

Λειτουργικά Συστήματα: Εργασία

2018-2019

Ζητείται να υλοποιήσετε ένα διερμηνέα γραμμής εντολών (command line interpreter) ή κελύφους (shell). Το κέλυφος θα πρέπει να μπορεί να λειτουργήσει με δυο τρόπους, interactive και batch.

Interactive λειτουργία

Στην **interactive** λειτουργία, το κέλυφος σας θα ξεκινά εκτελώντας το όνομα που θα αντιστοιχεί στο εκτελέσιμο αρχείο σας, πχ

```
$ ./myshell
```

Με την εκτέλεση του προγράμματός σας θα εμφανίζεται ένα prompt (μια συμβολοσειρά με το επίθετό σας και το αριθμό μητρώου σας πχ, kalogiannis_3695>) και ο χρήστης του shell θα πληκτρολογεί μια εντολή στη γραμμή, πχ

```
kalogiannis_3695> ls
```

Με την πληκτρολόγηση της εντολής (στο prompt του), το shell θα δημιουργεί μια διαδικασία παιδί (child process) που θα εκτελεί την εντολή που έχετε εισάγει και στη συνέχεια εφόσον ολοκληρωθεί η εκτέλεσή της να επιστρέφει στο prompt. Το shell που θα υλοποιήσετε, θα πρέπει να υποστηρίζει παραμέτρους εντολών πχ

```
kalogiannis_3695> ls -l
```

αλλά και τη δυνατότητα να δίνονται περισσότερες από μια εντολές διαχωρισμένες με το χαρακτήρα ; ή && , πχ

```
kalogiannis_3695> ls -l ; cat file
```

```
kalogiannis_3695> ls -l && cat file
```

Στη περίπτωση αυτή οι εντολές που διαχωρίζονται με τον χαρακτήρα ; θα πρέπει να εκτελούνται διαδοχικά, ανεξάρτητα από την επιτυχία ή μη ολοκλήρωση της πρώτης εντολής. Όταν διαχωρίζονται με τον χαρακτήρα && θα πρέπει να εκτελούνται οι διαδοχικές εντολές εφόσον η προηγούμενη επιστρέψει επιτυχώς. Το shell σας δε θα πρέπει να τυπώνει το prompt του (πχ kalogiannis_3695>), πριν ολοκληρωθεί η εκτέλεση όλων των διεργασιών (η χρήση των εντολών wait() και/ή waitpid() θα σας φανεί χρήσιμη στο σημείο αυτό).

Για παράδειγμα η εκτέλεση των εντολών

```
kalogiannis_3695> ls -l ; cat file ; grep hello file2
```

θα πρέπει να υλοποιηθεί με τέτοιο τρόπο, ώστε οι απαντήσεις των εντολών να εμφανίζονται στην οθόνη με την ίδια σειρά που δόθηκαν οι εντολές.

Πρέπει να υλοποιήσετε επίσης την δυνατότητα ανακατεύθυνσης της εισόδου ή της εξόδου μίας εντολής με τα σύμβολα < και >.

Bonus: Υλοποιήστε σωληνώσεις (pipes) μεταξύ προγραμμάτων με χρήση του συμβόλου |.

Για τη έξοδο από το shell θα πρέπει να πληκτρολογήσετε την λέξη quit (η χρήση της κλήσης συστήματος exit()) θα σας φανεί χρήσιμη στο σημείο αυτό).

Batch λειτουργία

Σε **batch** λειτουργία, το κέλυφος σας ξεκινά καθορίζοντας ένα αρχείο δέσμης (batchfile) στη γραμμή εντολών του, πχ

```
$ ./myshell batchfile
```

Το **batchfile** είναι ένα αρχείο κειμένου που θα περιλαμβάνει το σύνολο των εντολών που θα πρέπει να εκτελεστούν από το shell σας, πχ

```
ls -l
pwd
ps
touch hello
ls -l ; cat file ; grep foo file2
ls -l && cat file
quit
```

Το **batchfile** θα μπορεί να περιλαμβάνει είτε μια εντολή ανά γραμμή, είτε περισσότερες εντολές ανά γραμμή (με διαχωριστικό χαρακτήρα είτε το ; είτε το &&). Σε κάθε περίπτωση η λογική της εκτέλεσης των εντολών (mother-child) θα είναι ίδια με αυτή της περίπτωσης του Interactive mode.

Προδιαγραφές προγράμματος

Το πρόγραμμά σας θα πρέπει να υποστηρίζει και τις δυο μορφές λειτουργίας του, batch και interactive. Σε περίπτωση λαθεμένης εισαγωγής εντολών ή λάθους μορφής batchfile θα πρέπει να εμφανίζεται μήνυμα λάθους στον χρήστη διακόπτοντας τη εκτέλεση του προγράμματος σας (για παράδειγμα αν η εντολή που ζητάτε προς εκτέλεση δεν υπάρχει, έχει λάθος σύνταξη ή το batchfile είναι άδειο). Αναφορικά με το πλήθος των εντολών που μπορούν να δοθούν σε interactive mode ανά γραμμή θα πρέπει να φροντίσετε το μέγεθος να μην υπερβαίνει τους 512 χαρακτήρες συμπεριλαμβανόμενων και των κενών χαρακτήρων καθώς και των διαχωριστικών χαρακτήρων (;). Σε batchfile mode κάθε γραμμή θα πρέπει να έχει το μέγιστο 512 χαρακτήρες συμπεριλαμβανόμενων και των κενών χαρακτήρων καθώς και των διαχωριστικών χαρακτήρων (;).

Επιπρόσθετα θα πρέπει να φροντίσετε να διαχειριστείτε τυχαίες καταστάσεις που δεν είναι λανθασμένες όπως:

- Μια άδεια γραμμή σε batchfile ή άδεια εντολή σε interactive mode,
- Επιπλέον κενοί χαρακτήρες μέσα στο batchfile ή στο interactive mode.
- Batchfile που δεν τελειώνει με την εντολή quit() ή δεν την περιλαμβάνει.

Χρήσιμες πληροφορίες

Το κέλυφός σας στην ουσία θα είναι μία αέναη αναδρομή η οποία θα τυπώνει επαναληπτικά το prompt σας (στην περίπτωση του Interactive mode), θα κάνει «parsing» την είσοδο σας, θα χωρίζει το Input σας σε τμήματα (πλήθος εντολών), θα κάνει τους αναγκαίους έλεγχους περί ορθότητας των εντολών σας και θα τις εκτελεί με τη προαναφερθείσα λογική. Parsing Για εισαγωγή-ανάγνωση των γραμμών μπορείτε να χρησιμοποιήσετε την μέθοδο fgets(). Για να ανοίξετε ένα αρχείο δείτε την μέθοδο fopen(). Η μέθοδος strtok() θα σας φανεί ιδιαίτερα χρήσιμη καθώς διαβάζει μια ολόκληρη γραμμή και «τεμαχίζει» τη γραμμή σε ανάλογα τμήματα. Εκτέλεση εντολών Στα πλαίσια της εργασίας κάντε χρήση μόνο της εντολής execvp(). Θυμηθείτε ότι επιτυχής εκτέλεση

της εντολής δεν επιστρέφει κάτι, διαφορετικά (σε περίπτωση λάθους, πχ η εντολή που ζητήθηκε να εκτελεσθεί δεν υπάρχει) επιστρέφει -1. Το πιο ενδιαφέρον στο σημείο αυτό είναι η τοποθέτηση των ορισμάτων με τον σωστό τρόπο:

```
int execvp(const char *file, char *const argv[]);
```

Το πρώτο όρισμα άφορα το πρόγραμμα-εντολή που πρόκειται να εκτελεσθεί μαζί με την πλήρη διαδρομή του(path). Το δεύτερο όρισμα, μπορεί να προέρχεται από την ίδια τη συνάρτηση «main» του προγράμματος σας :

```
int main(int argc, char *argv[]);
```

Παραδοτέο υλικό

Για την εργασία αυτή θα πρέπει να παραδώσετε:

- Τον πηγαίο κωδικά σας (όχι object files ή executable)
- Ένα Makefile για τη μεταγλώττιση του κωδικά σας
- Ένα αρχείο README με ορισμένες βασικές πληροφορίες τεκμηρίωσης του κωδικά σας
- Αναφορά 3 σελίδων όπου θα δείχνετε πώς λειτουργεί το prompt που δημιουργήσατε