

Λειτουργικά Συστήματα

Άσκηση 1

Προθεσμία: Κυριακή 29 Νοεμβρίου 2015, 23:59

1 Εισαγωγικά

Το shell (κέλυφος) είναι ένα πρόγραμμα που δέχεται εντολές από το πληκτρολόγιο και δημιουργεί processes (διεργασίες) για να τις εκτελέσει. Όποτε πληκτρολογείτε μια εντολή στο τερματικό ενός συστήματος Unix (π.χ., ls, ps, date, wc, κτλ.), ουσιαστικά το πρόγραμμα που διαβάζει αυτά που γράφετε και εκτελεί τις αντίστοιχες εντολές είναι το shell.

Συνήθως τα shells παρέχουν πολλές επιπλέον λειτουργίες, όπως π.χ., την εκτέλεση scripts (μέσω ενός ενσωματωμένου interpreter), διαχείριση environment variables, και άλλες. Στην παρούσα άσκηση, ο στόχος είναι η υλοποίηση ενός μινιμαλιστικού shell, του οποίου η μόνη λειτουργία θα είναι να διαβάζει εντολές από το πληκτρολόγιο και να τις εκτελεί.

2 Υλοποίηση των shells

Θα πρέπει να υλοποιήσετε τέσσερα shells, που έχουν βαθμιαία μεγαλύτερη πολυπλοκότητα και δυσκολία. Τα shells αυτά θα πρέπει να ονομαστούν `mysh1`, `mysh2`, `mysh3`, και `mysh4`. Τα τέσσερα αυτά shells θα πρέπει να παρέχουν τις εξής λειτουργίες:

1. Το `mysh1` διαβάζει το όνομα ενός προγράμματος από το πληκτρολόγιο, και στη συνέχεια το εκτελεί. Σημειωτέον, το προς εκτέλεση πρόγραμμα μπορεί να βρίσκεται σε οποιοδήποτε directory του `$PATH`.

Για να το εκτελέσει, δημιουργεί μια νέα διεργασία (διεργασία-παιδί), η οποία εκτελεί το εν λόγω πρόγραμμα. Το shell (διεργασία-πατέρα) περιμένει να ολοκληρωθεί η εκτέλεση του προγράμματος πριν δεχτεί την επόμενη εντολή. Για παράδειγμα, αν δώσετε στο prompt του `mysh1` την εντολή `ls`, θα πρέπει να εκτελεστεί το γνωστό `/bin/ls` και να εμφανίσει στην οθόνη (πιο σωστά: στο standard output) τη λίστα αρχείων στο τρέχον directory.

Μια ειδική περίπτωση είναι η εντολή `exit` η οποία δεν αντιστοιχεί σε εκτελέσιμο πρόγραμμα, αλλά απλά τερματίζει το `mysh1`.

Κάθε shell, για να δείξει ότι είναι έτοιμο να δεχθεί την επόμενη εντολή, εμφανίζει στην οθόνη ένα prompt. Στα πλαίσια της άσκησης, το prompt του `mysh1` πρέπει να είναι το `“$”`, δηλαδή ακριβώς το σύμβολο του δολαρίου χωρίς καμία παραπάνω πληροφορία, και χωρίς κενό ούτε πριν ούτε μετά!

Προαιρετικά μπορείτε να προσθέσετε και την εντολή `cd` στο `mysh1` (και στα επόμενα). Αν το κάνετε, θα πρέπει να υποστηρίζει και absolute paths (π.χ., `cd /home/user`) και relative paths (π.χ., `cd ../Pictures`). Για την υλοποίηση του `cd`, θα πρέπει να διαβάσετε τα manual pages των `chdir(2)` και `getcwd(3)`.

2. Το `mysh2` θα αποτελεί μια επέκταση του `mysh1`. Συγκεκριμένα, πέρα από την εκτέλεση προγραμμάτων βάσει μόνο του ονόματος, θα πρέπει να υποστηρίζει και παραμέτρους οι οποίες θα περνούν στο πρόγραμμα που θα εκτελεστεί.

Για παράδειγμα, το νέο σας shell θα πρέπει να υποστηρίζει εντολές όπως η `“ls -l /tmp”`.

3. Το `mysh3` είναι περαιτέρω επέκταση του `mysh2`. Προσθέτει υποστήριξη για pipes, όπως η εντολή `ls /tmp | wc -l`. Μπορείτε να υποθέσετε ότι οι εντολές θα περιέχουν το πολύ ένα pipe, δηλαδή δεν χρειάζεται να υποστηρίξετε εντολές με δύο ή περισσότερα pipes όπως η `sort foo | uniq -c | wc -l`.



Κατά την υλοποίηση του `mysh3`, θα χρειαστείτε να χρησιμοποιήσετε το system call `dup(2)` ή εναλλακτικά το `dup2(2)`. Ανατρέξτε στα manual pages (`man dup`) για την ακριβή λειτουργία που παρέχουν. Για να κάνετε parsing της εντολής που έχει δώσει ο χρήστης, ίσως σας βοηθήσει η χρήση της `strtok(3)`, χωρίς όμως να είναι υποχρεωτικό.

4. Το `mysh4` θα είναι σαν το `mysh3` αλλά θα υποστηρίζει και πολλαπλά pipes, π.χ., εντολές όπως η `cut -f 2 myfile.txt | sort -n | uniq | wc`.



Προφανώς, η χρήση της εντολής `system()` απαγορεύεται σε αυτή την άσκηση. Εξίσου προφανώς απαγορεύεται το invocation ενός άλλου shell (π.χ., `/bin/sh`) που θα κάνει τη δουλειά για λογαριασμό σας ;-).



Σε περίπτωση λαθεμένου input, π.χ., αν η εντολή που πληκτρολογήθηκε δεν αντιστοιχεί σε υπάρχον εκτελέσιμο, εννοείται πως το shell σας δεν θα πρέπει να τερματίζει ή να κολλάει.

Σε τέτοια περίπτωση το shell σας ΔΕΝ ΘΑ ΠΡΕΠΕΙ ΝΑ ΒΓΑΖΕΙ ΚΑΝΕΝΑ OUTPUT στο stdout. Καλύτερα να μην βγάξει κανένα απολύτως μήνυμα, αλλά αν θέλετε να τυπώσετε οπωσδήποτε μήνυμα `Command not found`, αυτό επιτρέπεται μόνο στο stderr!!

3 Κανονισμοί

- Η άσκηση είναι **ατομική**.
- Οι μεταπτυχιακοί φοιτητές δεν υποχρεούνται παράδοσης, καθώς αξιολογούνται μόνο από τη θεωρία. Αν όμως την παραδώσουν, θα προσμετρηθεί και θα βοηθήσει στο να περάσουν το μάθημα.

4 Οδηγίες παράδοσης και αυτόματου feedback

Για την ορθότερη υλοποίηση της άσκησης και τη διόρθωση σφαλμάτων, έχετε στη διάθεσή σας ένα σύστημα αυτόματης διόρθωσης, το οποίο μπορείτε να χρησιμοποιήσετε όσες φορές θέλετε. Με το σύστημα αυτό, μας στέλνετε την υλοποίησή σας, και μέσα σε λίγα λεπτά σας επιστρέφονται συνοπτικά αποτελέσματα του ελέγχου που κάνουμε.

Τα τεστ στα οποία υποβάλλουμε τις υλοποιήσεις σας σε αυτό το στάδιο πιθανόν να είναι λιγότερα (και λίγο απλούστερα) από το σύνολο των τεστ που θα τρέξουμε για την βαθμολόγηση της άσκησης. Δηλαδή, το ότι περνάτε όλα τα τεστ επιτυχώς δεν συνεπάγεται αυτόματα 10 στα 10.

Για να μας στείλετε την άσκηση, ακολουθήστε τα παρακάτω βήματα:

1. Πηγαίνετε (`cd`) στο directory που περιέχει τον κώδικά σας (δηλ. τα `mysh1.c`, `mysh2.c`, ...)
2. Δημιουργήστε ένα αρχείο tar ως εξής:
`tar cvf homework.tar mysh?.c`
3. Στείλτε το αρχείο tar συνημμένο με email στη διεύθυνση **spyros+hw1@ceid.upatras.gr**

- Το subject του email ΠΡΕΠΕΙ να είναι αποκλειστικά και μόνο ο Αριθμός Μητρώου σας (π.χ. 1234), χωρίς έξτρα κενά, χωρίς "AM: 1234", χωρίς "Re:", χωρίς οτιδήποτε άλλο πέρα από τα τέσσερα αριθμητικά ψηφία του AM σας.
 - Το email πρέπει να σταλεί από τον λογαριασμό σας στο CEID.
 - Θα λάβετε άμεσα επιβεβαίωση για το email σας, και σε λίγα λεπτά θα λάβετε και δεύτερο email που θα σας δίνει μια ένδειξη του πόσα τεστ περάσατε. Το περιεχόμενο των τεστ είναι επίτηδες κρυφό, ώστε να διερευνήσετε όλες τις πιθανές αιτίες σφαλμάτων. Μερικά και μόνο μερικά τεστ παρέχουν κάποιο hint για το είδος του λάθους, αν δεν τα περάσατε.
4. Μπορείτε να στείλετε την υλοποίησή σας όσες φορές θέλετε. Η **τελευταία** υλοποίηση που θα λάβουμε θα είναι και αυτή που θα βαθμολογηθεί. Όλες οι προηγούμενες θα αγνοηθούν. Αν συνεχίσετε να στέλνετε υλοποιήσεις μετά το τέλος της προθεσμίας, για κάθε μέρα καθυστέρησης θα χρεώνεστε βαθμούς όπως έχει εξηγηθεί αναλυτικά στο πρώτο μάθημα (και στις αντίστοιχες διαφάνειες).
5. **ΣΗΜΑΝΤΙΚΟ:** Για την παράδοση της τελικής μορφής της άσκησης οφείλετε να συμπεριλάβετε μέσα στο αρχείο tar και ένα σύντομο **documentation 1-2 σελίδων** που να περιγράφει την λογική που ακολουθήσατε και προβλήματα που αντιμετωπίσατε, **υποχρεωτικά σε format PDF**. Γι' αυτό το tar θα πρέπει να γίνει ως εξής:
- ```
tar cvf homework.tar mysh?.c report.pdf
```
- έχοντας ήδηβάλει το report.pdf στο ίδιο directory.

**Καλή επιτυχία!**