

Λειτουργικά Συστήματα (HY-345)

Εαρινό Εξάμηνο 2017-2018

Άσκηση 1

Φροντιστήριο: 14/02/2018

Παράδοση: 02/03/2018

C Shell Implementation

Σε αυτή την άσκηση θα χρειαστεί να φτιάξετε ένα απλό πρόγραμμα που θα υλοποιεί ένα linux C shell ή αλλιώς command interpreter με όνομα "cs345sh" στο λειτουργικό σύστημα Linux. Σε αυτήν την άσκηση θα εξοικειωθείτε με τη δημιουργία, τον έλεγχο, την εκτέλεση διεργασιών, τις κλήσεις συστήματος (system calls) fork(), wait() και exec(), την επικοινωνία διεργασιών μέσω pipes, τη χρήση εταβλητών συστήματος και την ανακατεύθυνση dup2().

Το prompt του cs345sh shell θα είναι: <user>@cs345sh/<dir>/\$
όπου αντί για <user> θα τυπώνει το όνομα του χρήστη που είναι logged-in στο σύστημα, (δείτε την συνάρτηση βιβλιοθήκης getlogin()) και αντί για <dir> θα τυπώνει το τρέχον directory (δείτε την συνάρτηση βιβλιοθήκης getcwd()).

Με το cs345sh θα μπορεί ένας χρήστης:
να δίνει απλές εντολές και αυτές να εκτελούνται

Για να γίνει αυτό:

1. Αφού διαβάσει μια γραμμή εντολών, το cs345sh θα χρησιμοποιεί το system call fork(), για να δημιουργήσει μια καινούρια διεργασία-παιδί (child process) η οποία θα εκτελεί την εντολή του χρήστη. Η εκτέλεση της εντολής από τη διεργασία-παιδί θα γίνεται με τη χρήση ενός από τα system calls της οικογένειας exec().
2. Η πατρική διεργασία θα περιμένει τη διεργασία-παιδί να τελειώσει την εκτέλεσή της (χρησιμοποιώντας το system call wait()). Ύστερα, θα επιστρέφει στο prompt περιμένοντας τη νέα εντολή του χρήστη, εκτός αν ο χρήστης έχει πληκτρολογήσει το "&" στο τέλος της εντολής όπου αυτό σημαίνει ότι η διεργασία θα τρέχει ως daemon(background process), οπότε η πατρική διεργασία δεν θα περιμένει αλλά θα επιστρέφει αμέσως στο prompt για νέα εντολή. Σε αυτήν την περίπτωση πριν πιστρέψει στο prompt για να περιμένει νέα εντολή θα πρέπει πρώτα να τυπώνει το όνομα και το pid της διεργασίας που τρέχει στο παρασκήνιο.

Απλές εντολές:

- **cd:** Η εντολή αυτή θα δέχεται ένα όρισμα που δηλώνει τον κατάλογο (directory) ο οποίος θα πρέπει να γίνει το νέο τρέχον directory (δείτε την κλήση συστήματος chdir()).
- **Background process:**
 - Ο χρήστης μπορεί πατώντας το συνδυασμό Ctrl-z να κάνει μια διεργασία που τρέχει στο προσκήνιο (foreground process) να τρέχει στο παρασκήνιο (background process), οπότε η πατρική διεργασία δεν θα περιμένει αλλά θα επιτρέπει αμέσως στο prompt για νέα εντολή. Πριν επιστρέψει στο prompt θα πρέπει να τυπώνει το όνομα της διεργασίας που τρέχει στο παρασκήνιο.
 - Επιπλέον ο χρήστης πληκτρολογώντας 'fg' πρέπει να μπορεί να επαναφέρει τη διεργασία που έβαλε στο παρασκήνιο (background) στο προσκήνιο (foreground).
- **Terminal character control flow:**

- Ο χρήστης πατώντας το συνδυασμό Ctrl-s θα πρέπει να σταματάει τη ροή χαρακτήρων στην οθόνη (freeze screen).
- Για να επαναφέρει τη ροή χαρακτήρων ξανά, ο χρήστης θα πρέπει να πατήσει το συνδυασμό πληκτρών Ctrl-q (unfreeze screen).
- **exit**: Με την εντολή exit θα τερματίζεται η λειτουργία του cs345sh.

Προσοχή, το shell σας θα πρέπει να υλοποιεί όλες τις απλές εντολές όχι μόνο αυτές που αναφέρονται παραπάνω και να υποστηρίζονται και τα αντίστοιχα ορίσματα (arguments).
Π.χ. user@cs345sh/dir/\$ ls -a -l

Επιπλέον, θα πρέπει να υλοποιήσετε:

- **pipe**: Το pipe() δίνει την έξοδο μιας εντολής σαν είσοδο σε μια άλλη εντολή που υπάρχει στην ίδια γραμμή εντολών και διαχωρίζονται με το σύμβολο "|" μεταξύ τους.
Π.χ.: user@cs345sh/dir/\$ ps axl | grep zombie
που βρίσκει όλες τις zombie διεργασίες.
Για την υλοποίηση αυτής της λειτουργίας θα σας φανεί χρήσιμο το system call pipe().
- **redirection (ανακατεύθυνση)**: εισόδου "<" και εξόδου ">" και ">>".
 - user@cs345sh/dir/\$ command > output
η οποία "**γράφει**" την έξοδο της εντολής στο αρχείο output
Π.χ.: user@cs345sh/dir/\$ls -l my_files/ > output_file
 - user@cs345sh/dir/\$ command < input
η οποία "**διαβάζει**" την είσοδο για την εντολή από το αρχείο input
Π.χ.: user@cs345sh/dir/\$ cat < input_file

Για την υλοποίηση αυτής της λειτουργίας θα σας φανεί χρήσιμο το system call dup2().

Reference – man pages

Ένα man page περιγράφει τον τρόπο λειτουργίας ενός προγράμματος, ενός system call ή μιας library function. Η εμφάνιση ενός man page γίνεται με τη χρήση της εντολής: **man**

Για να δείτε το man page (σε Linux) που αναφέρεται στη foo(N), κάνετε:

```
$man -S N foo
```

Ο συμβολισμός foo(N) αναφέρεται στο man page που περιγράφει τη foo στην κατηγορία(section) 'N'.

Π.χ.: \$ man -S 2 fork → επιστρέφει το man page του system call fork(2)

Λίστα με χρήσιμα man pages για την Άσκηση 1:

Σας παραθέτουμε man pages με system calls που μπορεί να χρειαστείτε για την υλοποίηση της άσκησης. Η παρακάτω λίστα δεν είναι δεσμευτική. Μπορείτε να χρησιμοποιήσετε και εναλλακτικούς τρόπους:
fork(2), exec(3), execv(3), wait(2), waitpid(2), signal(2), kill(1), pipe(2), sh(1), bash(1), chdir(2), getcwd(2), getlogin(2), dup2(2), termios(3)

Παρατηρήσεις

1. Η άσκηση είναι **ατομική**. Τυχόν αντιγραφές μπορούν να ανιχνευθούν εύκολα από κατάλληλο πρόγραμμα και θα μηδενιστούν. Συμπεριλάβετε το όνομα σας και το λογαριασμό σας (account) σε όλα τα αρχεία.
2. Κατασκευάστε ένα αρχείο **Makefile**, έτσι ώστε πληκτρολογώντας `make all` να γίνεται η μεταγλώττιση (compilation) του προγράμματος και να παράγεται το εκτελέσιμο αρχείο. Επίσης πληκτρολογώντας `make clean` να καθαρίζονται όλα τα περιττά αρχεία, και να μένουν μόνο τα αρχεία που χρειάζονται για τη μεταγλώττιση.
3. Επιπλέον, γράψτε και ένα αρχείο **readme.txt** το πολύ 30 γραμμών που να περιέχει επεξηγήσεις για τον τρόπο υλοποίησης και παραδείγματα για τη χρήση του `cs345sh` που υλοποιήσατε.
4. Τοποθετήστε σε ένα κατάλογο όλα τα αρχεία που χρειάζονται για την άσκηση 1. Παραδώστε τα παραπάνω αρχεία χρησιμοποιώντας το πρόγραμμα `turnin` (πληκτρολογήστε “`turnin assignment_1@hy345 directory_name`” από τον κατάλογο που περιέχει τον κατάλογο `directory_name` με τα αρχεία της άσκησης).
5. Σε πολλές περιπτώσεις τα ονόματα των συναρτήσεων βιβλιοθήκης είναι ενδεικτικά. Μπορείτε να χρησιμοποιήσετε όποια σας βολεύουν.