

Λειτουργικά Συστήματα (HY-345)

Χειμερινό Εξάμηνο 2018-2019

Άσκηση 1

Φροντιστήριο: 05/10/2018

Παράδοση: 18/10/2018

Υλοποίηση Linux C Shell

TA : *shevtsov*(shevtsov@csd.uoc.gr)

Ώρες Γραφείου : Δευτέρα 14:00-16:00

Παρασκευή 14:00-16:00

Στις αίθουσες B208,B210,B212

Σε αυτή την άσκηση θα χρειαστεί να φτιάξετε ένα απλό πρόγραμμα που θα υλοποιεί ένα Linux C shell(command interpreter) με όνομα **'cs345sh'** για το λειτουργικό σύστημα Linux.

Μέσω της άσκησης θα εξοικειωθείτε με τη δημιουργία, τον έλεγχο, την εκτέλεση **διεργασιών**· τις κλήσεις συστήματος¹ **fork()**, **wait()** και **exec()**· την επικοινωνία διεργασιών μέσω **pipes**, τη χρήση **μεταβλητών συστήματος** και την ανακατεύθυνση **dup2()**.

Το prompt του **'cs345sh'** Shell θα είναι:

<user>@cs345sh/<dir>/\$

Σημειώστε πως αντί για **<user>**, το Shell θα εκτυπώνει το **όνομα** του χρήστη που είναι logged-in στο σύστημα² και αντί για **<dir>** θα τυπώνει το **τρέχον directory**³.

1. Ή αλλιώς System Calls
2. Δείτε την συνάρτηση βιβλιοθήκης **getlogin()**
3. Δείτε την συνάρτηση βιβλιοθήκης **getcwd()**

Με το **'cs345sh'**, ένας χρήστης θα έχει τη δυνατότητα να δίνει απλές εντολές και αυτές να εκτελούνται.

Προκειμένου να γίνει αυτό:

1. Αφότου διαβάσει μια γραμμή εντολών, το **'cs345sh'** θα χρησιμοποιεί το system call **fork()**, για να δημιουργήσει μια καινούρια **διεργασία-παιδί**⁴ 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 της⁵. Έπειτα 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 daemon⁶ · οπότε και η πατρική διεργασία δεν θα περιμένει, αλλά θα επιστρέφει αμέσως στο prompt για νέα εντολή. Σε αυτήν την περίπτωση, πριν επιστρέψει 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:

1. **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()**).
2. **Background Process**
 - Επιπλέον ο χρήστης πληκτρολογώντας **'fg'** πρέπει να μπορεί να επαναφέρει τη διεργασία που έβαλε στο **παρασκήνιο** (background) στο **προσκήνιο** (foreground).
 - Ο χρήστης μπορεί χρησιμοποιώντας το συνδυασμό **Ctrl+Z** να κάνει μια διεργασία που τρέχει στο **προσκήνιο** (foreground) να μεταφερθεί στο **παρασκήνιο** (background).

Έτσι η πατρική διεργασία δεν θα περιμένει, αλλά θα επιστρέφει αμέσως στο prompt για νέα εντολή. Πριν επιστρέψει στο prompt, θα πρέπει να **τυπώνει** το όνομα της διεργασίας που τρέχει στο **παρασκήνιο**.

4. Έλλιώς Child Process

5. Δείτε το System Call wait()

6. Background Process

3. Tab filename expansion

- Με την χρήση του **TAB**, στην περίπτωση που υπάρχουν αρχεία των οποίων το όνομα ξεκινά με το ίδιο **πρόθεμα** (prefix), το πρόγραμμα θα πρέπει να τυπώνει **όλα** τα αρχεία με αυτό το prefix -εάν είναι **πάνω από ένα**. Στην περίπτωση που υπάρχει μόνο ένα τέτοιο αρχείο, το πρόγραμμά σας πρέπει να συμπληρώνει αυτόματα (autocomplete) το όνομα του αρχείου.

4. Wildcard “*”

- Ο χρήστης μπορεί να χρησιμοποιεί ***** για εκτέλεση της εντολής για **πάνω από ένα αρχείο**.
Π.Χ. `rm *.txt`

5. Exit

- Με την εντολή **‘exit’** θα τερματίζεται η λειτουργία του **‘cs345sh’**.

Προσοχή, το Shell σας θα πρέπει να υλοποιεί όλες τις απλές εντολές ουχί μονό αυτές που αναφέρονται παραπάνω και να υποστηρίζονται και τα αντίστοιχα ορίσματα.

Π.Χ. `<user>@cs345sh/<dir>/ $ ls -a -l`

Επιπλέον, θα πρέπει να υλοποιήσετε:

6. Pipe

- Το **pipe()** τροφοδοτεί την έξοδο μιας εντολής ως είσοδο σε μια άλλη εντολή που συνυπάρχουν στη γραμμή εντολών και διαχωρίζονται με το σύμβολο **‘|’**.

Π.χ. `<user>@cs345sh/<dir>/ $ ps axl | grep zombie`

Η παραπάνω εντολή εντοπίζει όλες τις **zombie** διεργασίες. Για την υλοποίηση της συγκεκριμένης λειτουργίας μπορείτε να αξιοποιήσετε το **system call** **pipe()**.

7. Redirection

- Ανακατεύθυνση **εισόδου** '<' και **εξόδου** ('>', '>>')

Η εντολή:

```
<user>@cs345sh/<dir>/<$command> > output
```

“Γράφει” την έξοδο της εντολής αριστερά του '<' στο αρχείο output.

Π.Χ. user@cs345sh/dir/\$ls -l mydir > out_file

Από την άλλη, η εντολή :

```
user@cs345sh/dir/$ command < input
```

“Διαβάζει” την είσοδο για την εντολή αριστερά του '<' από το αρχείο input.

Π.Χ. user@cs345sh/dir/\$ cat < input_file

Για την υλοποίηση της συγκεκριμένης λειτουργίας μπορείτε να αξιοποιήσετε 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. Μπορείτε να εξερευνήσετε και εναλλακτικές επιλογές. Η παρακάτω λίστα δεν είναι δεσμευτική:

- 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 ευκολά 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 μονό 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 οποία σας βολεύουν.