Λειτουργικά Συστήματα

Τμήμα Πληροφορικής
Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

1η Εργασία
Χειμερινό Εξάμηνο 2021

Γενικές οδηγίες

Στόχος της εργασίας είναι η εξοικείωσή σας με τη γραμμή εντολών του Linux και με τη δημιουργία σεναρίων κελύφους (shell scripts) τα οποία μπορείτε να χρησιμοποιήσετε για τον αυτοματισμό εργασιών.

Για την παράδοση της εργασίας θα δημιουργήσετε ένα **private** αποθετήριο git, όπου θα τοποθετήσετε τα scripts που ζητούνται από τις δύο ασκήσεις της εργασίας. **ΠΡΟΣΟΧΗ**: αν το repository είναι public αποτελεί αιτία μηδενισμού της εργασίας σας!

Ακολουθήστε τις σχετικές οδηγίες με ονομασία «Οδηγίες δημιουργίας Git repository για την 1η εργασία» που βρίσκονται στην πλατφόρμα elearning.

Η ολοκλήρωση των παραπάνω θα πρέπει να έχει πραγματοποιηθεί μέχρι την Κυριακή 31/10/2021 και ώρα 23:59. Σε αντίθετη περίπτωση, δεν θα βαθμολογηθεί η 1η εργασία.

Για την υποβολή της εργασίας σας δημιουργήστε ένα αρχείο με όνομα

opsys2021-assignment1-1234.txt

(αντικαθιστώντας το 1234 με τον αριθμό μητρώου σας). Αυτό το αρχείο θα πρέπει να υποβάλετε στην ιστοσελίδα του μαθήματος στην πλατφόρμα elearning μέχρι τη λήξη της ημερομηνίας υποβολής της εργασίας σας. Το αρχείο θα πρέπει να περιέχει **μόνο** τη διεύθυνση του git repository σας, πχ

```
https://github.com/yourusername/assignment1
```

Δεν πρέπει να υποβάλετε κανένα άλλο αρχείο εκτός από αυτό. Ο κώδικας που υποβάλετε θα βρίσκεται στο git repository σας.

Μέσα στο git repository σας δημιουργήστε φακέλους με όνομα *askisi1*, *askisi2* και *askisi3* όπου και θα προσθέσετε τα αντίστοιχα αρχεία για τα τρία scripts που σας ζητούνται.

Δεν επιτρέπεται για καμία από τις τρεις ασκήσεις η συγγραφή κώδικα σε κάποια άλλη γλώσσα προγραμματισμού (c, python, perl, awk...). Η επίλυση των ασκήσεων θα πρέπει να

πραγματοποιηθεί μόνο με τη χρήση Bash scripts και των εργαλείων που υπάρχουν προεγκατεστημένα στις περισσότερες διανομές Linux. Σε περίπτωση που δεν είστε σίγουροι για τη χρήση κάποιου εργαλείου, μπορείτε να ρωτήσετε.

Απορίες σχετικά με την εργασία μπορείτε να υποβάλετε μέσω της πλατφόρμας elearning στο αντίστοιχο forum συζητήσεων.

1 Έλεγχος ενημέρωσης ιστοσελίδων

Σκοπός της άσκησης είναι η δημιουργία ενός bash script, το οποίο θα διαβάζει μια λίστα με διευθύνσεις ιστοσελίδων και θα ελέγχει αν κάποια ή κάποιες από αυτές έχουν ενημερωθεί σε σχέση με την προηγούμενη φορά που τρέξατε το script.

Η λίστα με τις διευθύνσεις των ιστοσελίδων θα βρίσκεται σε ένα απλό αρχείο κειμένου (επιλέξτε εσείς ένα κατάλληλο όνομα), το οποίο θα δίνεται ως παράμετρος στο script. Παράδειγμα περιεχομένου του αρχείου αυτού είναι το παρακάτω:

```
# List of addresses
http://www.tldp.org
https://en.wikipedia.org/wiki/Linux
https://www.gnu.org/software/bash/
```

Ο αριθμός των διευθύνσεων ιστοσελίδων που περιλαμβάνονται στο αρχείο μπορεί να είναι μία ή πολλές περισσότερες, χωρίς να υπάρχει κάποιο όριο. Αν υπάρχουν γραμμές οι οποίες ξεκινούν με τον χαρακτήρα #, θα πρέπει να τις αγνοείτε ως σχόλια.

Εκτελώντας το script, αυτό θα πρέπει να διαβάζει τις διευθύνσεις των ιστοσελίδων και να ελέγχει αν αυτές έχουν ενημερωθεί. Αν τα περιεχόμενα μιας ιστοσελίδας έχουν παραμείνει τα ίδια, το script *δεν πρέπει να τυπώνει τίποτα*. Αν όμως τα περιεχόμενα μιας ιστοσελίδας έχουν τροποποιηθεί σε σχέση με την προηγούμενη φορά που εκτελέστηκε το script, αυτό θα πρέπει να τυπώνει στην *τυπική έξοδο (stdout)*, τη διεύθυνση της ιστοσελίδας, π.χ.:

```
http://www.tldp.org
```

Αν είναι η πρώτη φορά που το script σας «βλέπει» μια ιστοσελίδα, τότε το script θα πρέπει να τυπώνει στην *τυπική έξοδο (stdout)*, τη διεύθυνση της ιστοσελίδας, μαζί με το μήνυμα *INIT*, π.χ.:

```
http://www.tldp.org INIT
```

Προφανώς την πρώτη φορά που θα εκτελέσετε το script και εφόσον δεν υπάρχει προηγούμενη πληροφορία για οποιαδήποτε ιστοσελίδα, το παραπάνω μήνυμα θα πρέπει να τυπώνεται για κάθε ιστοσελίδα.

Αν δεν είναι δυνατή η ανάγνωση των περιεχομένων μιας ιστοσελίδας, λόγω σφάλματος (δικτύου ή για οποιοδήποτε άλλο λόγο), θα πρέπει το script να τυπώνει στην *έξοδο σφαλμάτων (stderr)*, αντίστοιχο μήνυμα, π.χ.:

```
http://www.tldp.org FAILED
```

Παραδοχές που μπορείτε να κάνετε:

- Οι ιστοσελίδες που θα ελέγχετε θα αποτελούνται μόνο από στατικό περιεχόμενο. Ιστοσελίδες που ανανεώνονται δυναμικά με τη χρήση Javascript δεν σας απασχολούν.
- Αν δεν ήταν δυνατή η ανάγνωση των περιεχομένων μιας ιστοσελίδας την προηγούμενη φορά, αλλά στην τρέχουσα εκτέλεση είναι δυνατή η ανάγνωσή τους, θεωρήστε ότι τα περιεχόμενά της άλλαξαν.

- Αν δεν ήταν δυνατή η ανάγνωση των περιεχομένων μιας ιστοσελίδας την προηγούμενη φορά και δεν είναι επίσης δυνατή και στην τρέχουσα εκτέλεση, θεωρήστε ότι τα περιεχόμενά της δεν άλλαξαν.

Υποβάλετε δύο εκδόσεις του script σας. Η πρώτη έκδοση θα βρίσκεται σε ένα αρχείο με την ονομασία *script1a.sh* και θα ελέγχει τις ιστοσελίδες διαδοχικά, από την πρώτη μέχρι την τελευταία. Η δεύτερη έκδοση θα βρίσκεται σε ένα αρχείο με την ονομασία *script1b.sh* και θα ελέγχει όλες τις ιστοσελίδες *ταυτόχρονα*. Τοποθετήστε και τα δύο scripts στο αποθετήριο git που δημιουργήσατε για την εργασία σας.

Συμβουλές:

- Χρησιμοποιήστε τις εντολές curl ή wget για τη λήψη των ιστοσελίδων.
- Κάθε φορά που τρέχει το script, αποθηκεύστε την τρέχουσα κατάσταση κάθε ιστοσελίδας.
- Η τρέχουσα κατάσταση κάθε ιστοσελίδας μπορεί να είναι ολόκληρο το περιεχόμενό της, ή κάποιο άθροισμα των περιεχομένων της (π.χ. md5sum).

2 Κατέβασμα εργασιών με το Git

Σκοπός της άσκησης είναι η δημιουργία ενός bash script, το οποίο θα μπορεί να κατεβάσει όλες τις εργασίες που θα παραδοθούν από εσάς και τους συναδέλφους σας στα πλαίσια της παρούσας εργασίας. Έτσι, το script θα διαβάζει διευθύνσεις αποθετηρίων git, από πολλαπλά αρχεία, θα κατεβάζει το περιεχόμενο των αποθετηρίων και θα ελέγχει αν οι κατάλογοι και τα αρχεία που βρίσκονται σε κάθε ένα από τα αποθετήρια ακολουθούν την απαραίτητη δομή. Για κάθε ένα από αυτά, το script θα πρέπει να παρουσιάζει αναφορά για τα ευρήματά του.

Περισσότερο συγκεκριμένα, υποθέστε ότι σας παρέχεται ένα συμπιεσμένο αρχείο, τύπου .tar.gz. Μέσα σε αυτό το αρχείο, υπάρχει απροσδιόριστος αριθμός απλών αρχείων κειμένου, (τύπου .txt). Θεωρήστε επίσης ότι τα αρχεία ενδέχεται να βρίσκονται σε απροσδιόριστους καταλόγους μέσα στο συμπιεσμένο αρχείο, σε οποιοδήποτε βάθος. Ένα παράδειγμα της δομής των αρχείων μέσα στο συμπιεσμένο αρχείο .tar.gz θα μπορούσε να είναι το παρακάτω:

```
|- fileA.txt
|- fileB.txt
|- directoryA/
   |- anotherfile.txt
   |- notatxtfile.doc
|- anotherdirectory/
   |- nesteddirectory/
   |  |- fileinnesteddir.txt
   |- yetanotherfile.txt
...
```

Μέσα στο συμπιεσμένο αρχείο, ενδεχομένως υπάρχουν και άλλα αρχεία, διαφορετικού τύπου από απλών αρχείων κειμένου (όπως το αρχείο *notatxtfile.doc* στο παραπάνω παράδειγμα). Αυτά τα αρχεία θα πρέπει να τα αγνοείτε κατά την επεξεργασία σας.

Κάθε αρχείο txt περιέχει τη διεύθυνση ενός αποθετηρίου git σε https μορφή, π.χ.:

```
https://github.com/username/repo-name.git
```

Αν υπάρχουν γραμμές οι οποίες ξεκινούν με τον χαρακτήρα #, να τις θεωρήσετε ως γραμμές σχολίων και να τις απορρίψετε. Θα πρέπει να διαβάζετε και να επεξεργάζεστε μόνο τις γραμμές που ξεκινούν με `https`, ενώ αν υπάρχουν πολλαπλές τέτοιες γραμμές σε κάποιο αρχείο, θα πρέπει να χρησιμοποιείτε μόνο την πρώτη και να απορρίπτετε τις επόμενες.

Για κάθε αποθετήριο git που εντοπίζετε στα αρχεία txt, θα πρέπει να το κλωνοποιήσετε και να το τοποθετήσετε μέσα σε ένα κατάλογο με το όνομα *assignments* (τον οποίο κατάλογο θα πρέπει να βεβαιωθείτε ότι υπάρχει ή έχετε δημιουργήσει). Για κάθε αποθετήριο, αν η κλωνοποίησή του είναι επιτυχής, το script σας θα πρέπει να τυπώνει μια γραμμή στην *τυπική έξοδο (stdout)* όπως η παρακάτω:

```
https://github.com/username/repo-name.git: Cloning OK
```

ενώ αν για οποιοδήποτε λόγο αποτύχει η κλωνοποίησή του, θα πρέπει να τυπώνει στην *έξοδο σφαλμάτων (stderr)*, μια γραμμή όπως η παρακάτω:

```
https://github.com/username/repo-name.git: Cloning FAILED
```

Οποιαδήποτε άλλη έξοδος δημιουργείται από την εκτέλεση της εντολής `git` ή οποιαδήποτε άλλη, θα πρέπει να την απορρίπτετε.

Σε οποιαδήποτε περίπτωση, αφού κλωνοποιηθούν όλα τα αποθετήρια, θα έχετε δημιουργήσει μια δομή καταλόγων που μοιάζει με την παρακάτω, χωρίς τα ονόματα των καταλόγων να είναι προφανώς τα ίδια:

```
|- assignments/
   |- repo1
   |- repo2
   |- repo3
   ...
```

Στη συνέχεια το script σας θα πρέπει να εξετάζει αν μέσα σε κάθε ένα από τα αποθετήρια που κλωνοποιήσατε, η δομή των αρχείων είναι η απαιτούμενη. Αυτή για κάθε αποθετήριο θα πρέπει να είναι ακριβώς η παρακάτω, με τα ονόματα των αρχείων και καταλόγων να είναι τα ίδια με τα παρακάτω (τα οποία προφανώς δεν αντιστοιχούν ακριβώς με αυτά της παρούσας εργασίας):

```
|- dataA.txt
|- more/
   |- dataB.txt
   |- dataC.txt
```

Για κάθε αποθετήριο, το script σας θα πρέπει να τυπώνει συγκεντρωτικά αποτελέσματα για τον αριθμό των καταλόγων και τον συνολικό αριθμό των αρχείων που περιέχει. Έτσι, π.χ. αν υπάρχει ένα αποθετήριο με το όνομα `repo1` και δεδομένου ότι αυτό ακολουθεί την παραπάνω δομή, θα πρέπει να τυπώνει στην *τυπική έξοδο (stdout)*:

```
repo1:
Number of directories: 1
Number of txt files: 3
Number of other files: 0
```

Τέλος, θα πρέπει να ελέγχει αν ο κατάλογος `more` και τα αρχεία `dataA.txt`, `dataB.txt` και `dataC.txt` βρίσκονται στις σωστές τοποθεσίες και με ακριβώς τις σωστές ονομασίες. Αν όλα είναι εντάξει, τότε θα τυπώνει στην *τυπική έξοδο (stdout)*:

```
Directory structure is OK.
```

Σε αντίθετη περίπτωση θα πρέπει να τυπώνει στην *έξοδο σφαλμάτων (stderr)*:

```
Directory structure is NOT OK.
```

Ένα ολοκληρωμένο παράδειγμα εξόδου, για τρία αποθετήρια, το ένα εκ των οποίων δεν μπορεί αν κλωνοποιηθεί σωστά, ενώ μόνο το πρώτο από τα άλλα δύο έχει τη σωστή δομή, είναι το παρακάτω:

```
https://github.com/username1/repo1.git: Cloning OK
https://github.com/username2/repo2.git: Cloning FAILED
https://github.com/username3/repo3.git: Cloning OK
repo1:
Number of directories: 1
Number of txt files: 3
Number of other files: 0
Directory structure is OK.
repo3:
Number of directories: 1
Number of txt files: 2
Number of other files: 2
Directory structure is NOT OK.
```

Παραδοχές που μπορείτε να κάνετε:

- Δεν υπάρχουν συγκρούσεις όσον αφορά στα ονόματα των αποθετηρίων git. Κάθε αποθετήριο διαθέτει μοναδική ονομασία.
- Τα αρχεία και κατάλογοι δεν περιέχουν πουθενά στο όνομά τους τον κενό χαρακτήρα.
- Τα αποθετήρια git είναι όλα σε https μορφή.

Ονομάστε το script που θα δημιουργήσετε *script2.sh* και τοποθετήστε το στο αποθετήριο git που δημιουργήσατε για την εργασία σας.

3 Καταμέτρηση λέξεων σε βιβλία

Στην ιστοσελίδα Project Gutenberg (`https://www.gutenberg.org`) υπάρχει τεράστιος όγκος βιβλίων, τα οποία είναι ελεύθερα διαθέσιμα. Όλα τα βιβλία παρέχονται σε πολλές διαφορετικές μορφές, συμπεριλαμβανομένης και της μορφής απλού κειμένου (plain text). Ένα παράδειγμα τέτοιου βιβλίου είναι το «Alice's Adventure in Wonderland» του Lewis Caroll, το οποίο μπορεί να βρεθεί σε μορφή plain text εδώ: `https://www.gutenberg.org/ebooks/19033.txt.utf-8`

Σκοπός της άσκησης είναι η δημιουργία ενός bash script, το οποίο θα δέχεται ως παραμέτρους το όνομα αρχείου (το οποίο θα βρίσκεται τοπικά στο δίσκο σας) και έναν αριθμό n. Το script, θα αναλύει το βιβλίο, και θα παρουσιάζει στο χρήστη τις n λέξεις που εμφανίζονται πιο συχνά στο βιβλίο και θα τις παρουσιάζει σε φθίνουσα σειρά εμφάνισης.

Ένα παράδειγμα εκτέλεσης θα μπορούσε να είναι το παρακάτω (δεν αντιστοιχεί στο παραπάνω βιβλίο):

```
$ ./count_words.sh mybook.txt 4
the 12345
a 10345
to 6231
that 938
```

Αν κοιτάξετε προσεκτικά τα περιεχόμενα του βιβλίου «Alice's Adventure in Wonderland» που αναφέρθηκε προηγουμένως, θα δείτε ότι το πραγματικό κείμενο του βιβλίου ξεκινά μετά από μία γραμμή που αναγράφει:

```
*** START OF THIS PROJECT GUTENBERG EBOOK ALICE IN WONDERLAND ***
```

και ολοκληρώνεται πριν από μία γραμμή που αναγράφει:

```
*** END OF THIS PROJECT GUTENBERG EBOOK ALICE IN WONDERLAND ***
```

Θεωρείστε ότι όλα τα αρχεία τα οποία θα επεξεργάζεται το script μας προέρχονται από το Project Gutenberg και ότι ακολουθούν αντίστοιχη δομή, δηλαδή το πραγματικό κείμενό τους ξεκινά με μία γραμμή που αναγράφει

```
*** START OF THIS PROJECT GUTENBERG EBOOK ...
```

και ολοκληρώνεται πριν από μία γραμμή που αναγράφει

```
*** END OF THIS PROJECT GUTENBERG EBOOK ...
```

Το script που θα δημιουργήσετε θα πρέπει να το λαμβάνει αυτό υπόψιν και να προσμετρά μόνο λέξεις που βρίσκονται μέσα σε αυτά τα όρια.

Επίσης, το script σας θα πρέπει να *μην* διαχωρίζει μεταξύ κεφαλαίων/πεζών (π.χ. η λέξη «The» θα πρέπει να θεωρείται ίδια με τη λέξη «the») και να μη λαμβάνει υπόψιν σημεία στίξης (π.χ. η λέξη «yes!» θα πρέπει να προσμετράται ως απλά «yes») ενώ θα πρέπει να διαχωρίζει και σύνθετες λέξεις που χωρίζονται με αποστρόφους, όπως πχ το «it's» το οποίο θα πρέπει να προσμετράται απλά ως «it» (αγνοώντας το «'s»).

Τέλος, το script σας πρέπει να αποκρίνεται στη χρήση της παραμέτρου `--help` ή `-h` και να εμφανίζει αντίστοιχο μήνυμα βοήθειας.