



ECE 316 - Operating Systems and Networks Laboratory

Assignment 4

General Instructions

For each Assignment, a report (.pdf) and the source-code (e.g., .c, .cpp, .m, .bat etc.) of the solution should be submitted through Microsoft Teams “ECE316 – Operating Systems and Networks Laboratory” no later than the due date of the Assignment. The report should start with a cover page that clearly contains the assignment number and the Team number and the names of the members. In your report, include only the pseudocode, not the actual code, with any comments and description you may want to add, as well as a typical scenario that you used to test your programs. Please note that the report should be as concise as possible. **Caution:** You are not allowed to upload executables (.exe)!

If input test files are given, you are not allowed to make any changes to the provided input files.

Report File Naming Format: “Team#_Assignment#.pdf”

Βοηθητικές Οδηγίες:

- Για την τέταρτη άσκηση θα χρειαστείτε να εγκαταστήσετε στο προσωπικό σας υπολογιστή λειτουργικό σύστημα UNIX ή LINUX. Ένα πιθανό λειτουργικό σύστημα που μπορείτε να εγκαταστήσετε είναι το UBUNTU που μπορείτε να το κατεβάσετε από:
 - ο (Προτεινόμενο)Επιλογή 1: Εγκατάσταση του Ubuntu μέσω WSL2 σε Windows 10+ <https://ubuntu.com/wsl>.
 - ο Επιλογή 2: <https://www.ubuntu.com/desktop/developers> και να το εγκαταστήσετε παράλληλα με το λειτουργικό Windows ακολουθώντας τις εξής εντολές <https://vitux.com/how-to-install-ubuntu-18-04-along-with-windows-10/>.
- Τα scripts που γράφετε πρέπει αναγνωρίζουν τυχόν λανθασμένες εισόδους και να εμφανίζουν τα κατάλληλα μηνύματα λάθους.
- Στα πιο κάτω links βρίσκεται βοηθητικό υλικό για τη διεκπεραίωση της άσκησης:
 - ο https://en.wikibooks.org/wiki/Bash_Shell_Scripting
 - ο http://tldp.org/LDP/Bash-Beginners-Guide/html/sect_02_01.html
 - ο <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO.html>

Description:

1. Άσκηση 1 [25%]

Δημιουργήστε ένα shell script το οποίο θα καλείται κάθε φορά που κάνετε login στο σύστημα (πως θα γίνει αυτό;). Το script θα τυπώνει ένα χαιρετισμό, ανάλογα με την ώρα. Αν είναι πρωί (5πμ.-11:59πμ) θα τυπώνει "Good Morning “username”", αν είναι απόγευμα (12μμ.-5:59μμ) θα τυπώνει "Good Afternoon “username”", αν είναι βράδυ



(5:59μμ-11:59μμ) θα τυπώνει "Good Evening "username"" και αν είναι (12πμ-4:59πμ) θα τυπώνει "It's too late "username"! Go to sleep!".

Η φράση "username" θα πρέπει να τυπώνει αυτόματα το όνομα του χρήστη.

2. Άσκηση 2 [25%]

Δημιουργήστε shell scripts που θα δέχονται παράμετρο έναν αριθμό n και θα υπολογίζουν:

- Το παραγοντικό $n!$
- Αν ο αριθμός n είναι πρώτος (prime) ή όχι.
- Τον μέγιστο κοινό διαιρέτη (ΜΚΔ) του n και θα εκτυπώνεται κατάλληλο μήνυμα αν όντως υπάρχει.
- Το εκθετικό 2^n .
- Τον n -οστό όρο της ακολουθίας Fibonacci:

$$x_n = x_{n-1} + x_{n-2}, \forall n \geq 3, x_1 = 0, x_2 = 1.$$

Για ποιες τιμές του n παρατηρείτε σφάλμα στο αποτέλεσμα στις πιο πάνω περιπτώσεις; Τι συμπέρασμα εξαγάγετε για την αριθμητική του shell;

3. Άσκηση 3 [25%]

Δημιουργήστε ένα shell script με όνομα "jpgDateRename" το οποίο θα μετονομάζει όλα τα αρχεία φωτογραφιών τύπου ".jpg" μόνο, που βρίσκονται στον ίδιο κατάλογο/φάκελο (i.e., current directory), έτσι ώστε το όνομα τους να αρχίζει από την ημερομηνία της ημέρας που εκτελείτε το script σε μορφή DD-MM-YYYY. Για παράδειγμα, αν έχω μια φωτογραφία του σκύλου μου με όνομα "mydog.jpg" στο current directory, και σήμερα η ημερομηνία είναι 27-01-2021 τότε το όνομα της φωτογραφίας πρέπει να αλλάξει σε "27-01-2021-mydog.jpg" (χωρίς να μεταβληθεί η φωτογραφία).

4. Άσκηση 4 [25%]

Δημιουργήστε ένα shell script με όνομα mgrep το οποίο θα πρέπει να δουλεύει όπως η εντολή grep αλλά για πολλαπλά search patterns. Δηλαδή:

mgrep str1 str2 ... strn file

Θα εκτιμηθεί η όσο το δυνατό μεγαλύτερη συμφωνία με την αυθεντική grep. (π.χ. να χειρίζεται επιπλέον παραμέτρους όπως η grep).