

HY345 - Linux Shell

Φροντιστήριο : 7/10/2022

Παράδοση : 25/10/2022

Σκοπός της άσκησης είναι η υλοποίηση ενός απλού shell, με την χρήση ενός virtual machine, για την εξοικείωση και κατανόηση της λειτουργικότητας του. Στόχος είναι η κατανόηση της δημιουργίας, ελέγχου και εκτέλεσης διεργασιών με την χρήση κλήσεων συστήματος, όπως τα `fork()`, `wait()` και `exec()`, της κατανοησης του `signal handling` καθώς και των `environment variables`.

Για λεπτομέρειες σχετικά με την λειτουργικότητα των `system calls` που θα χρειαστείτε μπορείτε να ανατρέξετε στα `manual pages` τους. Στην τελευταία ενότητα δίνονται τα `man pages` για τα βασικά `system calls` που θα χρησιμοποιήσετε.

Σε αυτήν την άσκηση την άσκηση ζητείται να υλοποιήσετε ένα απλό Linux shell με την χρήση της γλώσσας C το οποίο θα ονομάζεται **cs345sh** και το prompt του θα είναι `<user>@cs345sh/<dir>$`. Στην θέση του `<user>` το shell θα τυπώνει το όνομα του χρήστη που είναι συνδεδεμένος στο σύστημα και στην θέση του `<dir>` το τρέχον directory. Για την υλοποίηση αυτών των δύο λειτουργιών μπορείτε να ανατρέξετε στα `manual pages` των `getlogin()` και `getcwd()`. Το shell σας θα πρέπει να υποστηρίζει τις εξής λειτουργίες:

1) Εκτέλεση Απλών Εντολών:

Το shell θα πρέπει να δέχεται ως input εντολές από τον χρήστη και να της εκτελεί. Κάθε φορά που το shell διαβάζει μια νέα εντολή, θα πρέπει να δημιουργεί με την χρήση της `fork()` μια διεργασία-παιδί, υπεύθυνη για την εκτέλεση της. Η εκτέλεση των εντολών θα γίνεται με την χρήση ενός από τα `system calls` της οικογένειας `exec()`.

Η πατρική διεργασία θα περιμένει την διεργασία-παιδί να ολοκληρώσει την εκτέλεση της, με την χρήση του `system call wait()`. Έπειτα, θα επιστρέφει στο prompt και θα περιμένει μια νέα εντολή του χρήστη.

Πέραν των απλών εντολών θα πρέπει επίσης να υποστηρίζονται και οι εξής εντολές:

1. **cd:** Η εντολή αυτή παίρνει ένα δεύτερο όρισμα το οποίο δηλώνει το `directory path` στο οποίο πρέπει να μεταφερθεί ο χρήστης από το τρέχον `directory`. Για

την υλοποίηση της εντολής αυτής μπορείτε να χρησιμοποιήσετε την συνάρτηση `chdir()` (μπορείτε να ανατρέξετε στο `man page` της εντολής για περαιτέρω λεπτομέρειες).

2. **exit:** Η εντολή αυτή δεν δέχεται κάποιο όρισμα και απλά τερματίζει την λειτουργία του shell.

2) Global Variables:

Όπως και σε ένα πρόγραμμα έτσι και ένα shell μπορεί να ορίσει τις δικές του μεταβλητές. Μια μεταβλητή ορίζεται ως ένα character string στο οποίο δίνουμε μια τιμή. Για παράδειγμα:

- `<user>@cs345sh/<dir>$ my_var="Hello world!"`

Για να κάνουμε access σε μια μεταβλητή χρησιμοποιούμε το prefix `$` πριν από το όνομα της μεταβλητής. Μπορούμε να τυπώσουμε την τιμή αυτής της μεταβλητής, μέσω της κλήσης της `echo`:

- `<user>@cs345sh/<dir>$ echo $my_var`
- **Output : Hello world!**

Για να θέσετε μια μεταβλητή στο περιβάλλον του shell σας μπορείτε να χρησιμοποιήσετε την `setenv()` και για να διαβάσετε την τιμή της μεταβλητής μπορείτε να χρησιμοποιήσετε την `getenv()`.

Οι local μεταβλητές δεν είναι προσβάσιμες μέσα από subshells. Για παράδειγμα:

- `<user>@cs345sh/<dir>$ my_var="Hello world!"`
- `<user>@cs345sh/<dir>$ echo $my_var`
- **Output : Hello world!**
- `<user>@cs345sh/<dir>$ bash (initialize a subshell)`
- `<user>@cs345sh/<dir>$ echo $my_var`
- **Output :**

Σκοπός αυτής της άσκησης είναι να μπορείτε να δημιουργείτε μεταβλητές οι οποίες θα είναι προσβάσιμες και μέσα από subshells (μπορείτε να κάνετε χρήση της εντολής `export()`)

3) Signal Handling:

Τα signals αποτελούν software interrupts τα οποία στέλνονται από το OS σε μια διεργασία. Τα signals μπορεί να προκύψουν είτε από actions του χρήστη είτε από μια άλλη διεργασία. Ένα παράδειγμα user invoked signal είναι το CTRL-C το οποίο τερματίζει την διεργασία η οποία τρέχει στο shell. Από εσάς ζητείται να υλοποιήσετε την λειτουργικότητα η οποία θα χειρίζεται τα εξής signals:

1. **CTRL-S:** freeze screen
2. **CTRL-Q:** unfreeze screen

3. **CTRL-Z**: Θετει την τρεχουσα διεργασια στο background
4. **fg**: Φερνει την διεργασια απο το background στο foreground

4) Υλοποίηση Pipes:

Τα Pipes επιτρέπουν σε μια εντολή να δέχεται σαν είσοδο το αποτέλεσμα της εκτέλεσης μια άλλης που βρίσκεται στην ίδια γραμμή εντολών. Οι εντολές στα pipes διαχωρίζονται με το σύμβολο “|” και εκτελούνται από τα αριστερά προς τα δεξιά με την κάθε επόμενη εντολή να δέχεται σαν είσοδο την έξοδο της προηγούμενης. Για την υλοποίησή τους θα πρέπει να χρησιμοποιήσετε το system call pipe().

Μερικά παραδείγματα:

1. **Single pipe:** `<user>@cs345sh/<dir>$ echo “Hello world” | wc -w`

Η πρώτη εντολή τυπώνει στο command prompt την πρόταση “Hello World”, το pipe “|” θα μεταφέρει την έξοδο αυτήν από το standard output στο input της επομένης εντολής η οποία μετράει τον αριθμό των λέξεων από το input και τυπώνει το νούμερο. Συνεπώς θα τυπώσει 2 στο command prompt.

2. **Multiple pipes:** `<user>@cs345sh/<dir>$ cat test.txt | sort | head -2`

1. **cat test.txt:** τυπώνει τα περιεχόμενα του φακέλου test.txt.
2. **sort:** κάνει sorting τα περιεχόμενα αλφαβητικά και τυπώνει το αποτέλεσμα.
3. **head -2:** τυπώνει τα πρώτα δύο στοιχεία από το input που θα δεχτεί (εδώ τα ταξινομημένα περιεχόμενα του test.txt).

Συνεπώς θα τυπωθούν τα πρώτα δύο στοιχεία από τα ταξινομημένα περιεχόμενα του αρχείου αυτού.

5) Linux on QEMU Emulator

Ένας emulator μας επιτρέπει να εγκαταστήσουμε ένα λειτουργικό σύστημα σε έναν υπολογιστή που τρέχει ένα άλλο λειτουργικό σύστημα, χωρίς αυτό να επηρεαστεί. Τον ίδιο υπολογιστή μπορούν να χρησιμοποιούν πολλοί χρήστες και ο καθένας μπορεί να χρησιμοποιεί διαφορετικό λειτουργικό σύστημα με έναν emulator χωρίς να

επηρεάζει τους υπόλοιπους χρήστες. Στην περίπτωση που μια εφαρμογή μας σε έναν κοινόχρηστο υπολογιστή οδηγήσει στην κατάρρευση του συστήματος (π.χ. kernel panic caused by a fatal error), εάν η αποτυχία συστήματος αυτή συμβεί στον emulator μπορούμε να ξεκινήσουμε ξανά το guest λειτουργικό του σύστημα (μετά από recovery ή debugging) χωρίς να επηρεαστεί το βασικό λειτουργικό σύστημα του υπολογιστή (host operating system).

Για την υλοποίηση της άσκησης θα χρησιμοποιήσετε τον QEMU, **Virtual Machine** όπου θα κατασκευάσετε το shell. Μπορείτε να κατασκευάσετε το shell εκτός του virtual machine σας αλλά το compilation και execution του shell θα γίνεται **πάντα εντός** του virtual machine.

Ο QEMU βρίσκεται 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 (TinyCore Linux distribution) για αρχιτεκτονική 32-bit x86 (i386) σε ένα virtual disk image που θα πρέπει να χρησιμοποιήσετε για την άσκηση σας.

Εσείς θα πρέπει να αντιγράψετε αυτό το disk image από την περιοχή του μαθηματος (~hy345/qemu-linux/hy345-development.img) σε έναν φακέλο (e.g. ~/Desktop/hy345/) στο home folder σας εκτελώντας την παρακάτω εντολή:

cp ~hy345/qemu-linux/hy345-development.img ~/<your_file_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.

Έχοντας locally αυτό το image μπορείτε να δοκιμάσετε να ξεκινήσετε το Linux OS με τον QEMU εκτελώντας την παρακάτω εντολή:

\$ qemu-system-i386 -hda hy345-development.img -curses

- **-hda hy345-development.img:** Υποδεικνύει στον QEMU να χρησιμοποιεί το αρχείο hy345-development.img σαν virtual disc image, το οποίο θα φανεί ως το device /dev/had στο emulated λειτουργικό σύστημα.
- **-curses:** library το οποίο τρέχει τον QEMU χωρίς γραφικό περιβάλλον (σε περίπτωση που δουλεύετε remotely με τον QEMU για βελτιστοποίηση και για να μην υπάρχει lag χρησιμοποιήστε το flag αυτό).

Εκτελώντας την παραπάνω εντολή θα δείτε να ξεκινάει το emulated Linux OS. Τα credentials για να συνδεθείτε είναι:

1. Guest:

a. **username: "hy345"**

b. password: “hy345”

Ορθη Χρηση του QEMU:

1. Για να βγειτε απο τον QEMU και να κλεισετε το Guest OS χρησιμοποιηστε την εντολη **turnoff**. Η εντολη αυτη εξασφαλιζει οτι το λειτουργικο συστημα θα κλεισει ομαλα. Διαφορετικα, υπαρχει κινδυνος να χασετε τα αρχεια σας η τις αλλαγες που εχετε κανει σε αυτα.
2. Έχει παρατηρηθει οτι κατα την αναπτυξη της ασκησης, οι φοιτητες κανουν σημαντικα λαθη με την δημιουργια threads με αποτελεσμα να κρασαρει το συστημα και να χαθουν αρχεια. Για τον λογο, σας παρεχουμε την εντολη **save** η οποια αποθηκευει τα αρχεια στο image. Καλο ειναι προτου εκτελεσετε την υλοποιηση σας να εκτελεσετε την εντολη αυτη καθως μειωνει τις πιθανοτητες να χασετε τα αρχεια σας σε περιπτωση που κανετε καποιο σημαντικό λάθος.
3. Το image που σας παρεχουμε ειναι υπο αναπτυξη και μπορει να προκυψουν προβληματα. Για τον λογο αυτο καλειστε να παρνετε συχνα αντιγραφα των αρχειων σας στο Host OS χρησιμοποιωντας το SCP. Για περισσοτερες πληροφοριες δειτε το κεφαλαιο 7 της εκφωνησης.

Προσοχή! Πάντα να κρατάτε αντίγραφα της εργασίας σας **εκτός του virtual machine**. Σε περίπτωση που δεν τερματίσει ομαλά (π.χ. κάνετε kill το process του QEMU) υπάρχει περίπτωση να καταστρέψετε τον εικονικό δίσκο χωρίς να υπάρχει δυνατότητα να ανακτήσετε τα έγγραφα σας. Χρησιμοποιήστε τον QEMU για το compilation και execution του shell για να αποφύγετε να επηρεάσετε το κοινόχρηστο μηχάνημα και πιθανά την δουλειά άλλων συναδέλφων σας.

6) 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 το μηχανημα π.χ. **despolo** θα σας επιστρέψει το command που χρειάζεται να εκτελέσετε για να συνδεθείτε στο π.χ.:

```
$ ssh username@despolo.csd.uoc.gr -p 22 -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 (προτεινόμενο):

\$ qemu-system-i386 -hda hy345-development.img -curses

❖ **Windows:**

Κατεβάστε το WinSCP από το παρακάτω link, κάνετε το εγκατάσταση (default installation) και τρέξτε το.

- <https://winscp.net/download/WinSCP-5.19.3-Setup.exe>

Στο WinSCP όταν το εκκινήσετε πρέπει να κάνετε τις παρακάτω ρυθμίσεις:

- Hostname: **your_name@despolo.csd.uoc.gr**
- Port : 22
- User_name : **your_name** (e.g. csd1234)
- Password : your password.

Θέτοντας τις παραπάνω ρυθμίσεις στο WinSCP θα μπορείτε να συνδεθείτε στο μηχάνημα της σχολής που επιθυμείτε και να μεταφέρετε αρχεία (drag and drop) από το μηχάνημα στο οποίο δουλεύετε σε αυτό της σχολής.

7) 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 directory με όνομα 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 test1.c **δεν** είναι τυπογραφικό λάθος, δηλώνει ότι η μεταφορά του test1.c θα γίνει στο παρόν directory (μπορείτε να αντικαταστήσετε με οποιοδήποτε path επιθυμείτε να μεταφέρετε το αρχείο σας)

7) Manual Pages:

Τα manual pages (man pages) περιγράφουν τον τρόπο λειτουργίας ενός system call ή μιας library function. Για την εμφάνιση της man page μιας συνάρτησης στα Linux systems μπορείτε να εκτελέσετε στο terminal την εντολή : **man foo**, όπου foo είναι το όνομα της συνάρτησης την οποία ψάχνετε.

Παραθέτουμε μια λίστα από man pages system calls και συναρτήσεων τα οποία μπορείτε να χρησιμοποιήσετε για την υλοποίηση της άσκησης (η λίστα **δεν** είναι δεσμευτική):

fork(2), exec(3), execv(3), wait(2), waitpid(2), pipe(2), dup2(2),chdir(2)

Επιπλέον μπορείτε να βρείτε και online τα man pages που θα χρειαστείτε.Μερικά ενδεικτικά παραδείγματα είναι:

1. **fork(2)**: <https://man7.org/linux/man-pages/man2/fork.2.html>
2. **exec(3)**: <https://man7.org/linux/man-pages/man3/exec.3.html>
3. **export()**:<https://www.ibm.com/docs/en/aix/7.1?topic=customization-exporting-shell-variables-export-shell-command>
4. **setenv(3)**: <https://man7.org/linux/man-pages/man3/setenv.3.html>
5. **getenv(3)**: <https://man7.org/linux/man-pages/man3/getenv.3.html>
6. **pipe(2)**: <https://man7.org/linux/man-pages/man2/pipe.2.html>
7. **dup2(2)**: <https://man7.org/linux/man-pages/man2/dup.2.html>
8. **termios(3)**:<https://man7.org/linux/man-pages/man3/termios.3.html>

Παρατηρησεις:

1. Η άσκηση αυτή είναι καθαρά ατομική και τυχόν αντιγραφές που θα διαπιστωθούν θα οδηγήσουν σε αυτόματο μηδενισμό.
2. Στα παραδοτέα συμπεριλάβετε το όνομα σας καθώς και το ιδρυματικό σας email (σαν σχόλια στα αρχεία που θα παραδώσετε)
3. Μαζί με τον κώδικα που θα παραδώσετε δημιουργήστε ένα README το οποίο περιγράφει την υλοποίησή σας.
4. Επιπλέον, δημιουργήστε ένα Makefile το οποίο με την χρήση της εντολής **make all** γίνεται το compilation του shell και με την χρήση της εντολής **make clean** καθαρίζονται τα περιττά αρχεία.
5. Τοποθετήστε σε έναν φάκελο όλα τα παραδοτέα (αρχεία .c,.h Makefile, README) και παραδώστε τον φάκελο με το πρόγραμμα turnin (**turnin assignment_1@hy345 directory_name** όπου directory_name είναι το όνομα/παθ του φακέλου που περιέχει τα παραδοτέα αρχεία σας).

6. Χρησιμοποιήστε την mailing list του μαθήματος για τυχόν απορίες (αποφύγετε τα προσωπικά email στους βοηθούς του μαθήματος).
7. Μην στέλνετε κομμάτια της υλοποίησης σας (κώδικα/screenshots) στην mailing list του μαθήματος.