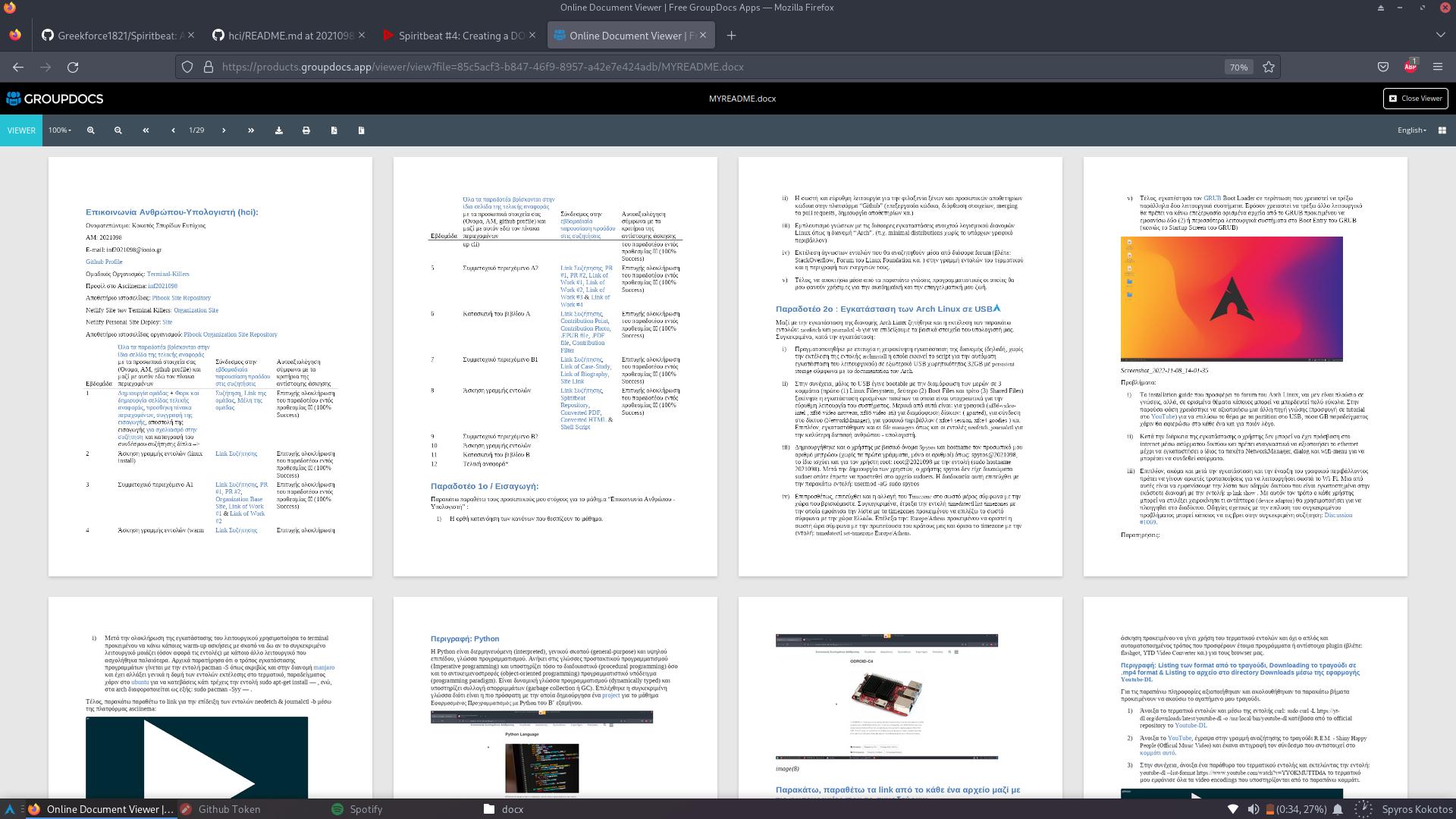
## IPYNB (.ipynb) Results:



Screenshot\_2022-11-28\_02-08-49

## Converting a markdown (.md) file into DOCX (.docx)

## DOCX (.docx) Results:



Screenshot\_2022-11-28\_02-07-03

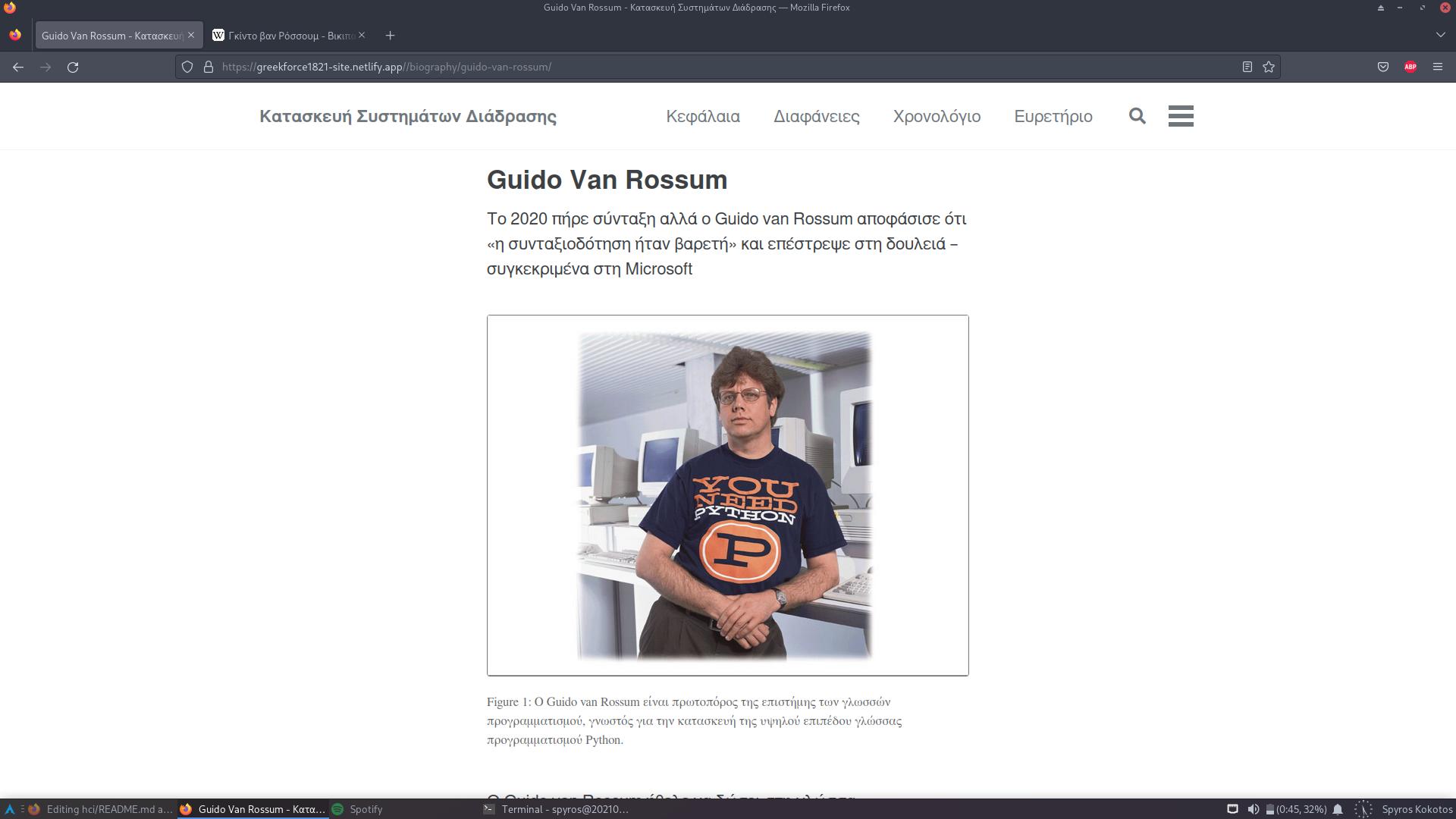
# Πίνακας των διάφορων link περί εργασίας:

| Links | Platform | Extension |
| --- | --- | --- |
| [Shell Script](https://github.com/Greekforce1821/Spiritbeat/blob/main/make.sh) | GitHub | .sh |
| [MYREADMEEXAMPLE.md](https://github.com/Greekforce1821/Spiritbeat/blob/main/MYREADME.md) | GitHub | .md |
| [Converted PDF File](https://github.com/Greekforce1821/Spiritbeat/blob/main/pdf/MYREADME.pdf) | GitHub | .pdf |
| [Converted HTML File](https://github.com/Greekforce1821/Spiritbeat/blob/main/html/README.html) | GitHub | .html |
| [Guides For Users - How To Use](https://github.com/Greekforce1821/Spiritbeat/blob/main/README.md) | GitHub | .md |
| [SpiritBeat Shell Script Repository](https://github.com/Greekforce1821/Spiritbeat) | GitHub | - |
| [PDF Conversion Using Terminal](https://asciinema.org/a/540266) | Asciinema | .cast |
| [HTML Conversion Using Terminal](https://asciinema.org/a/540268) | Asciinema | .cast |
| [Converted ODT File](https://github.com/Greekforce1821/Spiritbeat/blob/main/odt/MYREADME.odt) | GitHub | .odt |
| [Converted TXT File](https://github.com/Greekforce1821/Spiritbeat/blob/main/txt/MYREADME.txt) | GitHub | .txt |
| [Converted IPYNB File](https://github.com/Greekforce1821/Spiritbeat/blob/main/ipynb/MYREPORT.ipynb) | GitHub | .ipynb |
| [Converted DOCX File](https://github.com/Greekforce1821/Spiritbeat/blob/main/docx/MYREADME.docx) | GitHub | .docx |
| [ODT Conversion Using Terminal](https://asciinema.org/a/540846) | Asciinema | .cast |
| [TXT Conversion Using Terminal](https://asciinema.org/a/540844) | Asciinema | .cast |
| [IPYNB Conversion Using Terminal](https://asciinema.org/a/540847) | Asciinema | .cast |
| [DOCX Conversion Using Terminal](https://asciinema.org/a/540845) | Asciinema | .cast |

# Παραδοτέο 9ο: Συμμετοχικό περιεχόμενο Β2.

Στο συγκεκριμένο παραδοτέο, έπρεπε να δημιουργήσουμε μια νέα βιογραφία σχετική με τα παραδοτέα [Α1](https://github.com/courses-ionio/help/discussions/1185) και [Α2](https://github.com/courses-ionio/help/discussions/1461) σύμφωνα με τις [οδηγίες](https://courses-ionio.github.io/help/social/) του μαθήματος και να τα ανεβάσουμε στην [ιστοσελίδα](https://greekforce1821-site.netlify.app/) μας. Συγκεκριμένα, ασχολήθηκα με την βιογραφία: Guido Van Rossum, ο οποίος είναι ο δημιουργός της υψηλού επιπέδου γλώσσας προγραμματισμού Python. Παρακάτω επισυνάπτω την φωτογραφία από το bio.

## Guido Van Rossum Biography



Screenshot\_2022-11-19\_11-29-28

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

* Προσωπικό site από το αποθετήριο μου: [Greekforce1821](https://greekforce1821-site.netlify.app/)
* Biography του προσωπικού αποθετηρίου: [Biography Guido Van Rossum](https://greekforce1821-site.netlify.app//biography/guido-van-rossum/)
* Αρχείο βιογραφίας από το αποθετήριο: [.md File #Biography Thumbnail](https://github.com/Greekforce1821/site/blob/master/_biography/guido-van-rossum.md)
* Αρχείο βιογραφίας από το αποθετήριο: [.md File #Biography](https://github.com/Greekforce1821/site/blob/master/_includes/bio-van-rossum.md)

# Φωτογραφίες που αξιοποιήθηκαν στην παραπάνω εργασία:

* Φωτογραφία thumbnail για την βιογραφία του Guido Van Rossum: [Thumbnail-Guido](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/van-rossum-profile-thumb.jpg)
* Φωτογραφία για την βιογραφία του Guido Van Rossum: [Van-Rossum Picture](https://github.com/Greekforce1821/images/blob/be9e48f02a3c780971691d468f07062e500ef9ec/van-rossum-profile-1.jpg)

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

| Φωτογραφίες | Copyright Holders | License | Type of License |
| --- | --- | --- | --- |
| [Guido Van Rossum #1](https://greekforce1821-site.netlify.app//biography/guido-van-rossum/) | [Φωτογραφία #1](https://gvanrossum.github.io/images/guido-headshot-2019.jpg) | © Michael Cavotta | [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) |
| Guido Van Rossum #2 | [Φωτογραφία #2](https://gvanrossum.github.io/images/guido-portrait-dan-stroud.jpg) | © Dan Stroud | [CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/deed.en) |
| Guido Van Rossum #3 | [Φωτογραφία #3](https://gvanrossum.github.io/images/DO6GvRlo.gif) | © Unknown | [CC BY-NC-ND 4.0](https://creativecommons.org/licenses/by-nc-nd/4.0/) |

| Άρθρο | Πηγή | Copyright Holder |
| --- | --- | --- |
| [Βιογραφικό σημείωμα Guido Van Rossum](https://greekforce1821-site.netlify.app/biography/guido-van-rossum/) | https://bit.ly/3Xfnymx | [Ελληνικό Μουσείο Πληροφορικής](https://elmp.gr/) |

# Παραδοτέο 10: Command Line Exercise #3

Στο συγκεκριμένο παραδοτέο, καλούμαστε να συνεχίσουμε την: Άσκηση Γραμμής εντολών σύμφωνα με τις [οδηγίες](https://github.com/courses-ionio/hci/discussions/1839) του μαθήματος. Μιας και στην προηγούμενη άσκηση γραμμής εντολών υλοποίησα ένα script το οποίο δέχεται σαν είσοδο ένα οποιοδήποτε (.md) αρχείο και το μετατρέπει σε διάφορες μορφές όπως (.docx, .pdf, .html κα.). Επομένως, σε αυτήν την άσκηση επέλεξα να ασχοληθώ με το groff & το SpaceVim . Εγκατέστησα τα ανωτέρω μέσω τερματικού και, με την εντολή: groff > hcireport1 στο τερματικό, πρόσθεσα το κείμενο της αναφοράς μου (με πολύ λιγότερες λεπτομέρειες), δημιουργήθηκε το αρχείο hcireport1 και το έλεγξα με την εντολή: vim hci1report1. Παρακάτω παραθέτω το βίντεο υλοποίησης της αναφοράς μέσω της πλατφόρμας asciinema. Η άσκηση του groff, ήταν έμπνευση από του χρήστη @p18balt, το οποίο ευχαριστώ!

## GROFF and SpaceVim Overview:

# Παραδοτέο 11: Κατασκευή του Βιβλίου #2