

Λειτουργικά Συστήματα

Άσκηση 2

Προθεσμία: Παρασκευή 22 Μαΐου 2026, 23:59

1 Εισαγωγικά

Το shell (ιέλυφος) 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 τρία shells, που έχουν βαθμιαία μεγαλύτερη πολυπλοκότητα και δυσκολία. Τα shells αυτά θα πρέπει να ονομαστούν **mysh1** ως **mysh3**. Τα τρία 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-πατέρας) 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. Στο πλαίσιο της άσκησης, το prompt του **mysh1** πρέπει να είναι το “**\$**”, δηλαδή ακριβώς το σύμβολο του δολαρίου ακολουθούμενο από ένα κενό, χωρίς κανέναν άλλο χαρακτήρα, ούτε πριν ούτε μετά.

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 περνιούνται στο πρόγραμμα που θα εκτελεστεί.

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

Επίσης, το `mysh2` θα πρέπει να υποστηρίζει και την εντολή `cd`, η οποία 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)`.

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 pipes, 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.



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 λαθμεμένου input, π.χ., αν η εντολή που πληκτρολογήθηκε δεν αντιστοιχεί σε υπάρχον εκτελέσιμο, το shell σας δεν θα πρέπει να τερματίζει ή να κολλάει. Σε τέτοια περίπτωση το shell σας δεν θα πρέπει να βγάζει κανένα output στο stdout. Αν θέλετε να τυπώσετε ένα μήνυμα (π.χ., `Command not found` ή κάτι παρόμοιο), αυτό επιτρέπεται μόνο στο stderr.

3 Οδηγίες παράδοσης και αυτόματου feedback

Στείλτε μας συνημμένα στη διεύθυνση `voulgaris+hw2@aueb.gr` τα τρία αρχεία `mysh[123].c` καθώς και το `report.pdf`. Το email πρέπει να σταλεί αποκλειστικά από τον ιδρυματικό σας λογαριασμό (@aueb.gr). Τα αρχεία πρέπει να είναι συνημμένα (attachments) ως απλά αρχεία, δηλαδή ούτε μέσα σε zip, ούτε σε tar, ούτε ως links, ούτε σε οποιαδήποτε άλλη μορφή.

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. Μόνο μερικά τεστ παρέχουν κάποιο hint για το είδος του λάθους, αν δεν τα περάσετε.

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.

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).

Χρειάζονται και τα τρία mysh; NAI! Χρειάζονται και τα τρία! Αν έχετε φτιάξει όλη τη λειτουργικότητα σε ένα και μόνο shell (δηλ., το mysh3), μπορείτε κάλλιστα να το παραδώσετε και ως mysh1, mysh2, αντιγράφοντάς το και συνάπτοντάς το και με τα τρία ονόματα στην τελευταία σας υποβολή. Αν η τελευταία σας υποβολή το περιέχει μόνο ως mysh3, θα πάρετε μόνο τους βαθμούς που αντιστοιχούν σ' αυτό.

ΣΗΜΑΝΤΙΚΟ: Για την παράδοση της τελικής μορφής της άσκησης οφείλετε να συμπεριλάβετε στο email και ένα σύντομο **documentation 1-2 σελίδων** που να περιγράφει την λογική που ακολουθήσατε και προβλήματα που αντιμετωπίσατε, **υποχρεωτικά σε format PDF** με όνομα **report.pdf**.

Καλή επιτυχία!