



Πανεπιστήμιο Πελοποννήσου
Τμήμα Επιστήμης και Τεχνολογίας Τηλεπικοινωνιών

Λειτουργικά Συστήματα – Προγραμματισμός Συστήματος – 2011

1^η Σειρά Ασκήσεων

Προθεσμία παράδοσης: Δευτέρα 30 Μαΐου.

Τρόπος παράδοσης: αποστολή με e-mail στη διεύθυνση **gelioud@ieee.org**.

Οι ασκήσεις θα παραδοθούν είτε ατομικά είτε σε ομάδες των δύο ατόμων.

Άσκηση 1

Δημιουργήστε bash script, το οποίο θα εκτελεί τις παρακάτω λειτουργίες:

1. Θα δημιουργεί ένα κενό αρχείο, π.χ. myFile.
2. Θα δημιουργεί ένα καινούργιο directory (π.χ. myDir), κάτω από το home directory του χρήστη.
3. Θα μετακινεί το αρχείο myFile στο myDir.
4. Θα γράφει μέσα στο αρχείο μία λίστα με τα περιεχόμενα του home directory του χρήστη. Τα περιεχόμενα θα πρέπει να εμφανίζονται ακολουθώντας "long listing format".

Παραδοτέα:

- Ο κώδικας του προγράμματος (αρχείο *filename.sh*) – πρέπει να περιέχει επεξηγηματικά σχόλια!
- Το αρχείο – έξοδος του προγράμματος.

Άσκηση 2

Δημιουργήστε bash script παρόμοιο με εκείνο της Άσκησης 1, με τις εξής διαφορές:

- Τα ονόματα του αρχείου και του directory δε θα είναι "καρφωμένα" στο script, αλλά θα ζητούνται από το χρήστη κατά την εκτέλεση του script.

- Εκτός από τα περιεχόμενα του home directory του χρήστη, στο αρχείο θα εγγράφονται και τα περιεχόμενα του directory που μόλις έχει δημιουργηθεί. Για να ξεχωρίζουν μεταξύ τους, αφενός θα μεσολαβεί μία κενή γραμμή μεταξύ των δύο λιστών και αφετέρου κάθε λίστα θα ξεκινάει με το path του directory του οποίου τα περιεχόμενα ακολουθούν.

Παραδοτέα:

- Ο κώδικας του προγράμματος (αρχείο *filename.sh*) – πρέπει να περιέχει επεξηγηματικά σχόλια!
- Το αρχείο – έξοδος του προγράμματος.

Άσκηση 3

Δημιουργήστε bash script, το οποίο θα εκτελεί τις παρακάτω λειτουργίες:

1. Θα διαβάζει “γραμμή – γραμμή” τα περιεχόμενα ενός αρχείου “*renameFile*”, του οποίου η κάθε γραμμή θα πρέπει να είναι της μορφής:

```
filename newFileName
```

2. Για κάθε γραμμή του *renameFile* που διαβάζει:
 - a. Εάν στο home directory του χρήστη υπάρχει κάποιο αρχείο με όνομα *filename*, θα το μετονομάζει σε *newFileName* και θα τυπώνει σχετικό μήνυμα στην έξοδο (π.χ., “Μετονόμασα το ... σε ...”).
 - b. Διαφορετικά, θα τυπώνει μήνυμα λάθους, όπως “Δε βρήκα το ...”.

Φροντίστε να πραγματοποιηθεί τουλάχιστον μία μετονομασία.

Επιπλέον, το script θα πρέπει να δημιουργεί δύο αρχεία τα οποία θα περιέχουν τα περιεχόμενα του home directory του χρήστη πριν και μετά την εκτέλεση της σειράς των μετονομασιών, αντίστοιχα, ακολουθώντας “long listing format”. Τα αρχεία θα πρέπει να έχουν όνομα της μορφής: “όνομα χρήστη, ημερομηνία και ώρα”, π.χ.:

```
“gelioud Mon Jun 14 03:17:04 EEST 2010”
```

Για το σκοπό αυτό, θα πρέπει να γίνει χρήση των εντολών *whoami* και *date*. Προκειμένου να αποφύγετε το ενδεχόμενο τα δύο αρχεία να δημιουργηθούν το ίδιο δευτερόλεπτο, χρησιμοποιήστε την εντολή *sleep*.

Παραδοτέα:

- Ο κώδικας του προγράμματος (αρχείο *filename.sh*) – πρέπει να περιέχει επεξηγηματικά σχόλια!
- Την έξοδο του script. (μπορείτε να χρησιμοποιήσετε αρχείο)
- Το αρχείο εισόδου *renameFile*.
- Δύο αρχεία τα οποία θα περιέχουν τα περιεχόμενα του home directory του χρήστη πριν και μετά την εκτέλεση της σειράς των μετονομασιών.

Άσκηση 4

Δημιουργήστε bash script το οποίο, αφού ζητήσει από το χρήστη να εισάγει κάποιον αριθμό, θα υπολογίζει τους αριθμούς Fibonacci που είναι μικρότεροι από αυτόν τον αριθμό και θα τους εγγράφει σε αρχείο. Η μορφή του αρχείου εξόδου πρέπει να έχει ως εξής:

```
Fibonacci 0 = 0
Fibonacci 1 = 1
Fibonacci 2 = 1
Fibonacci 3 = 2
κ.ο.κ.,
```

ενώ το όνομα του αρχείου εξόδου θα πρέπει να είναι της μορφής "FibonacciNum", όπου Num ο αριθμός που δόθηκε ως είσοδος από το χρήστη (π.χ., Fibonacci65).

Σημείωση: Η σύνταξη για την πρόσθεση δύο αριθμών είναι: $a=\$b+\c , όπου a, b και c είναι μεταβλητές.

Παραδοτέα:

- Ο κώδικας του προγράμματος (αρχείο filename.sh) – πρέπει να περιέχει επεξηγηματικά σχόλια!
- Το αρχείο εξόδου, για τις εξής εισόδους: 0, 1, 10, 50, 100.
- Τι παρατηρείτε στην τελευταία εκτέλεση; Πώς το εξηγείτε; Καταγράψτε τα σχόλιά σας, εφόσον υπάρχουν.

Άσκηση 5

Δημιουργήστε bash script το οποίο θα τυπώνει την τρέχουσα ώρα σε διάφορες πόλεις του κόσμου. Η λογική του θα πρέπει να έχει ως εξής:

Θα περιλαμβάνει κάποια συνάρτηση (π.χ., "printTime()") η οποία θα λαμβάνει ως είσοδο το όνομα μίας πόλης καθώς και πληροφορία αναφορικά με τη ζώνη ώρας, υπό μορφή "UTC±X", και θα τυπώνει κάποιο σχετικό μήνυμα (π.χ., "Η ώρα στην Αθήνα είναι 03:07:35").

Για να παίρνει το script τις απαραίτητες πληροφορίες εισόδου, θα πρέπει να δημιουργηθεί αρχείο, η κάθε γραμμή του οποίου θα έχει τη μορφή:

```
City_name UTC±X
```

Το script θα καλεί τη συνάρτηση επαναληπτικά, για όσες γραμμές περιέχει το αρχείο αυτό (τουλάχιστον 5!).

Υπόδειξη:

- Θα κάνετε χρήση της bash εντολής date.
- Στην κλήση της εντολής date, για να δηλώσετε τη ζώνη ώρας θα χρησιμοποιήσετε τη μεταβλητή TZ. Παράδειγμα:

```
TZ="UTC+10" date
```

- Για να μορφοποιήσετε την έξοδο της `date` θα χρησιμοποιήσετε το σύμβολο “+” και έναν ή περισσότερους κωδικούς μορφοποίησης. Για παράδειγμα:

```
date +%k:%M      →    3:07
date +%k--%M     →    3--07
date +%r         →    03:07:25 AM
```

- Πληροφορίες για τις ζώνες ώρας θα βρείτε εδώ: http://en.wikipedia.org/wiki/Time_zone.

Παραδοτέα:

- Ο κώδικας του προγράμματος (αρχείο `filename.sh`) – πρέπει να περιέχει επεξηγηματικά σχόλια!
- Την έξοδο του `script`. (μπορείτε να χρησιμοποιήσετε αρχείο)
- Το αρχείο εισόδου.

Άσκηση 6

Δημιουργήστε `bash script` παρόμοιο με εκείνο της Άσκησης 5, το οποίο όμως θα τυπώνει την ώρα για την πόλη που θα υποδεικνύεται από το χρήστη. Δηλαδή, το `script` θα ρωτάει το χρήστη την πόλη της οποίας την ώρα θα τυπώσει. Οι πληροφορίες για τις ζώνες ώρας θα δίνονται από το ίδιο αρχείο της Άσκησης 5. Εφόσον για την πόλη που δηλώνει ο χρήστης δεν υπάρχουν πληροφορίες στο αρχείο, θα πρέπει να τυπώνεται κάποιο “ευγενικό” μήνυμα λάθους.

Παραδοτέα:

- Ο κώδικας του προγράμματος (αρχείο `filename.sh`) – πρέπει να περιέχει επεξηγηματικά σχόλια!
- Την έξοδο του `script`. (μπορείτε να χρησιμοποιήσετε αρχείο)
- Το αρχείο εισόδου (ίδιο με αυτό της Άσκησης 5).