

# Λειτουργικά Συστήματα (HY-345)

## Χειμερινό Εξάμηνο 2020

### Άσκηση 1

## C shell Implementation

Φροντιστήριο: 9/10/2020

Παράδοση: 23/10/2020

Σκοπός της άσκησης είναι να εξοικειωθείτε με τον τρόπο λειτουργίας και υλοποίησης ενός απλού shell καθώς και με την ανάπτυξη εφαρμογών στο Linux με την χρήση ενός virtual machine. Επίσης, στόχος είναι να κατανοήσετε τον τρόπο δημιουργίας, ελέγχου και εκτέλεσης διεργασιών χρησιμοποιώντας κλήσεις συστήματος (system calls) όπως τα `fork()`, `wait()` και `exec()` καθώς και την επικοινωνία διεργασιών με την χρήση pipes, την ανακατεύθυνση και την χρήση μεταβλητών περιβάλλοντος.

Σε αυτή την άσκηση θα χρειαστεί να κατασκευάσετε έναν απλό C shell (command interpreter) για το Linux. Το C shell που πρέπει να υλοποιήσετε θα ονομάζεται **cs345sh** και το prompt του θα είναι **[cs345sh][<user>][<dir>]~\$**. Στην θέση του **<user>** το shell θα τυπώνει το όνομα του χρήστη που είναι συνδεδεμένος στο σύστημα και στην θέση του **<dir>** θα τυπώνει το τρέχον directory. Για την υλοποίηση αυτών των λειτουργιών μπορείτε να ανατρέξετε στα man pages των συναρτήσεων `getlogin()` και `getcwd()`. Το shell θα πρέπει να υποστηρίζει όλες τις παρακάτω λειτουργίες:

### Εκτέλεση απλών εντολών

Το shell θα μπορεί να διαβάζει εντολές από τον χρήστη και να τις εκτελεί. Κάθε φορά που το shell θα διαβάζει μια γραμμή εντολών θα χρησιμοποιεί το system call `fork()`, για να δημιουργήσει μια καινούρια διεργασία-παιδί 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()`. 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. Για την υλοποίηση αυτών των λειτουργιών μπορείτε να ανατρέξετε στα man pages των συναρτήσεων `fork()`, `wait()` και `exec()`. Εκτός από τις εντολές του χρήστη, το shell θα πρέπει να υποστηρίζει και τις παρακάτω εντολές:

1. **cd**: Η εντολή αυτή θα δέχεται ένα όρισμα που δηλώνει το directory το οποίο θα πρέπει να γίνει το νέο τρέχον directory. Για την υλοποίηση αυτής της εντολής μπορείτε να ανατρέξετε στο man page της συνάρτησης chdir().
2. **exit**: Με την εντολή exit θα τερματίζεται η λειτουργία του shell.

## Pipes

Τα pipes επιτρέπουν μια εντολή να χρησιμοποιήσει την έξοδο μιας άλλης εντολής, που βρίσκεται στην ίδια γραμμή εντολών, ως είσοδο. Στο shell το οποίο θα υλοποιήσετε, τα pipes διαχωρίζουν τις εντολές με το σύμβολο ">". Για την υλοποίηση των pipes θα χρειαστείτε το system call pipe().

Παραδείγματα:

1. [cs345sh][<user>][<dir>]~\$ ls -al > wc -l  
Τυπώνει τον αριθμό των περιεχομένων του τρέχοντος directory. Πρώτα εκτελείται η εντολή "ls -al" και έπειτα η έξοδος της (τα περιεχόμενα του directory) δίνεται ως είσοδος στην εντολή "wc -l" η οποία τα απαριθμεί.
2. [cs345sh][<user>][<dir>]~\$ ls -al > sort -r -k 6 > head 5  
Τυπώνει τα 5 μεγαλύτερα αρχεία στο τρέχον directory ταξινομημένα κατά μέγεθος. Οι εντολές εκτελούνται από αριστερά προς δεξιά με την επόμενη εντολή να παίρνει σαν είσοδο την έξοδο της προηγούμενης. Το shell δεν θα πρέπει να έχει περιορισμό στα πόσα pipes θα μπορεί να χρησιμοποιήσει ο χρήστης.

## Redirection

Το shell θα πρέπει να υποστηρίζει ανακατεύθυνση εισόδου και εξόδου. Μέσω της ανακατεύθυνσης, μια εντολή μπορεί να πάρει είσοδο από αρχείο ή να γράψει την έξοδό της σε αρχείο. Για την υλοποίηση του redirection μπορείτε να ανατρέξετε στο man page της συνάρτησης dup2(). Στο shell το οποίο θα υλοποιήσετε, η ανακατεύθυνση εισόδου συμβολίζεται με τον χαρακτήρα "|". Η μονή ανακατεύθυνση εξόδου συμβολίζεται με "||" (δύο χαρακτήρες "|") και η διπλή ανακατεύθυνση εξόδου συμβολίζεται με "|||" (τρεις χαρακτήρες "|").

Παραδείγματα:

1. [cs345sh][<user>][<dir>]~\$ cat | data.txt  
Η εντολή "cat" "διαβάζει" την είσοδό της από το αρχείο "data.txt"
2. [cs345sh][<user>][<dir>]~\$ ls -al || log.txt  
Η έξοδος της εντολής "ls -al" γράφεται στο αρχείο "log.txt". Το αρχείο περιέχει πλέον μόνο την έξοδο της εντολής και ό,τι δεδομένα πιθανώς να υπάρχουν στο αρχείο "log.txt"

αντικαθίστανται.

3. [cs345sh][<user>][<dir>]~\$ ls -al |> log.txt

Η έξοδος της εντολής "ls -al" γράφεται στο τέλος του αρχείου "log.txt". Τα δεδομένα που πιθανώς να υπάρχουν στο αρχείο δεν επηρεάζονται καθώς η έξοδος της εντολής γράφεται στο τέλος του αρχείου.

## Shortcuts

Οι συντομεύσεις (shortcuts), είναι συνδυασμοί πλήκτρων οι οποίοι ενεργοποιούν διάφορες λειτουργικότητες ενός shell και συνήθως περιλαμβάνουν τα πλήκτρα Control (CTRL) ή ALT και ένα χαρακτήρα του λατινικού αλφάβητου. Πολλές συντομεύσεις τις συναντάμε συχνά κοινές σε όλα τα γνωστά και διαθέσιμα shells όμως σε κάποιες περιπτώσεις, περίπλοκα shells προσφέρουν επιπλέον συντομεύσεις. Το shell το οποίο θα υλοποιήσετε πρέπει να υποστηρίζει τις παρακάτω συντομεύσεις.

1. **CTRL-C**: Πληκτρολογώντας ταυτόχρονα τους χαρακτήρες CTRL και C, το shell θα πρέπει να τερματίζει (kill) άμεσα την εντολή που εκτελείται εκείνη την στιγμή.
2. **CTRL-H**: Πληκτρολογώντας ταυτόχρονα τους χαρακτήρες CTRL και H, το shell θα διαγράψει ένα χαρακτήρα αριστερά του κέρσορα. Η λειτουργία αυτή έχει το ίδιο αποτέλεσμα με το backspace.
3. **CTRL-S**: Πληκτρολογώντας ταυτόχρονα τους χαρακτήρες CTRL και S, το shell θα παγώνει την εμφάνιση του output της εντολής που εκτελείται εκείνη την στιγμή, χωρίς να τερματίζει την εντολή. [https://en.wikipedia.org/wiki/Software\\_flow\\_control](https://en.wikipedia.org/wiki/Software_flow_control)
4. **CTRL-Q**: Πληκτρολογώντας ταυτόχρονα τους χαρακτήρες CTRL και Q, το shell θα πρέπει να συνεχίζει την εμφάνιση του output της εντολής που εκτελείται εκείνη την στιγμή και της οποίας το output είχε προηγουμένως σταματήσει με την συντόμευση CTRL-S.

## Παράδειγμα

Μπορείτε να ελέγξετε τις παραπάνω συντομεύσεις χρησιμοποιώντας ένα απλό script στο default shell του συστήματός σας όπως το: `a=1; while [ 1 ]; do echo $a; ((a=a+1)); done`

Το παραπάνω script τυπώνει ένα μετρητή ο οποίος σε κάθε βήμα αυξάνεται κατά ένα και έπειτα τυπώνεται στην οθόνη. Η χρήση του **CTRL-S** σταματάει το output. Στην συνέχεια, με την χρήση

του **CTRL-Q** μπορείτε να συνεχίσετε το output του script. Η διακοπή της εκτέλεσής του, καθώς τρέχει σε ένα `while(1) loop`, μπορεί να γίνει με την χρήση της συντόμευσης **CTRL-C**.

## Χρήση των 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 στο Linux το man page που αναφέρεται στη συνάρτηση `foo` εκτελείται: **man foo**

Ο συμβολισμός `foo(N)` αναφέρεται στο man page που περιγράφει τη `foo` στη κατηγορία (section) 'N'. Για να δείτε στο Linux το man page που αναφέρεται στη συνάρτηση `foo` στο section N εκτελείται: **man -S N foo**. Για παράδειγμα μπορείτε να δείτε το man που αναφέρεται στην `open(2)` ως εξής: **man -S 2 open**.

Σας παραθέτουμε man pages με συναρτήσεις 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)`, `pipe(2)`, `dup2(2)`, `sh(1)`, `bash(1)`, `chdir(2)`, `getcwd(2)`, `getlogin(2)`

## Linux on QEMU Emulator

Οι emulators είναι διαδεδομένοι για πολλούς λόγους. Μας επιτρέπουν να εγκαταστήσουμε και να τρέξουμε ένα λειτουργικό σύστημα σαν απλοί χρήστες σε έναν υπολογιστή που έχει κάποιο άλλο λειτουργικό σύστημα, χωρίς να χρειαστεί να αλλάξουμε κάτι σε αυτό. Αυτόν τον υπολογιστή μπορεί να τον χρησιμοποιούν αρκετοί χρήστες και κάθε ένας μπορεί να τρέχει διαφορετικό λειτουργικό σύστημα με έναν emulator χωρίς να επηρεάζονται οι υπόλοιποι χρήστες. Ιδιαίτερα όταν θέλουμε να αναπτύξουμε εφαρμογές οι οποίες λόγω κάποιου προγραμματιστικού λάθους οδηγούν το σύστημα στο να καταρρεύσει (π.χ. kernel panic) μπορούμε εύκολα και γρήγορα να ξεκινήσουμε ξανά το guest λειτουργικό σύστημα με κάποια αλλαγή (debugging) χωρίς να επηρεαστεί το βασικό λειτουργικό σύστημα του υπολογιστή (host operating system).

Σε αυτή την άσκηση, θα χρησιμοποιήσετε τον QEMU για να δημιουργήσετε ένα Virtual Machine όπου θα κατασκευάσετε και θα δοκιμάσετε το shell. Αν θέλετε, μπορείτε να υλοποιήσετε το shell εκτός του Virtual Machine αλλά το compilation και το execution του shell θα πρέπει **πάντα** να γίνεται **εντός** του Virtual Machine.

Ο QEMU υπάρχει ήδη εγκατεστημένος στα μηχανήματα του τμήματος (man qemu για περισσότερες πληροφορίες). Ο QEMU emulator μπορεί να δημιουργήσει και να διαβάσει έναν εικονικό δίσκο (virtual disk image) και σε αυτόν μπορούμε να εγκαταστήσουμε ένα οποιοδήποτε

λειτουργικό σύστημα (π.χ. από ένα εικονικό cd rom). Για τους σκοπούς της άσκησης έχουμε εγκαταστήσει για σας ένα απλό Linux OS (ttylinux distribution) για αρχιτεκτονική 32-bit x86 (i386) σε ένα virtual disk image που θα πρέπει να χρησιμοποιήσετε για την άσκηση σας. Εσείς θα πρέπει αρχικά να αντιγράψετε αυτό το disk image (63 MB) από την περιοχή του μαθήματος (~hy345/qemu-linux/hy345-linux.img) σε έναν κατάλογο στο home folder σας:

```
$ cp ~hy345/qemu-linux/hy345-linux.img ~/<path>
```

Το παραπάνω disk image έχει εγκατεστημένο το Linux OS που θα χρησιμοποιήσετε. Περιέχει το root filesystem (/) στο οποίο υπάρχουν τα βασικά προγράμματα και tools του συστήματος. Περιέχει επίσης τον αρχικό Linux kernel 2.6.38.1 που χρησιμοποιεί για να ξεκινήσει το λειτουργικό σύστημα. Οπότε χρησιμοποιώντας το image αυτό μπορείτε να δοκιμάσετε να ξεκινήσετε αυτό το Linux OS με τον QEMU emulator, απλά με την παρακάτω εντολή:

```
$ qemu-system-i386 -hda hy345-linux.img -curses
```

Η παράμετρος -hda hy345-linux.img κάνει τον QEMU να χρησιμοποιεί το αρχείο hy345-linux.img σαν virtual disk image, το οποίο θα φαίνεται σαν το device /dev/hda στο emulated OS (guest operating system). Τρέχοντας την παραπάνω εντολή θα δείτε να ξεκινάει το Linux OS που προσομοιώνουμε. Όταν σας ζητήσει να κάνετε login χρησιμοποιήστε το account με username "user" και password "csdhy345", όπως θα σας τυπώσει και το σύστημα. Επίσης μπορείτε να κάνετε login και με το account του "root" με password "hy345".

**Προσοχή!** Πάντα να κρατάτε αντίγραφα (backups) της άσκησης σας εκτός του Virtual Machine! Σε περίπτωση που δεν τερματίσετε το Virtual Machine ομαλά (π.χ. Κάνετε αν kill το process του QEMU για οποιοδήποτε λόγο), υπάρχει περίπτωση να καταστρέψετε τον εικονικό του δίσκο με αποτέλεσμα να **μην** μπορείτε να ανακτήσετε τα αρχεία που βρίσκονται σε αυτόν.

Σε περίπτωση που φτάσετε σε αυτό το σημείο μπορείτε να σβήσετε το κατεστραμμένο hy345-linux.img, να πάρετε ένα καινούριο, καθαρό, αντίγραφο του από την περιοχή του μαθήματος και να συνεχίσετε την άσκηση από τα backups που έχετε κρατήσει.

## X11 Forwarding - Για να δουλεύετε remotely με QEMU

### Linux:

Αν δουλεύετε remotely σε κάποιο μηχάνημα του τμήματος, για να ξεκινήσετε το QEMU στο remote μηχάνημα θα πρέπει να συνδεθείτε με X11 forwarding από τον δικό σας υπολογιστή. Αν χρησιμοποιείτε μηχάνημα Linux, όταν κάνετε enable από το gate1/gate2 θα σας επιστρέψει το command που χρειάζεται να εκτελέσετε για να συνδεθείτε στο , π.χ.:

```
$ ssh username@gate1.csd.uoc.gr -p 17724 -Y
```

Δοκιμάστε να τρέξετε xterm, θα πρέπει να σας ανοίξει το xterm. Εφόσον λειτουργεί το xterm, μπορείτε να χρησιμοποιήσετε το QEMU. Εναλλακτικά, μπορείτε απλά να τρέχετε το QEMU χωρίς γραφικό περιβάλλον (πολύ λιγότερο lag) με την βιβλιοθήκη ncurses:

```
$ qemu-system-i386 -hda hy345-linux.img -curses
```

### Windows:

Κατεβάστε το Xming, εγκαταστήστε το και τρέξτε το:

<http://sourceforge.net/projects/xming/files/Xming/6.9.0.31/Xming-6-9-0-31-setup.exe/download>

Στο putty πρέπει να κάνετε τις παρακάτω ρυθμίσεις:

Session -> Hostname: gate1.csd.uoc.gr

Session -> Port: το port που θα σας επιστρέψει το "enable", π.χ. 17724

Connection -> ssh -> X11: tick στο enable X11 forwarding.

Μετά open στο putty για να συνδεθείτε. Αν έχετε firewall πρέπει να κάνετε unblock το Xming. Επίσης αν έχει lag δοκιμάστε να συνδεθείτε με VPN.

## Μεταφορά αρχείων μεταξύ guest OS και host OS

Για να μεταφέρετε αρχεία από το guest OS (που τρέχετε με το QEMU) στο host OS (που κάνετε την βασική σας υλοποίηση) και αντίστροφα, μπορείτε να χρησιμοποιήσετε το πρόγραμμα scp. Μέσα από το guest OS μπορείτε να προσπελάσετε το host OS με την (virtual) IP address 10.0.2.2. Για παράδειγμα, για να μεταφέρετε το αρχείο test1.c από το guest OS στο host OS στην περιοχή σας σε έναν κατάλογο hy345 μπορείτε απλά να κάνετε:

```
$ scp test1.c [username]@10.0.2.2:~/hy345
```

μέσα από το QEMU (guest OS). Το [username] είναι το username που έχετε στα μηχανήματα του τμήματος. Θα χρειαστεί να δώσετε το password που έχετε στα μηχανήματα του τμήματος για να ολοκληρωθεί η αντιγραφή με το scp. Αντίστοιχα, για να αντιγράψετε από το host OS (π.χ. ένα μηχάνημα του τμήματος) το αρχείο test1.c από τον κατάλογο hy345 που είναι στην περιοχή σας στο Linux OS που τρέχει στο QEMU, θα τρέξετε μέσα από το QEMU την εντολή:

```
$ scp [username]@10.0.2.2:~/hy345/test1.c .
```

**! Προσοχή** το «.» μετά το «test.c» δεν είναι τυπογραφικό λάθος !

## 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. Γράψτε ένα αρχείο README, το πολύ 30 γραμμών, με επεξηγήσεις για τον τρόπο υλοποίησης του shell.
3. Κατασκευάστε ένα αρχείο Makefile, έτσι ώστε πληκτρολογώντας make all να γίνεται η μεταγλώττιση (compilation) του shell. Επίσης πληκτρολογώντας make clean να καθαρίζονται όλα τα περιττά αρχεία.
4. Τοποθετήστε σε ένα κατάλογο όλα τα αρχεία προς παράδοση για την άσκηση 1 (Όλα τα απαραίτητα .c και .h αρχεία, το Makefile και το README). 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 αρχείων είναι ενδεικτικά. Μπορείτε να χρησιμοποιήσετε όποια σας βολεύουν.
6. Χρησιμοποιήστε την mailing list του μαθήματος για απορίες. Αποφύγετε τα προσωπικά μηνύματα προς τους βοηθούς.
7. Μην στέλνετε κομμάτια της υλοποίησης σας στην mailing list του μαθήματος.