Linux για αρχάριους/Βασικές γνώσεις τερματικού

1 Βασικές γνώσεις τερματικού

Συμβουλή: Ο καλύτερος τρόπος για να μάθετε την γραμμή εντολών, είναι να δοκιμάζετε τις εντολές με διάφορους τρόπους στο τερματικό σας, ταυτόχρονα με την ανάγνωση του παρακάτω οδηγού!!
Προειδοποίηση: Πρίν αρχίσετε αυτόν τον οδηγό, καλό θα ήταν να γνωρίζετε τα βασικά για το folder structure σε Unix-οειδή συστήματα.

1.1 Φάκελοι και Αρχεία (Folders and Files)

1.1.1 Ιεραρχία αρχείων και φακέλων

Στο Linux η βασική μονάδα ταξινόμησης των αρχείων είναι ο "**φάκελος**" (folder). Κάθε αρχείο βρίσκεται μέσα σε ένα φάκελο, ο οποίος φυσικά μπορεί να περιέχει πολλά αρχεία. Κάθε φάκελος με τη σειρά του βρίσκεται μέσα σε κάποιον άλλο φάκελο ο οποίος ονομάζεται "*γονικός φάκελος*" (parent folder). Η μόνη εξαίρεση είναι ο φάκελος / ο οποίος ονομάζεται "*ριζικός φάκελος*" (root folder) και είναι η κορυφή της ιεραρχίας των φακέλων, είναι δηλαδή ο φάκελος που περιέχει όλους του άλλους φακέλους και αρχεία.

Σημείωση: Γονικός φάκελος ονομάζεται ο ιεραρχικά ανώτερος φάκελος του τρέχοντα φακέλου, δηλαδή ο φάκελος που βρίσκεται ακριβώς από πάνω.

1.1.2 Everything is a file (τα πάντα είναι ένα αρχείο)

Στο Linux, όσον αφορά το ίδιο το Λειτουργικό Σύστημα δεν υπάρχει ουσιαστική διάκριση μεταξύ αρχείων και φακέλων. Για την ακρίβεια, τα ονόματα των αρχείων και των φακέλων αντιμετωπίζονται ακριβώς με τον ίδιο τρόπο. Το μόνο που διαχωρίζει το όνομα ενός αρχείου από ένα φάκελο είναι ότι στους φακέλους μπορεί να μπει ένα / στο τέλος του ονόματος, η ύπαρξη του οποίου όμως είναι προαιρετική. Πχ στο ακόλουθο παράδειγμα, ενώ για την πρώτη γραμμή είμαστε σίγουροι ότι το όνομα αναφέρεται σε ένα φάκελο, για την δεύτερη γραμμή δεν μπορούμε να ξέρουμε που ακριβώς αναφέρεται. Όταν θέλουμε να είμαστε απολύτως σίγουροι ότι αναφερόμαστε σε ένα φάκελο τότε βάζουμε πάντα ένα / στο τέλος του path.
<pre<noinclude></noinclude>>/home/john/this.is.a.folder/
/home/johy/this.is.a.just.a.name</pre<noinclude></noinclude>>

Προειδοποίηση: Στο Linux, εν αντιθέσει με τα Windows, οι καταλήξεις των ονομάτων των αρχείων και των φακέλων **δεν** παίζουν κανένα ρόλο. Έτσι μπορούμε να έχουμε ονόματα φακέλων με . όπως πχ το /home/john/myphoto.jpg/.
Προειδοποίηση: Στο Linux τα πεζά/κεφαλαία γράμματα **έχουν** σημασία, δηλαδή οι ακόλουθοι είναι δύο διαφορετικοί φάκελοι: /home/john/doc/ και /home/john/DOC/.

1.1.3 Εμφάνιση τρέχοντος φακέλου

Κάθε φορά που μπαίνουμε σε ένα περιβάλλον γραμμής εντολών (ονομάζεται και κονσόλα ή τερματικό) βρισκόμαστε πάντα στον προσωπικό φάκελο του user μας. Η εντολή pwd (από τα αρχικά της φράσης “print working directory”) μας δείχνει το όνομα του φακέλου στον οποίο βρισκόμαστε (τρέχον φάκελος). Π.χ. αν ο user μας ονομάζεται John, με την εκτέλεση της παραπάνω εντολής, η έξοδος που θα πάρουμε θα ήταν η εξής: <pre<noinclude></noinclude> style="margin-bottom: 0; border-bottom:none; padding-bottom:0.8em; background-color: LightCyan">$ pwd</pre<noinclude></noinclude>> <pre<noinclude></noinclude> style="margin-top: 0; border-top-style:dashed; padding-top: 0.8em; background-color: LightCyan">/home/john</pre<noinclude></noinclude>>

1.1.4 Paths (διαδρομές)

Το όνομα κάθε αρχείου και φακέλου ορίζεται από τη διαδρομή των φακέλων (path) που συνδέει το root folder με τον υπο-εξέταση φάκελο. Το όνομα αυτό είναι μοναδικό και ονομάζεται absolute path (απόλυτη διαδρομή). Μερικά παραδείγματα absolute paths είναι τα ακόλουθα:

- Ο φάκελος /home/john/Documents/photos
- Το αρχείο /home/john/Documents/photos/birthday1.jpg

- Το αρχείο /home/john/Documents/photos/birthday2.jpg
- Το αρχείο /home/john/Documents/photos/birthday3.jpg

Όπως γίνεται εύκολα κατανοητό η χρήση των absolute paths ορίζει μονοσήμαντα κάθε αρχείο/φάκελο, όμως μπορεί να γίνει σύντομα κουραστική λόγω της επανάληψης και του μεγέθους των paths. Για το λόγο αυτό υπάρχουν και τα λεγόμενα relative paths (σχετικές διαδρομές), μέσω των οποίων μπορούμε να αναφερθούμε σε ένα αρχείο/φάκελο, όχι σε σχέση με το root folder / αλλά σε σχέση με τον τρέχοντα φάκελο. Αυτό γίνεται μέσω δύο ειδικών αρχείων που περιέχονται σε κάθε φάκελο και τα οποία συμβολίζονται με μία και δύο τελείες αντίστοιχα (**.** και **..**). Το **.** υποδεικνύει τον τρέχοντα φάκελο, ενώ το **..** υποδεικνύει τον γονικό του φάκελο.

Έτσι, παραδείγματος χάρη, αν βρισκόμαστε μέσα στο φάκελο /home/john/:

- Ο τρέχον φάκελος είναι ο /home/john και μπορούμε να αναφερθούμε σε αυτόν μέσω του .
- Ο γονικός φάκελος του /home/john είναι ο /home και μπορούμε να αναφερθούμε σε αυτόν μέσω του ..
- Ο ο γονικός φάκελος του /home είναι ο / και μπορούμε να αναφερθούμε σε αυτόν μέσω του ../..

Σημείωση: Για λόγους ευκολίας, γίνεται η θεώρηση ότι ο γονικός φάκελος του / είναι το ίδιο το /.

1.1.5 Αλλαγή φακέλου

Η εντολή για να μεταβούμε σε έναν διαφορετικό φάκελο είναι η εντολή cd (από τα αρχικά της φράσης “change directory”). Για να μεταβούμε σε έναν φάκελο αρκεί να γράψουμε cd όνομα_φακέλου (δηλαδή “cd” (κενό) και το όνομα του φακέλου).

Αν το όνομα του φακέλου είναι ένα absolute path τότε ανεξαρτήτως ποιος είναι ο τρέχοντας φάκελος θα μεταφερθούμε στο φάκελο που δείχνει το absolute path (με την προϋπόθεση ότι αυτό το path υπάρχει φυσικά...). Για παράδειγμα, αν θέλουμε να μεταβούμε στον φάκελο /home/john/Documents/school τότε αρκεί να δώσουμε την εντολή:

<pre<noinclude></noinclude>>cd /home/john/Documents/school</pre<noinclude></noinclude>>

Η προηγούμενη εντολή θα δουλέψει ανεξαρτήτως του που βρισκόμαστε αυτήν την στιγμή. Αν όμως βρισκόμαστε σε έναν από τους γονικούς φακέλους του /home/john/Documents/school τότε υπάρχει και πιο εύκολος τρόπος για να μεταβούμε σε αυτόν, και δεν είναι άλλος από την χρήση των relative paths. Πχ, έστω ότι βρισκόμαστε στον φάκελο /home/john/Documents. Στην περίπτωση αυτή, για να μεταβούμε στον υποφάκελο school, αρκεί να δώσουμε: <pre<noinclude></noinclude>>cd school</pre<noinclude></noinclude>>

Προκειμένου να καταλάβουμε καλύτερα τη χρήση των relative paths, ας δούμε μερικά παραδείγματα ακόμα. Έστω για παράδειγμα ότι βρισκόμαστε στον φάκελο /home/john/Documents/. Αν δώσουμε cd . τότε *δεν* θα αλλάξουμε φάκελο καθώς, όπως είπαμε το . συμβολίζει τον τρέχοντα φάκελο. Εάν, από την άλλη, επιθυμούμε να μεταβούμε στον γονικό φάκελο, αρκεί να δώσουμε cd .. (προσοχή στο κενό πριν το ..!) και έτσι θα επιστρέψουμε στον /home/john. Εάν βρισκόμαστε, για παράδειγμα, στον φάκελο /home/john/Documents/school/ και θέλουμε με έναν εύκολο τρόπο να μεταβούμε αμέσως στον αρχικό μας φάκελο /home/john, αυτό μπορούμε να το επιτύχουμε δίνοντας cd ../.. (μεταφερόμαστε δηλαδή στον γονικό του γονικού φάκελο).

Μία πάρα πολύ χρήσιμη συντόμευση είναι η χρήση της cd σκέτης χωρίς κανένα path μετά από αυτήν. Κάθε φορά που δίνουμε αυτή την εντολή μεταφερόμαστε στον αρχικό φάκελο του χρήστη μας. Πχ στο προηγούμενο παράδειγμα, αντί της cd ../.., μπορούσαμε να δώσουμε απλά cd και να έχουμε το ίδιο αποτέλεσμα. Προφανώς, για λόγους ευκολίας προτιμούμε τη χρήση του cd.

1.1.6 Tab Completion (αυτοσυμπλήρωση)

Όταν γράφουμε paths με το χέρι είναι πολύ εύκολο να κάνουμε λάθη. Για το λόγο αυτό, όλα τα σύγχρονα τερματικά μας δίνουν βοήθεια, συμπληρώνοντας μόνα τους τα paths. Αυτό γίνεται ως ακολούθως.

Έστω ότι θέλουμε να πάμε στον φάκελο /home/john. Αρχικά γράφουμε στο τερματικό χωρίς να πατήσουμε όμως ακόμα το “Enter": <pre<noinclude></noinclude>>cd /h</pre<noinclude></noinclude>> και σε αυτό το σημείο πατάμε το πλήκτο TAB, και θα δούμε ότι θα συμπληρωθεί το υπόλοιπο μέρος του ονόματος του φακέλου από μόνο του και ξαφνικά στο τερματικό εμφανίζεται το: <pre<noinclude></noinclude>>cd /home/</pre<noinclude></noinclude>> Σε αυτό το σημείο συνεχίζουμε γράφοντας τον πρώτο χαρακτήρα του υποφακέλου του "/home/" που μας ενδιαφέρει, δηλαδή: <pre<noinclude></noinclude>>cd /home/j</pre<noinclude></noinclude>> και πατάμε πάλι “TAB” ώστε να συμπληρωθεί πάλι το όνομα του επόμενου φακέλου. Θα δούμε ότι μετά από αυτό στο τερματικό θα γράφει: <pre<noinclude></noinclude>>cd /home/john</pre<noinclude></noinclude>>

Την παραπάνω διαδικασία την συνεχίζουμε έως ότου

συμπληρωθεί η πλήρης διαδρομή που μας ενδιαφέρει. Το παραπάνω μπορούσε να αυτοσυμπληρωθεί, διότι στον φάκελο "/" υπάρχει συνήθως μόνο ένας φάκελος του οποίου το όνομα αρχίζει από “h”. Για να δούμε και μια άλλη περίπτωση, υποθέστε ότι ο ριζικός φάκελος περιέχει και έναν υποφάκελο “house”. Εάν πατούσαμε πάλι το “TAB”, τότε η αυτοσυμπλήρωση θα σταματούσε στο σημείο όπου οι φακέλοι θα είχαν διαφορετικούς χαρακτήρες στο όνομά τους. Για να καταλάβετε καλύτερα, οι φάκελοι “home” και “house” έχουν κοινούς χαρακτήρες έως το “ho”, άρα η αυτοσυμπλήρωση σταματάει σε αυτό το σημείο και στην οθόνη εμφανίζεται:

cd /ho

ενώ περιμένει από μας να εισάγουμε τον επόμενο (ή επόμενους) χαρακτήρες. Εάν πατήσουμε τώρα “u” και αμέσως μετά “TAB”, τότε η αυτοσυμπλήρωση μας δίνει, όπως θα περιμέναμε:

cd /house/

Ένα επιπρόσθετο βοηθητικό εργαλείο που μας προσφέρει το τερματικό είναι η χρήση του διπλού “TAB"! Έστω ότι έχουμε εισάγει:

cd /home/john/

και σε αυτό το σημείο πατήσουμε διπλό “TAB”, θα εμφανιστούν όλοι οι φακέλοι που είναι διαθέσιμοι στον κατάλογο “john”, δηλαδή θα μας εμφάνιζε:

cd /home/john/ Documents Downloads Music Pictures Videos (κλπ)

ώστε να ξέρουμε τι επιλογές έχουμε και να μας κάνει την ζωή πιο εύκολη :D

Ένα χρήσιμο σύμβολο που μας βοηθάει στην περιήγηση των φακέλων είναι το "~". Το σύμβολο αυτό αντικαθιστά την διαδρομή από τον ριζικό φάκελο έως τον προσωπικό μας φάκελο. Στην περίπτωσή μας δηλαδή "/home/john/". Για παράδειγμα βρισκόμαστε στον φάκελο "/house/" και θέλουμε να μεταβούμε γρήγορα στον φάκελο "/home/john/Documents/school/", ένας τρόπος για να το επιτύχουμε αυτό είναι να δώσουμε:

cd ~/Documents/school

Τέλος, ένα ακόμα χρήσιμο σύμβολο είναι το "-", το οποίο μας πηγαίνει στον αμέσως προηγούμενο φάκελο, στον οποίο βρισκόμασταν πρίν. Αυτή τη στιγμή, λοιπόν, βρισκόμαστε στον φάκελο "/home/john/Documents/school” και θέλουμε να επιστρέψουμε πάλι στο "/house” όπου ήμασταν πριν, με την εντολή:

cd -

Οι εντολές δεν είναι τίποτα παραπάνω από κανονικά προγράμματα όπως για παράδειγμα το γνωστό firefox. Θα μπορούσατε να δώσετε την εντολή

firefox

(εφόσον το έχετε εγκατεστημένο) και να σας άνοιγε ένα νέο παράθυρο του browser. Ο διαχωρισμός γίνεται μόνο σε εντολές (προγράμματα) που προορίζονται καθαρά για την γραμμή εντολών και τις υπόλοιπες που έρχονται και μαζί με ένα γραφικό περιβάλλον (GUI).

Σημείωση: Καμιά φορά χρησιμοποιούμε το τερματικό για να ανοίξουμε προγράμματα με GUI, όταν παρατηρούμε παράξενη ή δυσλειτουργική συμπεριφορά σε αυτά, διότι με αυτόν τον τρόπο στο τερματικό θα εκτυπωθούν τα σφάλματα των προγραμμάτων αυτών και αυτό μας βοηθάει πάρα πολύ στην αντιμετώπιση των προβλημάτων!

Για να καταλάβετε τις εντολές κάπως καλύτερα, θα πρέπει να γνωρίζετε την γενική τους μορφή η οποία είναι η εξής:

βασική_εντολή -παράμετρος1 τιμή_παραμέτρου1 -παράμετρος2 τιμή_παραμέτρου2 κλπ

ενώ συνήθως δεν χρειάζονται όλα αυτά. Στο παραπάνω παράδειγμα είδαμε την εντολή “cd” σε δύο πιό απλοικές μορφές

cd (σκέτη βασική_εντολή)

και

cd ~/Documents (βασική_εντολή τιμή_παραμέτρου1)

1.2 Περιεχόμενα φακέλου

Παραπάνω είδαμε το διπλό "TAB", για να εμφανίσουμε το σύνολο των διαθέσιμων υποφακέλων. Αυτός ο τρόπος χρησιμοποιείται κυρίως όταν γράφουμε την διαδρομή. Όταν βρισκόμαστε σε κάποιον φάκελο και θέλουμε να εμφανίσουμε τους υποφακέλους, αλλά και τα αρχεία, δίνουμε την εντολή

ls

(**list**) και θα μας τα εμφανίσει με παρόμοιο τρόπο του διπλού “TAB”. Δίνοντας την εντολή

ls -l

επιστρέφεται αυτή τη φορά μία πιό λεπτομερής λίστα των αρχείων και φακέλων, μαζί με ενδιαφέρουσα χαρακτηριστικά του καθενός. Από τα αριστερά προς τα δεξιά βλέπουμε τα δικαιώματα του αρχείου (δείτε σχετικά εδώ), τον αριθμό των συνδέσμων ή υποφακέλων μέσα σε αυτόν τον φάκελο, τον ιδιοκτήτη του αρχείου, το group στο οποίο ανήκει αυτό το αρχείο, το μέγεθος σε KB, την ημερομηνία τελευταίας τροποποίησης και τέλος το όνομα του αρχείου. Στις δυο παραπάνω εντολές δεν εμφανίζονται τα κρυφά αρχεία. Αυτό το επιτυγχάνουμε δίνοντας μαζί και την παράμετρο "-a”. Δηλαδή

ls -a

και

ls -l -a

αντίστοιχα. Στην δεύτερη περίπτωση επειδή οι παράμετροι δεν παίρνουν τιμή_παράμετρου μπορούμε να το γράψουμε και κάπως πιο μαζεμένα με αυτόν τον τρόπο:

ls -al

Τα γράμματα των παραμέτρων δεν θα μπορούσαν να ήταν τυχαία. Εδώ βλέπουμε ότι το "-a" σημαίνει "**a**ll" και το "-l" σημαίνει "**l**ist". Ίσως παρατηρήσατε, ότι στην εντολή “ls -l” (ή “ls -al”) τα μεγέθη των αρχείων δεν είναι και τόσο ευανάγνωστα. Για να το διορθώσουμε αυτό, μπορούμε να δώσουμε μαζί και την παράμετρο "-h" (human readable), δηλαδή

ls -alh

Επειδή δεν μπορούμε να θυμόμαστε πάντα όλες τις παραμέτρους απ'έξω, μαζί με σχεδόν όλες τις εντολές έρχεται και ένα manual μαζί. Για να δούμε λοιπόν, το manual της “ls” αρκεί να δώσουμε

man ls

και όταν θελήσουμε να βγούμε από αυτό, απλά πατάμε το “q"(**q**uit). Πολλά προγράμματα προσφέρουν επίσης την επιλογή (παράμετρο) "--help”. Παράδειγμα:

ls --help

όπου θα βρείτε αντίστοιχη βοήθεια με του “man”.

1.3 Αντιγραφή αρχείων

Εν συνεχεία, θα μάθουμε την εντολή αντιγραφής, την

cp

Για παράδειγμα, θέλουμε να πάρουμε backup ένα αρχείο κειμένου. Τότε δίνουμε:

cp κείμενο.txt κείμενο_backup.txt

Η πρώτη παράμετρος ορίζει την πηγή και η δεύτερη τον προορισμό. Εάν το δεύτερο αρχείο υπήρχε πρίν την παραπάνω εντολή, τότε τα περιεχόμενα του θα διαγραφούν και θα αντικατασταθούν με τα περιεχόμενα του πρώτου. Εάν δεν υπήρχε, τότε θα δημιουργηθεί ένα νέο αρχείο και θα πάρει πάλι τα περιεχόμενα του πρώτου.

Σημείωση: Στο linux η κατάληξη των αρχείων δεν έχει καμία απολύτως σημασία. Θα μπορούσε το αρχείο κείμενο.txt να είναι κάποιο βίντεο! Για λόγους ευκολίας όμως, συνήθως προσδίδονται οι κατάλληλες καταλήξεις στα ονόματα των αρχείων.

Εάν θέλουμε να αντιγράψουμε το κείμενο σε κάποιον άλλον φάκελο, γράφουμε:

cp κείμενο.txt ~/Documents/school/

1.4 Μετακίνηση αρχείων

Η επόμενη εντολή μας χρησιμεύει σε εργασίες μεταφοράς και μετονομασίας:

mv

Όπως και η “cp”, έτσι και αυτή δεν μπορεί να χρησιμοποιηθεί σκέτη. Θα πρέπει να πάρει λοιπόν πάλι δύο παραμέτρους, μία πηγής και μία προορισμού. Έτσι λοιπόν την χρησιμοποιούμε με αυτόν τον τρόπο:

mv κείμενο.txt λίστα_σουπερμάρκετ.txt

όπου μετονομάζουμε το αρχικό κείμενο σε κάτι πιο χρήσιμο. Εάν όμως υπήρχε και το αρχείο "λίστα_σουπερμάρκετ.txt” και σε αυτήν την περίπτωση το περιεχόμενο του θα διαγραφόταν και θα το αντικαταστούσε αυτό του "κείμενο.txt”. Για να μεταφέρουμε το κείμενο σε κάποιον άλλο φάκελο, για παράδειγμα στον γονεικό, δίνουμε την εντολή:

mv κείμενο.txt ../

ενώ αν θέλουμε δίνουμε και κάποια απόλυτη διαδρομή (σε κάποιο κρυφό φάκελο αν δεν θέλουμε να γίνεται εύκολα αντιληπτό αυτό το αρχείο):

mv κείμενο.txt /home/john/.my_secret_text_files/

1.5 Δημιουργία φακέλων

Θα πρέπει όμως να δημιουργήσουμε πρώτα αυτόν τον φάκελο. Πως γίνεται αυτό:

cd (για να επιστρέψουμε στον αρχικό μας φάκελο, εάν δεν βρισκόμαστε ήδη) mkdir .my_secret_text_files

Έστω ότι θέλουμε να δημιουργήσουμε τον φάκελο "/home/john/Music/discographies/MichaelJackson”, αλλά ο φάκελος “discographies” δεν υπάρχει. Σε αυτήν την περίπτωση θα πρέπει να χρησιμοποιήσουμε την παράμετρο "-p” (δημιούργησε τα **p**arents, εάν δεν υπάρχουν) του “mkdir":

mkdir -p /home/john/Music/discographies/MichaelJackson

1.6 Δημιουργία αρχείων

Εκτός από φακέλους θέλουμε φυσικά να δημιουργούμε και αρχεία. Ο πιο απλός τρόπος για να το πετύχουμε αυτό είναι με την εντολή touch η σύνταξη της οποίας είναι <pre<noinclude></noinclude>>touch filename</pre<noinclude></noinclude>> και η οποία δημιουργεί ένα κενό αρχείο με το όνομα filename. Το αρχείο αυτό δεν είναι αρχείο κειμένου. Θα μπορούσε να είναι οτιδήποτε, αλλά προς το παρόν είναι απλά empty. Παρακάτω θα του εισάγουμε κείμενο με έναν text editor και θα γίνει εν τέλει αρχείο κειμένου.

1.7 Διαγραφή αρχείων

Προειδοποίηση: Η επόμενη εντολή είναι από τις πιό επικίνδυνες που υπάρχουν. Ένα πολύ μικρό συντακτικό λάθος (ένα λάθος κενό ή μία τελεία) μπορεί να έχει καταστροφικές συνέπειες! Θα πρέπει να την χρησιμοποιείτε πολύ προσεκτικά!

Και αυτή δεν είναι άλλη από την

rm

(**rem**ove!) Η εντολή “rm” εξυπηρετεί την διαγραφή αρχείων και φακέλων.

Σημείωση: Υπάρχει και η εντολή “rmdir” για την διαγραφή φακέλων, αλλά επειδή μπορούμε να την χρησιμοποιήσουμε μόνο
σε κενούς φακέλους, σπάνια προτιμάται.

Για να διαγράψουμε ένα αρχείο εκτελούμε:

rm όνομα_αρχείου

Μπορούμε φυσικά να δώσουμε την πλήρη διαδρομή ενός αρχείου:

rm /home/john/Documents/κείμενο.txt

Αν θέλουμε να διαγράψουμε πολλά συγκεκριμένα αρχεία στον τρέχοντα φάκελο:

rm κείμενο.txt φωτογραφία.png κλπ

Παραπάνω αναφερθήκαμε στην διαγραφή κενών φακέλων. Σε περίπτωση που υπάρχει έστω και ένα αρχείο μέσα στον φάκελο που θέλουμε να διαγράψουμε, θα πρέπει να χρησιμοποιήσουμε την παράμετρο "-r” (**r**ecursive - αναδρομικά, δηλαδή ψάχνει σε υποφακέλους, σε υποφακέλους των υποφακέλων, μέχρι να "πιάσει πάτο" και διαγράφει τα πάντα μέσα σε αυτούς):

rm -r /home/john/Documents/school

:D Μέχρι τώρα είδαμε ότι πρέπει να δώσουμε τα συγκεκριμένα ονόματα των αρχείων και των φακέλων ως τιμή_παράμετρου στις διάφορες εντολές. Αυτό όμως δεν μας προσφέρει καθόλου ευελιξία. Γι'αυτό υπάρχουν και κάποια βοηθητικά σύμβολα (τα λεγόμενα Wildcards!), που μας βοηθάνε να εκτελούμε τις εντολές σε μικρές ή μεγάλες ομάδες αρχείων:

?

Αντικαθιστάται με έναν μόνο χαρακτήρα (αλλά όχι τον κενό χαρακτήρα!). Έστω ότι μέσα στον φάκελο υπάρχουν τα αρχεία:

κείμενο1.txt κείμενο4.txt κείμενο5.txt κείμενο.txt

Αν δώσουμε:

rm κείμενο?.txt

θα διαγράψει τα τρία πρώτα αρχεία, ενώ το "κείμενο.txt” θα παραμείνει εκεί.

*

Αντικαθιστάται με οποιοδήποτε πλήθος χαρακτήρων (μαζί με τον κενό!) Είναι το πιό συχνά χρησιμοποιούμενο σύμβολο. Αυτή τη φορά στο φάκελο έχουμε:

κείμενο1.txt κείμενο4.txt κείμενο5.txt κείμενο.txt φωτογραφία.png

Για να διαγράψουμε όλα τα αρχεία που τελειώνουν σε ".txt” δίνουμε:

rm *.txt

Για να διαγράψουμε όλα τα περιεχόμενα του φακέλου:

rm *

το οποίο μπορούμε να το γράψουμε και ως:

rm ./*

Παραπάνω Wildcards μπορείτε να βρείτε εδώ. Αν θέλουμε να εκτυπώσουμε τα περιεχόμενα ενός αρχείου κειμένου στα γρήγορα στο τερματικό πληκτολογούμε:

cat κείμενο.txt

(con**cat**enate) Η εντολή “cat” έχει πάρα πολλές δυνατότητες και η παραπάνω χρήση της είναι η απλούστερη μορφή. Αν θέλετε να δείτε τι παραπάνω μπορεί να κάνει αρκεί να δώσετε:

man cat

όπως άλλωστε και με τις περισσότερες εντολές. Αν θέλουμε να έχουμε την δυνατότητα να επεξεραστούμε το παραπάνω κείμενο, θα χρειαστεί να επιστρατεύσουμε κάποιον text editor και από τους πιο φιλικούς προς αρχάριους είναι ο:

nano

Αν τον δώσουμε σκέτο, δεν έχει επιλέξει κάποιο συγκεκριμένο αρχείο προς επεξεργασία και αυτό θα δημιουργηθεί/αποθηκευτεί κατά την έξοδο μας από τον editor. Εισάγουμε το κείμενο μας όπως στους περισσότερους editors που έχετε συνηθίσει εως τώρα. Όταν τελειώσουμε πατάμε “Ctrl+X” για να βγούμε από το πρόγραμμα. Σε αυτό το σημείο θα μας ρωτήσει εαν θέλουμε να αποθηκευτεί το αρχείο κειμένου. Πατάμε “y” ή “n” αντίστοιχα. Αν πατήσαμε “y” θα πρέπει να δώσουμε το όνομα με το οποίο θα αποθηκευτεί και πατάμε “Enter”. Για να ανοίξουμε ένα ήδη υπάρχον αρχείο (για παράδειγμα το αρχείο που δημιουργήσαμε παραπάνω με την εντολή “touch"!) προς επεξεργασία απλά δίνουμε:

nano αρχείο_κειμένου

και κατά την αποθήκευση το ίδιο όνομα έχει συμπληρωθεί από μόνο του και αν δεν θέλουμε να το αλλάξουμε απλά πατάμε πάλι “Enter”.

Σημείωση: Η χρήση του “touch” πριν το “nano” δεν είναι απαραίτητη. Το αρχείο θα δημιουργηθεί έτσι και αλλιώς από το “nano”.
Το “touch” χρησιμοποιείται συνήθως σε άλλες περιπτώσεις.

Στο κάτω μέρος του “nano” θα βρείτε τα δίαφορα χρήσιμα shortcuts που μας προσφέρει (όπου "^" πατάμε Ctrl!). Ένα ακόμη χρήσιμο shortcut είναι το “Ctrl+W” για αναζήτηση κειμένου. Πιο προχωρημένοι χρήστες προτιμούν πολύ πιο δυνατούς editors όπως ο “vim” ή ο “emacs”, αλλά δεν θεωρούνται καθόλου φιλικοί προς αρχάριους, όμως έρχονται με πάρα πολλές δυνατότητες. Εκτός από τα διάφορα tutorials και man pages που μπορείτε να αναζητήσετε μόνοι σας (δείτε σχετικά εδώ), υπάρχει και ένας διασκεδαστικός τρόπος να μάθετε τον “vim”, παίζοντας στον browser σας vim adventures!!

Οι παραπάνω τρόποι είναι για να επεξεργαστούμε τα κείμενα μας μόνο στο τερματικό. Πολλές φορές όμως οι χρήστες επιλέγουν να ανοίξουν τα αρχεία τους με κάποιον editor που έχει και GUI. Ο συνηθέστερος αυτών είναι ο “gedit”. Ανοίγετε το αρχείο σας:

gedit text_file

και όταν τελειώσετε, αποθηκεύετε το αρχείο σας (π.χ. με Ctrl+S ή από το αντίστοιχο button που προσφέρει ο gedit πάνω αριστερά) και όταν κλείστε το παράθυρο, επιστρέφετε πάλι στο τερματικό και συνεχίζετε από εκεί που ήσασταν. Όταν εκκινείτε από το τερματικό κάποιο πρόγραμμα με GUI, δεν μπορείτε να συνεχίσετε να χρησιμοποιείτε το τερματικό εως ότου κλείστε την εφαρμογή. Αυτό μπορούμε να το παρακάμψουμε εισάγοντας το σύμβολο "&" στο τέλος κάποιας εντολής:

gedit λίστα_σουπερμάρκετ &

Η τελευταία εντολή που θα μας απασχολήσει σε αυτόν τον εισαγωγικό οδηγό είναι η “echo”. Αυτή αναλαμβάνει να εκτυπώσει τα περιεχόμενα από κάποια πηγή που της δίνουμε στο τερματικό. Για παράδειγμα:

echo “Hello World!"

και στο τερματικό θα εκτυπωθεί φυσικά:

Hello World!

Το σύστημά μας χρησιμοποιεί και διάφορες μεταβλητές που είναι χρήσιμες για κάποιες εργασίες. Μία από τις πιό ενδιαφέρουσες είναι η μεταβλητή “PATH” η οποία περιέχει όλες τις διαδρομές οι οποίες περιέχουν εκτελέσιμα προγράμματα. Κάθε φορά που δίνουμε μία εντολή, το τερματικό ψάχνει σε όλους τους φακέλους του “PATH” και αν την βρεί, την εκτελεί. Για να δούμε τα περιεχόμενα του “PATH”, εκτελούμε:

echo $PATH

και στην οθόνη μας θα εμφανιστεί κάτι τέτοιο:

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:
/bin:/usr/games:/usr/local/games

το "$" υποδηλώνει ότι θα πρέπει να εκτυπωθούν τα περιεχόμενα του “PATH” (λόγω του ότι εδώ έχουμε μια μεταβλητή), επειδή χωρίς το "$" στην οθόνη μας θα εκτυπωνόταν απλά ένα:

PATH

Υπάρχουν προγράμματα που δεν είναι αποθηκευμένα σε κάποιον από τους φακέλους του PATH. Συνήθως επειδή μπορεί να έχουμε κατεβάσει κάποιο πρόγραμμα χειροκίνητα από κάποια ιστοσελίδα (το οποίο είναι σπάνιο - εάν υπάρχει δυνατότητα εγκατάστασης αυτού το προγράμματος μέσω κάποιου package manager (δείτε σχετικά εδώ) θα πρέπει να ακολουθείτε πάντα αυτόν τον τρόπο!). Για να εκτελέσουμε, λοιπόν ένα τέτοιο πρόγραμμα θα πρέπει να δώσουμε την πλήρη διαδρομή του ξεκινώντας από τον ριζικό φάκελο. Όμως ο πιο συνήθης τρόπος είναι ότι βρισκόμαστε ήδη μέσα στον φάκελο αυτού το προγράμματος και σε αυτήν την περίπτωση αρκεί να δώσουμε:

./όνομα_προγράμματος

Προσέξτε όμως ότι το αρχείο αυτό θα πρέπει να είναι εκτελέσιμο! (δείτε σχετικά εδώ). Επίσης δεν θα πρέπει να κατεβάζετε τέτοια προγράμματα από άγνωστες πηγές (δείτε γιατί!!).

Άλλες δύο χρήσιμες μεταβλητές είναι η $HOME που έχει το ίδιο περιεχόμενο με το σύμβολο "~" που είδαμε παραπάνω και η $EDITOR που ορίζει τον βασικό editor του συστήματός μας (πχ vim ή nano).

1.8 Εγκατάσταση προγραμμάτων

With great power comes great responsibility (όπως λένε και στο χωριό του Βαιμάκη). Μπορεί να φαίνεται ωραίο το να έχει κανείς πολλά δικαιώματα, αλλά από την άλλη πλευρά αυτό αποτελεί και μικρότερη ασφάλεια. Η μεγαλύτερη πηγή προβλημάτων στο linux είναι ο ίδιος ο χρήστης, όσο και αν αυτό μπορεί να σας φαίνεται παράξενο. Ίσως να έχετε ακούσει το ακρωνύμιο “PEBKAC” (Problem Exists Between Keyboard And Chair), ενώ ιοί και malwares έρχονται σε δεύτερη μοίρα. Για αυτό, οι περισσότερες διανομές δεν σας συνδέουν με τον υπερχρήστη, αλλά δημιουργούν έναν δεύτερο χρήστη με περιορισμένα δικαιώματα. Βέβαια, πολλές φορές θα πρέπει να εκτελέσετε κάποιες εντολές με δικαιώματα υπερχρήστη, αλλά αυτό δεν σημαίνει ότι θα πρέπει να συνδεθείτε με τον λογαριασμό αυτόν. Συνήθως τέτοια δικαιώματα απαιτούνται από εντολές που μπορούν να έχουν σημαντικές επιπτώσεις στο σύστημά σας.

Ένα καλό παράδειγμα είναι η χρήση του package manager της διανομής σας. Καθώς η (απ)εγκατάσταση/αναβάθμιση πακέτων είναι ζωτικής σημασίας, οι αντίστοιχες έντολες έχουν προστατευτεί ώστε να μην μπορούν να εκτελεστούν από τον απλό χρήστη. Όμως υπάρχει η δυνατότητα από τον απλό χρήστη να εκτελέσει μία και μόνο εντολή προσωρινά με δικαιώματα υπερχρήστη, τοποθετώντας την λέξη “sudo” πριν από την εντολή

που θα ήθελε να εκτελέσει. Έτσι, αν δίναμε:

apt-get install gimp

το τερματικό θα μας ειδοποιούσε ότι δεν έχουμε τα κατάλληλα δικαιώματα για να την εκτελέσουμε. Αντ' αυτού θα πρέπει να δώσουμε:

sudo apt-get install gimp

Στο σημείο αυτό θα ζητηθεί ο κωδικός του χρήστη μας, και μόλις τελειώσει δεν θα έχουμε πλέον δικαιώματα υπερχρήστη. Πολλές φορές ίσως να ξεχάσετε το sudo και αντί να ξαναγράψετε το παραπάνω, ή να "παίζετε" με βελάκια, υπάρχει ένας πιο σύντομος τρόπος:

sudo !!

όπου το "!!" αντικαθιστά την προηγούμενη εντολή, στην περίπτωσή μας το “apt-get install gimp”.

Τέλος, μία λίστα με χρήσιμες εντολές με σύντομη περιγραφή. Για παραπάνω πληροφορίες σχετικά με αυτές θα πρέπει να συμβουλευτείτε είτε τις man pages τους ή online tutorials κλπ. Καλό θα ήταν να μάθετε καλά τις επιλογές που σας προσφέρει ο package manager της διανομής σας (περισσότερες πληροφορίες εδώ):

2 Διάφορες εντολές

2.1 Διαχειριστές πακέτων

2.2 Περιήγηση στους φακέλους / Εκτύπωση πληροφοριών / Επεξεργασία αρχείων

2.3 Λογαριασμοί χρηστών / Δικαιώματα

2.4 Σύστημα

2.5 Αναζήτηση

2.6 Δίκτυα

2.7 Λοιπές

2.8 Χρήσιμα tips για το τερματικό

Εάν κάποια εντολή θέλει πολλή ώρα για να ολοκληρωθεί και θέλετε για κάποιο λόγω να την διακόψετε οριστικά πατάτε τον συνδυασμό “Ctrl+C”.

Για αντιγραφή και επικόλληση κειμένου μέσα στο τερματικό χρησιμοποιείτε τους συνδυασμούς “Shift+Ctrl+C” και “Shift+Ctrl+V”. Ειδάλως, μπορείτε να χρησιμοποιήσετε και το ποντίκι με το δεξί κλίκ.

Εάν έχετε ταυτόχρονα ανοιχτό και έναν file manager μπορείτε μέσω drag'n'drop να τραβήξετε μέσα στο τερματικό κάποιον φάκελο ή αρχείο και στο τερματικό θα εμφανιστεί αυτομάτως η πλήρης διαδρομή αυτού.

Συχνά μπορεί να ξεχάσετε το “sudo” μπροστά από μια εντολή. Ένας γρήγορος τρόπος για να επαναλάβετε την προηγούμενη εντολή με “sudo” από μπροστά είναι απλά η ολόκληρη εντολή “sudo !!", όπου το "!!" αντικαθιστά την προηγούμενη εντολή.

Ο συνδυασμός “Alt+.” μας δίνει την τελευταία παράμετρο από την προηγούμενη εντολή. Αν δώσαμε π.χ. “cat text.txt” και θέλουμε να επεξεργαστούμε αυτό το αρχείο με μία γρήγορη εντολή αρκεί να δώσουμε “nano” ακολουθούμενο από αυτόν τον συνδυασμό.

2.9 Επικίνδυνες εντολές

Η κονσόλα είναι δύναμη.

Η κονσόλα/τερματικό είναι ένα πολύ ισχυρό εργαλείο. Μπορεί να λύσει τα χέρια, αλλά μπορεί και να καταστρέψει το σύστημα μας. Αυτό δεν είναι χαρακτηριστικό του Linux. Ακριβώς το ίδιο ισχύει σε όλα τα λειτουργικά συστήματα. Ακολούθουν μερικές εντολές οι οποίες μπορούν να καταστρέψουν το σύστημα σας. Φροντίστε να μην τις δώσετε, εκτός και αν έχετε ένα καλό λόγο να το κάνετε. Μη φοβάστε πάντως, οι περισσότερες από αυτές απαιτούν να δοθούν με δικαιώματα υπερχρήστη (root) οπότε είναι δύσκολο να τις γράψετε κατά λάθος. Το σύστημα permissions του Linux σας προστατεύει.

Αυτές είναι κάποιες από τις εντολές που μπορούν να κάνουν ζημιά στο σύστημα σας. Σαφώς και υπάρχουν και άλλες (φαντασία να έχει κανείς να βρίσκει τρόπους να κάνει ζημιά...), μα αυτές είναι οι πιο επιζήμιες.

Έβαλα τις εντολές σε έναν πίνακα. Αν ξέχασα καμία συμπληρώστε την. Το αν χρειάζονται δικαιώματα root η όχι ίσως να μπορεί να γίνει ξεχωριστή στήλη. Στο σχολιασμό μπορεί να λέω και βλακείες.))--Pmav99 (συζήτηση) 13:18, 12 Ιανουαρίου 2013 (UTC)

wget http://some_untrusted_source O | sh

Με αυτό κατεβάζουμε (η εντολη wget) διάφορα από το διαδίκτυο. Να ΜΗΝ το κάνετε αν η σελιδα που παραπέμπει είναι άγνωστη η τέλος πάντων, αν το βρήκατε σε καποια ιστοσελίδα που δεν γνωρίζετε και δεν έχει καποιο κύρος.

: (){ :|:& };:

Το… χαιδευτικό αυτής της εντολής είναι: fork bomb και λέει στο σύστημα σας να κάνει τόσες διεργασίες που και το CERN θα χάζευε. Το σύστημα παθαίνει κοκομπλόκο, χάνετε δεδομενα.

rm /bin/init cd / ; find -iname init -exec rm -rf {} \;

Διαγράφει ζωτικά αρχεία και συγκεκριμένα τα έχοντα τη κατάληξη: init, όπου φυσικά ανάμεσα τους και τα /sbin/init.

3 Δείτε επίσης

- http://www.labtestproject.com/

4 Text and image sources, contributors, and licenses

4.1 Text

- **Linux για αρχάριους/Βασικές γνώσεις τερματικού** *Πηγή:* http://el.wikibooks.org/wiki/Linux%20%CE%B3%CE%B9%CE%B1%20%CE%B1%CF%81%CF%87%CE%AC%CF%81%CE%B9%CE%BF%CF%85%CF%82/%CE%92%CE%B1%CF%83%CE%B9%CE%BA%CE%AD%CF%82%20%CE%B3%CE%BD%CF%8E%CF%83%CE%B5%CE%B9%CF%82%20%CF%84%CE%B5%CF%81%CE%BC%CE%B1%CF%84%CE%B9%CE%BA%CE%BF%CF%8D?oldid=24108 *Συνεισφέροντες:* Pmav99, Tr3quart1sta και Ανώνυμες συνεισφορές: 1

4.2 Images

4.3 Content license