ΜΑΘΑΙΝΩ ΤΟ ΚΕΛΥΦΟΣ (SHELL)

Και τώρα τι γίνεται;

Εγκαταστήσατε το GNU/ Linux και τρέχει μια χαρά. Η γραφική διεπαφή (GUI) δουλεύει σαν ... Ελβετικό ρολόι, αλλά σας κουράζει να αλλάζετε τα θέματα στην επιφάνεια εργασίας. Βλέπετε συνέχεια αυτό το πράγμα που λέγεται "τερματικό" (terminal) και αναρωτιέστε.

Μην ανησυχείτε, θα σας δείξουμε τι ακριβώς να κάνετε.

Γιατί να σας νοιάζει;

Τέλος πάντων, γιατί χρειάζεται να μάθετε τη γραμμή εντολών; Επιτρέψτε μου, λοιπόν, να σας διηγηθώ μια ιστορία. Πρόσφατα, είχαμε ένα πρόβλημα στο χώρο της δουλειάς μου. Είχαμε έναν κοινό σκληρό δίσκο (shared drive) σε έναν από τους διακομιστές αρχείων, ο οποίος συνεχώς γέμιζε. Δεν θα αναφερθώ στο γεγονός ότι το παρωχημένο (legacy) λειτουργικό σύστημα που έτρεχε εκεί, δεν υποστήριζε τη χρήση ξεχωριστών περιοχών για τον κάθε χρήστη (user quotas), διότι αυτή είναι μια άλλη ιστορία. Αλλά ο διακομιστής συνεχώς γέμιζε και εμπόδιζε τον κόσμο να κάνει τη δουλειά του. Ένας από τους μηχανικούς λογισμικού της εταιρείας μας σπατάλησε σχεδόν μια εργάσιμη ημέρα, γράφοντας ένα πρόγραμμα σε C++ για να μπορεί να διερευνήσει τους φακέλλους όλων των χρηστών, και να προσθέσει το χώρο που χρησιμοποιούσαν, βάζοντας τα αποτελέσματα σε έναν κατάλογο. Από τότε που αναγκάσθηκα να χρησιμοποιώ αυτό το παρωχημένο λειτουργικό σύστημα, για όσο χρόνο βρέθηκα στο χώρο της δουλειάς, εγκατέστησα μια έκδοση του κελύφους bash που να δουλεύει σε αυτό. Όταν άκουσα για το πρόβλημα αυτό, συνειδητοποίησα ότι μπορούσα να κάνω όλη τη δουλειά που είχε καταφέρει ο μηχανικός μας, χρησιμοποιώντας μόνον την εξής μία γραμμή εντολής:

```
du -s * | sort -nr > $HOME/space_report.txt
```

Οι Γραφικές Διεπαφές Χρήστη (GUI) βοηθούν σε πολλές δουλειές, αλλά δεν ταιριάζουν για όλες. Εδώ και καιρό, μου φαινόταν ότι οι περισσότεροι υπολογιστές σήμερα δεν χρησιμοποιούν ηλεκτρισμό. Αντιθέτως, μοιάζει σαν να τροφοδοτούνται από την κίνηση του ποντικιού! Υποτίθεται ότι οι υπολογιστές θα μας απελευθέρωναν από τη χειρωνακτική δουλειά. Πόσες φορές, όμως, δεν κάνατε μια δουλειά που ήσασταν σίγουροι ότι ο υπολογιστής θα έπρεπε να είναι σε θέση να την κάνει για εσάς; Στο τέλος, καταντούσε να εκτελείτε αυτή την εργασία βαριεστημένα, με τα κλικ του ποντικιού. Δείχνοντας με το ποντίκι και πατώντας τα πλήκτρα του, και ξαναδείχνοντας και ξαναπατώντας τα και πάλι.

Άκουσα κάποτε ένα συγγραφέα να κάνει την παρατήρηση πως όταν είσαι παιδί, χρησιμοποιείς τον υπολογιστή κοιτώντας τις εικόνες. Όταν μεγαλώνεις, μαθαίνεις γραφή και ανάγνωση. Καλωσήλθατε στη σειρά μαθημάτων 101, για την εξοικείωση και εκμάθηση υπολογιστή. Και τώρα, ας αρχίσουμε τη δουλειά.

Περιεχόμενα

1. Τι είναι το "κέλυφος" (shell);
    1. Τι είναι το xterm, το gnome-terminal, το konsole, κλπ;

    2. Πως ξεκινάμε ένα Τερματικό
    3. Δοκιμή του πληκτρολογίου
    4. Χρήση του ποντικιού
2. Πλοήγηση
    1. Οργάνωση Συστήματος Αρχείων
    2. pwd
    3. cd
3. Ψάχνοντας τριγύρω
    1. ls
    2. less
    3. file
4. Μια καθοδηγούμενη ξενάγηση
    1. /
    2. /boot
    3. /etc
    4. /bin, /usr/bin
    5. /sbin, /usr/sbin
    6. /usr
    7. /usr/local
    8. /var
    9. /lib
    10. /home
    11. /root
    12. /tmp
    13. /dev
    14. /proc
    15. /mnt
5. Διαχείριση Αρχείων
    1. Χαρακτήρες μπαλαντέρ (Wildcards)
    2. cp
    3. mv
    4. rm
    5. mkdir
6. Ανακατεύθυνση Εισόδου/Εξόδου
    1. Κανονική Έξοδος (Output)
    2. Κανονική Είσοδος (Input)
    3. Σωληνώσεις (Pipes)
    4. Φίλτρα
7. Δικαιώματα
    1. Δικαιώματα αρχείων
    2. chmod
    3. Δικαιώματα Καταλόγων
    4. Γίνε Υπερχρήστης (superuser) για λίγο
    5. Αλλαγή δικαιωμάτων ιδιοκτησίας αρχείου
    6. Αλλαγή δικαιωμάτων ιδιοκτησίας ομάδας
8. Έλεγχος Εργασιών (Job Control)
    1. Ένα πρακτικό παράδειγμα
    2. Πως να θέσετε ένα πρόγραμμα στο παρασκήνιο (background)
    3. Εμφάνιση των διεργασιών σας (processes) σε λίστα
    4. Πως να τερματίσετε (να "σκοτώσετε") μια διεργασία
    5. Λίγες παραπάνω λεπτομέρειες για το "σκότωμα"

1

Τι είναι το “κέλυφος” (shell) ;

Με απλά λάγια, το κέλυφος είναι ένα πρόγραμμα που παραλαμβάνει τις δικές σας εντολές από το πληκτρολόγιο και τις παραδίδει στο Λειτουργικό Σύστημα (Λ.Σ.) προς εκτέλεση. Στα παλιότερα χρόνια, αυτή ήταν και η μόνη διαθέσιμη διεπαφή σε έναν υπολογιστή που έτρεχε σε Unix.

Στις μέρες μας, πέρα από τις Διεπαφές Γραμμής Εντολών *(Command line interfaces, CLI), σαν το Κέλυφος,* έχουμε στη διάθεσή μας και τις Γραφικές Διεπαφές Χρήστη (G*raphical User Interfaces, GUI)*.

Στα περισσότερα συστήματα GNU/Linux, υπάρχει ένα πρόγραμμα που λέγεται `bash` (που σημαίνει Bourne Again Shell (“Αναγεννημένο Κέλυφος”), που αποτελεί μια βελτιωμένη έκδοση του αρχικού προγράμματος Bourne shell, του `sh`, που είχε γράψει ο Steve Bourne) και το οποίο λειτουργεί σαν πρόγραμμα Κελύφους. Υπάρχουν διάφορα άλλα πρόσθετα προγράμματα κελύφους που προσφέρονται σε ένα τυπικό σύστημα GNU/Linux. Ανάμεσα σε αυτά, περιλαμβάνονται τα `ksh`, `tcsh` και `zsh`.

Τι είναι το xterm, το gnome-terminal, το konsole, κλπ.;

Αυτά είναι γνωστά και ως "προσομοιωτές τερματικού” (terminal emulators). Πρόκειται για προγράμματα που εμφανίζουν ένα παράθυρο και σας επιτρέπουν να αλληλεπιδράτε με το κέλυφος. Υπάρχουν πολλοί τέτοιοι προσομοιωτές τερματικού που μπορείτε να χρησιμοποιήσετε. Οι περισσότερες διανομές GNU/Linux, προσφέρουν διάφορους από αυτούς, όπως τους: `xterm`, `rxvt`, `konsole`, `kvt`, `gnome-terminal`, `nxterm`, και το `eterm`.

Πως ξεκινάμε ένα τερματικό

Ο διαχειριστής παραθύρων του λειτουργικού σας συστήματος, πιθανώς έχει έναν τρόπο για να ξεκινά τα προγράμματα από ένα μενού. Εξετάστε τον κατάλογο προγραμμάτων, για να δείτε αν περιέχει κάτι που να μοιάζει με πρόγραμμα προσομοίωσης τερματικού. Στο KDE, μέσα στο μενού των Εργαλείων Συστήματος, μπορείτε να βρείτε την Κονσόλα ("konsole"). Στο Gnome, μέσα στο μενού των Βοηθημάτων, μπορείτε να βρείτε το Τερματικό ("gnome-terminal"). Μπορείτε να εκκινήσετε όσα από αυτά θέλετε και να παίξετε μαζί τους. Αν και υπάρχουν πολλοί και διάφοροι από αυτούς τους προσομοιωτές τερματικού, όλοι τους κάνουν το ίδιο πράγμα. Σας προσφέρουν πρόσβαση σε μια συνεδρία κελύφους. Μάλλον θα αναπτύξετε μια προτίμηση για έναν από αυτούς, που θα βασίζεται στα διάφορα κολπάκια και τα χαρακτηριστικά που προσφέρει ο καθένας.

Δοκιμάζοντας το πληκτρολόγιο

Ωραία, ας προσπαθήσουμε να πληκτρολογήσουμε λίγο. Ανοίξτε ένα παράθυρο τερματικού. Θα πρέπει, τώρα, να βλέπετε την προτροπή του κελύφους (shell prompt), που περιέχει το δικό σας όνομα χρήστη και το όνομα του υπολογιστή, ακολουθούμενα από ένα σύμβολο δολλαρίου.

Κάτι σαν και αυτό που ακολουθεί παρακάτω:

```
[me@linuxbox me]$
```

Πολύ ωραία! Τώρα πληκτρολογείστε μερικούς τυχαίους χαρακτήρες χωρίς νόημα (nonsense characters) και πατήστε το πλήκτρο “Enter”.

```
[me@linuxbox me]$ kdkjflajfks
```

Αν όλα πήγαν καλά, θα πρέπει να έχετε λάβει ένα μήνυμα σφάλματος, που θα παραπονείται ότι δεν μπορεί να σας κατανοήσει:

```
[me@linuxbox me]$ kdkjflajfks
bash: kdkjflajfks: command not found
```

Υπέροχα! Τώρα, πατήστε το κουμπί με το βέλος προς τα επάνω (up-arrow). Δείτε πως η προηγούμενη εντολή μας "kdkjflajfks" επιστρέφει. Ναι, έχουμε ένα *ιστορικό εντολών (command history)*. Πατήστε το κουμπί με το βέλος προς τα κάτω (down-arrow) και θα λάβουμε και πάλι την κενή γραμμή (blank line).

Ανακαλέστε την εντολή "kdkjflajfks" χρησιμοποιώντας, αν χρειάζεται, το κουμπί με το βέλος προς τα επάνω. Τώρα, δοκιμάστε το κουμπί με το αριστερό και το δεξί βέλος. Μπορείτε να βάλετε τον κέρσορα του κειμένου οπουδήποτε μέσα στη γραμμή εντολών. Αυτό σας επιτρέπει να διορθώνετε εύκολα τα σφάλματα.

Δεν έχετε συνδεθεί σαν root, σωστά;

Μη χρησιμοποιείτε τον υπολογιστή σαν υπερχρήστης (superuser). Θα πρέπει να γίνετε υπερχρήστης (superuser) μόνον όταν είναι αυστηρώς απαραίτητο. Διαφορετικά, κάθε άλλη στάση, είναι επικίνδυνη, ανόητη, και κακόγουστη. Δημιουργήστε έναν λογαριασμό χρήστη για τον εαυτό σας τώρα!

Χρήση του ποντικιού

Ακόμη και αν το κέλυφος είναι μια εντολή σε γραμμή εντολών, μπορείτε να χρησιμοποιείτε το ποντίκι σας για διάφορα πράγματα. Αυτό, βέβαια, ισχύει εφόσον έχετε ένα ποντίκι με 3-κουμπιά. Και θα πρέπει να έχετε ένα τέτοιο ποντίκι με 3-κουμπιά, αν θέλετε να χρησιμοποιείτε GNU/Linux.

Κατ' αρχήν, μπορείτε να χρησιμοποιείτε το ποντίκι για να μετακινηθείτε (scroll) προς τα εμπρός και προς τα πίσω, μέσα στο κείμενο του παραθύρου του τερματικού. Για λόγους επίδειξης, κρατήστε πατημένο το κουμπί “Enter” μέχρι την μετακίνηση εκτός του παραθύρου. Τώρα, με το ποντίκι σας, μπορείτε να χρησιμοποιήσετε την πλαϊνή μπάρα κύλισης (scroll bar) του παραθύρου του τερματικού, για την μετακίνηση των περιεχομένων του παραθύρου προς τα πάνω και προς τα κάτω. Αν χρησιμοποιείτε το `xterm`, μπορεί να σας φανεί δύσκολο, αφού απαιτείται το μεσαίο κουμπί για τη λειτουργία αυτή. Αν έχετε ένα ποντίκι με 2-κουμπιά, μπορεί να ρυθμισθεί, για να προσομοιώνει ένα ποντίκι με 3-κουμπιά. Αυτό σημαίνει ότι το μεσαίο κουμπί μπορεί να προσομοιωθεί πατώντας, συγχρόνως, τόσο το αριστερό, όσο και το δεξιό κουμπί.

Στη συνέχεια, μπορείτε να αντιγράψετε κείμενο με το ποντίκι. Σύρτε το ποντίκι σας πάνω σε ένα κείμενο (π.χ., "kdkjflajfks", ακριβώς εδώ, στο παράθυρο του περιηγητή), κρατώντας πατημένο το αριστερό κουμπί. Το κείμενο θα πρέπει να φωτισθεί. Τώρα, μετακινείστε τον δείκτη του ποντικιού σας στο παράθυρο του τερματικού και πατήστε το μεσαίο κουμπί του ποντικιού. Το κείμενο που φωτίσατε στο παράθυρο του περιηγητή, θα πρέπει να έχει αντιγραφεί στη γραμμή εντολών. Α! Σας το είπα ότι θα χρειασθείτε ένα ποντίκι με 3-κουμπιά;

Λίγα λόγια για την εστίαση...

Όταν κάνατε την εγκατάσταση του συστήματός σας GNU/Linux και του διαχειριστή παραθύρων του (κατά πάσα πιθανότητα του Gnome ή του KDE), είχε ρυθμισθεί να συμπεριφέρεται κατά κάποιο τρόπο σαν το γνωστό και παρωχημένο (legacy) λειτουργικό σύστημα.

Συγκεκριμμένα, η *τακτική εστίασης (focus policy)* μάλλον ρυθμίσθηκε στο “κάνε κλικ για να γίνει εστίαση” (click to focus). Αυτό σημαίνει ότι για να μπορεί ένα παράθυρο να αποκτήσει εστίαση (δηλ. για να γίνει ενεργό), θα πρέπει να κάνετε κλικ μέσα στο παράθυρο. Αυτό είναι αντίθετο με την παραδοσιακή συμπεριφορά του X windows. Αν υιοθετήσετε τη συμβουλή μου και πάρετε ένα ποντίκι με 3-κουμπιά, θα θελήσετε να ρυθμίσετε την τακτική εστίασης στο στυλ "η εστίαση ακολουθεί το ποντίκι". Αυτό θα κάνει πολύ πιο εύκολη την χρήση της αντιγραφής κειμένου του X windows. Μπορεί, στην αρχή, να σας φανεί παράξενο ότι τα παράθυρα δεν σηκώνονται μπροστά όταν αποκτούν εστίαση (αφού

θα πρέπει να κάνετε κλικ στη γραμμή τίτλου/title bar για να το πετύχετε), αλλά θα σας αρέσει το ότι θα μπορείτε πλέον να δουλεύετε με περισσότερα παράθυρα, ταυτοχρόνως, χωρίς το κάθε ενεργό παράθυρο να κρύβει τα υπόλοιπα. Δοκιμάστε το και δώστε του μια ευκαιρία. Νομίζω ότι θα σας αρέσει. Μπορείτε να βρείτε αυτή τη ρύθμιση μέσα στα εργαλεία ρύθμισης του διαχειριστή παραθύρων.

2

Πλοήγηση

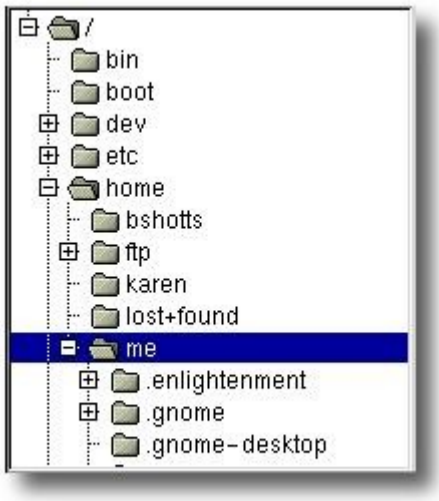
Σε αυτό το μάθημα, θα σας παρουσιάσω τις τρεις πρώτες σας εντολές: το **pwd** (print working directory/ εκτύπωση καταλόγου εργασίας), το **cd** (change directory/ αλλαγή καταλόγου), και το **ls** (list files and directories/ παράθεση αρχείων και καταλόγων σε λίστα).

Αν δεν έχετε ξαναδουλέψει στο παρελθόν με τη γραμμή εντολών, θα πρέπει να δώσετε ιδιαίτερη προσοχή σε αυτό το μάθημα, μιας και θα πάρει λίγο χρόνο για την εξοικείωση με αυτές τις έννοιες.

Οργάνωση συστήματος αρχείων

Όπως και σε αυτό το παρωχημένο (legacy) λειτουργικό σύστημα, τα αρχεία σε ένα σύστημα GNU/Linux αρχειοθετούνται σε αυτό που ονομάζεται *ιεραρχική δομή φακέλλων*. Αυτό σημαίνει ότι οι κατάλογοι, που σε άλλα συστήματα ονομάζονται φάκελλοι, οργανώνονται σε μία μορφή διακλαδώσεων δένδρου (tree-like pattern), που μπορούν να περιέχουν αρχεία και άλλους φακέλλους. Ο πρώτος *κατάλογος* στο σύστημα αρχείων, αποκαλείται *κατάλογος συστήματος*. Ο *κατάλογος συστήματος* περιέχει αρχεία και υποφακέλλους, που περιέχουν περισσότερα αρχεία και υποφακέλλους κ.ο.κ..

Τα περισσότερα γραφικά περιβάλλοντα που κυκλοφορούν σήμερα, περιλαμβάνουν ένα πρόγραμμα διαχείρισης αρχείων για την προβολή και διαχείριση των περιεχομένων του συστήματος αρχείων. Συχνά, θα δείτε την αναπαράσταση του συστήματος αρχείων όπως φαίνεται πιο κάτω:



Μια σημαντική διαφορά ανάμεσα στο γνωστό παρωχημένο (legacy) λειτουργικό σύστημα και το Unix/Linux, είναι ότι το GNU/Linux δεν χρησιμοποιεί το σύστημα γραμμάτων για τις συσκευές αποθήκευσης (δηλ. για τους οδηγούς). Ενώ τα γράμματα των οδηγών (drive letters) διασπούν το σύστημα αρχείων σε μια σειρά διαφορετικών δένδρων (ένα για κάθε οδηγό), το GNU/Linux έχει, πάντα, ένα μοναδικό δένδρο. Οι διάφορες συσκευές αποθήκευσης, μπορεί να περιέχουν διάφορους κλάδους του δένδρου, αλλά υπάρχει πάντοτε ένα ενιαίο δένδρο.

pwd

Αφού η διεπαφή γραμμής εντολών δεν μπορεί να προσφέρει γραφικές εικόνες της δομής του συστήματος αρχείων, θα πρέπει να διαθέτει έναν διαφορετικό τρόπο αναπαράστασής της. Φαντασθείτε το δένδρο του συστήματος αρχείων σαν έναν λαβύρινθο, και εσείς βρίσκεσθε μέσα του. Σε κάθε δεδομένη στιγμή, βρίσκεσθε και στέκεσθε σε ένα μεμονωμένο κατάλογο. Μέσα σε αυτό τον κατάλογο, μπορείτε να δείτε τα αρχεία του, καθώς και τη διαδρομή προς τον γονεϊκό του κατάλογο και τις διαδρομές προς τους υποκαταλόγους του καταλόγου εντός του οποίου βρίσκεσθε.

Ο κατάλογος μέσα στον οποίο βρίσκεσθε, ονομάζεται τρέχων *κατάλογος εργασίας*. Για να βρείτε το όνομα του τρέχοντος καταλόγου εργασίας, χρησιμοποιείστε την εντολή **pwd**.

```
[me@linuxbox me]$ pwd
/home/me
```

Όταν συνδέεσθε για πρώτη φορά στο σύστημα GNU/Linux, ο κατάλογος εργασίας ρυθμίζεται να είναι ο

κατάλογος χρήστη (home directory). Εκεί είναι που αποθηκεύετε τα αρχεία σας. Στα περισσότερα συστήματα, ο κατάλογος χρήστη θα ονομάζεται /home/your_user_name, αλλά μπορεί να ονομάζεται και οτιδήποτε, ανάλογα με τις επιθυμίες του διαχειριστή του συστήματος.

Για να προβάλλετε σε κατάλογο τα αρχεία του φακέλλου εργασίας, χρησιμοποιείστε την εντολή **ls**.

```
[me@linuxbox me]$ ls

Desktop     Xrootenv.0    linuxcmd
GNUstep     bin           nedit.rpm
GUILG00.GZ  hitni123.jpg  nsmail
```

Θα επιστρέψω στην εντολή ls στο επόμενο μάθημα. Υπάρχουν πολλά διασκεδαστικά πράγματα που μπορείτε να κάνετε με αυτή, αλλά πρέπει πρώτα να μιλήσω για ονόματα διαδρομών και καταλόγων.

cd

Για να αλλάξετε τον τρέχοντα κατάλογο εργασίας (δηλ., το σημείο στο οποίο βρίσκεσθε μέσα στον λαβύρινθο) χρησιμοποιήστε την εντολή **cd**. Για να το κάνετε αυτό, πληκτρολογείστε cd ακολουθούμενο από το *όνομα διαδρομής* του επιθυμητού καταλόγου εργασίας. Το όνομα διαδρομής είναι το μονοπάτι που παίρνετε κατά μήκος των κλαδιών του δένδρου, για να πάτε στο κατάλογο που θέλετε. Τα ονόματα διαδρομών μπορούν να διευκρινισθούν με έναν από δύο διαφορετικούς τρόπους: *Απόλυτα ονόματα διαδρομών* (absolute pathnames) ή *Σχετικά ονόματα διαδρομών* (relative pathnames). Ας ασχοληθούμε πρώτα με τα απόλυτα ονόματα διαδρομών.

Ένα απόλυτο όνομα διαδρομής αρχίζει με τον κατάλογο συστήματος και ακολουθεί το δένδρο κλαδί προς κλαδί, μέχρι να ολοκληρωθεί η διαδρομή προς τον επιθυμητό κατάλογο. Για παράδειγμα, στο σύστημά σας υπάρχει ένας κατάλογος μέσα στον οποίο εγκαθίστανται προγράμματα για το σύστημα X window. Το όνομα διαδρομής του καταλόγου είναι /usr/X11R6/bin. Αυτό σημαίνει ότι, στον κατάλογο συστήματος, που αντιπροσωπεύεται με μία αρχική κάθετο μπροστά από το όνομα της διαδρομής, υπάρχει ένας κατάλογος που ονομάζεται "usr" που περιέχει έναν κατάλογο με όνομα "X11R6" που περιέχει ένα κατάλογο με όνομα "bin".

Ας δοκιμάσουμε το εξής:

```
[me@linuxbox me]$ cd /usr/X11R6/bin

[me@linuxbox bin]$ pwd
/usr/X11R6/bin

[me@linuxbox bin]$ ls
Animate              import               xfwp
AnotherLevel         lbxproxy             xg3
Audio                listres              xgal
Auto                 lndir                xgammon
Banner               makedepend           xgc
Cascade              makeg                xgetfile
Clean                mergelib             xgopher
Form                 mkdirhier            xhexagons
Ident                mkfontdir            xhost
Pager                mkxauth              xieperf
Pager_noxpm          mogrify              xinit
RunWM                montage              xiterm
RunWM.AfterStep      mtv                  xjewel
RunWM.Fvwm95         mtvp                 xkbbell
RunWM.MWM            nxterm               xkbcomp
και πολλά περισσότερα...
```

Τώρα, μπορούμε να δούμε ότι αλλάξαμε τον τρέχοντα κατάλογο εργασίας στο /usr/X11R6/bin και ότι είναι γεμάτος με αρχεία. Προσέξατε ότι άλλαξε το σύμβολο της προτροπής σας (prompt); Για πρακτικούς λόγους, συνήθως έχει ρυθμισθεί να εμφανίζει το όνομα του καταλόγου εργασίας.

Όπου ένα απόλυτο όνομα διαδρομής αρχίζει από τον κατάλογο συστήματος (root directory) και οδηγεί στον προορισμό τους, ένα σχετικό όνομα διαδρομής αρχίζει από τον τρέχοντα φάκελλο εργασίας. Για να γίνει αυτό, χρησιμοποιεί δύο ειδικά σύμβολα για να αναπαραστήσει τις σχετικές θέσεις μέσα στο δένδρο αρχείων. Αυτά τα ειδικά σύμβολα είναι η τελεία "." (dot) και η διπλή τελεία ".." (dot dot).

Το σύμβολο της τελείας "." αναφέρεται στον τρέχοντα κατάλογο εργασίας, ενώ το σύμβολο της διπλής τελείας ".." αναφέρεται στον γονεϊκό κατάλογο του τρέχοντος καταλόγου εργασίας. Εδώ παρακάτω, μπορείτε να δείτε πως δουλεύει. Ας αλλάξουμε, ξανά, τον τρέχοντα κατάλογο εργασίας στο /usr/X11R6/bin :

```
[me@linuxbox me]$ cd /usr/X11R6/bin
[me@linuxbox bin]$ pwd
/usr/X11R6/bin
```

Εντάξει, τώρα ας πούμε ότι θέλουμε να αλλάξουμε τον τρέχοντα κατάλογο εργασίας στον γονεϊκό κατάλογο του /usr/X11R6/bin που είναι το /usr/X11R6. Αυτό μπορούμε να το πετύχουμε με δύο (2) διαφορετικούς τρόπους. Πρώτα, με τη βοήθεια ενός απολύτου ονόματος διαδρομής:

```
[me@linuxbox bin]$ cd /usr/X11R6
[me@linuxbox X11R6]$ pwd
/usr/X11R6
```

Είτε, διαφορετικά, με ένα σχετικό όνομα διαδρομής:

```
[me@linuxbox bin]$ cd ..
[me@linuxbox X11R6]$ pwd
/usr/X11R6
```

Πρόκειται για δύο διαφορετικές μεθόδους με τα ίδια αποτελέσματα. Ποια από τις δύο θα πρέπει να χρησιμοποιήσετε; Προφανώς εκείνη που απαιτεί τη λιγότερη πληκτρολόγηση!

Παρομοίως, μπορούμε να αλλάξουμε τον τρέχοντα κατάλογο εργασίας από το /usr/X11R6 στο /usr/X11R6/bin με 2 διαφορετικούς τρόπους. Ο πρώτος είναι με τη χρήση ενο απολύτου ονόματος διαδρομής.

```
[me@linuxbox X11R6]$ cd /usr/X11R6/bin
[me@linuxbox bin]$ pwd
/usr/X11R6/bin
```

Ο άλλος είναι με τη χρήση ενός σχετικού ονόματος διαδρομής:

```
[me@linuxbox X11R6]$ cd ./bin
[me@linuxbox bin]$ pwd
/usr/X11R6/bin
```

Τώρα, υπάρχει κάτι σημαντικό που θα πρέπει να υπογραμμίσω εδώ. Σχεδόν σε όλες τις περιπτώσεις, μπορείτε να παραλείψετε το "./". Απλώς αυτό υπονοείται. Πληκτρολογώντας:

```
[me@linuxbox X11R6]$ cd bin
```

θα πετύχουμε το ίδιο πράγμα. Σε γενικές γραμμές, αν δεν ορίσετε ένα όνομα διαδρομής προς μία δεδομένη τοποθεσία, τότε εξυπακούεται ότι θα πρόκειται για τον τρέχοντα φάκελλο εργασίας. Υπάρχει μόνο μια σημαντική εξαίρεση σε αυτό τον κανόνα, αλλά δεν θα την βρούμε μπροστά μας για λίγο.

Δύο συντομεύσεις

Αν πληκτρολογήσετε **cd** και αυτό δεν ακολουθείται από τίποτε, τότε η εντολή `cd` θα αλλάξει τον

τρέχοντα φάκελλο εργασίας στο δικό σας αρχικό κατάλογο χρήστη (home directory).

Μια σχετική συντόμευση είναι το να πληκτρολογήσετε την εντολή **`cd ~user_name`**. Σε αυτή την περίπτωση, η εντολή `cd` θα αλλάξει τον τρέχοντα φάκελλο εργασίας στον αρχικό φάκελλο (home directory) του χρήστη που ορίσατε.

Σημαντικά δεδομένα για τα ονόματα αρχείων

1. Τα ονόματα αρχείων που αρχίζουν με μια τελεία, είναι κρυφά αρχεία. Αυτό σημαίνει, απλώς, ότι η εντολή `ls` δεν θα σας τα εμφανίζει, εκτός και αν διευκρινίσετε `ls -a`. Όταν είχε δημιουργηθεί ο λογαριασμός σας, πολλά κρυφά αρχεία είχαν τοποθετηθεί στον αρχικό φάκελλό σας (home directory), για τη ρύθμιση διαφόρων παραμέτρων του λογαριασμού σας. Στην πορεία, θα εξετάσουμε λεπτομερέστερα μερικά από αυτά τα αρχεία, για να δούμε πως μπορείτε να παραμετροποιήσετε το περιβάλλον σας. Επιπρόσθετα, μερικές εφαρμογές θα τοποθετήσουν και αυτές αρχεία των δικών τους ρυθμίσεων μέσα στον αρχικό σας κατάλογο (home directory), υπό μορφή κρυφών αρχείων.
2. Τα ονόματα αρχείων στο GNU/Linux, όπως γενικότερα ισχύει στο Unix, είναι ευαίσθητα στη χρήση μικρών ή κεφαλαίων γραμμάτων (case sensitive). Π.χ., τα ονόματα αρχείων "Αρχείο1" και "αρχείο1" αναφέρονται σε δύο διαφορετικά αρχεία.
3. Στο GNU/Linux δεν υφίσταται η έννοια της επέκτασης αρχείου (file extension), όπως στα ιδιοταγή, κλειστά Λ.Σ.. Μπορείτε να δίνετε στα αρχεία ότι ονόματα θέλετε. Τα περιεχόμενα και ο σκοπός του αρχείου, καθορίζονται με άλλους τρόπους.
4. Παρόλο που το GNU/Linux υποστηρίζει τα μεγάλα ονόματα αρχείων, που μπορεί να περιέχουν ενσωματωμένα κενά διαστήματα ή σημεία στίξεως, περιορίστε την χρήση σημείων στίξεως στις τελείες, τις παύλες, και την υπόπαυλα (κάτω παύλα). *Και προπάντων, μην συμπεριλαμβάνετε κενά διαστήματα στα ονόματα των αρχείων σας*. Αν, σε ένα όνομα αρχείου, θέλετε να αναπαραστήσετε τα κενά διαστήματα ανάμεσα σε λέξεις, τότε χρησιμοποιήστε υπόπαυλες. Αργότερα, θα ευγνωμονείτε τον εαυτό σας για αυτό.

3

Ψάχνοντας τριγύρω

Τώρα που ξέρετε πως να μετακινείσθε από τον έναν κατάλογο εργασίας προς τον άλλο, θα δοκιμάσουμε να κάνουμε μια περιήγηση στο GNU/Linux σύστημά σας και, στην πορεία, να μάθουμε μερικά πράγματα για το πως λειτουργεί. Πριν, όμως, αρχίσουμε, πρέπει να σας δείξω μερικά εργαλεία που θα μας είναι πολύ χρήσιμα σε αυτή την περιπέτειά μας. Αυτά είναι:

- `ls` (εμφάνιση λίστας αρχείων και καταλόγων)
- `less` (προβολή των αρχείων κειμένου)
- `file` (κατάταξη περιεχομένων ενός αρχείου)

ls

Η εντολή `ls` χρησιμοποιείται για την εμφάνιση των περιεχομένων ενός καταλόγου. Είναι μάλλον η πιο πολυχρησιμοποιημένη εντολή στο GNU/Linux. Μπορεί να χρησιμοποιηθεί κατά πολλούς τρόπους. Ακολουθούν ορισμένα παραδείγματα:

Παραδείγματα της εντολής **ls**

Εντολή	***Αποτέλεσμα***
`ls`	Εμφάνιση των αρχείων του τρέχοντος καταλόγου εργασίας, σε μορφή λίστας
`ls /bin`	Εμφάνιση σε λίστα των αρχείων του καταλόγου /bin (ή του οποιουδήποτε άλλου καταλόγου που εσείς ορίσετε)
`ls -l`	Εμφάνιση σε λίστα των αρχείων του καταλόγου εργασίας, με αναλυτική μορφή (long format)
`ls -l /etc /bin`	Εμφάνιση σε λίστα, με αναλυτική μορφή (long format), των αρχείων του καταλόγου /bin και του καταλόγου /etc
`ls -la ..`	Εμφάνιση σε λίστα όλων των αρχείων του καταλόγου που είναι ένα γονεϊκός για τον κατάλογο εργασίας (ακόμη και εκείνων με όνομα που αρχίζει με μια τελεία, και που είναι συνήθως κρυφά), σε αναλυτική μορφή (long format)

Αυτά τα παραδείγματα τονίζουν, επίσης, και μία άλλη σημαντική έννοια για τις εντολές. Οι περισσότερες εντολές λειτουργούν ως εξής:

```
command -options arguments (εντολή -παράμετροι αρθρώματα)
```

όπου *εντολή (command)* είναι το όνομα της εντολής, το *-options* είναι μια ή παραπάνω ρυθμίσεις στη

συμπεριφορά της εντολής και τα *αρθρώματα (arguments)* είναι ένα ή παραπάνω “πράγματα” πάνω στα οποία εφαρμόζεται η εντολή.

Στην περίπτωση του `ls`, βλέπουμε ότι το όνομα της εντολής είναι `ls` και ότι μπορεί να έχει δύο ή παραπάνω παραμέτρους, όπως το `-a` και το `-l` και ότι μπορεί να λειτουργήσει σε έναν ή και περισσότερους καταλόγους/.

Μια προσεκτική ματιά στην Αναλυτική Μορφή (Long Format)

Αν χρησιμοποιείτε την παράμετρο `-l` με την εντολή `ls`, θα λάβετε μια λίστα αρχείων με έναν πλούτο πληροφοριών σχετικά με τα αρχεία που παρατίθενται σε λίστα. Ακολουθεί ένα παράδειγμα:

```
-rw-------   1 bshotts  bshotts       576 Apr 17  1998 weather.txt
drwxr-xr-x   6 bshotts  bshotts      1024 Oct  9  1999 web_page
-rw-rw-r--   1 bshotts  bshotts    276480 Feb 11 20:41 web_site.tar
-rw-------   1 bshotts  bshotts      5743 Dec 16  1998 xmas_file.txt

----------     -------  -------  -------- ------------ -------------
    |             |        |         |          |             |
    |             |        |         |          |         Όνομα αρχείου
    |             |        |         |          |
    |             |        |         |          +---  Χρονική στιγμή τροποποίησης
    |             |        |         |
    |             |        |         +-------------   Μέγεθος (σε bytes)
    |             |        |
    |             |        +------------------------        Ομάδα
    |             |
    |             +--------------------------------        Ιδιοκτήτης
    |
    +-----------------------------------------------  Δικαιώματα στο αρχείο
```

Όνομα Αρχείου (File Name)

Το όνομα του αρχείου ή του καταλόγου.

Χρόνος Τροποποίησης (Modification Time)

Είναι η τελευταία χρονική στιγμή κατά την οποία τροποποιήθηκε το αρχείο. Αν η τελευταία τροποποίηση έγινε παραπάνω από 6 μήνες πριν, τότε εμφανίζονται η ημερομηνία και το έτος. Διαφορετικά, εμφανίζεται η χρονική στιγμή της ημέρας.

Μέγεθος (Size)

Το μέγεθος του αρχείου σε bytes.

Ομάδα (Group)

Το όνομα της ομάδας που έχει δικαιώματα αρχείων, πέραν του ιδιοκτήτη των αρχείων.

Ιδιοκτήτης (Owner)

Το όνομα του ιδιοκτήτη του αρχείου.

Δικαιώματα Αρχείων (File Permissions)

Μια αναπαράσταση των δικαιωμάτων πρόσβασης των αρχείων. Ο πρώτος χαρακτήρας αντιστοιχεί στον τύπο αρχείου. Μία παύλα "-" δείχνει ότι πρόκειται για ένα (συνηθισμένο) αρχείου. Ένα "d" υποδεικνύει έναν κατάλογο. Η δεύτερη ομάδα των τριών χαρακτήρων αναπαριστά τα δικαιώματα ανάγνωσης, εγγραφής και εκτέλεσης του ιδιοκτήτη του αρχείου. Οι επόμενοι τρεις χαρακτήρες αντιπροσωπεύουν τα δικαιώματα της ομάδας των αρχείων, ενώ οι τελευταίοι τρεις αντιπροσωπεύουν τα δικαιώματα που παραχωρούνται προς οποιονδήποτε άλλον.

less

Το **less** είναι ένα πρόγραμμα που σας επιτρέπει να προβάλλετε αρχεία κειμένου. Αυτό είναι πολύ βολικό, αφού πολλά από τα αρχεία που χρησιμοποιούνται για τον έλεγχο και τη ρύθμιση του GNU/Linux είναι αναγνώσιμα από τον άνθρωπο (σε αντίθεση με εκείνο που συμβαίνει στα ιδιοταγή λειτουργικά συστήματα).

Τι είναι το "κείμενο";

Υπάρχουν πολλοί τρόποι για την αναπαράσταση μιας πληροφορίας σε έναν υπολογιστή. Όλες οι μέθοδοι περιλαμβάνουν τον καθορισμό των σχέσεων ανάμεσα στις πληροφορίες και ορισμένους αριθμούς που θα χρησιμοποιηθούν για την αναπαράστασή τους. Οι υπολογιστές, σε τελευταία ανάλυση, αντιλαμβάνονται μόνον τους αριθμούς και τα δεδομένα που είναι μετατρέψιμα σε αριθμητικά στοιχεία.

Μερικά από αυτά τα συστήματα αναπαράστασης είναι πολύ σύνθετα (όπως τα συμπιεσμένα αρχεία εικόνας), ενώ κάποια άλλα είναι μάλλον απλά. Ένα από πιο αρχικά και τα πιο απλά λέγεται *κείμενο ASCII*. Το ASCII (που προφέρεται "Άσκι") είναι μια συντόμευση που σημαίνει American Standard Code for Information Interchange. Αυτό είναι ένα σχήμα απλής κωδικοποίησης που είχε πρωτοχρησιμοποιηθεί στις μηχανές Teletype, για να αντιστοιχίσει τους χαρακτήρες του πληκτρολογίου με αριθμούς.

Το κείμενο είναι μια απλή αντιστοίχηση, ένα-προς-ένα, των χαρακτήρων με τους αριθμούς. Είναι πολύ συμπαγής. Πενήντα χαρακτήρες κειμένου μεταφράζονται σε πενήντα bytes δεδομένων. Σε ένα σύστημα GNU/Linux, πολλά αρχεία αποθηκεύονται υπό μορφή κειμένου και υπάρχουν πολλά εργαλεία GNU/Linux που δουλεύουν με αρχεία κειμένου. Ακόμη και τα ιδιοταγή λειτουργικά συστήματα αναγνωρίζουν τη σημασία αυτής της μορφής. Το πασίγνωστο πρόγραμμα NOTEPAD.EXE είναι ένα πρόγραμμα επεξεργασίας απλών κειμένων ASCII.

Το πρόγραμμα `less` καλείται απλώς πληκτρολογώντας:

```
less text_file
```

Αυτό θα εμφανίσει το αρχείο.

Έλεγχος του less

Όταν ξεκινήσει, το `less` θα εμφανίζει το αρχείο κειμένου σε μορφή μίας σελίδας ανά κάθε στιγμή. Μπορείτε να χρησιμοποιήσετε τα πλήκτρα Page Up και Page Down, για να μετακινηθείτε μέσα στο αρχείο κειμένου. Για να βγείτε από το `less`, πληκτρολογείστε "q". Ακολουθούν μερικές εντολές που θα γίνουν δεκτές από το `less`:

Εντολές πληκτρολογίου για το πρόγραμμα less

Εντολή	***Ενέργεια***
Page Up ή b	Μετακίνηση προς τα πίσω κατά μια σελίδα

Page Down ή κενό διάστημα (space)	Μετακίνηση προς τα εμπρός κατά μια σελίδα
G	Μεταβείτε στο τέλος του αρχείου κειμένου
1G	Μεταβείτε στην αρχή του αρχείου κειμένου
/*characters*	Αναζήτηση προς τα εμπρός στο αρχείο κειμένου, για μια συγκεκριμμένη σειρά/ αλληλουχία κάποιων *χαρακτήρων*
n	Επανάληψη της προηγούμενης αναζήτησης
q	Έξοδος

file

Καθώς περιηγείσθε ολόγυρα μέσα στο σύστημα GNU/Linux, είναι χρήσιμο να ξέρετε τι περιέχει ένα αρχείο, πριν ακόμη το ανοίξετε. Εδώ είναι που υπεισέρχεται η εντολή `file`. Η εντολή `file` θα εξετάσει το αρχείο και θα σας πει για τι είδους αρχείο πρόκειται.

Για να χρησιμοποιήσετε το πρόγραμμα `file`, απλώς πληκτρολογείστε:

```
file name_of_file
```

Η εντολή `file` μπορεί να αναγνωρίσει τους περισσότερους τύπους αρχείων, όπως:

Διάφορα είδη αρχείων

Τύπος Αρχείου	***Περιγραφή***	***Μπορεί να προβληθεί ως κείμενο;***
Κείμενο ASCII	Το όνομα τα λέι όλα	Ναι
Κείμενο σεναρίου (script) για Κέλυφος Bourne-Again	Ένα σενάριο (script) για το `bash`	Ναι
Βασικό (core) αρχείο ELF 32-bit LSB	Ένα βασικό αρχείο dump (θα δημιουργηθεί από κάποιο πρόγραμμα, όταν αυτό καταρρεύσει)	Όχι
Εκτελέσιμο ELF	Δυαδικό εκτελέσιμο πρόγραμμα	Όχι

32-bit LSB		
Κοινό αντικείμενο ELF 32-bit LSB	Κοινή βιβλιοθήκη	Όχι
Συμπιεσμένο αρχείο GNU tar	Ένα αρχείο tape archive. Συνηθισμένος τρόπος για την αποθήκευση ομάδων αρχείων.	Όχι, χρήση του `tar tvf` για εμφάνιση των περιεχομένων
Συμπιεσμένα δεδομένα gzip	Αρχείο συμπιεσμένο με το πρόγραμμα `gzip`	Όχι
Έγγραφο κειμένου HTML	Μια ιστοσελίδα	Ναι
Δεδομένα εικόνας JPEG	Συμπιεσμένη εικόνα σε μορφή JPEG	Όχι
Έγγραφο κειμένου PostScript	Αρχείο σε μορφή Υστερόγραφου (PostScript)	Ναι
RPM	Αρχείο Διαχειριστή Πακέτων Red Hat Package Manager	Όχι. Χρήση `rpm -q` για εξέταση περιεχομένου
Δεδομένα αρχείου Zip	Αρχείο συμπιεσμένο με το πρόγραμμα `zip`	Όχι

Αν και μπορεί να μοιάζει ωσάν τα περισσότερα είδη αρχείων να μην μπορούν να προβληθούν ως κείμενο, θα σας εκπλήξει για πόσα από αυτά είναι, τελικά, εφικτό. Αυτό αληθεύει ιδιαιτέρως για τα σημαντικά αρχεία ρυθμίσεων. Κατά τη διάρκεια της περιπέτειάς μας, θα διαπιστώσετε, επίσης, ότι πολλά από τα χαρακτηριστικά του λειτουργικού συστήματος, ελέγχονται από σενάρια κελύφους (shell scripts). Στο GNU/Linux, δεν υπάρχουν μυστικά!

4

Μια καθοδηγούμενη ξενάγηση

Είναι η σωστή στιγμή να κάνουμε την ξενάγησή μας. Ο παρακάτω πίνακας παραθέτει ορισμένες ενδιαφέρουσες τοποθεσίες προς εξερεύνηση. Δεν πρόκειται σε καμία περίπτωση για έναν πλήρη και εξαντλητικό κατάλογο, αλλά θα βοηθήσει σίγουρα να έχουμε μια ενδιαφέρουσα εμπειρία. Για καθέναν από τους παρακάτω φακέλλους, κάντε τα εξής:

- `cd` σε κάθε κατάλογο.
- Χρησιμοποιήστε το `ls` για την παράθεση των περιεχομένων του καταλόγου.
- Αν δείτε ένα ενδιαφέρον αρχείο, χρησιμοποιήστε την εντολή `file` για την εξακρίβωση του περιεχομένου του.
- Στην περίπτωση αρχείων κειμένου, χρησιμοποιήστε την εντολή `less` για να τα δείτε.

Ενδιαφέροντες κατάλογοι και τα περιεχόμενά τους

Φάκλλος	***Περιγραφή***
`/`	Ο κατάλογος συστήματος (root directory) όπου αρχίζει το σύστημα των αρχείων. Στις περισσότερες περιπτώσεις, περιέχει μόνον υποκαταλόγους.
`/boot`	Εδώ είναι το μέρος όπου φυλάσσονται ο πυρήνας Linux και το πρόγραμμα φόρτωσης της εκκίνησης (boot loader). Ο πυρήνας είναι το αρχείο με το όνομα `vmlinuz`.

`/etc`	Ο κατάλογος `/etc` περιέχει τα αρχεία ρύθμισης του συστήματος. Όλα τα αρχεία του `/etc` πρέπει να είναι αρχεία κειμένου. Σημαντικά σημεία:
	
	`/etc/passwd`
	Το αρχείο `passwd` περιέχει όλες τις σημαντικές πληροφορίες για τον κάθε χρήστη. Εδώ είναι το μέρος όπου ορίζονται οι χρήστες.
	`/etc/fstab`
	Το αρχείο `fstab` περιέχει έναν πίνακα συσκευών που προσαρτώνται κατά την εκκίνηση του συστήματος. Αυτό είναι το αρχείο που ορίζει τους δίσκους αποθήκευσης δεδομένων (disk drives) σας.
	`/etc/hosts`
	Αυτό το αρχείο περιέχει τον κατάλογο με τα ονόματα δικτύου (host names) και τις διευθύνσεις IP που είναι εσωτερικά γνωστές στο σύστημα.
	`/etc/init.d`
	Αυτός ο κατάλογος περιέχει τα σενάρια (scripts) που εκκινούν διάφορες υπηρεσίες συστήματος, τυπικά κατά τη φάση εκκίνησης.

`/bin`, `/usr/bin`	Αυτοί οι δύο κατάλογοι περιέχουν τα περισσότερα προγράμματα του συστήματος. Ο κατάλογος `/bin` έχει τα βασικά προγράμματα που απαιτούνται για τη λειτουργία του συστήματος, ενώ το `/usr/bin` περιέχει τις εφαρμογές των χρηστών του συστήματος.
`/sbin`,	Οι κατάλογοι `sbin` περιέχουν προγράμματα για τη διαχείριση συστήματος, κυρίως

/usr/sbin	προς χρήση από τον υπερχρήστη (superuser).
/usr	Ο φάκελλος /usr περιέχει διάφορα πράγματα που υποστηρίζουν τις εφαρμογές των χρηστών. Μερικά παραδείγματα: /usr/share/X11 Υποστηρίζει αρχεία για το σύστημα X Windows /usr/share/dict Λεξικά για τον ορθογραφικό διορθωτή. Στοίχημα πως δεν γνωρίζατε ότι το GNU/ Linux είχε ορθογραφικό διορθωτή. Δείτε το look και το ispell. /usr/share/doc Διάφορα αρχεία τεκμηρίωσης, σε μια ποικιλία μορφών. /usr/share/man Οι σελίδες man φυλάσσονται εδώ. /usr/src Αρχεία πηγαίου κώδικα. Αν εγκαταστήσατε το πακέτο με τις πηγές του πυρήνα, τότε θα βρείτε εδώ τον πηγαίο κώδικα όλου του πυρήνα Linux.
/usr/local	Το /usr/local και όλοι οι υποκατάλογοί του, χρησιμοποιούνται για την εγκατάσταση λογισμικού και άλλων αρχείων, προς χρήση στο τοπικό μηχάνημα. Αυτό σημαίνει ότι όλα εκείνα τα λογισμικά που δεν αποτελούν τμήμα της επίσημης διανομής (η οποία συνήθως πάει στο /usr/bin) θα τα βρείτε εδώ. Όταν θα βρείτε ενδιαφέροντα προγράμματα να εγκαταστήσετε στο σύστημά σας, θα πρέπει να εγκατασταθούν σε ένα από τους φακέλλους του /usr/local. Τις περισσότερες φορές, ο κατάλογος επιλογής θα είναι το /usr/local/bin.
/var	Ο κατάλογος /var περιέχει αρχεία που αλλάζουν, καθώς το σύστημα τρέχει. Αυτά περιλαμβάνουν: /var/log Κατάλογος με τα αρχεία καταγραφής (log files). Αυτά ενημερώνονται καθώς το σύστημα τρέχει. Θα πρέπει να τα βλέπετε κάθε τόσο, για να μπορείτε έτσι να επιβλέπετε την γενικότερη υγεία του συστήματός σας. /var/spool Αυτός ο κατάλογος χρησιμοποιείται για να κρατά αρχεία που μπαίνουν σε ουρά για κάποιες διεργασίες, όπως τα μηνύματα ταχυδρομείου και οι εργασίες εκτύπωσης. Όταν ένα μήνυμα ταχυδρομείου του χρήστη φθάσει πρώτη φορά στο τοπικό σύστημα (εφόσον προϋποθέσουμε πως έχετε τοπικό ταχυδρομείο), τα μηνύματα αυτά θα αποθηκευθούν πρώτα στο κατάλογο /var/spool/mail
/lib	Οι κοινές - διαμοιραζόμενες (shared) βιβλιοθήκες (όμοιες με τα DLL εκείνου του άλλου λειτουργικού συστήματος) φυλάσσονται εδώ.
/home	Το /home είναι το μέρος όπου οι χρήστες φυλάσσουν τις προσωπικές τους εργασίες. Γενικά, είναι το μόνο μέρος όπου επιτρέπεται στους χρήστες να

	εγγράφουν αρχεία. Αυτό το σύστημα κρατά τα πράγματα όμορφα σε τάξη :-)
`/root`	Αυτός είναι ο αρχικός κατάλογος του υπερχρήστη (superuser).
`/tmp`	Το `/tmp` είναι ένας κατάλογος μέσα στον οποίο τα προγράμματα μπορούν να γράφουν τα προσωρινά τους αρχεία.
`/dev`	Ο κατάλογος `/dev` είναι ένας ειδικός κατάλογος, αφού δεν περιέχει πραγματικά αρχεία, με την συνηθισμένη, συμβατική έννοια, παρά μόνο συσκευές (devices), που διατίθενται στο σύστημα. Το GNU/Linux (όπως το Unix), διαχειρίζεται τις συσκευές σαν αρχεία. Μπορείτε να διαβάσετε ή να γράψετε συσκευές, σαν να ήταν αρχεία. Για παράδειγμα, το `/dev/fd0` είναι ο πρώτος οδηγός ανάγνωσης μαλακής δισκέτας (floppy disk drive), ενώ το`/dev/sda` (μπορεί να ονομάζεται και `/dev/hda`, σε παλιότερα συστήματα) είναι ο πρώτος σκληρός δίσκος IDE (hard drive). Όλες οι συσκευές που αναγνωρίζει ο πυρήνας, αντιπροσωπεύονται εδώ.
`/proc`	Ο κατάλογος `/proc` είναι και αυτός ιδιαίτερος. Αυτός δεν περιέχει αρχεία. Στην πραγματικότητα, αυτός ο κατάλογος δεν υπάρχει καν! Είναι εξ' ολοκλήρου εικονικός. Ο κατάλογος `/proc` περιέχει μικρές “τρυπίτσες”, από όπου μπορείτε να κοιτάξετε μέσα στον ίδιο τον πυρήνα. Εδώ θα βρείτε μια ομάδα αριθμημένων καταχωρήσεων, που αντιστοιχούν σε όλες τις διεργασίες που τρέχουν στο σύστημα. Επιπλέον, υπάρχουν και διάφορες ονοματισμένες καταχωρήσεις που επιτρέπουν την πρόσβαση στις τρέχουσες ρυθμίσεις του συστήματος. Μπορείτε να δείτε τις περισσότερες από αυτές. Προσπαθείστε να δείτε το αρχείο `/proc/cpuinfo`. Αυτή η καταχώρηση θα σας πει τι νομίζει ο πυρήνας για τον επεξεργαστή σας (CPU).
`/media, /mnt`	Τέλος, ας έρθουμε και στο `/media`, έναν κανονικό κατάλογο, που χρησιμοποιείται με έναν ειδικό τρόπο. Ο κατάλογος `/media` χρησιμοποιείται για τα *σημεία προσάρτησης*. Όπως μάθαμε στο δεύτερο μάθημα, οι διάφορες συσκευές φυσικής αποθήκευσης (πχ. σκληροί δίσκοι), προσαρτώνται στο σύστημα του δένδρου των αρχείων, σε διάφορα σημεία. Αυτή η διαδικασία “προσκόλλησης” μιας συσκευής στο δένδρο αρχείων, ονομάζεται *προσάρτηση*. Για να μπορεί μια συσκευή να είναι διαθέσιμη, πρέπει πρώτα να προσαρτηθεί. Όταν το σύστημά σας κάνει εκκίνηση, διαβάζει μια λίστα με τις οδηγίες προσάρτησης, από το αρχείο `/etc/fstab`, που περιγράφει ποια συσκευή είναι προσαρτημένη, σε ποιο ακριβώς σημείο του δένδρου των αρχείων. Αυτό θα φροντίσει τους σκληρούς δίσκους, αλλά μπορείτε επίσης να έχετε συσκευές που θεωρούνται προσωρινές, όπως ένα CD-ROM ή μια μαλακή δισκέτα. Αφού πρόκειται για αφαιρέσιμα μέσα, δεν θα είναι προσαρτημένα συνεχώς. Ο κατάλογος `/media` χρησιμοποιείται από τους μηχανισμούς αυτόματης προσάρτησης συσκευών, που προσφέρουν οι σύγχρονες διανομές GNU/Linux για επιτραπέζιους υπολογιστές. Σε συστήματα που απαιτούν την χειροκίνητη προσάρτηση των αφαιρέσιμων συσκευών, ο κατάλογος `/mnt` προσφέρει μια βολική θέση για την προσάρτηση αυτών των προσωρινών συσκευών. Θα δείτε συχνά τους καταλόγους `/mnt/floppy` και `/mnt/cdrom`. Για να ελέγξετε τι συσκευές και ποια σημεία προσάρτησης χρησιμοποιούν, πληκτρολογήστε την εντολή `mount`.

Ένα παράξενο είδος αρχείου...

Κατά την ξενάγησή σας, ίσως θα παρατηρήσατε ένα παράξενο είδος αρχείου, ειδικά στους καταλόγους `/boot` και `/lib`. Όταν αυτοί εμφανίζονται με τη χρήση της εντολής `ls -l`, θα προσέξατε κάτι σαν κι' αυτό:

```
lrwxrwxrwx     25 Jul  3 16:42 System.map -> /boot/System.map-2.0.36-3
-rw-r--r-- 105911 Oct 13  1998 System.map-2.0.36-0.7
-rw-r--r-- 105935 Dec 29  1998 System.map-2.0.36-3
-rw-r--r-- 181986 Dec 11  1999 initrd-2.0.36-0.7.img
-rw-r--r-- 182001 Dec 11  1999 initrd-2.0.36.img
lrwxrwxrwx     26 Jul  3 16:42 module-info -> /boot/module-info-2.0.36-3
-rw-r--r--  11773 Oct 13  1998 module-info-2.0.36-0.7
-rw-r--r--  11773 Dec 29  1998 module-info-2.0.36-3
lrwxrwxrwx     16 Dec 11  1999 vmlinuz -> vmlinuz-2.0.36-3
-rw-r--r-- 454325 Oct 13  1998 vmlinuz-2.0.36-0.7
-rw-r--r-- 454434 Dec 29  1998 vmlinuz-2.0.36-3
```

Προσέξτε τα αρχεία: `System.map, module-info` και `vmlinuz`. Βλέπετε τις περίεργες σημειώσεις που ακολουθούν μετά τα ονόματα των αρχείων;

Αυτά τα τρία αρχεία λέγονται *συμβολικοί δεσμοί*. Οι συμβολικοί δεσμοί (symbolic links) είναι ένα ιδιαίτερος τύπος αρχείων, που παραπέμπουν προς κάποιο άλλο αρχείο. Με τους συμβολικούς δεσμούς, είναι δυνατό ένα οποιοδήποτε μεμονωμένο αρχείο, να διαθέτει πολλαπλά ονόματα. Εξηγούμε παρακάτω πως λειτουργεί αυτό: Οποτεδήποτε εμείς δώσουμε στο σύστημα ένα όνομα αρχείου που είναι ένας συμβολικός δεσμός, τότε αυτό το καταχωρεί διαφανώς (transparently maps) στο αρχείο προς το οποίο γίνεται η παραπομπή.

Σε τι ακριβώς μας χρειάζεται; Πρόκειται για ένα πολύ βολικό χαρακτηριστικό. Ας πάρουμε για παράδειγμα τα περιεχόμενα του παραπάνω καταλόγου (είναι ο κατάλογος `/boot` ενός παλιού συστήματος Red Hat 5.2). Αυτό το σύστημα είχε πολλές εγκατεστημένες εκδόσεις του πυρήνα Linux. Αυτό μπορούμε να το διαπιστώσουμε από τα αρχεία `vmlinuz-2.0.36-0.7` και `vmlinuz-2.0.36-3`. Τα ονόματα αυτών των αρχείων υποδεικνύουν ότι είναι εγκατεστημένες τόσο η έκδοση 2.0.36-0.7, όσο και η 2.0.36-3. Επειδή τα ονόματα των αρχείων περιέχουν την έκδοση, είναι εύκολο να δούμε τις διαφορές, στα περιεχόμενα του καταλόγου. Αυτό θα προκαλούσε, πάντως, σύγχυση, στην περίπτωση προγραμμάτων που έχουν ένα σταθερό όνομα για το αρχείο του πυρήνα. Αυτά τα προγράμματα μπορεί να θεωρούν πως ο πυρήνας θα λέγεται απλώς `"vmlinuz"`. Εδώ είναι που υπεισέρχεται η ομορφιά του συμβολικού δεσμού. Δημιουργώντας ένα συμβολικό δεσμό με το όνομα `vmlinuz`, που να παραπέμπει στο `vmlinuz-2.0.36-3`, επιλύουμε το πρόβλημα.

Για να δημιουργήσετε συμβολικούς δεσμούς, χρησιμοποιήστε την εντολή `ln`.

5

Διαχείριση Αρχείων

Αυτό το μάθημα θα σας εισαγάγει στις εξής εντολές:

- `cp` – αντιγραφή (copy) αρχείων και καταλόγων
- `mv` – μετακίνηση (move) ή μετονομασία αρχείων και καταλόγων
- `rm` – απομάκρυσνη (διαγραφή/ remove) αρχείων και καταλόγων
- `mkdir` – δημιουργία καταλόγων (make directories)

Αυτές είναι ανάμεσα στις πιο πολυχρησιμοποιημένες εντολές στο περιβάλλον GNU/Linux. Πρόεκιται για τις πιο βασικές εντολές για τη διαχείριση αρχείων, αλλά και ολόκληρων καταλόγων.

Τώρα, για να είμαστε ειλικρινείς, μερικές από τις εργασίες που εκτελούν αυτές οι εντολές, γίνονται πιο εύκολα με έναν διαχειριστή αρχείων σε γραφικό περιβάλλον. Με ένα διαχειριστή αρχείων, μπορείτε να σύρετε και να αφήσετε (drag 'n drop) ένα αρχείο, από το έναν κατάλογο στον άλλον, να κάνετε αποκοπή και επικόλληση (cut and paste) αρχείων, διαγραφή αρχείων, κλπ. Συνεπώς, για ποιο λόγο να χρησιμοποιεί κανείς αυτά τα παλιά προγράμματα από τη γραμμή εντολών;

Η απάντηση βρίσκεται στην ισχύ και την ευελιξία. Αν και η εκτέλεση απλών χειρισμών με τα αρχεία, με ένα διαχειριστή αρχείων σε γραφικό περιβάλλον, είναι εύκολη, εν τούτοις, οι πιο σύνθετες εργασίες μπορεί να είναι ευκολότερες με τα προγράμματα από τη γραμμή εντολών. Για παράδειγμα, πως θα μπορούσατε να αντιγράψετε όλα τα αρχεία HTML από τον έναν κατάλογο στον άλλον, αλλά αντιγράφοντας μόνον τα αρχεία που δεν βρίσκονται στον κατάλογο προορισμού, ή μόνο τα αρχεία που είναι νεότερα από τις εκδόσεις του καταλόγου προορισμού; Πολύ δύσκολο να το πετύχετε με ένα διαχειριστή αρχείων. Αντιθέτως, πολύ εύκολο με την γραμμή εντολών:

```
[me@linuxbox me]$ cp -u *.html destination
```

Χαρακτήρες μπαλαντέρ (Wildcards)

Πριν αρχίσω με τις εντολές μας, θα ήθελα να μιλήσω για ένα χαρακτηριστικό του κελύφους που καθιστά αυτές τις εντολές τόσο ισχυρές. Επειδή το κέλυφος χρησιμοποιεί τα ονόματα αρχείων τόσο πολύ, παρέχει ειδικούς χαρακτήρες για να σας βοηθήσει να ορίσετε γρήγορα ομάδες με ονόματα αρχείων. Αυτοί οι ειδικοί χαρακτήρες ονομάζονται *χαρακτήρες μπαλαντέρ (wildcards)*. Αυτοί οι χαρακτήρες μπαλαντέρ, σας επιτρέπουν να επιλέγετε ονόματα αρχείων που βασίζονται σε πρότυπα (patterns) χαρακτήρων. Ο παρακάτω πίνακας παραθέτει σε λίστα τους διάφορους χαρακτήρες μπαλαντέρ και τι ακριβώς κάνουν:

Σύνοψη των χαρακτήρων μπαλαντέρ (wildcards) και της σημασίας τους

Μπαλαντέρ	***Σημασία***
*	Ταιριάζει σε οποιουσδήποτε χαρακτήρες
?	Ταιριάζει με οποιονδήποτε μεμονωμένο χαρακτήρα
[characters]	Ταιριάζει με οποιονδήποτε χαρακτήρα που ανήκει σε μια ομάδα *χαρακτήρων*. Η ομάδα χαρακτήρων μπορεί να εκφρασθεί και ως ένας *χαρακτήρας κλάσης*

	POSIX, όπως ένας από τους παρακάτω: Ομάδες χαρακτήρων Posix *[:alnum:]* Αλφαριθμητικοί χαρακτήρες *[:alpha:]* Αλφαβητικοί χαρακτήρες *[:digit:]* Αριθμητικά ψηφία *[:upper:]* Κεφαλαίοι αλφαβητικοί χαρακτήρες *[:lower:]* Αλφαβητικοί χαρακτήρες με μικρά γράμματα
[! characters]	Ταιριάζει με οποιονδήποτε χαρακτήρα δεν ανήκει σε μια ομάδα *χαρακτήρων.*

Με τη χρήση των wildcards, είναι εφικτή η δημιουργία πολύ σύνθετων κριτηρίων επιλογής για τα ονοματα αρχείων. Ακολουθούν ορισμένα παραδείγματα για patterns και με τι αντιστοιχούν:

Παραδείγματα αντιστοίχησης των χαρακτήρων μπαλαντέρ (wildcards)

Πρότυπο	***Αντιστοιχήσεις***
*	Όλα τα ονόματα αρχείων
*g**	Όλα τα ονόματα αρχείων που αρχίζουν με το γράμμα "g"
b.txt*	Όλα τα ονόματα αρχείων που αρχίζουν με το γράμμα "b" και τελειώνουν με τα γράμματα ".txt"
Data???	Οποιοδήποτε όνομα αρχείου που αρχίζει με τους χαρακτήρες "Data", ακολουθούμενο από ακριβώς τρεις ακόμη χαρακτήρες.
*[abc]**	Οποιοδήποτε όνομα αρχείου που αρχίζει με το "a" ή το "b" ή το "c", ακολουθούμενο από οποιουσδήποτε χαρακτήρες
*[[:upper:]]**	Οποιοδήποτε όνομα αρχείου που αρχίζει με ένα κεφαλαίο γράμμα. Αυτό είναι ένα παράδειγμα μιας ομάδας χαρακτήρων.
BACKUP. [[:digit:]] [[:digit:]]	Άλλο ένα παράδειγμα ομάδας χαρακτήρων. Αυτό το πρότυπο ταιριάζει με οποιοδήποτε όνομα αρχείου που αρχίζει με τους χαρακτήρες "BACKUP.", ακολουθούμενο από ακριβώς δύο αριθμητικά ψηφία.
[![:lower:]]	Οποιοδήποτε όνομα αρχείου δεν τελειώνει με ένα μικρό γράμμα.

Μπορείτε να χρησιμοποιείτε χαρακτήρες μπαλαντέρ (wildcards) με οποιαδήποτε εντολή που δέχεται ορίσματα ονόματος αρχείων.

cp

Το πρόγραμμα **cp** αντιγράφει αρχεία και καταλόγους. Στην απλούστερη μορφή τους, αντιγράφει ένα μεμονωμένο αρχείο:

```
[me@linuxbox me]$ cp file1 file2
```

Μπορεί επίσης να χρησιμοποιηθεί για την αντιγραφή πολλαπλών αρχείων σε έναν διαφορετικό κατάλογο:

```
[me@linuxbox me]$ cp file1 file2 file3 directory
```

Άλλα χρήσιμα παραδείγματα της εντολής cp και των επιλογών της περιλαμβάνουν:

Παραδείγματα της εντολής cp

Εντολή	***Αποτελέσματα***
cp file1 file2	Αντιγραφή των περιεχομένων του *file1* στο *file2*. Αν το *file2* δεν υπάρχει, τότε θα δημιουργηθεί. Αλλιώς, τα περιεχόμενα του το *file1* εγγράφονται πάνω από τα περιεχόμενα του *file2*.
cp -i file1 file2	Όπως και πιο πάνω, εφόσον επιλεγεί το όρισμα "-i" (διαδραστικό), αν το *file2* υπάρχει, τότε ο χρήστης προειδοποιείται, πριν γίνει αντικατάσταση με τα περιεχόμενα του *file1*.
cp file1 dir1	Αντιγραφή των περιεχομένων του *file1* (σε ένα αρχείο με το όνομα *file1*) μέσα στον κατάλογο *dir1*.
cp -R dir1 dir2	Αντιγραφή των περιεχομένων του καταλόγου *dir1*. Αν ο κατάλογος *dir2* δεν υπάρχει, τότε θα δημιουργηθεί. Αλλιώς, θα δημιουργηθεί ένας κατάλογος με το όνομα *dir1* μέσα στον κατάλογο *dir2*.

mv

Η εντολή **mv** εκτελεί δύο διαφορετικές λειτουργίες, ανάλογα με το πως χρησιμοποιείται.. Είτε θα μετακινήσει ένα ή περισσότερα αρχεία σε έναν διαφορετικό κατάλογο, ή θα μετονομάσει ένα αρχείο ή έναν κατάλογο. Για τη μετονομασία ενός αρχείου, χρησιμοποιείται ως εξής:

```
[me@linuxbox me]$ mv filename1 filename2
```

Για τη μετακίνηση αρχείων σε έναν διαφορετικό κατάλογο:

```
[me@linuxbox me]$ mv file1 file2 file3 directory
```

Παραδείγματα της εντολής mv και των ορισμάτων της, περιλαμβάνουν:

Παραδείγματα της εντολής mv

Εντολή	***Αποτελέσματα***

mv file1 file2	Αν το *file2* δεν υπάρχει, τότε το *file1* μετονομάζεται σε *file2*. Αν το *file2* υπάρχει, τα περιεχόμενά του αντικαθίστανται με τα περιεχόμενα του *file1*.
mv -i file1 file2	Όπως και πιο πάνω, εφόσον επιλεγεί το όρισμα "-i" (διαδραστικό), αν το *file2* υπάρχει, ο χρήστης προειδοποιείται πριν γίνει αντικατάσταση με τα περιεχόμενα του *file1*.
mv file1 file2 file3 dir1	Τα αρχεία *file1, file2, file3* μετακινούνται στον κατάλογο *dir1*. Το *dir1* πρέπει να υπάρχει, ειδάλλως το `mv` θα δώσει μια έξοδο με ένα σφάλμα.
mv dir1 dir2	Αν το *dir2* δεν υπάρχει, τότε το *dir1* μετονομάζεται σε *dir2*. Αν το *dir2* υπάρχει, τότε δημιουργείται ο κατάλογο *dir1*, μέσα στον κατάλογο *dir2*.

rm

Η εντολή `rm` διαγράφει (απομακρύνει) αρχεία και καταλόγους.

```
[me@linuxbox me]$ rm file
```

Μπορεί, επίσης, να χρησιμοποιηθεί για τη διαγραφή ενός καταλόγου:

```
[me@linuxbox me]$ rm -r directory
```

Παραδείγματα της εντολής `rm` και των ορισμάτων της, περιλαμβάνουν:

Παραδείγματα της εντολής rm

Εντολή	***Αποτελέσματα***
rm file1file2	Διαγραφή του *file1* και του *file2*.
rm-ifile1file2	Όπως και πιο πάνω, εφόσον έχει επιλεγεί το όρισμα "-i" (διαδραστικό), ο χρήστης προειδοποιείται πριν τη διαγραφή του κάθε αρχείου.
rm-rdir1dir2	Οι κατάλογοι *dir1* και *dir2* διαγράφονται, μαζί με όλα τα περιεχόμενά τους.

Να είστε προσεκτικοί με την εντολή rm!

Στο GNU/Linux δεν υπάρχει εντολή αντιστροφής ή αναίρεσης της διαγραφής. Άπαξ και διαγράψετε ένα αρχείο με την εντολή `rm`, πάει, πέταξε! Μπορείτε να επιφέρετε τρομερή ζημιά στο σύστημά σας με την εντολή `rm` αν δεν είστε προσεκτικοί, ιδιαιτέρως με τη χρήση χαρακτήρων μπαλαντέρ.

Πριν λοιπόν χρησιμοποιήσετε το `rm` με τους χαρακτήρες μπαλαντέρ, προσπαθήστε να εφαρμόσετε το εξής κόλπο: δώστε, αντιθέτως, την εντολή σας με τη χρήση του `ls`. Κάνοντάς το, μπορείτε να δείτε το αποτέλεσμα των wildcards πριν τη διαγραφή των αρχείων σας. Μετά τη δοκιμή της εντολής σας με το `ls`, ανακαλέστε την εντολή με το πλήκτρο με το βέλος προς τα επάνω (up-arrow) και, μετά, αντικαταστήστε το `rm` με το `ls`, μέσα στην εντολή.

mkdir

Η εντολή `mkdir` χρησιμοποιείται για τη δημιουργία καταλόγων. Για να τη χρησιμοποιήσετε, απλώς πληκτρολογείστε:

```
[me@linuxbox me]$ mkdir directory
```

6

Δουλεύοντας με εντολές

Μέχρι τώρα, είδατε μερικές εντολές και τις μυστηριώδεις τους παραμέτρους και τα αρθρώματα. Σε αυτό το μάθημα, θα προσπαθήσουμε να απαλείψουμε ένα κομμάτι αυτού του μυστηρίου. Αυτό το μάθημα θα σας εισαγάγει στις εξής εντολές.

- `type` – Εμφανίζει πληροφορίες για το είδος της εντολής
- `which` – Εντοπισμός μιας εντολής
- `help` – Εμφάνιση σελίδας αναφοράς για εντολή εσωτερική του κελύφους
- `man` – Εμφάνιση ενός on-line καταλόγου αναφοράς με επεξηγήσεις των εντολών

Τι είναι οι "εντολές" ;

Οι εντολές μπορούν να ανήκουν σε ένα από τα παρακάτω διαφορετικά είδη:

1. **Ένα εκτελέσιμο πρόγραμμα**, σαν όλα εκείνα τα αρχεία που είδαμε στο /usr/bin. Στα πλαίσια αυτής της κατηγορίας, τα προγράμματα μπορεί να είναι μεταγλωττισμένα *δυαδικά (compiled binaries)*, σαν τα προγράμματα που είναι γραμμένα σε C ή σε C++, ή σαν τα προγράμματα που είναι γραμμένα σε *γλώσσες σεναριακού προγραμματισμού (scripting languages)* όπως το κέλυφος, η Perl, η Python, η Ruby, κλπ.
2. **Μια εντολή ενσωματωμένη μέσα στο ίδιο το κέλυφος.** Το κέλυφος (bash) προσφέρει διάφορες εσωτερικές εντολές που λέγονται *shell builtins*. Η εντολή `cd`, για παράδειγμα, είναι ένα shell builtin.
3. **Μια συνάρτηση κελύφους** (shell function). Πρόκειται για μινιατούρες σεναριακών προγραμμάτων (miniature shell scripts), που είναι ενσωματωμένα στο *περιβάλλον*. Θα καλύψουμε αυτό το θέμα, κάνοντας ρυθμίσεις περιβάλλοντος και γράφοντας συναρτήσεις κελύφους, στα επόμενα μαθήματα, αλλά προς το παρόν, απλώς να έχετε υπ' όψη σας ότι υπάρχουν.
4. **Ένα ψευδώνυμο** (alias). Εντολές που μπορείτε να ορίσετε εσείς οι ίδιοι, που φτιάχνονται από άλλες εντολές. Αυτές θα καλυφθούν σε ένα μεταγενέστερο μάθημα.

Ταυτοποίηση εντολών

Είναι συχνά χρήσιμο να ξέρουμε ποιο ακριβώς από τα τέσσερα είδη εντολών χρησιμοποιείται και το GNU/Linux προσφέρει δύο τρόπους για να το βρούμε.

type

Η εντολή `type` είναι μια εσωτερική εντολή κελύφους που εμφανίζει ποιο είδος εντολής χρησιμοποιεί το κέλυφος, με δεδομένο ένα συγκεκριμμένο όνομα εντολής. Λειτουργεί ως εξής:

```
type command
```

όπου “command” είναι το όνομα της εντολής που θέλουμε να εξετάσουμε. Ακολουθούν μερικά παραδείγματα:

```
[me@linuxbox me]$ type type
type is a shell builtin
[me@linuxbox me]$ type ls
```

```
ls is aliased to `ls --color=tty'
```

```
[me@linuxbox me]$ type cp
cp is /bin/cp
```

Εδώ, βλέπουμε τα αποτελέσματα τριών διαφορετικών εντολών. Παρατηρείστε εκείνα για το ls (που προέρχονται από ένα σύστημα Fedora) και πως η εντολή ls είναι πράγματι ένα ψευδώνυμο (alias) για την εντολή ls με την προσθήκη της παραμέτρου "-- color=tty". Τώρα ξέρουμε γιατί η έξοδος από το ls εμφανίζεται έγχρωμα!

which

Μερικές φορές, υπάρχει παραπάνω από μια έκδοση ενός εκτελέσιμου προγράμματος που είναι εγκατεστημένο σε ένα σύστημα. Αν και αυτό δεν είναι πολύ συχνό σε ένα σύστημα επιτραπέζιου συστήματος, δεν είναι ασυνήθιστο σε μεγάλους διακομιστές. Για την εντόπιση της ακριβούς τοποθεσίας ενός δεδομένου εκτελέσιμου, χρησιμοποιείται η εντολή `which`:

```
[me@linuxbox me]$ which ls
/bin/ls
```

Η εντολή `which` λειτουργεί μόνον για εκτελέσιμα προγράμματα, ούτε για εσωτερικές εντολές, ούτε για ψευδώνυμα (aliases) που είναι υποκατάστατα των πραγματικών εκτελέσιμων προγραμμάτων.

Λήψη τεκμηρίωσης εντολών

Με αυτή τη γνώση για το τι είναι μια εντολή, μπορούμε τώρα να αναζητήσουμε την τεκμηρίωση που είναι διαθέσιμη για κάθε είδος εντολής.

help

Το `bash` έχει ένα ενσωματωμένο εργαλείο βοήθειας που είναι διαθέσιμο για κάθε μια από τις εσωτερικές εντολές. Για να το χρησιμοποιήσετε, πληκτρολογείστε την εντολή "help", ακολουθούμενη από το όνομα της εσωτερικής εντολής κελύφους. Προαιρετικά, μπορείτε να προσθέσετε το όρισμα -m για να αλλάξετε τη μορφή της εξόδου. Για παράδειγμα:

```
[me@linuxbox me]$ help -m cd
NAME
    cd – Αλλαγή του τρέχοντος καταλόγου εργασίας του κελύφους.

ΣΥΝΟΨΗ
    cd [-L|-P] [dir]

ΠΕΡΙΓΡΑΦΗ
    Αλλαγή του τρέχοντος καταλόγου εργασίας του κελύφους.

    Αλλαγή του τρέχοντος καταλόγου στο DIR.  Το προεπιλεγμένο DIR είναι η τιμή της
μεταβλητής HOME του κελύφους.

    Η μεταβλητή CDPATH ορίζει τη διαδρομή αναζήτησης για τον κατάλογο που περιέχει
το DIR.  Τα εναλλακτικά ονόματα καταλόγου στο CDPATH χωρίζονται από άνω κα κάτω
τελεία (:).

    Ένα null όνομα φακέλλου είναι το ίδιο με εκείνο του τρέχοντος καταλόγου.  Αν
το DIR αρχίζει με μία κάθετο (/), τότε δεν χρησιμοποιείται το CDPATH.

    Αν ο κατάλογος δεν βρεθεί και ορισθεί η παράμετρος κελύφους `cdable_vars',
τότε η λέξη εκλαμβάνεται ως το όνομα μιας μεταβλητής.  Αν αυτή η μεταβλητή έχει μια
τιμή, τότε, η τιμή της χρησιμοποιείται για το DIR.
```

```
    Παράμετροι (ορίσματα):
        -L      εξαναγκασμός παρακολούθησης των συμβολικών δεσμών
        -P      χρήση της δομής του φυσικού καταλόγου χωρίς την παρακολούθηση των
συμβολικών δεσμών

    Η προεπιλογή είναι η παρακολούθηση των συμβολικών δεσμών, σαν να είχε ήδη
ορισθεί το `-L'.

    Κατάσταση Εξόδου:
    Επιστρέφει ένα 0 αν αλλάξει ο κατάλογος. Διαφορετικά, μη-μηδέν.

ΔΕΙΤΕ ΕΠΙΣΗΣ
    bash(1)

ΥΛΟΠΟΙΗΣΗ
    GNU bash, έκδοση 4.1.5(1)-release (i486-pc-linux-gnu)
    Copyright (C) 2009 Ίδρυμα Ελεύθερου Λογισμικού (FSF Inc.)
```

Μια σημείωση για το notation: Όταν στην περιγραφή της σύνταξης μιας εντολής εμφανίζονται τετράγωνες παρενθέσεις, υποδεικνύουν προαιρετικά ορίσματα. Μια κατακόρυφη κάθετος υποδεικνύει αμοιβαία αποκλειόμενα θέματα. Στην περίπτωση της παραπάνω εντολής `cd`:

```
cd [-L|-P] [dir]
```

Αυτή η ονοματολογία (notation) λέει ότι η εντολή `cd` μπορεί να ακολουθείται προαιρετικά, είτε από ένα "-L" ή από ένα "-P" και, περαιτέρω, να ακολουθείται, προαιρετικά, από το όρισμα "dir".

--help

Πολλά εκτελέσιμα προγράμματα υποστηρίζουν την παράμετρο "--help" που εμφανίζει μια περιγραφή της σύνταξης που υποστηρίζεται απο την εντολή και τα ορίσματα. Για παράδειγμα:

```
[me@linuxbox me]$ mkdir --help
Usage: mkdir [OPTION] DIRECTORY...
Δημιουργία του DIRECTORY(ies), αν δεν υπάρχουν ήδη.

   -Z, --context=CONTEXT (SELinux) καθορισμός του πλαισίου ασφάλειας στο CONTEXT
Τα υποχρεωτικά ορίσματα για τις μεγάλες παραμέτρους είναι υποχρεωτικά και για τις
μικρές παραμέτρους.
   -m, --mode=MODE   καθορισμός τρόπου του αρχείου (όπως στο chmod), όχι a=rwx –
               umask
   -p, --parents     no error if existing, δημιουργία γονεϊκών καταλόγων,
                     αναλόγως των αναγκών
   -v, --verbose     εκτύπωση ενός μηνύματος για κάθε κατάλογο που δημιουργήθηκε
   --help            εμφάνιση αυτής της βοήθειας και έξοδος
   --version         Πληροφορίες για την έκδοση εξόδου και έξοδος
```

Μερικά προγράμματα δεν υποστηρίζουν την παράμετρο "--help", αλλά δοκιμάστε την σε κάθε περίπτωση. Συχνά, καταλήγει σε ένα μήνυμα σφάλματος, που θα αποκαλύψει παρόμοιες πληροφορίες χρήσης.

man

Τα περισσότερα εκτελέσιμα προγράμματα που προορίζονται για χρήση από την γραμμή εντολών, προσφέρουν μια επίσημη τεκμηρίωση, που λέγεται *manual* ή *man page*. Για να την δούμε, χρησιμοποιούμε ένα ειδικό πρόγραμμα εμφάνισης αυτών των σελίδων, που λέγεται `man`. Χρησιμοποιείται ως εξής:

```
man program
```

όπου “program” είναι το όνομα της εντολής που πρέπει να εμφανισθεί. Οι σελίδες man ποικίλουν σε κάποιο βαθμό ως προς τη μορφή τους, αλλά, σε γενικές γραμμές, περιέχουν έναν τίτλο, μια σύνοψη της σύνταξης της εντολής, μια περιγραφή του σκοπού της εντολής, καθώς και μια λίστα και μια περιγραφή της κάθε μιας από τις παραμέτρους της εντολής. Πάντως , οι σελίδες man δεν περιλαμβάνουν συνήθως παραδείγματα, μιας και προορίζονται για χρήση ως κείμενο αναφοράς και όχι σαν ένα είδος οδηγού βοήθειας. Σαν παράδειγμα, ας προσπαθήσουμε να εμφανίσουμε την σελίδα man για την ετολή `ls`:

```
[me@linuxbox me]$ man ls
```

Στα περισσότερα συστήματα GNU/ Linux, το `man` χρησιμοποιεί το `less` για την εμφάνιση της σελίδας man, έτσι ώστε όλες οι συνηθισμένες εντολές `less` λειτουργούν ενώ εμφανίζουν την σελίδα.

Το README και άλλα αρχεία τεκμηρίωσης

Πολλά πακέτα λογισμικού που είναι εγκατεστημένα στο σύστημά σας έχουν αρχεία τεκμηρίωσης που βρίσκονται στον κατάλογο /usr/share/doc. Τα περισσότερα από αυτά είναι αποθηκευμένα σε μορφή απλού κειμένου και μπορούν να εμφανισθούν με το `less`. Μερικά από αυτά τα αρχεία βρίσκονται σε μορφή HTML και μπορούν να εμφανισθούν μέσα στον φυλλομετρητή σας. Μπορεί να συναντήσετε κάποια αρχεία που έχουν μια κατάληξη σε “.gz”. Αυτό υποδεικνύει ότι έχουν συμπιεσθεί με το πρόγραμμα συμπίεσης `gzip`. Το πακέτο gzip περιλαμβάνει μια ειδική έκδοση του `less` που λέγεται `zless`, η οποία θα εμφανίσει τα περιεχόμενα των αρχείων κειμένου που είναι συμπιεσμένα με το gzip.

7

Ανακατεύθυνση Εισόδου/Εξόδου

Σε αυτό το μάθημα, θα διερευνήσουμε ένα ισχυρό χαρακτηριστικό, που χρησιμοποιείται σε πολλά προγράμματα της γραμμής εντολών, και ονομάζεται *ανακατεύθυνση εισόδου/εξόδου*. Όπως είδαμε, πολλές εντολές όπως το `ls`, εμφανίζουν την έξοδό τους στην οθόνη. Αυτό, πάντως, δεν είναι απαραίτητο. Με τη χρήση κάποιας ειδικής σήμανσης (notation) μπορούμε να *ανακατευθύνουμε* την έξοδο πολλών εντολών προς αρχεία, συσκευές, ακόμη και προς την είσοδο άλλων εντολών.

Κανονική Έξοδος (Standard Output)

Τα περισσότερα προγράμματα της γραμμής εντολών που εμφανίζουν τα αποτελέσματά τους, το πετυχαίνουν αποστέλλοντάς τα προς μια οντότητα που ονομάζεται *Κανονική Έξοδος (standard output)*. Από προεπιλογή, η κανονική έξοδος κατευθύνει τα αποτελέσματά της προς την οθόνη. Για να ανακατευθυνθεί η κανονική έξοδος προς ένα αρχείο, ο χαρακτήρας ">" χρησιμοποιείται ως εξής:

```
[me@linuxbox me]$ ls > file_list.txt
```

Σε αυτό το παράδειγμα, η εντολή `ls` εκτελείται και τα αποτελέσματα γράφονται σε ένα αρχείο με το όνομα file_list.txt. Αφού η έξοδος του `ls` ανακατευθύνθηκε προς το αρχείο, δεν θα εμφανισθεί στην οθόνη.

Κάθε φορά που επαναλαμβάνεται η εντολή, το file_list.txt αντικαθίσταται (από την αρχή) με την έξοδο της εντολής `ls`. Αν, αντιθέτως, θέλετε τα νέα αποτελέσματα να επισυνάπτονται *(appended)* στο αρχείο, τότε χρησιμοποιείστε το ">>" ως εξής:

```
[me@linuxbox me]$ ls >> file_list.txt
```

Όταν τα αποτελέσματα επισυναφθούν (appended), τα νέα αποτελέσματα προστίθενται στο τέλος του αρχείου, κάνοντας έτσι το αρχείο μακρύτερο, κάθε φορά που επαναλαμβάνεται η εντολή. Αν το αρχείο δεν υπάρχει όταν προσπαθείτε να το επισυνάψετε (append) στην ανακατευθυνόμενη έξοδο, τότε το αρχείο θα δημιουργηθεί.

Κανονική Είσοδος (Standard Input)

Πολλές εντολές μπορούν να δεχθούν μία είσοδο (input) από ένα εργαλείο που λέγεται *κανονική είσοδος (standard input)*. Από προεπιλογή, η κανονική είσοδος προσλαμβάνει τα περιεχόμενά της από το πληκτρολόγιο, αλλά, σαν κανονική είσοδος, μπορεί να ανακατευθυνθεί. Για την ανακατεύθυνση μιας κανονικής εισόδου από ένα αρχείο, αντί για το πληκτρολόγιο, ο χαρακτήρας "<" χρησιμοποιείται ως εξής:

```
[me@linuxbox me]$ sort < file_list.txt
```

Στο παραπάνω παράδειγμα, χρησιμοποιήσαμε την εντολή `sort` για να γίνει επεξεργασία του περιεχομένου του αρχείου file_list.txt. Τα αποτελέσματα είναι μια έξοδος επι της οθόνης, αφού η κανονική έξοδος δεν έχει ανακατευθυνθεί, σε αυτό το παράδειγμα. Θα μπορούσαμε να ανακατευθύνουμε την κανονική έξοδο σε ένα άλλο αρχείο, με τον εξής τρόπο:

```
[me@linuxbox me]$ sort < file_list.txt > sorted_file_list.txt
```

Όπως μπορείτε να δείτε, τόσο η είσοδος, όσο και η έξοδος μια εντολής, μπορούν κάλλιστα να ανακατευθυνθούν. Σημειώστε ότι η φορά της ανακατεύθυνσης δεν έχει σημασία. Η μόνη απαίτηση είναι ότι οι τελεστές της ανακατεύθυνσης, το "<" και το ">", θα πρέπει να εμφανίζονται μετά από τις άλλες μεταβλητές και τα αρθρώματα της εντολής.

Σωληνώσεις (Pipes)

Με διαφορά, το πιο χρήσιμο και δυνατό πράγμα που μπορείτε να κάνετε με την ανακατεύθυνση Εισόδου/ Εξόδου (I/O redirection), είναι να συνδέσετε πολλές εντολές μαζί με κάτι που ονομάζεται *σωληνώσεις (pipes)*. Με τις σωληνώσεις, η κανονική έξοδος μιας εντολής, διοχετεύεται σαν κανονική είσοδος μιας άλλης εντολής. Ακολουθεί το αγαπημένο μου παράδειγμα:

```
[me@linuxbox me]$ ls -l | less
```

Σε αυτό το παράδειγμα, η έξοδος της εντολής `ls` διοχετεύεται στο `less`. Χρησιμοποιώντας αυτό το κόλπο με το `"| less"`, μπορείτε να κάνετε οποιαδήποτε εντολή να έχει μια μετακυλιόμενη έξοδο (scrolling output). Εγώ χρησιμοποιώ συνεχώς αυτή την τεχνική.

Συνδέοντας πολλές εντολές μαζί, μπορείτε να καταφέρετε καταπληκτικά πράγματα. Παρακάτω θα βρείτε μερικά παραδείγματα που μπορεί να θελήσετε να δοκιμάσετε:

Παραδείγματα εντολών που χρησιμοποιούνται μαζί, με σωληνώσεις

Εντολή	***Τι ακριβώς κάνει***
`ls -lt | head`terminal	Εμφανίζει τα 10 πιο πρόσφατα αρχεία στον τρέχοντα κατάλογο εργασίας.
`du | sort -nr`	Εμφανίζει ένα κατάλογο φακέλλων και πόσο χώρο καταλαμβάνουν, κατεταγμένα από το μεγαλύτερο προς το μικρότερο.
`find . -type f -print | wc -l`	Εμφανίζει τον συνολικό αριθμό αρχείων στον τρέχοντα κατάλογο εργασίας και όλους τους υποκαταλόγους του.

Φίλτρα

Μια ομάδα προγραμμάτων που μπορείτε να χρησιμοποιείτε με τις σωληνώσεις, ονομάζονται *φίλτρα*. Τα φίλτρα αυτά παίρνουν την κανονική είσοδο και εκτελούν επάνω της μια λειτουργία και, κατόπιν, στέλνουν τα αποτελέσματα στην κανονική έξοδο. Με αυτό τον τρόπο, μπορούν να χρησιμοποιηθούν για την επεξεργασία πληροφορίας με πολύ δυνατούς τρόπους. Ακολουθούν μερικά από τα πιο συνηθισμένα προγράμματα που μπορούν να λειτουργήσουν σαν φίλτρα:

Συνηθισμένες εντολές φίλτρου

Πρόγραμμα	***Τι ακριβώς κάνει***
`sort`	Κατατάσσει την κανονική είσοδο και, μετά, βγάζει το κατεταγμένο αποτέλεσμα προς την κανονική έξοδο.
`uniq`	Σε μια δεδομένη ροή κατεταγμένων δεδομένων από μια κανονική είσοδο, διαγράφει τις διπλοκαταχωρημένες γραμμές δεδομένων (δηλ, εξασφαλίζει ότι η

	κάθε γραμή είναι μοναδική).
grep	Εξετάζει την κάθε γραμμή δεδομένων που λαμβάνει από την κανονική είσοδο και βγάζει στην έξοδο κάθε γραμμή που έχει ένα καθορισμένο πρότυπο χαρακτήρων.
fmt	Διαβάζει κείμενο από μια κανονική είσοδο και, μετά, βγάζει το μορφοποιημένο κείμενο στην κανονική έξοδο.
pr	Παίρνει input κειμένου από μια κανονική είσοδο και σπάει τα δεδομένα με διαχωριστικά σελίδας, επικεφαλίδες και υποσέλιδα, προετοιμάζοντας την εκτύπωση.
head	Βγάζει τις πρώτες λίγες γραμμές της εισόδου του. Χρήσιμο για τη λήψη της επικεφαλίδας ενός αρχείου.
tail	Βγάζει τις τελυταίες λίγες γραμμές της εισόδου του. Χρήσιμο πχ. για τη λήψη των πιο πρόσφατων καταχωρήσεων ενός αρχείου καταγραφής.
tr	Μεταφράζει χαρακτήρες. Μπορεί να χρησιμοποιηθεί για την εκτέλεση εργασιών όπως τις μετατροπές κεφαλαίων/ μικρών, ή την αλλαγή των χαρακτήρων τερματισμού γραμμής, από τον έναν τύπο στον άλλο (π.χ., μετατροπή αρχείων κειμένου DOS σε αρχεία κειμένου στυλ Unix).
sed	Πρόγραμμα επεξεργασίας ροής (stream editor). Μπορεί να κάνει πιο εξεζητημένες μεταφράσεις κειμένου σε σύγκριση με το `tr`.
awk	Πρόκειται για μια ολόκληρη προγραμματιστική γλώσσα που σχεδιάσθηκε για τη δημιουργία φίλτρων. Είναι εξαιρετικά ισχυρή.

Εκτέλεση εργασιών με τις σωληνώσεις (pipes)

1. *Εκτύπωση από τη γραμμή εντολών.* Το GNU/Linux παρέχει ένα πρόγραμμα που λέγεται `lpr` το οποίο δέχεται κανονική είσοδο και την στέλνει στον εκτυπωτή. Χρησιμοποιείται συχνά με τις σωληνώσεις (pipes) και τα φίλτρα. Ακολουθούν δύο παραδείγματα:

   ```
   cat poorly_formatted_report.txt | fmt | pr | lpr

   cat unsorted_list_with_dupes.txt | sort | uniq | pr | lpr
   ```

   Στο πρώτο παράδειγμα, χρησιμοποιούμε το πρόγραμμα `cat` για να διαβάσουμε το αρχείο και να

το βγάλουμε σε κανονική έξοδο, η οποία στη συνέχεια, θα σωληνωθεί σαν κανονική είσοδος για το `fmt`. Το `fmt` μορφοποιεί το κείμενο σε τακτοποιημένες παραγράφους και το βγάζει σαν κανονική έξοδο, η οποία σωληνώνεται στην κανονική είσοδο του `pr`. Το `pr` σπάει όμορφα το κείμενο σε σελίδες και το βγάζει σε κανονική έξοδο, που με τη σειρά της, σωληνώνεται στην κανονική είσοδο του `lpr`. Το `lpr` παίρνει την κανονική του είσοδο και τη στέλνει στον εκτυπωτή.

Το δεύτερο παράδειγμα αρχίζει με μια μη κατεταγμένη λίστα δεδομένων που περιέχει διπλοκαταχωρήσεις. Κατ' αρχάς, το `cat` στέλνει τη λίστα στο `sort`, το οποίο την κατατάσσει και μετά την διοχετεύει στο `uniq` που, με τη σειρά του, διαγράφει τα διπλοαντίγραφα. Κατόπιν, χρησιμοποιούνται το `pr` και το `lpr` για τη σελιδοποίηση και την εκτύπωση της λίστας.

2. *Ανάγνωση περιεχομένων των αρχείων tar:* Συχνά θα βρείτε λογισμικό που διανέμεται ως ένα αρχείο *gzipped tar*. Πρόκειται για ένα παραδοσιακό στυλ αρχείου Unix (tape archive file), που δημιουργήθηκε με το πρόγραμμα `tar` και συμπιέσθηκε με το `gzip`. Μπορείτε να αναγνωρίσετε εύκολα αυτά τα αρχεία από τις παραδοσιακές τους καταλήξεις, π.χ. ".tar.gz" ή ".tgz". Μπορείτε να χρησιμοποιείτε την παρακάτω εντολή για να δείτε τον κατάλογο ενός τέτοιου αρχείου σε ένα σύστημα GNU/Linux:

   ```
   tar tzvf name_of_file.tar.gz | less
   ```

8

Επέκταση (Expansion)

Κάθε φορά που εσείς πληκτρολογείτε μια γραμμή εντολής και πατάτε το πλήκτρο enter, το bash εκτελεί διαφορες διεργασίες στο κείμενο, πριν εφαρμόσει την εντολή σας. Εξετάσαμε δύο περιπτώσεις όπου μια απλή σειρά χαρακτήρων, π.χ. “*”, μπορεί να σημαίνει πολλά πράγματα για το κέλυφος. Η διεργασία που καθιστά εφικτή την εφαρμογή αυτών των πραγμάτων, ονομάζεται *επέκταση (expansion)*. Με την επέκταση, ενώ εσείς πληκτρολογείτε κάτι, αυτό επεκτείνεται και επεκτείνεται σε κάτι άλλο, πριν το κέλυφος αναλάβει δράση επάνω του. Για να δείξουμε τι ακριβώς εννοούμε, ας εξετάσουμε λίγο την εντολή `echo`. Η εντολή `echo` είναι μια εσωτερική εντολή κελύφους, που εκτελεί μια πολύ απλή ενέργεια. Εκτυπώνει τα ορίσματα κειμένου (text arguments) σε μια κανονική έξοδο (standard output):

```
[me@linuxbox me]$ echo this is a test
this is a test
```

Αυτό είναι αρκετά κατανοητό. Οποιοδήποτε όρισμα δοθεί στο `echo` θα εμφανισθεί. Ας δοκιμάσουμε ένα διαφορετικό παράδειγμα:

```
[me@linuxbox me]$ echo *
Desktop Documents ls-output.txt Music Pictures Public Templates
Videos
```

Τι συνέβη λοιπόν; Γιατί το `echo` δεν εκτύπωσε το “*”; Όπως θα θυμάσθε από το μάθημά μας για τους χαρακτήρες μπαλαντέρ, ο χαρακτήρας “*” σημαίνει το ταίριασμα οποιωνδήποτε χαρακτήρων μέσα σε ένα όνομα αρχείου, αλλά αυτό που δεν είδαμε στην αρχική μας συζήτηση ήταν ο τρόπος με τον οποίο το κέλυφος το καταφέρνει. Η απλή απάντηση είναι ότι το κέλυφος επεκτείνει το “*” σε κάτι άλλο (σε αυτή την περίπτωση, τα ονόματα των αρχείων στον τρέχοντα κατάλογο εργασίας) πριν την εκτέλεση της εντολής `echo`. Όταν πατηθεί το πλήκτρο enter, το κέλυφος επεκτείνει αυτομάτως οποιουσδήποτε ταιριαστούς χαρακτήρες στη γραμμή εντολών, πριν την εκτέλεση της εντολής, οπότε η εντολή `echo` δεν είδε ποτέ το “*”, παρά μόνον το αποτέλεσμα της επέκτασής του. Γνωρίζοντάς το, μπορούμε να κατανοήσουμε ότι το `echo` συμπεριφέρθηκε όπως αναμενόταν.

Επέκταση του ονόματος διαδρομής

Ο μηχανισμός με τον οποίο λειτουργούν οι χαρακτήρες μπαλαντέρ (wildcards), ονομάζεται *επέκταση ονόματος διαδρομής (pathname expansion)*. Αν δοκιμάσουμε κάποιες από τις τεχνικές που είχαμε εφαρμόσει στα προηγούμενα μαθήματά μας, θα δούμε ότι πρόκειται πραγματικά για επεκτάσεις (expansions). Δοθέντος ενός συγκεκριμμένου αρχικού καταλόγου (home directory), αυτό μοιάζει ως εξής:

```
[me@linuxbox me]$ ls
Desktop
ls-output.txt
Documents Music
Pictures
Public
Templates
Videos
```

θα κάναμε τις εξής expansions:

```
[me@linuxbox me]$ echo D*
Desktop Documents
```

και:

```
[me@linuxbox me]$ echo *s
Documents Pictures Templates Videos
```

ή, ακόμη και:

```
[me@linuxbox me]$ echo [[:upper:]]*
Desktop Documents Music Pictures Public Templates Videos
```

και κοιτώντας πέρα από τον αρχικό μας κατάλογο:

```
[me@linuxbox me]$ echo /usr/*/share
/usr/kerberos/share /usr/local/share
```

Επέκταση συμβόλου (Tilde Expansion)

Όπως μάλλον θα θυμάσθε από την εισαγωγή μας στην εντολή `cd`, ο χαρακτήρας με το σύμβολο (“~”) έχει μια ειδική σημασία. Όταν χρησιμοποιείται στην αρχή μιας λέξης, επεκτείνεται στο όνομα του αρχικού καταλόγου του κατονομαζόμενου χρήστη, ή αν δεν υπάρχει κανένας κατονομασμένος χρήστης, στον αρχικό κατάλογο του τρέχοντος χρήστη:

```
[me@linuxbox me]$ echo ~
/home/me
```

Αν ο χρήστης “foo” έχει έναν λογαριασμό, τότε:

```
[me@linuxbox me]$ echo ~foo
/home/foo
```

Αριθμητική επέκταση (Arithmetic Expansion)

Το κέλυφος επιτρέπει την διεξαγωγή αριθμητικών πράξεων μέσω της επέκτασης. Αυτό μας δίνει την δυνατότητα να χρησιμοποιούμε την προτροπή του κελύφους σαν μια αριθμομηχανή:

```
[me@linuxbox me]$ echo $((2 + 2))
4
```

Η αριθμητική επέκταση χρησιμοποιεί την φόρμα:

```
$((expression))
```

όπου η λέξη expression είναι μια αριθμητική έκφραση που αποτελείται από τιμές και αριθμητικούς τελεστές.

Η αριθμητική επέκταση υποστηρίζει μόνον ακέραιους αριθμούς (ολόκληρους αριθμούς, χωρίς δεκαδικούς), αλλά μπορεί να εκτελέσει αρκετές διαφορετικές λειτουργίες.

Τα κενά διαστήματα δεν έχουν σημασία στις αριθμητικές εκφράσεις και οι εκφράσεις μπορεί να είναι φωλιασμένες. Για παράδειγμα, για να πολλαπλασιάσουμε το 5 εις το τετράγωνο με το 3:

```
[me@linuxbox me]$ echo $(($((5**2)) * 3))
75
```

Οι μονές παρενθέσεις μπορούν να χρησιμοποιηθούν για την ομαδοποίηση πολλαπλών υποεκφράσεων. Με αυτή την τεχνική, μπορούμε να ξαναγράψουμε το παραπάνω παράδειγμα και να έχουμε το ίδιο αποτέλεσμα, χρησιμοποιώντας μόνον μια επέκταση, αντί για δύο:

```
[me@linuxbox me]$ echo $(((5**2) * 3))
75
```

Ακολουθεί ένα παράδειγμα με την χρήση της διαίρεσης και των υπόλοιπων τελεστών. Σημειώστε το αποτέλεσμα της διαίρεσης ακέραιων αριθμών:

```
[me@linuxbox me]$ echo Five divided by two equals $((5/2))
Five divided by two equals 2
[me@linuxbox me]$ echo with $((5%2)) left over.
with 1 left over.
```

Επέκταση αγκύλης (Brace Expansion)

Ίσως, η πιο παράξενη επέκταση λέγεται *επέκταση αγκύλης (brace)*. Με αυτή, μπορείτε να δημιουργήσετε πολλαπλές συμβολοσειρές, μέσα σε ένα πρότυπο που περιέχει αγκύλες. Ακολουθεί ένα παράδειγμα:

```
[me@linuxbox me]$ echo Front-{A,B,C}-Back
Front-A-Back Front-B-Back Front-C-Back
```

Τα πρότυπα που προορίζονται για επέκταση αγκύλης μπορεί να περιέχουν μια αρχική ενότητα, που λέγεται *πρόλογος,* και μια τελική ενότητα που λέγεται *υστερόγραφο (postscript)*. Η έκφραση σε αγκύλες, αυτή καθ' αυτή, μπορεί να περιέχει είτε μια λίστα από συμβολοσειρές διαχωρισμένες με κόμμα, ή ένα φάσμα ακέραιων αριθμών, ή μεμονωμένων χαρακτήρων. Το πρότυπο δεν μπορεί να περιέχει ενσωματωμένα κενά διαστήματα. Ακολουθεί ένα παράδειγμα με τη χρήση ενός φάσματος ακέραιων αριθμών:

```
[me@linuxbox me]$ echo Number_{1..5}
Number_1 Number_2 Number_3 Number_4 Number_5
```

΄Ενα φάσμα αριθμών σε αντίστροφη σειρά:

```
[me@linuxbox me]$ echo {Z..A}
Z Y X W V U T S R Q P O N M L K J I H G F E D C B A
```

Οι εκφράσεις σε αγκύλες μπορούν να είναι φωλιασμένες:

```
[me@linuxbox me]$ echo a{A{1,2},B{3,4}}b
```

```
aA1b aA2b aB3b aB4b
```

Σε τι χρησιμεύουν; Η πιο συνηθισμένη εφαρμογή είναι η δημιουργία λιστών με αρχεία ή καταλόγους. Για παράδειγμα, αν ήσασταν φωτογράφος και είχατε μια μεγάλη συλλογή εικόνων που θέλετε να οργανώσετε κατά έτος και ανά μήνα, το πρώτο πράγμα που θα μπορούσατε να κάνετε είναι να δημιουργήσετε μια σειρά καταλόγων, αριθμημένων σε αριθμητική μορφή "Έτος-Μήνας". Με αυτό τον τρόπο, τα ονόματα καταλόγου θα καταταχθούν σε χρονολογική σειρά. Θα μπορούσατε να πληκτρολογήσετε μία πλήρη λίστα καταλόγων, αλλά αυτό συνεπάγεται πολλή δουλειά και προσφέρεται, επίσης, για σφάλματα. Αντιθέτως, θα μπορούσατε να κάνετε το εξής:

```
[me@linuxbox me]$ mkdir Photos
[me@linuxbox me]$ cd Photos
[me@linuxbox Photos]$ mkdir {2007..2009}-0{1..9} {2007..2009}-{10..12}
[me@linuxbox Photos]$ ls
2007-01 2007-07 2008-01 2008-07 2009-01 2009-07
2007-02 2007-08 2008-02 2008-08 2009-02 2009-08
2007-03 2007-09 2008-03 2008-09 2009-03 2009-09
2007-04 2007-10 2008-04 2008-10 2009-04 2009-10
2007-05 2007-11 2008-05 2008-11 2009-05 2009-11
2007-06 2007-12 2008-06 2008-12 2009-06 2009-12
```

Πολύ κομψό!

Επέκταση παραμέτρων (Parameter Expansion)

Θα αναφερθούμε μόνο με συντομία στην *επέκταση παραμέτρων* σε αυτό το μάθημα, αλλά θα το αναλύσουμε σε περισσότερη έκταση αργότερα. Πρόκειται για ένα χαρακτηριστικό που είναι πιο χρήσιμο σε σενάρια κελύφους (shell scripts) παρά άμεσα στη γραμμή εντολών. Πολλές από τις δυνατότητές της έχουν σχέση με την ικανότητα του συστήματος για αποθήκευση μικρών τμημάτων δεδομένων και για απόδοση ενός ονόματος στο κάθε ένα τμήμα. Πολλά από αυτά τα τμήματα, που αποκαλούνται σωστότερα *μεταβλητές*, είναι στη διάθεσή σας προς εξέταση. Για παράδειγμα, η μεταβλητή που λέγεται "USER" περιέχει το δικό σας όνομα χρήστη. Για να καλέσετε την επέκταση παραμέτρου και να αποκαλύψετε τα περιεχόμενα του USER θα κάνατε τα εξής:

```
[me@linuxbox me]$ echo $USER
me
```

Για να δείτε μια λίστα των διαθέσιμων μεταβλητών, προσπαθήστε το εξής:

```
[me@linuxbox me]$ printenv | less
```

Μπορεί να παρατηρήσατε ότι με άλλα είδη επεκτάσεων, αν πληκτρολογήσετε εσφαλμένα ένα πρότυπο, τότε η επέκταση δεν θα λάβει χώρα και η εντολή echo απλώς θα εμφανίσει το λάθος πληκτρολογημένο πρότυπο. Με την επέκταση παραμέτρων, αν κάνετε λάθος στην ορθογραφία του ονόματος της μεταβλητής, η επέκταση θα λάβει και πάλι χώρα, αλλά θα καταλήξει σε μια κενή συμβολοσειρά:

```
[me@linuxbox me]$ echo $SUER
[me@linuxbox ~]$
```

Αντικατάσταση εντολής (Command Substitution)

Η αντικατάσταση εντολής (command substitution) μας επιτρέπει να χρησιμοποιούμε την έξοδο μιας εντολής ως μια επέκταση:

```
[me@linuxbox me]$ echo $(ls)
Desktop Documents ls-output.txt Music Pictures Public Templates
Videos
```

Ένα από τα αγαπημένα μου είναι κάπως έτσι:

```
[me@linuxbox me]$ ls -l $(which cp)
-rwxr-xr-x 1 root root 71516 2007-12-05 08:58 /bin/cp
```

Εδώ περάσαμε τα αποτελέσματα των οποίων το `cp` ως ένα όρισμα της εντολής `ls`, και άρα, πήραμε τον κατάλογο του προγράμματος `cp` χωρίς να χρειάζεται να γνωρίζουμε το πλήρες όνομα της διαδρομής της. Δεν περιροριζόμαστε σε απλές εντολές. Μπορούμε να χρησιμοποιήσουμε ολόκληρες σωληνώσεις/ pipelines (εμφανίζεται μόνον η μερική έξοδος):

```
[me@linuxbox me]$ file $(ls /usr/bin/* | grep bin/zip)
/usr/bin/bunzip2:
/usr/bin/zip:      ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.15, stripped
/usr/bin/zipcloak: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.15, stripped
/usr/bin/zipgrep:  POSIX shell script text executable
/usr/bin/zipinfo:  ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.15, stripped
/usr/bin/zipnote:  ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.15, stripped
/usr/bin/zipsplit: ELF 32-bit LSB executable, Intel 80386, version 1
(SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.15, stripped
```

Σε αυτό το παράδειγμα, τα αποτελέσματα της σωλήνωσης (pipeline) έγιναν η λίστα αρθρωμάτων (argument list) του αρχείου της εντολής. Υπάρχει και μια εναλλακτική σύνταξη για την αντικατάσταση εντολής, σε παλιότερα προγράμματα κελύφους, που υποστηρίζεται και από το `bash`. Χρησιμοποιούν ανάστροφα εισαγωγικά (back-quotes), αντί για το σήμα του δολλαρίου και τις παρενθέσεις:

```
[me@linuxbox me]$ ls -l `which cp`
-rwxr-xr-x 1 root root 71516 2007-12-05 08:58 /bin/cp
```

Χρήση εισαγωγικών (Quoting)

Τώρα που είδαμε με πόσους τρόπους το κέλυφος μπορεί να εκτελεί επεκτάσεις, είναι ώρα να μάθουμε πως μπορούμε να το ελέγχουμε. Πάρτε για παράδειγμα:

```
[me@linuxbox me]$ echo this is a     test
this is a test
```

ή:

```
[me@linuxbox me]$ [me@linuxbox ~]$ echo The total is $100.00
The total is 00.00
```

Στο πρώτο παράδειγμα, ο διαχωρισμός λέξης εκ μέρους του κελύφους, απομάκρυνε τις έξτρα κενά διαστήματα από την λίστα ορισμάτων της εντολής echo. Στο δεύτερο παράδειγμα, η επέκταση παραμέτρου αντικατέστησε ένα κενό string με την τιμή του "$1" διότι επρόκειτο για μια μη-προσδιορισμένη μεταβλητή. Το κέλυφος προσφέρει έναν μηχανισμό που λέγεται quoting, για την επιλεκτική καταστολή των ανεπιθύμητων επεκτάσεων.

Διπλά εισαγωγικά (Double Quotes)

Το πρώτο είδος χρήσης εισαγωγικών (quoting) το οποίο θα εξετάσουμε, είναι τα διπλά εισαγωγικά. Αν τοποθετήσετε ένα κείμενο μέσα σε διπλά εισαγωγικά, όλοι οι ειδικοί χαρακτήρες που χρησιμοποιούνται από το κέλυφος, θα χάσουν την ιδιαίτερη σημασία τους και θα εκλαμβάνονται ως κανονικοί χαρακτήρες. Οι εξαιρέσεις είναι "$", "\" (backslash), και το "`" (back- quote). Αυτό σημαίνει ότι ο διαχωρισμός λέξεων, η επέκταση παραμέτρων, η επέκταση συμβόλων και η επέκταση αγκύλης,

καταστέλλονται, ενώ η επέκταση παραμέτρου, η αριθμητική επέκταση και η αντικατάσταση εντολής, συνεχίζουν να διεξάγονται. Με τη χρήση διπλών εισαγωγικών, μπορούμε να διεκπεραιώνουμε ονόματα αρχείων που περιέχουν ενσωματωμένα διαστήματα. Ας υποθέσουμε ότι ήσασταν το ατυχές θύμα ενός αρχείου με το όνομα two words.txt. Αν δοκιμάζατε να το χρησιμοποιήσετε στη γραμμή εντολών, ο διαχωρισμός λέξεων θα οδηγούσε σε μια διαχείριση σαν να επρόκειτο για δύο ξεχωριστά ορίσματα, παρά σαν ένα επιθυμητό ενιαίο όρισμα:

```
[me@linuxbox me]$ ls -l two words.txt
ls: cannot access two: No such file or directory
ls: cannot access words.txt: No such file or directory
```

Με τη χρήση των διπλών εισαγωγικών, μπορείτε να σταματήσετε τον διαχωρισμός λέξεων και να έχετε τα επιθυμητά αποτελέσματα. Επιπλέον, μπορείτε ακόμη και να επισκευάσετε τη βλάβη:

```
[me@linuxbox me]$ ls -l "two words.txt"
-rw-rw-r-- 1 me me 18 2008-02-20 13:03 two words.txt
[me@linuxbox me]$ mv "two words.txt" two_words.txt
```

Ορίστε! Τώρα, δεν χρειάζεται να συνεχίζετε να πληκτρολογείτε αυτά τα ενοχλητικά διπλά εισαγωγικά. Να θυμάστε ότι η επέκταση παραμέτρου, η αριθμητική επέκταση και η αντικατάσταση εντολών, συνεχίζουν να διεξάγονται μέσα στα διπλά εισαγωγικά:

```
[me@linuxbox me]$ echo "$USER $((2+2)) $(cal)"
me 4
February 2008
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
```

Θα πρέπει να αφιερώσουμε λίγο χρόνο για να εξετάσουμε το αποτέλεσμα των διπλών εισαγωγικών στην αντικατάσταση εντολών. Κατ' αρχήν, ας κοιτάξουμε λίγο βαθύτερα τον τρόπο λειτουργίας του διαχωρισμού λέξεων. Στο προηγούμενο παράδειγμά μας, είδαμε πως ο διαχωρισμός λέξεων μοιάζει να διαγράφει τα έξτρα κενά διαστήματα στο κείμενό μας:

```
[me@linuxbox me]$ echo this is a     test
this is a test
```

Από προεπιλογή, ο διαχωρισμός λέξεων κοιτά για την παρουσία κενών διαστημάτων, tabs, και νέων γραμμών (χαρακτήρες linefeed) και τα μεταχειρίζεται σαν διαχωριστικά μεταξύ λέξεων. Αυτό σημαίνει ότι τα διαστήματα χωρίς εισαγωγικά, tabs, και νέες γραμμές, δεν θεωρούνται ως τμήμα του κειμένου. Χρησιμεύουν μόνον ως διαχωριστικά. Αφού διαχωρίζουν τις λέξεις σε ξεχωριστά ορίσματα, η γραμμή εντολών του παραδείγματός μας περιέχει μια εντολή που ακολουθείται από τέσσερα ανεξάρτητα ορίσματα. Αν προσθέσουμε διπλά εισαγωγικά:

```
[me@linuxbox me]$ echo "this is a     test"
this is a     test
```

τότε ο διαχωρισμός λέξεων καταστέλλεται και τα ενσωματωμένα διαστήματα δεν συμπεριφέρονται ως διαχωριστικά, αλλά μάλλον γίνονται τμήμα του ορίσματος. Όταν τα διπλά εισαγωγικά προστεθούν, η γραμμή εντολών μας περιέχει μια εντολή που ακολουθείται από ένα μεμονωμένο όρισμα. Το γεγονός ότι οι νέες γραμμές θεωρούνται ως διαχωριστικά από τον μηχανισμό διαχωρισμού λέξεων, δημιουργεί ένα ενδιαφέρον αλλά διακριτικό αποτέλεσμα στην αντικατάσταση εντολών. Λάβετε υπ' όψη τα εξής:

```
[me@linuxbox me]$ echo $(cal)
February 2008 Su Mo Tu We Th Fr Sa 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

```
15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
[me@linuxbox me]$ echo "$(cal)"
February 2008
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29
```

Σε πρώτο στάδιο, η εκτός εισαγωγικών αντικατάσταση εντολής, οδήγησε σε μια γραμμή εντολών που περιέχει τριανταοκτώ ορίσματα. Κατά δεύτερο λόγο, μια γραμμή εντολών με ένα όρισμα που συμπεριλαμβάνει τα ενσωματωμένα διαστήματα και τις νέες γραμμές.

Απλά εισαγωγικά

Αν πρέπει να καταστείλετε όλες τις επεκτάσεις, τότε χρησιμοποιήστε τα απλά εισαγωγικά. Ακολουθεί μια σύγκριση χωρίς εισαγωγικά, με διπλά εισαγωγικά και με απλά εισαγωγικά:

```
[me@linuxbox me]$ echo text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER
text /home/me/ls-output.txt a b foo 4 me
[me@linuxbox me]$ echo "text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER"
text ~/*.txt {a,b} foo 4 me
[me@linuxbox me]$ echo 'text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER'
text ~/*.txt {a,b} $(echo foo) $((2+2)) $USER
```

Όπως μπορείτε να δείτε, με το κάθε επόμενο επίπεδο εισαγωγικών, καταστέλλονται ολοένα και περισσότερες επεκτάσεις.

Χαρακτήρες διαφυγής (Escaping Characters)

Μερικές φορές θέλετε μόνο να παραθέσετε έναν μεμονωμένο χαρακτήρα. Για να το πετύχετε, μπορείτε να προτάξετε πριν τον χαρακτήρα, μια ανάποδη κάθετο (backslash), που σε αυτές τις περιστάσεις ονομάζεται *χαρακτήρας διαφυγής (escape character)*. Αυτό γίνεται, συχνά, μέσα σε διπλά εισαγωγικά, για να αποφευχθεί επιλεκτικά ένα expansion:

```
[me@linuxbox me]$ echo "The balance for user $USER is: \$5.00"
The balance for user me is: $5.00
```

Είναι επίσης συνηθισμένη η χρήση χαρακτήρων διαφυγής για την απαλοιφή της ειδικής σημασίας ενόας χαρακτήρος μέσα σε ένα όνομα αρχείου. Είναι δυνατή, για παράδειγμα, η χρήση χαρακτήρων σε ονόματα αρχείων που έχουν συνήθως ένα ειδικό νόημα για το κέλυφος. Αυτοί περιλαμβάνουν τα "$", "!", "&", " " και άλλα. Για να συμπεριλάβετε έναν ειδικό χαρακτήρα μέσα σε ένα όνομα αρχείου, μπορείτε να κάνετε ως εξής:

```
[me@linuxbox me]$ mv bad\&filename good_filename
```

Για να επιτρέψετε να εμφανισθεί μια ανάποδη κάθετος (backslash), κάντε διαφυγή πληκτρολογώντας "\\". Σημειώστε ότι μέσα στα απλά εισαγωγικά, η ανάποδη κάθετος (backslash) χάνει την ειδική της σημασία και έχει την μεταχείριση ενός συνηθισμένου χαρακτήρα.

Κι' άλλα κόλπα με την ανάποδη κάθετο (Backslash)

Αν δείτε τις σελίδες man οποιουδήποτε προγράμματος που έχει γραφεί στα πλαίσια του έργου GNU, θα παρατηρήσετε ότι, πέραν των παραμέτρων της γραμμής εντολών που συνίστανται σε μια παύλα και ένα

μεμονωμένο γράμμα, υπάρχουν επίσης μεγάλα ονόματα παραμέτρων που αρχίζουν με δύο παύλες. Για παράδειγμα, τα παρακάτω είναι ισοδύναμα:

```
ls -r
ls --reverse
```

Γιατί υποστηρίζονται και τα δύο; Η συντομευμένη μορφή προορίζεται για αυτούς που είναι τεμπέληδες στον χειρισμό του πληκτρολογίου στη γραμμή εντολών, ενώ η μακρά μορφή προορίζεται επί το πλείστον για σενάρια (scripts), αν και μερικά ορίσματα μπορεί να υφίστανται μόνον στην μακρά μορφή. Εγώ χρησιμοποιώ, ενίοτε, ορίσματα σκοτεινής φύσεως και βρίσκω την μακρά μορφή χρήσιμη όταν πρέπει να κάνω ανασκόπηση ενός σεναρίου (script) μερικούς μήνες μετά τη συγγραφή του. Όταν βλέπω την μακρά μορφή, αντιλαμβάνομαι τι ακριβώς κάνει, κι' έτσι γλυτώνω ένα ταξιδάκι ως την σελίδα `man`. Ας πληκτρολογήσουμε λίγο περισσότερο τώρα, για να καταφέρουμε να έχουμε λιγότερη δουλειά αργότερα. Συντηρείται η τεμπελιά.

Όπως θα μπορούσατε να φαντασθείτε, η χρήση ορισμάτων στην μακρά μορφή τους μπορεί να κάνει μία μεμονωμένη γραμμή εντολών πολύ μεγάλη. Για να καταπολεμήσετε αυτό το πρόβλημα, μπορείτε να χρησιμοποιήσετε μια ανάποδη κάθετο, για να υποχρεώσετε το κέλυφος να αγνοήσει έναν χαρακτήρα μιας νέας γραμμής σαν κι' αυτόν:

```
ls -l \
   --reverse \
   --human-readable \
   --full-time
```

Αυτή η χρήση της ανάποδης καθέτου μας επιτρέπει την ενσωμάτωση νέων γραμμών στη γραμμή εντολών. Σημειώστε ότι για να δουλέψει αυτό το κόλπο, η νέα γραμμή πρέπει να πληκτρολογηθεί αμέσως μετά την ανάποδη κάθετο. Αν παρεμβάλετε ένα κενό διάστημα μετά την ανάποδη κάθετο, το διάστημα αυτό θα αγνοηθεί, όχι όμως και η νέα γραμμή. Ο ανάποδες κάθετοι χρησιμοποιούνται, επίσης, για την εισαγωγή ειδικών χαρακτήρων στο κείμενό μας. Αυτές λέγονται *χαρακτήρες διαφυγής ανάποδης καθέτου*. Ακολουθούν οι συχνότεροι από αυτούς:

Χαρακτήρας διαφυγής	*Όνομα*	*Πιθανές χρήσεις*
\n	newline	Προσθήκη κενών γραμμών στο κείμενο
\t	tab	Εισαγωγή οριζόντιων tabs στο κείμενο
\a	alert	Κάνει το τερματικό σας να χτυπά beep
\\	backslash	Εισάγει μια ανάποδη κάθετο (backslash)
\f	formfeed	Αν το αποστείλετε στον εκτυπωτή σας, θα αποβάλλει τη σελίδα

Η χρήση χαρακτήρων διαφυγής με ανάποδη κάθετο είναι πολύ συχνή. Αυτή η ιδέα εμφανίσθηκε, αρχικά, στην προγραμματιστική γλώσσα C. Σήμερα, το κέλυφος, η C++, η perl, η python, η awk, η tcl και πολλές άλλες γλώσσες προγραμματισμού, χρησιμοποιούν αυτή την έννοια. Η χρήση της εντολής echo μαζί με το όρισμα -e θα μας επιτρέψει να αποδείξουμεε:

```
[me@linuxbox me]$ echo -e "Inserting several blank lines\n\n\n"
Inserting several blank lines



[me@linuxbox me]$ echo -e "Words\tseparated\tby\thorizontal\ttabs."
Λέξεις που χωρίζονται από οριζόντια  tabs

[me@linuxbox me]$ echo -e "\aMy computer went \"beep\"."

My computer went "beep".

[me@linuxbox me]$ echo -e "DEL C:\\WIN2K\\LEGACY_OS.EXE"

DEL C:\WIN2K\LEGACY_OS.EXE
```

9

Δικαιώματα

Το λειτουργικό σύστημα Unix (και, παρομοίως, το GNU/Linux) διαφέρει από άλλα υπολογιστικά περιβάλλοντα, ως προς το ότι δεν πρόκειται απλώς για ένα *σύστημα multitasking,* αλλά είναι, επίσης, και ένα σύστημα *πολλαπλών χρηστών (multi-user)*.

Τι ακριβώς σημαίνει αυτό; Σημαίνει ότι περισσότεροι του ενός χρήστες μπορούν να λειτουργούν, ταυτόχρονα, τον υπολογιστή. Παρ' όλο που ο υπολογιστής σας δεν διαθέτει παρά μόνον ένα πληκτρολόγιο και μια οθόνη, μπορεί εξ' ίσου καλά να χρησιμοποιηθεί από περισσότερους του ενός χρήστες. Αν, για παράδειγμα, ο υπολογιστής σας συνδεθεί σε ένα δίκτυο, ή στο Διαδίκτυο, τότε οι απομεμακρυσμένοι χρήστες μπορούν να συνδεθούν σε αυτόν, μέσω του `telnet` ή του `ssh` (secure shell/ ασφαλές κέλυφος) και να τον λειτουργούν. Πράγματι, οι απομεμακρυσμένοι αυτοί χρήστες μπορούν να εκτελούν X εφαρμογές και να έχουν το ίδιο γραφικό περιβάλλον που προβάλλεται και στον απομεμακρυσμένο υπολογιστή. Αυτό υποστηρίζεται από το σύστημα διαχείρισης παραθύρων X Windows.

Η ικανότητα εξυπηρέτησης πολλαπλών χρηστών του Unix δεν είναι κάποια πρόσφατη "καινοτομία". Πρόκειται, μάλλον, για ένα χαρακτηριστικό που είναι βαθιά ριζωμένο στο σχεδιασμό τον ίδιο του λειτουργικού αυτού συστήματος. Αν θυμηθείτε σε ποιο περιβάλλον αναπτύχθηκε το Unix, τότε θα δείτε ότι αυτό είναι πολύ λογικό. Πριν πολλά χρόνια, πριν ακόμη οι υπολογιστές γίνουν "προσωπικοί", ήταν τεράστια, δαπανηρά και κεντρικοποιημένα μηχανήματα. Ένας τυπικός υπολογιστής σε ένα Πανεπιστήμιο, συνίστατο σε ένα μεγάλο υπολογιστή τύπου mainframe, που εφιλοξενείτο σε κάποιο κτίριο του Πανεπιστημίου, ενώ τα *τερματικά* ήταν διάσπαρτα σε όλη την Πανεπιστημιούπολη (campus), και το καθένα από αυτά ήταν συνδεδεμένο σε ένα μεγάλο κεντρικό υπολογιστή. Συνεπώς, ένας υπολογιατής, έπρεπε να υποστηρίζει πολλούς χρήστες ταυτοχρόνως.

Για να γίνει αυτό πρακτικό, έπρεπε να βρεθεί μια μέθοδος που να προστατεύει τον χρήστη από όλους τους υπόλοιπους. Προφανώς, δεν θα μπορούσατε να επιτρέψετε στις πράξεις ενός μεμονωμένου χρήστη να οδηγήσουν σε κατάρρευση όλο τον υπολογιστή, ούτε και θα μπορούσατε να επιτρέψετε σε έναν χρήστη να έχει πρόσβαση στα αρχεία που ανήκαν σε κάποιον άλλον χρήστη.

Αυτό το μάθημα θα καλύψει τις παρακάτω εντολές:

- `chmod` – τροποποίηση των δικαιωμάτων πρόσβασης των αρχείων
- `su` – πως να γίνετε προσωρινά υπερχρήστης (superuser)
- `chown` – αλλαγή ιδιοκτησίας αρχείων
- `chgrp` – αλλαγή της ομάδας στην οποία ανήκει ένα αρχείο

Δικαιώματα αρχείου

Το GNU/Linux χρησιμοποιεί το ίδιο σχήμα δικαιωμάτων σαν το Unix. Σε κάθε αρχείο και σε κάθε φάκελλο στο σύστημά σας, αποδίδονται δικαιώματα πρόσβασης στον ιδιοκτήτη του αρχείου, στα μέλη μιας ομάδας συσχετιζόμενων χρηστών, ή και σε οποιονδήποτε άλλον. Τα δικαιώματα μπορούν να αποδοθούν μόνο για την ανάγνωση ενός αρχείου, για την εγγραφή στο αρχείο, και για την εκτέλεση ενός αρχείου (δηλ, το αρχείο αυτό να μπορεί να εκτελείται ως ένα πρόγραμμα).

Για ν δείτε τις ρυθμίσεις δικαιωμάτων ενός αρχείου, μπορούμε να χρησιμοποιήσουμε την εντολή `ls`, όπως φαίνεται στο παρακάτω παράδειγμα:

```
[me@linuxbox me]$ ls -l some_file
```

```
-rw-rw-r-- 1 me   me   1097374 Sep 26 18:48 some_file
```

Μπορούμε να αντλήσουμε πολλές πληροφορίες από την προσεκτική εξέταση των αποτελεσμάτων της εντολής αυτής:

- Το αρχείο με το όνομα "some_file" ανήκει στον χρήστη "me"
- Ο χρήστης "me" έχει το δικαίωμα ανάγνωσης και εγγραφής πάνω σε αυτό το αρχείο
- Το αρχείο ανήκει και στην ομάδα με όνομα "me"
- Τα μέλη της ομάδας "me" μπορούν, επίσης, να διαβάζουν και να γράφουν σε αυτό το αρχείο
- Ο καθένας μπορφεί να διαβάσει αυτό το αρχείο

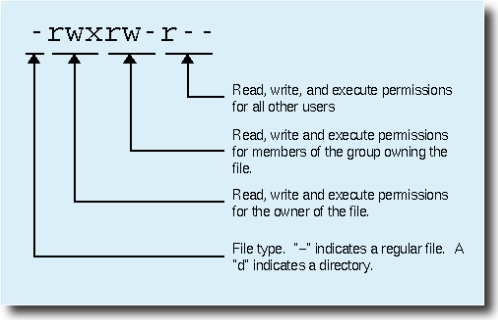
Ας δοκιμάσουμε ένα διαφορετικό παράδειγμα. Θα ρίξουμε μια ματιά στο πρόγραμμα `bash` που βρίσκεται στονβ φάκελλο `/bin` :

```
[me@linuxbox me]$ ls -l /bin/bash

-rwxr-xr-x 1 root root  316848 Feb 27  2000 /bin/bash
```

Εδώ μπορούμε να δούμε ότι:

- Το αρχείο "/bin/bash" ανήκει στον χρήστη "root"
- Ο υπερχρήστης έχει δικαίωμα ανάγνωσης, εγγραφής και εκτέλεσης σε αυτό το αρχείο
- Το αρχείο ανήκει στην ομάδα "root"
- Τα μέλη της ομάδας "root" μπορούν να διαβάζουν και να εκτελούν το αρχείο αυτό
- Ο καθένας μπορεί να διαβάζει και να εκτελεί αυτό το αρχείο



Στο διπλανό διάγραμμα, βλέπουμε την ερμηνεία του πρώτου τμήματος του καταλόγου. Υπάρχει ένας χαρακτήρας που δείχνει τον τύπο αρχείου, ακολουθούμενος από τρεις ομάδες των τριών χαρακτήρων, που μεταφέρουν τα δικαιώματα ανάγνωσης, εγγραφής και εκτέλεσης, για τον ιδιοκτήτη, την ομάδα και για τον οποιονδήποτε άλλον.

chmod

Η εντολή `chmod` χρησιμοποιείται για την αλλαγή των δικαιωμάτων ενός αρχείου ή ενός φακέλλου. Για να τη χρησιμοποιήσετε, καθορίστε ποιες είναι οι επιθυμητές ρυθμίσεις δικαιωμάτων, καθώς και ποιο αρχείο ή φάκελλο επιθυμείτε να τροποποιήσετε. Υπάρχουν δύο τρόποι για να ορίσετε τα δικαιώματα, αλλά πρόκειται να σας διδάξω μόνο τον έναν από τους δύο:

Εϊναι εύκολο να φαντασθεί κανείς τις ρυθμίσεις των δικαιωμάτων σαν μια σειρά από bits (που είναι, εξ' άλλου, ο τρόπος με τον οποίον τα βλέπει και ο υπολογιστής). Δείτε πως λειτουργεί:

```
rwx rwx rwx = 111 111 111
rw- rw- rw- = 110 110 110
rwx --- --- = 111 000 000

και ούτω καθ' εξής...

rwx = 111 στο δυαδικό σύστημα αριθμών (binary) = 7
rw- = 110 στο δυαδικό σύστημα αριθμών (binary) = 6
r-x = 101 στο δυαδικό σύστημα αριθμών (binary) = 5
```

```
r-- = 100 στο δυαδικό σύστημα αριθμών (binary) = 4
```

Τώρα, αν αναπαραστήσετε την κάθε μία από αυτές τις ομάδες δικαιωμάτων (τον ιδιοκτήτης, την ομάδα και τους άλλους) σαν ένα μόνον ψηφίο, τότε θα έχετε έναν πολύ βολικό τρόπο για να εκφράζετε τις διάφορες ρυθμίσεις δικαιωμάτων. Αν, για παράδειγμα, θέλαμε να ορίσουμε για το αρχείο `some_file`, να έχει δικαιώματα ανάγνωσης και εγγραφής για τον ιδιοκτήτη, αλλά θέλαμε να κρατήσουμε το αρχείο απρόσβατο για τους άλλους, τότε θα δίναμε την εξής εντολή:

```
[me@linuxbox me]$ chmod 600 some_file
```

Ακολουθεί ένας πίνακας αριθμών που καλύπτει όλες τις συνηθισμένες ρυθμίσεις. Εκείνες που αρχίζουν με το "7" χρησιμοποιούνται με προγράμματα (αφού επιτρέπουν την εκτέλεση), ενώ οι υπόλοιπες είναι για τα άλλα είδη αρχείων.

Τιμή	*Σημασία*
777	*(rwxrwxrwx)* Χωρίς περιορισμούς δικαιωμάτων. Ο οποιοσδήποτε μπορεί να κάνει οτιδήποτε. Σε γενικές γραμμές, δεν είναι μια επιθυμητή ρύθμιση.
755	*(rwxr-xr-x)* Ο ιδιοκτήτης του αρχείου μπορεί να διαβάζει, να γράφει και να εκτελεί το αρχείο. Όλοι οι υπόλοιποι μπορούν να διαβάζουν και να εκτελούν το αρχείο. Αυτή η ρύθμιση είναι συνηθισμένη για προγράμματα που χρησιμοποιούνται από όλους τους χρήστες.
700	*(rwx------)* Ο ιδιοκτήτης του αρχείου μπορεί να διαβάζει, να γράφει και να εκτελεί το αρχείο. Κανένας άλλος δεν διαθέτει οποιαδήποτε δικαιώματα. Αυτή η ρύθμιση είναι χρήσιμη για προγράμματα τα οποία μόνον ο ιδιοκτήτης μπορεί να χρησιμοποιεί και πρέπει να φυλάσσονται απρόσβατα προς τους άλλους.
666	*(rw-rw-rw-)* Όλοι οι χρήστες μπορούν να διαβάσουν και να γράψουν στο αρχείο.
644	*(rw-r--r--)* Ο ιδιοκτήτης μπορεί να διαβάζει και να γράφει σε αυτό το αρχείο, ενώ όλοι οι υπόλοιποι μπορούν μόνον να το διαβάζουν. Είναι μια συχνή ρύθμιση για αρχεί δεδομένων που ο ακθένας μπόρεί να διαβάσει, αλλά μόνον ο ιδιοκτήτης τους μπορεί να τα αλλάξει.
600	*(rw-------)* Ο ιδιοκτήτης μπορεί να διαβάζει και να γράφει σε αυτό το αρχείο, ενώ όλοι οι άλλοι δεν έχουν κανένα δικαίωμα. Είναι μια συχνή ρύθμιση για την περίπτωση όπου ο ιδιοκτήτης των αρχείων δεδομένων επιθυμεί να τα κρατά ιδιωτικά.

Δικαιώματα φακέλλου

Η εντολή **chmod** μπορεί να χρησιμοποιηθεί και για τον έλεγχο των δικαιωμάτων πρόσβασης στους φακέλλους. Ως επί το πλείστον, το σχήμα απόδοσης δικαιωμάτων στους φακέλλους λειτουργεί με τον ίδιο τρόπο όπως και στην περίπτωση των αρχείων. Τα δικαιώματα εκτέλεσης, όμως, χρησιμοποιούνται με ένα διαφορετικό τρόπο. Παρέχουν τον έλεγχο πρόσβασης στον κατάλογο αρχείων και σε άλλα

πράγματα. Ακολουθούν ορισμένες χρήσιμες ρυθμίσεις για τους φακέλλους:

Τιμή	*Σημασία*
777	*(rwxrwxrwx)* Χωρίς περιορισμούς δικαιωμάτων. Ο οποιοσδήποτε μπορεί να έχει τον κατάλογο των αρχείων, να δημιουργεί νέα αρχεία μέσα στον φάκελλο και να διαγράφει αρχεία μέσα σε αυτόν. Σε γενικές γραμμές, δεν είναι μια καλή ρύθμιση.
755	*(rwxr-xr-x)* Ο ιδιοκτήτης του φακέλλου έχει πλήρη πρόσβαση. Όλοι οι άλλοι μπορούν να έχουν καταλόγους του φακέλλου, αλλά χωρίς να μπορούν να δημιουργούν ή να διαγράφουν αρχεία. Αυτή η ρύθμιση είναι συχνή για του φακέλλους που επιθυμείτε να μοιραστείτε με άλλους χρήστες.
700	*(rwx------)* Ο ιδιοκτήτης του φακέλλου έχει πλήρη πρόσβαση. Κανείς άλλος δεν έχει οποιαδήποτε δικαιώματα. Αυτή η ρύθμιση είναι χρήσιμη για τους φακέλλους που μόνον ο ιδιοκτήτης μπορεί να χρησιμοποιεί και που πρέπει να διατηρούνται απρόσβατοι προς τους άλλους.

Γίνετε υπερχρήστης (superuser) για λίγο

Είναι, συχνά, χρήσιμο να γίνεστε ο υπερχρήστης (*superuser*), για να μπορείτε να εκτελέσετε σημαντικές εργασίες διαχείρισης του συστήματος. Έχετε, όμως, προειδοποιηθεί (και όχι μόνον από εμένα!), πως δεν θα πρέπει να παραμένετε συνδεδεμένος σαν υπερχρήστης. Στις περισσότερες διανομές, υπάρχει ένα πρόγραμμα που μπορεί να σας δίνει προσωρινή πρόσβαση στα προνόμια του υπερχρήστη. Αυτό το πρόγραμμα λέγεται **su** (που ειναι η συντόμευση των λέξεων: *substitute user*) και μπορεί να χρησιμοποιηθεί σε εκείνες τις περιπτώσεις που χρειάζεσθε να είστε ο υπερχρήστης για λίγες εργασίες. Για να γίνετε υπερχρήστης, απλώς πληκτρολογείστε την εντολή `su`. Θα σας ζητηθεί αμέσως ο κωδικός υπερχρήστης:

```
[me@linuxbox me]$ su
Password:
[root@linuxbox me]#
```

Μετά την εκτέλεση της εντολής `su`, εισέρχεσθε σε μια νέα συνεδρία κελύφους (shell session) με την ιδιότητα του υπερχρήστη. Για να βγείτε από αυτήν, πληκτρολογήστε `exit` και θα επιστρέψετε στην προηγούμενη συνεδρία σας.

Σε μερικές διανομές, μεταξύ των οποίων η γνωστότερη είναι το Ubuntu, χρησιμοποιείται μια εναλλακτική μέθοδος. Αντί για τη χρήση του `su`, αυτά τα συστήματα χρησιμοποιούν, αντιθέτως, την εντολή `sudo`. Με το `sudo`, παραχωρούνται δικαιώματα υπερχρήστη σε έναν ή περισσότερους χρήστες, αναλόγως των αναγκών. Για την εκτέλεση μιας εντολής με την ιδιότητα του υπερχρήστη, πληκτρολογούμε πρώτα την εντολή `sudo,` και στη συνέχεια ακολουθεί απλώς η επιθυμητή εντολή. Μετά την πληκτρολόγηση της εντολής, θα ζητηθεί από τον χρήστη να δώσει τον κωδικό χρήστη, αντί για τον κωδικό υπερχρήστη:

```
[me@linuxbox me]$ sudo some_command
Password:
[me@linuxbox me]$
```

Αλλαγή ιδιοκτησίας αρχείου

Μπορείτε να αλλάξετε τον ιδιοκτήτη ενός αρχείου, χρησιμοποιώντας την εντολή `chown`. Ακολουθεί ένα παράδειγμα: Ας υποθέσουμε ότι ήθελα να αλλάξω τον ιδιοκτήτη του αρχείου `some_file` από τον "me" στο "you". Θα μπορούσα να γράψω:

```
[me@linuxbox me]$ su
Password:
[root@linuxbox me]# chown you some_file
[root@linuxbox me]# exit
[me@linuxbox me]$
```

Σημειώστε ότι για να αλλάξω τον ιδιοκτήτη ενός αρχείου, πρέπει να είστε ο υπερχρήστης. Για να το κάνετε αυτό, στο παράδειγμά μας χρησιμοποιήσαμε την εντολή `su`, μετά, εκτελέσαμε το `chown`, και τελικά, πληκτρολογήσαμε `exit` για να επιστρέψουμε στην προηγούμενη συνεδρία μας.

Η εντολή `chown` λειτουργεί με τον ίδιο τρόπο στους καταλόγους, όπως και στα αρχεία.

Αλλαγή ιδιοκτησίας ομάδας

Η ιδιοκτησία ομάδας ενός αρχείου ή ενός καταλόγου, μπορούν να αλλάξουν με την εντολή `chgrp`. Αυτή η εντολή χρησιμοποιείται ως εξής:

```
[me@linuxbox me]$ chgrp new_group some_file
```

Στο παραπάνω παράδειγμα, αλλάξαμε την ιδιοκτησία ομάδας του αρχείου `some_file`, από την προηγούμενη πμάδα στην οποία ανήκε, σε μία νέα ομάδα "new_group". Πρέπει να είστε ο ιδιοκτήτης του αρχείου ή του καταλόγου, για να εκτελέσετε την λειτουργία `chgrp`.

10

Έλεγχος εργασιών

Στο προηγούμενο μάθημα, εξετάσαμε μερικές πλευρές της ικανότητας του GNU/ Linux να είναι ένα λειτουργικό σύστημα που επιτρέπει την πρόσβαση σε πολλούς χρήστες ταυτοχρόνως. Σε αυτό το μάθημα, θα εξετάσουμε την ικανότητα του GNU/Linux για ταυτόχρονη εκτέλεση πολλών εργασιών (multitasking) και πως μπορούμε να τη χειριστούμε μέσω της γραμμής εντολών.

Όπως και κάθε λειτουργικό σύστημα ικανό να εκτελεί συγχρόνως πολλές εργασίες (multitasking), έτσι και το GNU/Linux εκτελεί πολλές και παράλληλες διεργασίες. Τέλος πάντων, μοιάζουν να είναι ταυτόχρονες. Στην πραγματικότητα, ένας μεμονωμένος επεξεργαστής ενός υπολογιστή, μπορεί μόνο να φέρει εις πέρας μία διεργασία κάθε φορά, αλλά ο πυρήνας Linux καταφέρνει να παρουσιάζει την κάθε διεργασία με τη σειρά στον επεξεργαστή και έτσι, μοιάζει σαν να τρέχουν όλες συγχρόνως.

Υπάρχουν διάφορες εντολές που μπορείτε να χρησιμοποιήσετε για τον έλεγχο αυτών των διαδικασιών. Είναι οι εξής:

- `ps` – παράθεση σε κατάλογο των διαδικασιών που τρέχουν στο σύστημα
- `kill` – αποστολή ενός σήματος σε μια ή περισσότερες διεργασίες, συνήθως για να τερματίσετε (ή να "σκοτώσετε") μια διεργασία.
- `jobs` – ένας εναλλακτικός τρόπος για να πάρετε ένα κατάλογο των διεργασιών σας
- `bg` – βάλτε μια διεργασία στο παρασκήνιο
- `fg` - βάλτε μια διεργασία στο προσκήνιο

Ένα πρακτικό παράδειγμα

Ενώ αυτό το παράδειγμα μπορεί να φαίνεται αρκετά δυσνόητο, μπορεί όμως, από την άλλη πλευρά, να είναι πολύ πρακτικό για τον μέσο χρήστη, ο οποίος δουλεύει κυρίως με την γραφική διεπαφή χρήστη (GUI). Μπορεί και να μην το γνωρίζατε αυτό, αλλά τα περισσότερα (αν όχι όλα) τα γραφικά προγράμματα, μπορούν να εκκινηθούν από την γραμμή εντολών. Να ένα παράδειγμα: υπάρχει ένα προγραμματάκι που παρέχεται με το σύστημα διαχείρισης παραθύρων X Windows, που ονομάζεται `xload` και το οποίο εμφανίζει μια γραφική παράσταση που δείχνει τον φόρτο του συστήματος. Μπορείτε να εκτελέσετε αυτό το πρόγραμμα, πληκτρολογώντας τα εξής:

```
[me@linuxbox me]$ xload
```

Σημειώστε ότι εμφανίζεται το μικρό παράθυρο του `xload` και αρχίζει να δείχνει την γραφική αναπαράσταση με τον φόρτο του συστήματος. Σημειώστε, επίσης, ότι το σύμβολο προτροπής (prompt) δεν επανεμφανίσθηκε μετά την εκκίνηση του προγράμματος. Το κέλυφος περιμένει να τελειώσει πρώτα το πρόγραμμα, πριν περάσει πάλι τον έλεγχο σε εσάς. Αν κλείσετε το παράθυρο με το `xload`, το πρόγραμμα `xload` τερματίζει και το σύμβολο προτροπής επιστρέφει.

Πως να θέσετε ένα πρόγραμμα στο παρασκήνιο

Τώρα, για να κάνουμε τη ζωή μας λίγο ευκολότερη, θα εκκινήσουμε ξανά το πρόγραμμα `xload`, αλλά αυτή τη φορά, θα το βάλουμε στο υπόβαθρο, έτσι ώστε να επιστρέψει και πάλι το σύμβολο προτροπής (prompt). Για να το πετύχουμε, εκτελούμε το xload ως εξής:

```
[me@linuxbox me]$ xload &
[1] 1223

[me@linuxbox me]$
```

Σε αυτή την περίπτωση, η προτροπή (prompt) επέστρεψε, διότι η διεργασία είχε τεθεί στο παρασκήνιο.

Τώρα, φαντασθείτε ότι ξεχάσατε να χρησιμοποιείτε το σύμβολο "&" για να βάλετε το πρόγραμμα στο υπόβαθρο. Υπάρχει ακόμη ελπίδα. Μπορείτε να πληκτρολογήσετε control-z και η διεργασία θα ανασταλεί. Η διεργασία θα υπάρχει ακόμη, αλλά σε αδρανή κατάσταση. Για να ξαναξεκινήσει η διεργασία στο υπόβαθρο, πληκτρολογήστε την εντολή `bg` (συντόμευση της λέξης background). Ακολουθεί ένα παράδειγμα:

```
[me@linuxbox me]$ xload
[2]+ Stopped xload

[me@linuxbox me]$ bg
[2]+ xload &
```

Εμφάνιση των διαδικασιών σας σε κατάλογο

Τώρα που έχουμε μια διεργασία στο υπόβαθρο, θα ήταν χρήσιμο να εμφανίσουμε έναν κατάλογο των διαδικασιών που εκκινήσαμε. Για να το πετύχουμε, μπορούμε να χρησιμοποιήσουμε ή την εντολή **`jobs`**, είτε την ακόμη πιο ισχυρή εντολή **`ps`**.

```
[me@linuxbox me]$ jobs
[1]+ Running xload &

[me@linuxbox me]$ ps
PID TTY TIME CMD
1211 pts/4 00:00:00 bash
1246 pts/4 00:00:00 xload
1247 pts/4 00:00:00 ps

[me@linuxbox me]$
```

Πως να τερματίσετε (να “σκοτώσετε”) μια διεργασία

Ας υποθέσουμε ότι έχουμε ένα πρόγραμμα που γεν ανταποκρίνεται (χμμμ.. το Netscape έρχεται στο μυαλό μας ;-) Πως μπορείτε να το ξεφορτωθείτε; Ασφαλώς, θα χρησιμοποιήσετε την εντολή `kill`. Ας το δοκιμάσουμε με το xload. Κατ' αρχάς, πρέπει να βρείτε ποια ακριβώς είναι η διεργασία που θέλετε να σκοτώσετε. Για να το πετύχετε, μπορείτε να χρησιμοποιήσετε είτε το `jobs` είτε το `ps`. Αν χρησιμοποιήσετε το **`jobs`**, θα σας επιστρέψει έναν αριθμό εργασίας (job number). Με το **`ps`**, θα λάβετε έναν αριθμό ταυτοποίησης της διεργασίας (PID). Θα δοκιμάσουμε και με τους δύο τρόπους:

```
[me@linuxbox me]$ xload &
[1] 1292

[me@linuxbox me]$ jobs
[1]+ Running xload &

[me@linuxbox me]$ kill %1

[me@linuxbox me]$ xload &
[2] 1293
[1] Terminated xload

[me@linuxbox me]$ ps
PID TTY TIME CMD
1280 pts/5 00:00:00 bash
```

```
1293 pts/5 00:00:00 xload
1294 pts/5 00:00:00 ps
[me@linuxbox me]$ kill 1293
[2]+ Terminated xload
[me@linuxbox me]$
```

Λίγες παραπάνω λεπτομέρειες για το “σκότωμα”

Αν και η εντολή **`kill`** χρησιμοποιείται για το “σκότωμα” μιας διεργασίας, ο πραγματικός της σκοπός είναι να στέλνει *σήματα* στις διεργασίες. Τις περισσότερες φορές, το σήμα λέει στην διεργασία να σταματήσει και να φύγει, αλλά η εντολή περιέχει πολλά παραπάνω από αυτό. Τα προγράμματα (αν είναι σωστά γραμμένα) ακούν τα σήματα από το λειτουργικό σύστημα και απαντούν σε αυτά, επιτρέποντας τις περισσότερες φορές την εφαρμογή μιας χαριτωμένης μεθόδου για τον τερματισμό. Για παράδειγμα, ένας επεξεργαστής κειμένου μπορεί να ακούει για σήματα που υποδεικνύουν ότι ο χρήστης αποσυνδέεται, ή ότι ο υπολογιστής κλείνει. Όταν λάβει αυτό το σήμα, αποθηκεύει την εργασία που βρίσκεται σε εξέλιξη, πριν την έξοδο. Η εντολή `kill` μπορεί να στείλει μια ποικιλία σημάτων στις διεργασίες. Αν πληκτρολογήσετε: kill -l

θα λάβετε έναν κατάλογο των σημάτων που υποστηρίζονται. Τα περισσότερα είναι μάλλον σκοτεινής φύσεως, αλλά πολλά είναι χρήσιμο να τα γνωρίζετε:

Αριθμός Σήματος	*Όνομα*	*Περιγραφή*
1	*SIGHUP*	Σήμα για κλείσιμο. Τα προγράμματα μπορούν να ακούσουν αυτό το σήμα και να αναλάβουν δράση (ή κα όχι) ανταποκρινόμενα σε αυτό.
2	*SIGINT*	Σήμα διακοπής (interrupt signal). Αυτό το σήμα αποστέλλεται στις διεργασίες για τον τερματισμό τους. Τα προγράμματα μπορούν να επεξεργασθούν αυτό το σήμα και να δράσουν με βάση αυτό. Μπορείτε, επίσης, να στείλετε αυτό το σήμα με άμεσο τρόπο, πληκτρολογώντας ***control-c*** στο παράθυρο του τερματικού όπου τρέχει το πρόγραμμα.
15	*SIGTERM*	Σήμα τερματισμού (termination signal). Αυτό το σήμα αποστέλλεται στις διεργασίες για τον τερματισμό τους. Τα προγράμματα μπορούν, και πάλι, να επεξεργασθούν αυτό το σήμα και να δράσουν με βάση αυτό. Μπορείτε, επίσης, να στείλετε αυτό το σήμα με άμεσο τρόπο, πληκτρολογώντας control-c στο παράθυρο του τερματικού, όπου τρέχει το πρόγραμμα. Αυτό είναι το σήμα που αποστέλλεται από προεπιλογή εκ μέρους της ενοτολής `kill`, αν δεν διευκρινισθεί κανένα σήμα.
9	*SIGKILL*	Σήμα τερματισμού (Kill signal). Αυτό το σήμα προκαλεί τον άμεσο τερματισμό της διεργασίας από τον πυρήνα Linux. Τα προγράμματα δεν είναι σε θέση να ακούν για το σήμα αυτό.

Τώρα, ας υποθέσουμε ότι έχετε ένα πρόγραμμα που έχει κρεμάσει απελπιστικά (π.χ. το Netscape) και ότι θέλετε να απαλλαγείτε από αυτό. Μπορείτε να δείτε παρακάτω τι να κάνετε:

1. Χρησιμοποιήστε την εντολή `ps` για να λάβετε τον αριθμό ταυτοποίησης (PID) της διεργασίας που θέλετε να τερματίσετε.
2. Δώστε μια εντολή `kill` για το PID αυτό.
3. Αν η διεργασία αρνείται να τερματίσει (δηλ. αγνοεί το σήμα), συνεχίστε να στέλνετε όλο και πιο σκληρά σήματα, μέχρι να τερματίσει.

```
[me@linuxbox me]$ ps x
PID TTY STAT TIME COMMAND
2931 pts/5 SN 0:00 netscape
[me@linuxbox me]$ kill -SIGTERM 2931
[me@linuxbox me]$ kill -SIGKILL 2931
```

Στο παραπάνω παράδειγμα, χρησιμοποίησα την εντολή `kill` με τον επίσημο τρόπο. Στην πραγματικότητα, είναι πολύ πιο συνηθισμένο να το κάνει κανείς με τον εξής τρόπο, αφού το προεπιλεγμένο σήμα που στέλνει το `kill`, είναι το SIGTERM, ενώ το `kill` μπορεί και να χρησιμοποιήσει τον αριθμό του σήματος, αντί για το όνομα του σήματος:

```
[me@linuxbox me]$ kill 2931
```

Στη συνέχεια, αν η διεργασία δεν τερματίσει, αναγκάστε την με το σήμα SIGKILL:

```
[me@linuxbox me]$ kill -9 2931
```

Αυτό ήταν!

Εδώ ολοκληρώθηκε η σειρά μαθημάτων "Μαθαίνοντας το κέλυφος". Στην επόμενη σειρά, "Πως να γράφετε σενάρια κελύφους (shell scripts)," θα εξετάσουμε πως να αυτοματοποιούμε κάποιες εργασίες με το κέλυφος.

Μετάφραση στα Ελληνικά: Κωστής Μουσαφείρης
Επίβλεψη ορολογίας: Γιώργος Βλαχάβας