



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 (15% του βαθμού της Εργασίας 2)

Σας δίνονται, μαζί με την Εργασία 2, δύο αρχεία `temperature.txt` και `humidity.txt`. Αυτά τα αρχεία περιλαμβάνουν μετρήσεις θερμοκρασίας και υγρασίας αντίστοιχα, από ένα δίκτυο αισθητήρων (sensor network). Και τα δύο αρχεία είναι οργανωμένα ως εξής:

- Κάθε στήλη ενός αρχείου έχει τις μετρήσεις θερμοκρασίας και υγρασίας, αντίστοιχα, ενός sensor. Άρα η στήλη 5 του `temperature.txt` έχει τις μετρήσεις θερμοκρασίας του sensor 5 και η στήλη 5 του `humidity.txt` έχει τις μετρήσεις υγρασίας του sensor 5.
- Κάθε γραμμή ενός αρχείου αντιστοιχεί σε ένα γύρο δειγματοληψίας (epoch) μετρήσεων θερμοκρασίας και υγρασίας αντίστοιχα, για όλους τους sensors. Άρα για παράδειγμα η γραμμή 10 του `temperature.txt` έχει όλες τις μετρήσεις θερμοκρασίας για το sensor network κατά το epoch 10 και η γραμμή 10 του `humidity.txt` έχει όλες τις μετρήσεις υγρασίας για το sensor network κατά το epoch 10.
- Σε κάθε γραμμή οι μετρήσεις διαφορετικών sensors είναι χωρισμένες με κενό (white space).

Στο `PLH211_Project2_2023_2024.ipynb` template που σας δίδεται, στο Θέμα 1, υπάρχουν κάποια Ερωτήματα 1.1 έως 1.12. Καλείστε να απαντήσετε με τα Linux Commands που ζητούνται στα ερωτήματα στα αντίστοιχα code cells του template.

Περιορισμοί: Τα ερωτήματα στο Θέμα 1 πρέπει να απαντηθούν στα αντίστοιχα code cells χρησιμοποιώντας ΜΟΝΟ 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: Aggregative News Analyzer σε Bash Script (30% του βαθμού της Εργασίας 2)

Σε αυτό το θέμα σας ζητείται να υλοποιήσετε τον Aggregative News Analyzer της Εργασίας 1, αλλά αυτή τη φορά MONO σε Bash Script.

Οι λειτουργίες που θα παρέχει ο Aggregative News Analyzer συνοψίζονται στον ακόλουθο πίνακα:

Εντολή	Περιγραφή
@ <category> <k>	Δείξε μου τα <k> πιο σχετικά (βάσει Jaccard Index) stems (όχι tid) για την κατηγορία <category>
# <stem> <k>	Δείξε μου τα <k> πιο σχετικά (βάσει Jaccard Index) categories για το <stem>
\$ <stem> <category>	Δείξε μου το Jaccard Index του συγκεκριμένου (stem, category) ζεύγους
P <did> -c -t	Αν στην εντολή χρησιμοποιείται το switch “-t” τότε δείξε μου όλα τα stem που περιέχει το document με κωδικό did. Αν στην εντολή χρησιμοποιείται το switch “-c” τότε δείξε μου όλες τις κατηγορίες που έχουν ανατεθεί στο document με κωδικό did.
C <did> -c -t	Αν στην εντολή χρησιμοποιείται το switch “-t” μέτρησε και δείξε μου στην οθόνη πόσα (count) μοναδικά terms περιέχει το document με κωδικό did. Αν στην εντολή χρησιμοποιείται το switch “-c” τότε μέτρησε και δείξε μου στην οθόνη πόσες (count) κατηγορίες έχουν ανατεθεί στο document με κωδικό did.

Περιορισμοί και Οδηγίες:

- Δεν επιτρέπεται να κάνετε κλήση σε κώδικα Python μέσα από Bash Script.
- Μπορείτε να χρησιμοποιήσετε ικανά υποσύνολα των αρχείων που σας έχουν δοθεί και υποσύνολα των γραμμών του κάθε αρχείου επίσης.
- 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 τα δεδομένα στη μνήμη κατά την έναρξη εκτέλεσης του προγράμματός σας.
- Θα πρέπει να φτιάξετε ξεχωριστές μεθόδους για κάθε μια από τις εντολές του Aggregative News Analyzer και να τις παραμετροποιείτε/καλείτε κατάλληλα.

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

Μαζί με την εργασία, σας δίνονται ακόμη ένα αρχείο positive_stems.txt και ένα negative_stems.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 ένα-ένα κάθε document (δλδ μια-μια κάθε γραμμή) του lyl2004_vectors_train.dat.txt (υπενθύμιση: το document id είναι η πρώτη τιμή σε κάθε γραμμή) και:

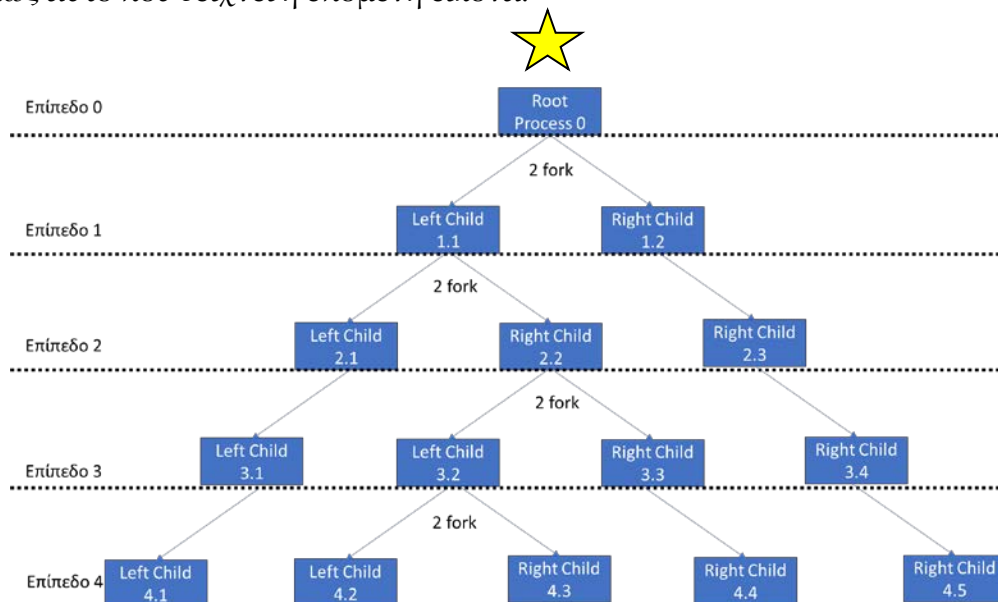
- Αντιστοιχεί τα term ids που περιέχει με τα stems στο stem.termid.idf.map.txt
- Μετράει αριθμό θετικών (positive) stem του document της γραμμής εκείνης
- Μετράει αριθμό αρνητικών (negative) stem του document της γραμμής εκείνης
- Γράφει σε ένα αρχείο sentimentpernewsitem.txt, για κάθε document που

επεξεργάζεται, το επικρατές sentiment. Π.χ. για το document στη γραμμή 1 του `lyrl2004_vectors_train.dat.txt`, στο `sentimentpertweet.txt` θα γραφτεί:

- ο “Document 2286 has a positive sentiment” αν ο αριθμός των positive stem στο document είναι μεγαλύτερος από τον αριθμό των negative
- ο “Document 2286 has a neutral sentiment” αν ο αριθμός των negative stem στο document είναι ίσος με τον αριθμό των positive (ακόμα και αν είναι και οι δύο μηδέν)
- ο “Document 2286 has a negative sentiment” αν δεν ισχύει κάτι από τα παραπάνω

Θέμα 4: Python Process Programming – Christmas Tree Process Generator (20% του βαθμού της Εργασίας 2)

Στο θέμα αυτό σας ζητείται να γράψετε ένα Christmas Tree Process Generator σε Python με χρήση των `os` και `sys` module. Το δέντρο διεργασιών που θα φτιάξετε θα πρέπει να είναι όπως αυτό που δείχνει η επόμενη εικόνα:



Προδιαγραφές:

- Δε θα πρέπει να φαίνεται στο output διεργασία με Parent id : 1
- Κάθε forked διεργασία θα πρέπει να τυπώνει στην οθόνη μια γραμμή όπως φαίνεται παρακάτω (τα PIDs είναι παραδείγματα):

```
I am Parent[0] of all, my PID is 107736
I am Child[1.1] with pid: 107738 and my Parent id: 107736
I am Child[1.2] with pid: 107739 and my Parent id: 107736
I am Child[2.1] with pid: 107740 and my Parent id: 107738
I am Child[2.3] with pid: 107741 and my Parent id: 107739
I am Child[3.1] with pid: 107742 and my Parent id: 107740
I am Child[3.4] with pid: 107743 and my Parent id: 107741
I am Child[4.1] with pid: 107744 and my Parent id: 107742
I am Child[4.5] with pid: 107745 and my Parent id: 107743
I am Child[2.2] with pid: 107746 and my Parent id: 107738
I am Child[3.2] with pid: 107747 and my Parent id: 107746
I am Child[4.2] with pid: 107748 and my Parent id: 107747
I am Child[4.3] with pid: 107749 and my Parent id: 107747
I am Child[3.3] with pid: 107750 and my Parent id: 107746
I am Child[4.4] with pid: 107751 and my Parent id: 107750
```

Υποδείξεις:

- Χρησιμοποιήστε κατάλληλα τις fork, wait και sleep.

Θέμα 5: Python Pipes: Mixed Piping Implementation (15% του βαθμού Εργασίας 2)

Έχουμε ένα source process που μεταδίδει πληροφορία διαβάζοντας elements από ένα tuple όπως φαίνεται στην παρακάτω εικόνα και γράφοντάς τα αντίστοιχα elements ένα-ένα σε ένα anonymous pipe.

Ένα δεύτερο process, διαδραματίζει το ρόλο του transformer process, όπου διαβάσει όσα γράφει το source process στο anonymous pipe, κάνει tokenize (ξεχωρίζει τις λέξεις βάσει white spaces) τα μηνύματα και γράφει το αποτέλεσμα σε ένα FIFO (aka named pipe).

Ένα τελευταίο output process,, διαβάσει όσα έχουν γραφτεί στο named pipe και τα τυπώνει στην οθόνη.

Υλοποιήστε σε Python την παραπάνω inter-process επικοινωνία.

Περιορισμοί:

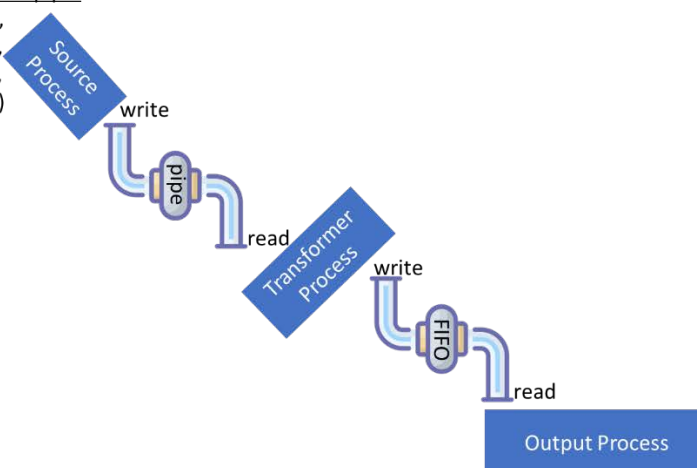
- Η υλοποίησή σας επιτρέπεται να κάνει import μόνο os, sys και time modules. Κανένα άλλο module.
- Πρέπει να κάνετε κατάλληλο χειρισμό εξαιρέσεων στο πρόγραμμά σας.
- Κανένα pipe ή process δε θα πρέπει να κάνει hang όταν τελειώσει τη δουλειά του. Με άλλα λόγια το πρόγραμμά σας πρέπει να τερματίζεται μόνο του ομαλά.

Υποδείξεις:

- Κάθε μήνυμα στο tuple MESSAGES όπως σας δίδεται στο ipynb είναι 23 bytes. Μην αλλάξετε τα περιεχόμενα του tuple.

Source Process writes the messages of this tuple to the anonymous pipe:

```
("I love PLH211",  
"The professor is boring",  
"but the curriculum is",  
"interesting & useful")
```



Output Process On screen:

```
I  
love  
PLH211  
The  
Professor  
Is  
boring,  
but  
the  
curriculum  
is  
Interesting  
&  
useful
```

Παραδοτέα και Υποβολή

- Ένας από την ομάδα σας, θα κατεβάσει και θα αποθηκεύσει το αρχείο PLH211_Project2_2023_2024.ipynb στον υπολογιστή του από την ενότητα Εργασίες του eclass του μαθήματος: <https://www.eclass.tuc.gr/courses/HMMY314/>
- Πρώτη του δουλειά μετά από αυτό θα είναι να ανεβάσει το PLH211_Project2_2023_2024.ipynb στο δικό του Colab και *****ΝΑ ΤΟ ΜΕΤΟΝΟΜΑΣΕΙ 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 αkéραιοι.

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 **04/01/2024**.
- **Τρόπος Παράδοσης:** Ομαδική υποβολή μέσω eclass σε ένα .ipynb αρχείο με κατάλληλο όνομα όπως περιγράφεται παραπάνω.
- **Ημερομηνία Προφορικής Εξέτασης:** Κατόπιν ανακοίνωσης, εντός της εβδομάδας 08/01/2024 – 16/01/2024, 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=HMMY314&forum=47633>
- να απευθύνεστε στο διδάσκοντα μέσω email:
 - ο Θέμα Email : Project2PLH211 – ΟΝΟΜΑΤΕΠΩΝΥΜΟ & ΑΜ φοιτητή
 - ο Παραλήπτης: **ngiatrakos@tuc.gr**

Καλή Δουλειά!