



Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Πολυτεχνείο Κρήτης
Ακαδημαϊκό Έτος 2022-2023 (Χειμερινό Εξάμηνο)
Μάθημα: Εργαλεία Ανάπτυξης Λογισμικού &
Προγραμματισμός Συστημάτων

Διδάσκων: Νίκος Γιατράκος

Εργασία 2 (20% της συνολικής βαθμολογίας του μαθήματος)

ΟΜΑΔΕΣ ΙΔΙΕΣ ΜΕ ΤΗΝ ΕΡΓΑΣΙΑ 1

Θέμα: Linux Commands, Bash Scripting, Process Programming

Σκοπός Εργασίας

Σκοπός της εργασίας είναι η εξοικείωσή σας με τις εντολές Linux, τον προγραμματισμό σε Bash Shell και τον προγραμματισμό διεργασιών (Process Programming). Προκειμένου να επιτευχθεί αυτό, θα υλοποιήσετε μια σειρά προγραμμάτων σε θέματα με αντίστοιχη κατανομή στις ενότητες του μαθήματος.

Θέματα Εργασίας

Θέμα 1: Linux Commands (10% του βαθμού της Εργασίας 2)

Σας δίνεται, μαζί με την Εργασία 2, ένα καινούριο tweetsmall.txt αρχείο το οποίο περιλαμβάνει κείμενο από ένα tweet ανά γραμμή. Όχι json αυτή τη φορά. Απλό κείμενο, 1 tweet ανά γραμμή. Στο PLH211_Project2_2022_2023.ipynb template που σας δίδεται, στο Θέμα 1, υπάρχουν κάποια Ερωτήματα 1.1 έως 1.10. Καλείστε να απαντήσετε στα ερωτήματα στα αντίστοιχα code cells του template.

***Περιορισμοί:** Τα ερωτήματα στο Θέμα 1 πρέπει να απαντηθούν στα αντίστοιχα code cells χρησιμοποιώντας MONO Linux Commands (όχι Python, όχι Bash Scripts). Οι εντολές σας θα πρέπει να εκτελούνται ΣΕ ΜΙΑ ΓΡΑΜΜΗ. Μπορείτε να χρησιμοποιήσετε ;, &&, |, >, <, >> για να κάνετε command chaining και ανακατεύθυνση input/output. ΔΕΝ ΕΠΙΤΡΕΠΕΤΑΙ η χρήση εντολών τύπου sed, awk. Απαντήσεις που δεν ακολουθούν αυτούς τους κανόνες, δε βαθμολογούνται.*

Θέμα 2: Twitter Viewer & Editor σε Bash Script (25% του βαθμού της Εργασίας 2)

Σε αυτό το θέμα σας ζητείται να υλοποιήσετε τον Tweeter Viewer & Editor της Εργασίας 1, αλλά αυτή τη φορά MONO σε Bash Script χρησιμοποιώντας το tweetsmall.txt. Θεωρούμε ξανά ότι ο προσδιοριστής του κάθε tweet είναι ο αριθμός γραμμής στην οποία εμφανίζεται.

Οι λειτουργίες που παρέχει ο Tweeter Viewer & Editor παραμένουν ίδιες με την Εργασία 1 και συνοψίζονται στον ακόλουθο πίνακα:

Εντολή	Περιγραφή
c	Create tweet by giving its text. Every created tweet is appended at the end of the rest (in memory and, if saved, in the twitter file). Sets current tweet ID to be the created one.
r <number>	Read the tweet at line/with ID <number>. Sets current tweet ID to be the read one.
u <number>	Update the tweet at line/with ID <number>, by giving its new text. Sets current tweet ID to be the updated one.
d	Delete current tweet (Hint: adjust tweet IDs when needed).
\$	Read the last tweet. Sets current tweet ID to be the last one.
-	Read one tweet up from the current tweet. Updates current tweet ID accordingly.
+	Read one tweet down from the current tweet. Updates current tweet ID accordingly.
=	Print current tweet ID.
q	Quit without save.
w	(Over)write file to disk
x	Exit and save.

Θα πρέπει να κρατάτε με κάποιο τρόπο το current tweet ID και ο χρήστης θα μπορεί να αλλάζει με βάση τις παραπάνω εντολές το current tweet ID.

Όταν θέλουμε να αλλάξουμε (να κάνουμε update) κάποιο tweet, διαγράφουμε την παλιά έκδοση και δημιουργούμε νέα έκδοση, εγγράφουμε τη νέα έκδοση στη σωστή γραμμή/θέση βάσει του ID/γραμμή που βρισκόταν το tweet που αλλάξαμε (στη μνήμη ή αν τα σώσουμε τελικά στο αρχείο).

Περιορισμοί και Οδηγίες:

- Δεν επιτρέπεται να κάνετε κλήση σε κώδικα *Python* μέσα από *Bash Script*.
- Δε χρειάζεται ο κώδικας σας να είναι βελτιστοποιημένος ως προς χρήση μνήμης, ταχύτητα επεξεργασίας κλπ. Αρκεί να είναι λειτουργικά σωστός. Μπορείτε, για παράδειγμα, να φορτώσετε όλα τα tweets στη μνήμη κατά την έναρξη εκτέλεσης του προγράμματός σας.
- Θα πρέπει να φτιάξετε ξεχωριστές μεθόδους για κάθε μια από τις εντολές του *Twitter Viewer & Editor* και να τις παραμετροποιείτε/καλείτε κατάλληλα.
- Προσοχή στο ότι οι αλλαγές του χρήστη δε σώζονται πάντα, αλλά μόνο όταν επιλεγεί «w» ή «x» από τις εντολές σας, όπως και στην Εργασία 1.

Θέμα 3: Sentiment Analyser σε Bash Script (20% του βαθμού της Εργασίας 2)

Μαζί με το tweetsmall.txt, σας δίνονται: ένα αρχείο positive.txt και ένα negative.txt. Τα αρχεία αυτά έχουν λέξεις (μία ανά γραμμή) στις οποίες έχει αποδοθεί θετική ή αρνητική σημασία, αντίστοιχα. Γράψτε ένα πρόγραμμα Bash Script που διαβάζει μια-μια κάθε γραμμή του tweetsmall.txt και για κάθε γραμμή που διαβάζει:

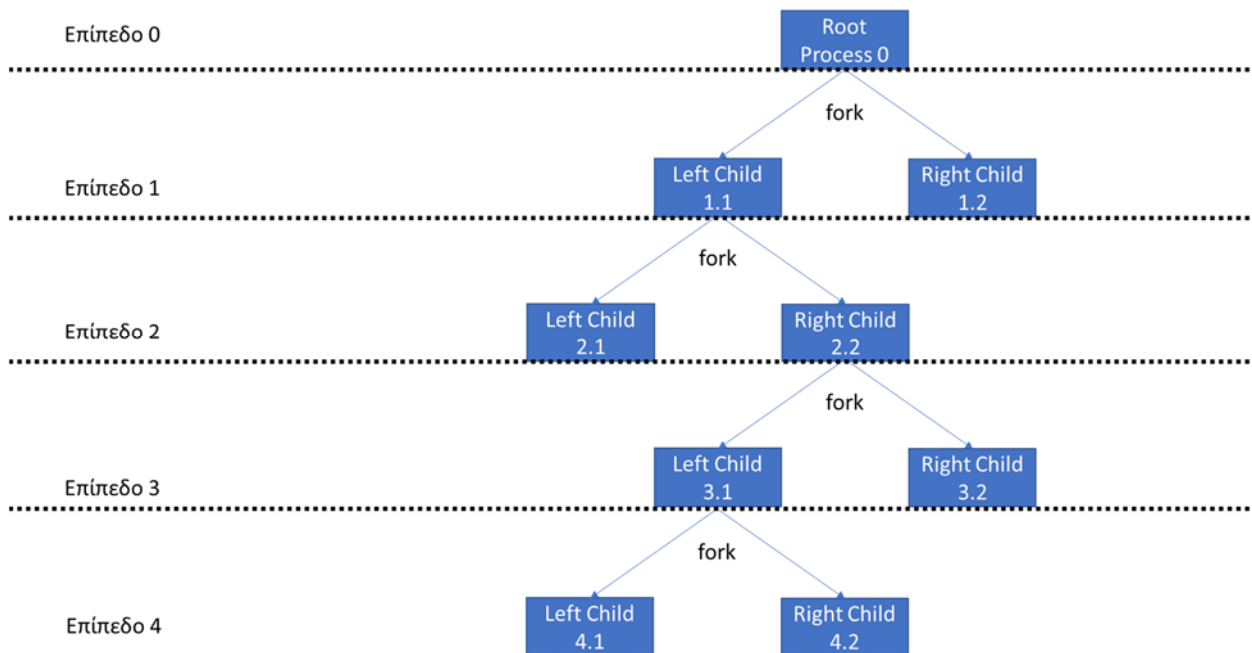
- Μετράει αριθμό θετικών (positive) λέξεων του tweet της γραμμής εκείνης
- Μετράει αριθμό αρνητικών (negative) λέξεων του tweet της γραμμής εκείνης
- Γράφει σε ένα αρχείο sentimentpertweet.txt, για κάθε tweet που επεξεργάζεται, το επικρατές sentiment. Π.χ. για το tweet στη γραμμή 1 του tweetsmall.txt, στο sentimentpertweet.txt θα γραφτεί:
 - Tweet at line #1 has a positive sentiment αν ο αριθμός των positive words στο tweet είναι μεγαλύτερος από τον αριθμό των negative

- Tweet at line #1 has a neutral sentiment αν ο αριθμός των positive words στο tweet είναι ίσος τον αριθμό των negative
- Tweet at line #1 has a negative sentiment αν δεν ισχύει κάτι από τα παραπάνω

Θέμα 4: Python Process Programming - Zig Zag Process Generator (25% του βαθμού της Εργασίας 2)

Στο θέμα αυτό σας ζητείται να γράψετε ένα Zig Zag Process Generator σε Python με χρήση του `os` module. Τι σημαίνει αυτό (ακολουθεί σχηματική αναπαράσταση με εικόνα):

Το πρόγραμμά σας θα ξεκινάει με μια parent διεργασία που θα βρίσκεται στο επίπεδο 0 και θα γεννά (fork) 2 άλλες διεργασίες παιδιά (βλ. εικόνα). Το αριστερό child και το δεξί child. Αυτά τα παιδιά βρίσκονται στο επίπεδο 1. Στο επόμενο επίπεδο μόνο ένα από αυτά τα δύο παιδιά θα κάνει fork. Αν ο αριθμός του τωρινού επιπέδου είναι μονός αριθμός (1, 3, 5, 7, 9), θα κάνει fork μόνο το αριστερό παιδί. Αλλιώς μόνο το δεξί.



Προδιαγραφές:

- Ο αριθμός των επιπέδων (βήματα zig zag) θα περνάει ως παράμετρος από τη γραμμή εντολών κατά την εκτέλεση του προγράμματός σας.
- Ο αριθμός των επιπέδων (βήματα zig zag) δεν επιτρέπεται να υπερβαίνει τα 10.
- Πρέπει να κάνετε κατάλληλο χειρισμό εξαιρέσεων στο πρόγραμμά σας.
- Κάθε forked process θα πρέπει να τυπώνει στην οθόνη μια γραμμή με το εξής μήνυμα (τα PIDs είναι παραδειγματικά):

I am process with PID = 11 standing at Level 1 of the zig zag path, my parent is PID=1002 and I am its left child

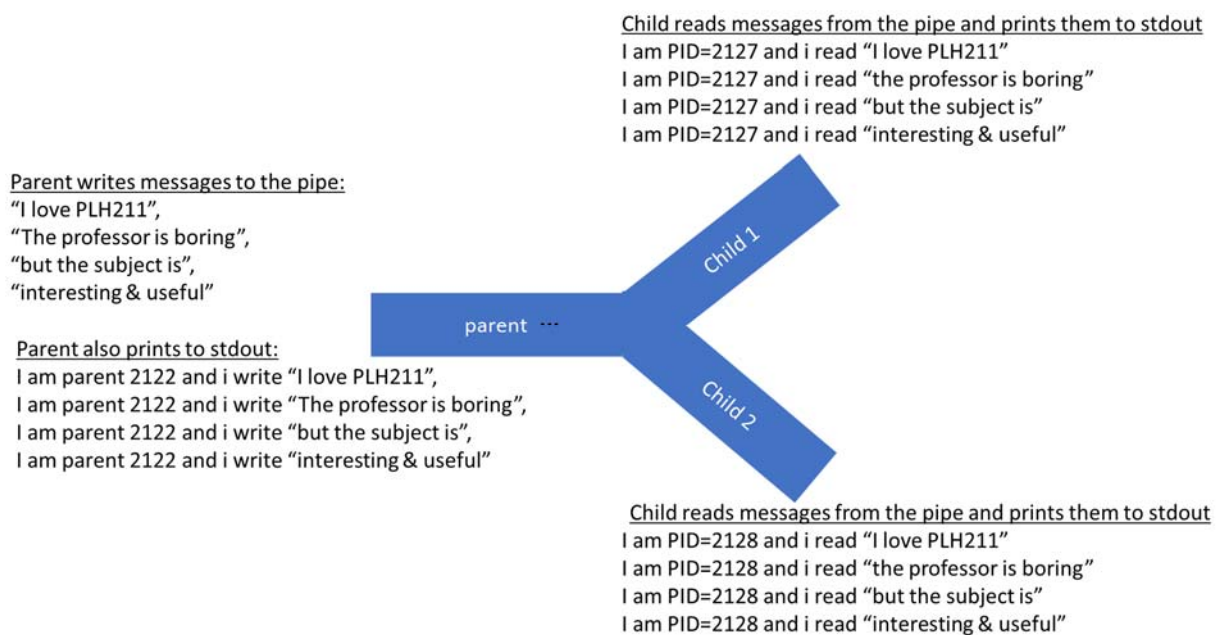
I am process with PID = 12 standing at Level 1 of the zig zag path, my parent is PID=1002 and I am its right child

...

Υποδείξεις: εκμεταλλευτείτε κατάλληλα το συνδυασμό `fork` και κάποιας από τις `exec*()`.

Θέμα 5: Python Pipes: Three-way Pipe Implementation (20% του βαθμού Εργασίας 2)

Τα three-way pipes είναι pipes στα οποία υπάρχει ένα parent process το οποίο δέχεται (στη δική μας περίπτωση από ένα tuple) μηνύματα ως input και τα κάνει pipe σε δύο παιδιά του. Σε αυτή την περίπτωση το parent process βρίσκεται στην πλευρά του write και τα δύο παιδιά του στην πλευρά του read του pipe. Ότι γράφει το parent process ως περιεχόμενο από το tuple (ένα-ένα στοιχείο του tuple), μεταφέρεται σε κάθε ένα από τα παιδιά του (Δλδ και τα δύο παιδιά διαβάζουν ακριβώς τα ίδια πράγματα). Αυτή η περιγραφή αποτυπώνεται στην επόμενη εικόνα.



Γράψτε ένα Python πρόγραμμα που υλοποιεί αυτό το three-way pipe. Πρέπει να κάνετε κατάλληλο χειρισμό εξαιρέσεων στο πρόγραμμά σας και να ανοίξετε, κλείσετε τους `file descriptors` σε κατάλληλα σημεία του κώδικά σας. Στο πρόγραμμά σας, το parent process θα τυπώνει στην οθόνη τα στοιχεία που θα διαβάζει από ένα tuple και θα κάνει write στη μια άκρη του pipe. Τα παιδιά θα τυπώνουν τα μηνύματα που έκαναν read από το pipe. Στο πρόγραμμά σας, θα πρέπει να εκτυπώνονται αντίστοιχες γραμμές με αυτές που φαίνονται στην παραπάνω εικόνα για parent και children.

Περιορισμοί: Η υλοποίησή σας επιτρέπεται να κάνει `import MONO os` και `sys modules`. Δεν μπορείτε να κάνετε άλλα imports και δε μπορείτε να χρησιμοποιήσετε άλλα modules.

Βοήθεια: Δείτε την `dup2` (<https://docs.python.org/3/library/os.html>)

Παραδοτέα και Υποβολή

- Ένας από την ομάδα σας, θα κατεβάσει και θα αποθηκεύσει το αρχείο

PLH211_Project2_2022_2023.ipynb στον υπολογιστή του από την ενότητα Εργασίες του eclass του μαθήματος: <https://www.eclass.tuc.gr/modules/work/?course=HMMY120>

- **Πρώτη του δουλειά** μετά από αυτό θα είναι να ανεβάσει το PLH211_Project2_2022_2023.ipynb στο δικό του Colab και *****NA TO METONOMΑΣΕΙ ΟΠΩΣΔΗΠΟΤΕ***** με τους Αριθμούς Μητρώου των μελών της ομάδας σας χωρισμένους με underscore. Αν για παράδειγμα η ομάδα σας έχει το Φοιτητή 1 με AM:11111, το Φοιτητή 2 με AM:22222 και το Φοιτητή 3 με AM:33333, το αρχείο σας θα πρέπει να μετονομαστεί σε 11111_22222_33333.ipynb
- Στη συνέχεια το μέλος της ομάδας θα κάνει share από το Google Drive του (εκεί αποθηκεύονται έτσι κι αλλιώς by default τα Colab Notebooks) με τα υπόλοιπα μέλη της ομάδας σας. Κάθε μέλος μπορεί να δημιουργήσει copy του .ipynb αρχείου αυτού αν θέλει.
- Οι ομάδες σε αυτή την εργασία παραμένουν ίδιες με την πρώτη εργασία.
- **Όταν τελειώσετε την εργασία σας, κάθε ομάδα θα ανεβάσει ΤΟ ΕΝΑ ΚΑΙ ΤΕΛΙΚΟ .ipynb αρχείο στο eclass. Αυτό είναι το παραδοτέό σας.** Όλος ο κώδικας και η τεκμηρίωση της εργασίας σας θα πρέπει να είναι σε code cells και text cells αυτού του .ipynb αρχείου. Το .ipynb αρχείο είναι δομημένο ώστε να λειτουργεί ως template για τη δομή των απαντήσεών σας. Ακολουθήστε αυτό το template.
- Στο Colab θα είστε έτοιμοι να δουλέψετε όλοι μαζί επί της εργασίας σας, αλλά προσοχή, όχι ταυτόχρονα. Το Colab δεν είναι Git ή Github (θα τα γνωρίσουμε στη συνέχεια στο μάθημα). Δεν μπορείτε να δουλεύετε όλοι μαζί ταυτόχρονα στο ίδιο .ipynb αρχείο (θα σώζει τις αλλαγές ενός, στους άλλους θα δίνει warnings ότι κάποιος άλλος άλλαξε το αρχείο).

Τρόπος Βαθμολόγησης

Σε κάθε ένα από τα Θέματα 1 έως 5 υποδεικνύεται η γλώσσα που θα προγραμματίσετε και ο τρόπος (οδηγίες, υποδείξεις, περιορισμοί) που θα πρέπει να λυθούν. Λύσεις που έχουν κώδικα σε άλλη γλώσσα από αυτή που ζητείται στο αντίστοιχο θέμα ή δεν ακολουθούν τις οδηγίες, δε βαθμολογούνται.

Η έμφαση στη βαθμολόγηση, είναι σε σωστή υλοποίηση. Ως εκ τούτου:

Ο κώδικας των ομάδων ελέγχεται με λογισμικού εντοπισμού αντιγραφών. Ομάδες που φαίνεται να έχουν αντιγράψει οποιοδήποτε μέρος της εργασίας μηδενίζονται (όλα τα μέλη και των δύο ομάδων, όλα τα θέματα - ανεξαρτήτως ποιος αντέγραψε από ποιόν).

Κώδικας που δεν κάνει compile, δίνει runtime errors ή δεν είναι σωστός ως προς τη λειτουργικότητα που ζητείται να αποδοθεί, βαθμολογείται με έως μηδενικό βαθμό ακόμη και αν η τεκμηρίωση και ο σχολιασμός του είναι αkéραια.

Όλες οι άλλες περιπτώσεις (ελλιπής αλλά σωστή λειτουργικότητα, ελλιπής αλλά καταληπτή τεκμηρίωση κλπ) που δεν εμπίπτουν παραπάνω, αξιολογούνται ξεχωριστά ανάλογα με την περίπτωση.

Σε αντίθεση με την Εργασία 1, **στην Εργασία 2, προκειμένου να μάθετε και να**

εξασκηθείτε όλοι σε όλα τα σχετικά θέματα (Linux Commands, Bash Script, Process Programming) θα πρέπει να δουλέψετε όλα τα μέλη της ομάδας μαζί σε όλα τα θέματα **ΚΑΙ θα εξεταστείτε προφορικά ΟΛΟΙ σε ΟΛΑ τα θέματα της Εργασίας 2.**

Διαδικαστικά

- **Αριθμός Μελών Ομάδας Εργασίας:** Οι ομάδες παραμένουν ακριβώς ίδιες με την πρώτη εργασία.
- **Ημερομηνία Παράδοσης Εργασίας:** έως τα μεσάνυχτα της **05/01/2023**
- **Τρόπος Παράδοσης:** Ομαδική υποβολή μέσω eclass σε ένα .ipynb αρχείο με κατάλληλο όνομα όπως περιγράφεται παραπάνω.
- **Ημερομηνία Προφορικής Εξέτασης:** Κατόπιν ανακοίνωσης, το πιθανότερο εντός της εβδομάδας 09/01/2023 – 16/01/2023, θα εξεταστείτε ΚΑΙ επί των 2 εργασιών που υλοποιήσατε, εκτελώντας demo λειτουργίας της υλοποίησης των εργασιών σας και του κώδικά σας. Για το λόγο αυτό **οι ομάδες ΔΕ μπορούν να αλλάξουν από την 1^η στη 2^η εργασία.**
- Η εργασία ισχύει για την εξεταστική Ιανουαρίου και Σεπτεμβρίου (δηλαδή δεν υπάρχει επαν-υποβολή – βελτίωση για το Σεπτέμβρη). Δεν κρατείται βαθμός εργασιών για το επόμενο ακαδημαϊκό έτος.

Ερωτήσεις/Απορίες Σχετικά με την Εργασία

Για τις ερωτήσεις - απορίες σας μπορείτε

- να χρησιμοποιείτε την αντίστοιχη κατηγορία στην περιοχή συζητήσεων του μαθήματος στο eclass:
<https://www.eclass.tuc.gr/modules/forum/viewforum.php?course=HMMY120&forum=40232>
- να απευθύνεστε στο διδάσκοντα μέσω email:
 - ο Θέμα Email : Project2PLH211 – ONOMATEΠΩΝΥΜΟ & ΑΜ φοιτητή
 - ο Παραλήπτης: **ngiatrakos@tuc.gr**

Καλή Δουλειά!