

Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα (Multimedia)



www.settemuse.it

Jean Léon Gerome: Pyrrhic dance

Λειτουργικό σύστημα Unix

v1.0

Σωκράτης Λεβέντης - Διονύσης Πυριόχος



Για να μάθετε για τις πηγές πληροφορίας και να ενημερωθείτε για τον τρόπο χρήσης, μπορείτε να δείτε το αντίστοιχο τμήμα των σημειώσεων. Με μία λέξη, οι σημειώσεις αυτές είναι CC BY-NC-SA 2.0



UNIX PEOPLE ARE HAPPY

Περιεχόμενα

Τι είναι το Unix;.....	3	Λίστα Διεργασιών που Έχουν Διακοπεί ή Εκτελούνται στο Παρασκήνιο.....	27
Τα Είδη του UNIX.....	3	Τερματίζοντας μια Διεργασία.....	27
Το Λειτουργικό Σύστημα UNIX.....	3	Εντολές.....	28
Αρχεία και Διεργασίες.....	4	Διάφορα (και όχι αδιάφορα).....	30
Δομή Καταλόγων.....	5	Άλλες Χρήσιμες Εντολές του UNIX	30
Ανοίγουμε ένα τερματικό του UNIX.....	5	Παροχέτευση (spooling).....	32
Εμφάνιση Αρχείων και Καταλόγων.....	7	Προγραμματισμός κελύφους.....	34
Δημιουργία Καταλόγων.....	8	Μεταβλητές στο UNIX.....	34
Αλλαγή σε Διαφορετικό Κατάλογο.....	8	Μεταβλητές Περιβάλλοντος.....	34
Οι Κατάλογοι . και	9	Μεταβλητές Κελύφους.....	34
Διαδρομές.....	9	Πως χρησιμοποιούμε και πως θέτουμε Μεταβλητές.....	35
Επιπλέον πληροφορίες για Καταλόγους και Διαδρομές.....	10	Μεταβλητές Περιβάλλοντος στο Αρχείο .cshrc.....	36
Εντολές.....	11	Ρύθμιση του path.....	36
Αρχεία.....	12	Σύνδεσμοι.....	38
Αντιγραφή Αρχείων.....	12	Επεξεργασία για σκληροπυρηνικούς - vi..	39
Μεταφορά και Μετονομασία Αρχείων..	12	Φορτώστε.....	39
Διαγραφή Αρχείων και Καταλόγων.....	13	Κατάσταση εντολών και Κατάσταση εισαγωγής.....	39
Εμφάνιση Περιεχομένων Αρχείου στην Οθόνη.....	13	Έξω τώρα!.....	40
Αναζήτηση στα Περιεχόμενα Ενός Αρχείου	14	Μετακίνηση.....	41
Εντολές.....	15	Μετακίνηση.....	41
Ανακατεύθυνση / Φίλτρα / Σωλήνωση....	17	Μαζική εκτέλεση εντολής.....	42
Ανακατεύθυνση της Εξόδου.....	17	Διαγραφή κειμένου.....	42
Ανακατεύθυνση της Εισόδου	18	Αναζήτηση.....	42
Φίλτρα.....	19	Αντικατάσταση.....	42
Σωληνώσεις.....	19	Αντιγραφή και επικόλληση.....	43
Εντολές.....	20	Γραφικό περιβάλλον.....	44
Χαρακτήρες Μπαλαντέρ - Βοήθεια.....	21	Κανονισμός Κατάρτισης Ειδικότητας.....	45
Χαρακτήρες Μπαλαντέρ.....	21	Ανάλυση Επαγγελματικών Δραστηριοτήτων.....	45
Συμβάσεις στα Ονόματα Αρχείων.....	21	Τελικές Εξετάσεις Πιστοποίησης.....	45
Αναζήτηση στη Βοήθεια.....	22	Κατάλογος ερωτήσεων.....	46
Εντολές.....	22	Η μεγάλη εικόνα.....	46
Ασφάλεια.....	23	Ερωτήσεις πιστοποίησης 2008.....	46
Χρήστες και ομάδες.....	23	Εκδόσεις - Διάθεση.....	48
Ασφάλεια Συστήματος Αρχείων (Δικαιώματα Πρόσβασης).....	23	Διάθεση και Δικαιώματα.....	48
Αλλαγή Δικαιωμάτων Πρόσβασης.....	24	Εκδόσεις.....	48
Διεργασίες.....	26		
Διεργασίες και Εργασίες.....	26		

Τι είναι το Unix;

Το UNIX είναι ένα λειτουργικό σύστημα που ξεκίνησε να αναπτύσσεται τη δεκαετία του 1960 και βρίσκεται από τότε σε διαρκή ανάπτυξη. Η έννοια του **Λειτουργικού Συστήματος** (operating system) αναφέρεται σε ένα αριθμό προγραμμάτων που κάνει έναν υπολογιστή να λειτουργεί¹. Το UNIX είναι ένα δοκιμασμένο σύστημα το οποίο έχει την ικανότητα να εκτελεί πολλά προγράμματα ταυτόχρονα (**πολυδιεργαστικό / multitasking**) και να εξυπηρετεί πολλούς χρήστες (**πολυχρηστικό / multiuser**) την ίδια στιγμή. Είναι κατάλληλο για εξυπηρετητές (servers), μηχανήματα γραφείου (desktops), σταθμούς εργασίας (workstations), φορητούς υπολογιστές και τηλέφωνα - για τα πάντα, με λίγα λόγια!



Τα συστήματα UNIX που απευθύνονται σε μηχανήματα γραφείου διαθέτουν γραφικό περιβάλλον εργασίας (GUI), παρόμοιο με αυτό που χρησιμοποιείται στα Microsoft Windows, πλήρως εξοπλισμένο με κουμπάκια, μενού, παράθυρα κι όλα τα συναφή. Έτσι μας παρέχει ένα εύκολο τρόπο εργασίας. Ωστόσο η πραγματική δύναμη του Unix βρίσκεται στη γραμμή εντολών. Από τη γραμμή εντολών μπορούν να τρέξουν εντολές διαχείρισης που δεν βρίσκονται σε κανένα γραφικό περιβάλλον, μπορεί ο προχωρημένος χρήστης να δημιουργήσει τα δικά του προγράμματα, μπορεί ο διαχειριστής να έχει πλήρη πρόσβαση σε απομακρυσμένα συστήματα.

Τα Είδη του UNIX



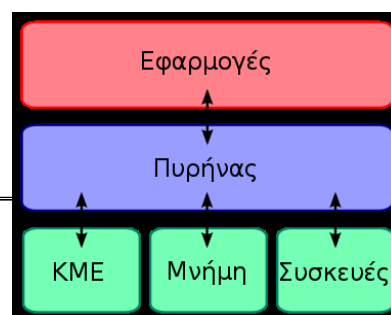
Υπάρχουν αρκετές διαφορετικές εκδόσεις του UNIX, αν και οι περισσότερες έχουν αρκετές ομοιότητες. Οι πιο συνηθισμένες σήμερα είναι το GNU/Linux, διάφορες εκδόσεις BSD, το MacOS X, το Sun Solaris, το Android και το iOS. Αυτές είναι εκδόσεις που απευθύνονται σε κάθε τύπο υπολογιστικής μηχανής, κάθε μεγέθους και απαίτησης.

Το Λειτουργικό Σύστημα UNIX

Το λειτουργικό σύστημα UNIX αποτελείται από τρία βασικά τμήματα: Τον πυρήνα, το κέλυφος και τα προγράμματα (εφαρμογές).

Ο Πυρήνας

Ο **πυρήνας** του UNIX είναι το κέντρο του λειτουργικού συστήματος. Αλληλεπιδρά με το υλικό ώστε να διαχειρίζεται τους πόρους του συστήματος και να γίνεται σωστή κατανομή (μοίρασμα) στις εφαρμογές.



Διαθέτει χρόνο και μνήμη στα προγράμματα και χειρίζεται το σύστημα αρχείων και τις επικοινωνίες, ανταποκρινόμενο στις κλήσεις συστήματος.

Ο πυρήνας είναι στενά δεμένος με την αρχιτεκτονική στην οποία απευθύνεται. Έτσι έχουμε άλλο πυρήνα για μηχανήματα i386, άλλο για x64, άλλο για Atom, άλλο για κοκ.

¹ Αυτός είναι ένας σύντομος ορισμός του ΛΣ και δεν είναι βέβαια πλήρης.

Το Κέλυφος

Το **κέλυφος (shell)** είναι μία εφαρμογή που ενεργεί ως διεπαφή (ενδιάμεσος) μεταξύ του χρήστη και του πυρήνα. Όταν ο χρήστης εισέλθει στο σύστημα, το πρόγραμμα που είναι υπεύθυνο για την αναγνώριση του (το login) ελέγχει το όνομα χρήστη και τον κωδικό του και κατόπιν ξεκινάει το κέλυφος.



Το κέλυφος είναι ένα **πρόγραμμα ερμηνείας εντολών** (command line interpreter ή CLI). Ερμηνεύει τις εντολές που γράφει ο χρήστης και κανονίζει για την εκτέλεση τους. Οι εντολές του χρήστη είναι και αυτές προγράμματα: όταν τερματίζουν, το κέλυφος δείχνει ξανά την **προτροπή** (% ή \$) στο χρήστη δείχνοντας ότι μπορεί να συνεχίσει με την εκτέλεση άλλων εντολών.

Ως παράδειγμα του τρόπου με τον οποίο συνεργάζονται το κέλυφος και ο πυρήνας, ας υποθέσουμε ότι ένας χρήστης γράφει την εντολή `rm myfile` (η οποία έχει ως αποτέλεσμα τη διαγραφή του αρχείου `myfile`). Το κέλυφος ψάχνει στο σύστημα αρχείων για το αρχείο που περιέχει το πρόγραμμα `rm` και έπειτα ζητάει από τον πυρήνα, μέσω κλήσεων συστήματος να εκτελέσει το `rm` στο `myfile`. Όταν ολοκληρωθεί η διεργασία `rm myfile` το κέλυφος επιστρέφει την προτροπή % στο χρήστη, δείχνοντας του έτσι ότι είναι έτοιμο να δεχθεί άλλες εντολές.

Αφού το κέλυφος είναι απλά μία εφαρμογή, μπορείτε να φανταστείτε ότι έχουν φτιαχτεί πολλά και διάφορα κελύφη. Υπάρχει η δυνατότητα χρήσης διαφορετικού κελύφους ακόμα και σε ένα μόνο μηχάνημα. Το συνηθέστερο κέλυφος που χρησιμοποιούμε από προεπιλογή είναι το `bash`.



Το `bash` έχει κάποια χαρακτηριστικά που βοηθούν το χρήστη όταν εισάγει εντολές:

- **Αυτόματη συμπλήρωση ονόματος αρχείου** - Πληκτρολογώντας μόνο ένα μέρος μιας εντολής ή ενός ονόματος αρχείου και πιέζοντας το Tab, το κέλυφος θα συμπληρώσει αυτόματα το υπόλοιπο μέρος. Αν το κέλυφος βρει περισσότερα από ένα ονόματα που να ταιριάζουν με τα γράμματα που έχουν πληκτρολογηθεί, θα ακουστεί ένας χαρακτηριστικός ήχος (μπιπ) που σημαίνει ότι πρέπει να γράψετε ακόμα μερικά γράμματα και να πιέσετε ξανά το tab.
- **Ιστορικό** - Το κέλυφος διατηρεί μια λίστα των εντολών που έχετε πληκτρολογήσει. Αν πρέπει να επαναλάβετε μια εντολή μπορείτε να χρησιμοποιήσετε τα πάνω και κάτω βελάκια στο πληκτρολόγιο για να δείτε και να εκτελέσετε ξανά τις προηγούμενες εντολές. Μπορείτε επίσης να γράψετε την εντολή [history](#) για να δείτε μια λίστα με τις προηγούμενες εντολές σας.

Όποιος χρήστης διαθέτει ικανές γνώσεις μπορεί να προσαρμόσει τη συμπεριφορά του κελύφους του.

Αρχεία και Διεργασίες

Στο UNIX τα πάντα είναι είτε αρχείο είτε διεργασία.

Μια **διεργασία (process)** είναι ένα πρόγραμμα που εκτελείται και μπορεί να αναγνωριστεί από ένα μοναδικό αριθμό διεργασίας, PID (process identifier).

Ένα **αρχείο (file)** είναι μια συλλογή δεδομένων. Τα αρχεία δημιουργούνται από τους χρήστες με την βοήθεια συντακτών κειμένου, μεταγλωττιστών και άλλων εφαρμογών.

Παραδείγματα αρχείων:

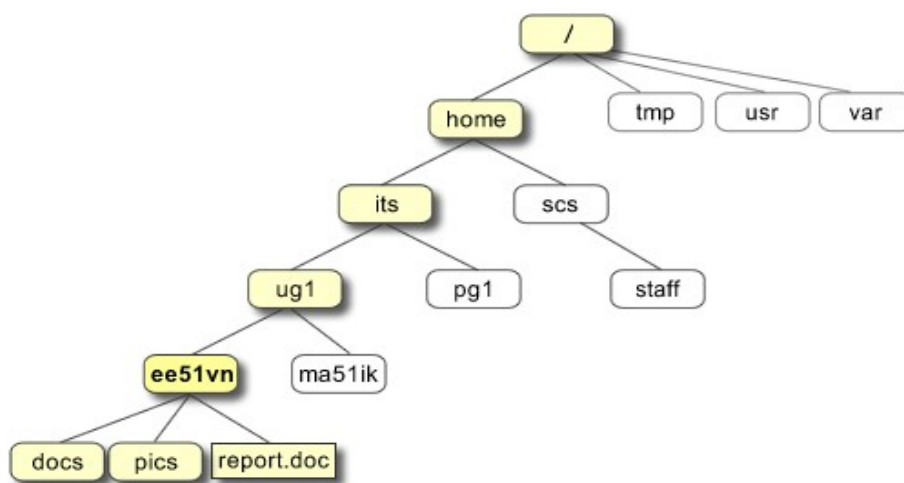
- Ένα έγγραφο (αναφορά, άρθρο κ.λ.π.)

- Το κείμενο ενός προγράμματος όπως έχει γραφεί σε μια γλώσσα προγραμματισμού υψηλού επιπέδου
- Οδηγίες που είναι απευθείας κατανοητές από το μηχάνημα αλλά ακατανόητες από ένα φυσιολογικό χρήστη. Για παράδειγμα μια συλλογή από δυαδικά ψηφία (ένα εκτελέσιμο ή δυαδικό αρχείο).
- Ένας κατάλογος που περιέχει πληροφορίες σχετικά με τα περιεχόμενα του. Τα περιεχόμενα μπορεί να είναι ένα μείγμα από αρχεία και άλλους καταλόγους (υποκαταλόγους).

Δομή Καταλόγων

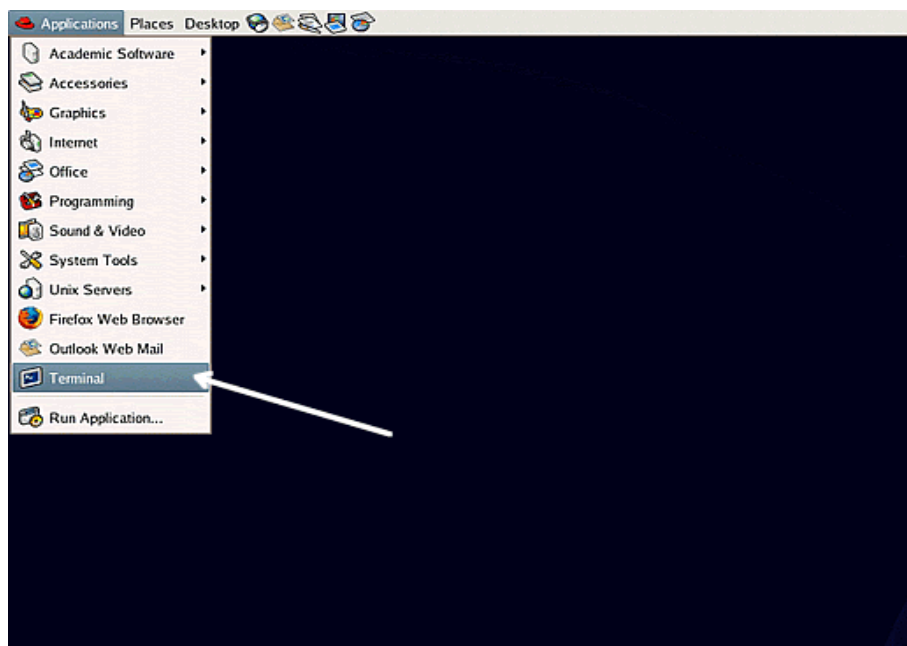
Τα αρχεία ομαδοποιούνται σε μια **δομή καταλόγων (directories)**. Το σύστημα αρχείων έχει ιεραρχική δομή, όπως ένα ανάστροφο δέντρο. Η κορυφή της ιεραρχίας κατά παράδοση ονομάζεται **root** (και γράφεται ως μια κάθετος /)

Στο διάγραμμα, βλέπουμε ότι ο προσωπικός κατάλογος (home) του προπτυχιακού φοιτητή "ee51vn" περιέχει δύο υποκαταλόγους (docs και pics) και ένα αρχείο, το report.doc.



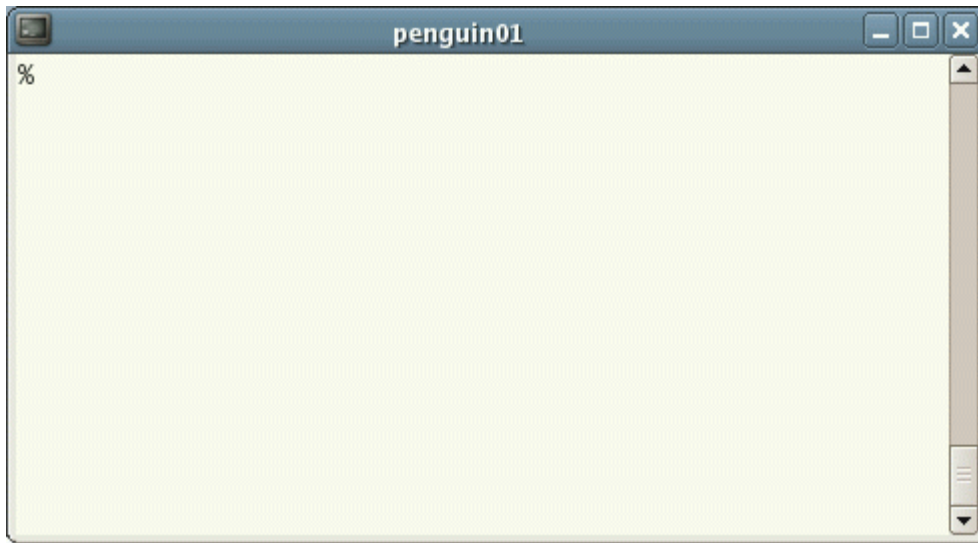
Η πλήρης **διαδρομή (path)** για το αρχείο report.doc είναι `"/home/its/ug1/ee51vn/report.doc"`

Ανοίγουμε ένα τερματικό του UNIX



Για να ανοίξετε ένα **τερματικό (terminal)** του UNIX, επιλέξτε το εικονίδιο "Terminal" στο μενού Applications/Accessories (Η ακριβής τοποθεσία μπορεί να είναι διαφορετική ανάλογα με το γραφικό περιβάλλον που χρησιμοποιείτε).

Θα εμφανιστεί ένα παράθυρο τερματικού με μια **προτροπή (prompt)**. Η προτροπή μπορεί να είναι κάτι απλό όπως % ή \$ ή και κάτι αρκετά πιο πολύπλοκο (μπορεί να αλλάξει). Τώρα, μπορείτε να αρχίσετε να γράφετε εντολές.



Εμφάνιση Αρχείων και Καταλόγων

ls (list)

Μόλις εισέλθετε στο σύστημα, βρισκόσαστε μέσα στον **προσωπικό σας κατάλογο**, γνωστό και ως **home directory**. Ο προσωπικός σας κατάλογος έχει το ίδιο όνομα με το όνομα χρήστη που έχετε, για παράδειγμα ee91ab ή mitsaras24, και σε αυτόν αποθηκεύονται τα αρχεία και οι υποκατάλογοι που δημιουργείτε.

Για να δείτε τι υπάρχει στον προσωπικό σας κατάλογο, πληκτρολογήστε:

```
% ls
```

Η εντολή `ls` δείχνει μια λίστα των περιεχομένων του τρέχοντος καταλόγου.



Ανάλογα με το πως κάνει εγκατάσταση η διανομή που βλέπετε, ίσως να μην υπάρχουν ορατά αρχεία στον κατάλογο σας, και σε αυτή την περίπτωση θα δείτε απλώς ξανά την προτροπή του κελύφους σας.

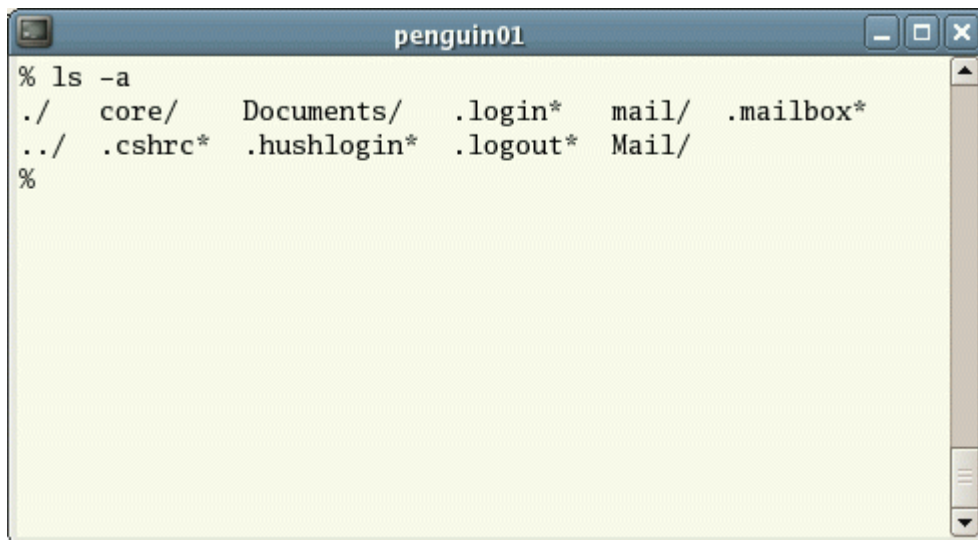
Η `ls` στην πραγματικότητα δεν δείχνει όλα τα αρχεία του καταλόγου σας, αλλά μόνο αυτά που δεν είναι κρυφά. Στο unix, αν θέλουμε ένα αρχείο να είναι **κρυφό** αρχείο, αρχίζουμε το όνομά του με τελεία (`.`), για παράδειγμα `.thisishidden`. Τα κρυφά αρχεία δε φανερώνονται γιατί συνήθως περιέχουν σημαντικές ρυθμίσεις προγραμμάτων και, συνεπώς, δεν πρέπει να τα αλλάζετε, εκτός αν είστε εξοικειωμένος με το UNIX!!!

Για να δείτε όλα τα αρχεία στον κατάλογο σας, συμπεριλαμβανομένων και αυτών που ξεκινάνε με την τελεία, πληκτρολογήστε:

```
% ls -a
```

Όπως μπορείτε να δείτε, το `ls -a` δείχνει και τα αρχεία που κανονικά είναι κρυφά.

Η εντολή `ls` είναι ένα καλό παράδειγμα εντολής που μπορεί να δεχθεί προαιρετικά ορίσματα: το `-a` είναι ένα τέτοιο παράδειγμα. Τα **προαιρετικά ορίσματα ή παράμετροι** αλλάζουν τη συμπεριφορά μιας εντολής. Το UNIX διαθέτει ένα ενσωματωμένο σύστημα βοήθειας το οποίο μπορεί να σας πει ποιες παραμέτρους δέχεται κάθε εντολή και πως κάθε μια από αυτές μεταβάλλουν τη συμπεριφορά της εντολής (θα το δούμε αργότερα στο μάθημα).



```
penguin01
% ls -a
./  core/  Documents/  .login*  mail/  .mailbox*
../  .cshrc*  .hushlogin*  .logout*  Mail/
%
```

Στη συντριπτική πλειοψηφία των εντολών, για να δείτε πως συντάσσεται μία εντολή, δίνετε την παράμετρο `--help` (για παράδειγμα `ls --help`). Θα εμφανιστεί η λίστα με τις παραμέτρους και μία σύντομη εξήγηση για την καθεμία.

Δημιουργία Καταλόγων

mkdir (make directory)

Θα δημιουργήσουμε τώρα ένα υποκατάλογο μέσα στον προσωπικό σας κατάλογο, για να κρατάτε τα αρχεία τα οποία θα δημιουργήσετε κατά τη διάρκεια αυτών των μαθημάτων. Για να δημιουργήσετε ένα υποκατάλογο με το όνομα `unixstuff` στον τρέχοντα κατάλογο, πληκτρολογήστε:

```
% mkdir unixstuff
```

Για να δείτε τον κατάλογο που μόλις δημιουργήσατε, πληκτρολογήστε:

```
% ls
```

Αλλαγή σε Διαφορετικό Κατάλογο

cd (change directory)

Η εντολή [cd directory](#) αλλάζει τον τρέχοντα κατάλογο σε αυτόν που αναφέρεται ως 'directory'. Μπορείτε να σκεφτείτε τον τρέχοντα κατάλογο ως αυτόν που βρίσκεστε τη δεδομένη στιγμή (τη θέση σας δηλ. μέσα στο δέντρο του συστήματος αρχείων που είδαμε προηγουμένως).

Για να μετακινηθείτε στον κατάλογο που μόλις δημιουργήσατε, γράψτε:

```
% cd unixstuff
```

Γράψτε `ls` για να δείτε τα περιεχόμενα του²

Μόνοι σας

Φτιάξτε ένα άλλο κατάλογο μέσα στον `unixstuff`, με το όνομα `backups`. Διαπιστώστε ότι όντως τον

² δε θα δείτε τίποτα αφού είναι άδειος!!!

φτιάξατε!

Οι Κατάλογοι . και ..

Ενώ βρίσκεστε στον κατάλογο `unixstuff` πληκτρολογήστε:

```
% ls -a
```

Όπως μπορείτε να δείτε, στον κατάλογο `unixstuff` (αλλά και σε όλους τους άλλους καταλόγους), υπάρχουν δύο ειδικοί κατάλογοι με τις ονομασίες `(.)` και `(..)`

Ο Τρέχων Κατάλογος (.)

Στο UNIX, η `(.)` σημαίνει τον **τρέχοντα κατάλογο**, έτσι αν πληκτρολογήσετε (`!!!` υπάρχει ένα κενό μεταξύ της εντολής `cd` και της τελείας)

```
% cd .
```

σημαίνει ότι θα μείνετε στον κατάλογο που είστε (στον `unixstuff`).

Αρχικά αυτό μάλλον δεν θα σας φανεί χρήσιμο, αλλά θα δείτε αργότερα ότι χρησιμοποιώντας το `(.)` ως το όνομα του τρέχοντος καταλόγου, θα σας γλυτώσει από άσκοπη πληκτρολόγηση.

Ο Γονικός Κατάλογος (..)

Οι δύο τελείες `(..)` σημαίνουν το **γονικό** του τρέχοντος καταλόγου. Η εντολή:

```
% cd ..
```

θα σας πάει ένα κατάλογο επάνω στην ιεραρχία (στη συγκεκριμένη περίπτωση, στον προσωπικό σας κατάλογο). Δοκιμάστε το τώρα.

Αν πληκτρολογήσετε `cd` χωρίς κανένα όρισμα, θα γυρίσετε πάντα στον προσωπικό σας κατάλογο. Αυτό είναι χρήσιμο αν έχετε χαθεί μέσα στο σύστημα αρχείων.

Διαδρομές

`pwd` (print working directory)

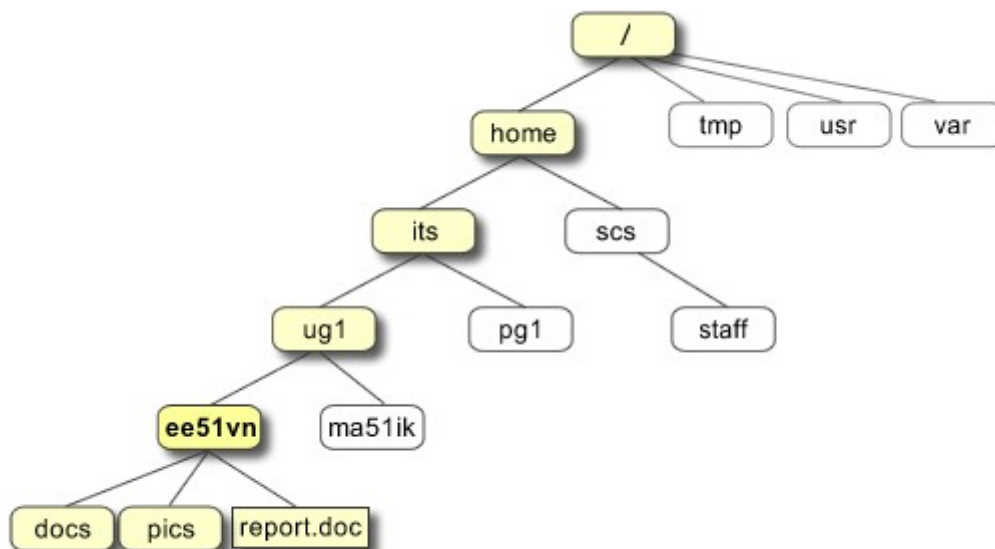
Οι **διαδρομές (paths)** σας επιτρέπουν να ανακαλύψετε που βρίσκεστε σε σχέση με όλο το σύστημα αρχείων. Για παράδειγμα, για να βρείτε την απόλυτη διαδρομή του προσωπικού σας καταλόγου, γράψτε `cd` για να επιστρέψετε σε αυτόν, και έπειτα πληκτρολογήστε:

```
% pwd
```

Η πλήρης διαδρομή θα είναι κάτι σαν το παρακάτω:

```
/home/its/ug1/ee51vn
```

το οποίο σημαίνει ότι ο `ee51vn` (ο προσωπικός σας κατάλογος) βρίσκεται μέσα στον υποκατάλογο `ug1` (τον κατάλογο της ομάδας), ο οποίος με τη σειρά του βρίσκεται μέσα στον υποκατάλογο `its` που βρίσκεται στον υποκατάλογο `home` ο οποίος βρίσκεται μέσα στο ριζικό κατάλογο, γνωστό ως `" / "`.



Μόνοι σας

Χρησιμοποιήστε τις εντολές `cd`, `ls` and `pwd` για να εξερευνήσετε το σύστημα αρχείων.

(Θυμηθείτε ότι αν καθείτε, μπορείτε να χρησιμοποιήσετε την εντολή `cd` χωρίς ορίσματα για να επιστρέψετε άμεσα στον προσωπικό σας κατάλογο).

Ποιοί άλλοι χρήστες είναι σημμένοι στο μηχάνημα που βρισκόσαστε; Τι καταλόγους έχουν στους προσωπικούς τους καταλόγους;

Επιπλέον πληροφορίες για Καταλόγους και Διαδρομές

Κατανόηση των Διαδρομών

Πληκτρολογήστε πρώτα `cd` για να επιστρέψετε στον προσωπικό σας κατάλογο, και έπειτα:

```
% ls unixstuff
```

για να δείτε τα περιεχόμενα του καταλόγου `unixstuff`

Πληκτρολογήστε τώρα:

```
% ls backups
```

Θα δείτε ένα μήνυμα όπως το παρακάτω:

```
backups: No such file or directory
```

Ο λόγος που βλέπετε αυτό το μήνυμα, είναι ότι το `backups` δεν βρίσκεται στον τρέχον κατάλογο σας. Για να χρησιμοποιήσετε μια εντολή σε ένα αρχείο (ή κατάλογο) που δεν βρίσκεται στον τρέχοντα κατάλογο (σε αυτόν που είστε μέσα), θα πρέπει είτε να χρησιμοποιήσετε την εντολή `cd` για να μετακινηθείτε στο σωστό κατάλογο, ή να καθορίσετε την πλήρη διαδρομή του. Για να δείτε τα περιεχόμενα του καταλόγου `backups`, θα πρέπει να γράψετε:

```
% ls unixstuff/backups
```

~ (ο προσωπικός σας κατάλογος)

Μπορείτε να χρησιμοποιήσετε τον χαρακτήρα ~ για να αναφερθείτε στον προσωπικό σας κατάλογο. Μπορείτε να χρησιμοποιήσετε αυτό το χαρακτήρα για να προσδιορίσετε διαδρομές που ξεκινούν από τον κατάλογο σας. Αν πληκτρολογήσετε:

```
% ls ~/unixstuff
```

θα δείτε τα περιεχόμενα του καταλόγου σας unixstuff, άσχετο με το που βρίσκεστε τη δεδομένη στιγμή μέσα στο σύστημα αρχείων.

Μόνοι σας

Τι πιστεύετε ότι θα δείξει η εντολή:

```
% ls ~
```

Τι πιστεύετε ότι θα δείξει η εντολή:

```
% ls ~/.
```

Εντολές

Εντολή	Σημασία
ls	Δείχνει αρχεία και καταλόγους
ls -a	Δείχνει όλα τα αρχεία και τους καταλόγους
mkdir	Δημιουργεί ένα κατάλογο
cd directory	Σας μεταφέρει στον κατάλογο που αναγράφεται
cd	Σας μεταφέρει απευθείας στον προσωπικό σας κατάλογο
cd ~	Σας μεταφέρει στον προσωπικό σας κατάλογο
cd ..	Σας μεταφέρει στον προηγούμενο στην ιεραρχία κατάλογο (γονικό)
pwd	Δείχνει την πλήρη διαδρομή του τρέχοντος καταλόγου

Αρχεία

Αντιγραφή Αρχείων

cp (copy)

Η εντολή `cp file1 file2` δημιουργεί ένα αντίγραφο του file1 στον τρέχοντα κατάλογο και το ονομάζει file2.

Τώρα, θα αντιγράψουμε ένα αρχείο που υπάρχει σε κάποιο σημείο του συστήματος αρχείων (και στο οποίο έχουμε πρόσβαση) μέσα στον κατάλογο unixstuff που έχουμε δημιουργήσει, χρησιμοποιώντας την εντολή cp.

Αρχικά χρησιμοποιήστε την εντολή `cd` για να μετακινηθείτε στον κατάλογο unixstuff.

```
% cd ~/unixstuff
```

Έπειτα πληκτρολογήστε στην προτροπή:

```
% cp /vol/examples/tutorial/science.txt .
```

Μην ξεχάσετε την τελεία . στο τέλος. Θυμηθείτε ότι η τελεία στο UNIX συμβολίζει τον τρέχοντα κατάλογο.

Η παραπάνω εντολή σημαίνει "αντέγραψε το αρχείο science.txt στον τρέχοντα κατάλογο, διατηρώντας το ίδιο όνομα".

Μόνοι σας

Δημιουργήστε ένα αντίγραφο ασφαλείας του αρχείου science.txt, αντιγράφοντας το σε ένα αρχείο με όνομα science.bak

Μεταφορά και Μετονομασία Αρχείων

mv (move)

Η εντολή `mv file1 file2` μεταφέρει (ή μετονομάζει) το file1 στο file2.

Για να μετακινησετε ένα αρχείο από μια θέση σε μια άλλη, χρησιμοποιήστε την εντολή `mv`. Αυτό έχει ως αποτέλεσμα την μετακίνηση αντί για την αντιγραφή του αρχείου, έτσι στο τέλος θα έχετε ένα αντίγραφο του αρχείου αντί για δύο.

Μπορείτε επίσης να χρησιμοποιήσετε την εντολή `mv` για να μετονομάσετε ένα αρχείο. Αυτό γίνεται αν το μετακινησετε στον ίδιο κατάλογο αλλά με διαφορετικό όνομα.

Θα μετακινήσουμε τώρα το αρχείο science.bak στον κατάλογο backup που δημιουργήσαμε προηγουμένως.

Αρχικά μετακινηθείτε στον κατάλογο unixstuff (ελπίζουμε να θυμόσαστε πως γίνεται). Έπειτα, καθώς είστε μέσα στον κατάλογο unixstuff, πληκτρολογήστε:

```
% mv science.bak backups/.
```

Πληκτρολογήστε `ls` και `ls backups` για να δείτε αν η εντολή λειτούργησε κανονικά.

Διαγραφή Αρχείων και Καταλόγων

rm (remove), rmdir (remove directory)

Για να διαγράψετε ένα αρχείο, χρησιμοποιήστε την εντολή [rm](#). Ως παράδειγμα, θα δημιουργήσουμε ένα αντίγραφο του αρχείου `science.txt` και έπειτα θα το διαγράψουμε.

Μέσα στον κατάλογο `unixstuff`, πληκτρολογήστε:

```
% cp science.txt tempfile.txt
% ls
% rm tempfile.txt
% ls
```

Μπορείτε να χρησιμοποιήσετε την εντολή [rmdir](#) για να διαγράψετε ένα κατάλογο (βεβαιωθείτε πρώτα ότι είναι άδειος). Προσπαθήστε να διαγράψετε τον κατάλογο `backups`. Δεν θα μπορέσετε να το κάνετε, καθώς το UNIX δεν θα σας αφήσει να διαγράψετε ένα κατάλογο που δεν είναι άδειος.

Μόνοι σας

Δημιουργήστε ένα κατάλογο με το όνομα `tempstuff` χρησιμοποιώντας την εντολή [mkdir](#), και έπειτα διαγράψτε τον χρησιμοποιώντας την εντολή [rmdir](#).

Εμφάνιση Περιεχομένων Αρχείου στην Οθόνη

clear (καθαρισμός οθόνης)

Πριν ξεκινήσετε με την επόμενη ενότητα, ίσως θέλετε να καθαρίσετε το παράθυρο του τερματικού από τις προηγούμενες εντολές, ώστε η έξοδος των εντολών που ακολουθούν να είναι ξεκάθαρη.

Στην προτροπή, πληκτρολογήστε:

```
% clear
```

Αυτό θα καθαρίσει το παράθυρο και θα σας αφήσει μόνο με την προτροπή στο πάνω μέρος.

cat (concatenate)

Μπορείτε να χρησιμοποιήσετε την εντολή [cat](#) για να δείτε τα περιεχόμενα ενός αρχείου (ή παραπάνω) στην οθόνη σας. Πληκτρολογήστε:

```
% cat science.txt
```

Όπως μπορείτε να δείτε, το αρχείο είναι μεγαλύτερο από το μέγεθος του παραθύρου και έτσι σύντομα κυλάει προς τα πάνω με αποτέλεσμα να μην μπορούμε να το δούμε ολόκληρο.

less

Η εντολής [less](#) γράφει τα περιεχόμενα ενός αρχείου στην οθόνη, μια σελίδα κάθε φορά. Πληκτρολογήστε:


```
% less science.txt
```

Στο τέλος κάθε σελίδας πιάστε το [space-bar] αν θέλετε να δείτε την επόμενη σελίδα, ή το [q] για να τερματίσετε την ανάγνωση. Όπως καταλαβαίνετε, η `less` είναι προτιμότερη από την `cat` για μεγάλα αρχεία.

head

Η εντολή `head` γράφει τις δέκα πρώτες γραμμές ενός αρχείου στην οθόνη.

Καθαρίστε πρώτα την οθόνη και μετά πληκτρολογήστε:

```
% head science.txt
```

Έπειτα πληκτρολογήστε:

```
% head -5 science.txt
```

Τι διαφορά προκάλεσε το -5 στην εντολή `head`;

tail

Η εντολή `tail` εμφανίζει τις δέκα τελευταίες γραμμές ενός αρχείου στην οθόνη.

Καθαρίστε την οθόνη και πληκτρολογήστε:

```
% tail science.txt
```

Ερώτηση: Πως μπορείτε να δείτε τις 15 τελευταίες γραμμές ενός αρχείου;

Αναζήτηση στα Περιεχόμενα Ενός Αρχείου

Απλή Αναζήτηση με τη Χρήση της less

Χρησιμοποιώντας την εντολή `less`, μπορείτε να ψάξετε το αρχείο για μια λέξη κλειδί. Για παράδειγμα, για να ψάξετε το αρχείο `science.txt` για τη λέξη 'science', πληκτρολογήστε:

```
% less science.txt
```

και καθώς εκτελείται η `less`, πληκτρολογήστε μια κάθετο [/] ακολουθούμενη από τη λέξη που ψάχνετε

```
/science
```

Όπως μπορείτε να δείτε η `less` βρίσκει και φωτίζει τη λέξη. Πληκτρολογήστε [n] για να βρείτε το επόμενο σημείο που εμφανίζεται η λέξη.

grep (μη ρωτήσετε γιατί τη λένε έτσι)

Η εντολή `grep` αποτελεί ένα από τα τυποποιημένα βοηθητικά εργαλεία του UNIX. Ψάχνει αρχεία για καθορισμένες λέξεις ή patterns. Αρχικά καθαρίστε την οθόνη και έπειτα πληκτρολογήστε:

```
% grep science science.txt
```

Όπως μπορείτε να δείτε, η `grep` εμφάνισε κάθε γραμμή που περιέχει την λέξη `science`.

Η μήπως όχι;

Δοκιμάστε το παρακάτω:

```
% grep Science science.txt
```

Η εντολή `grep` διαχωρίζει μικρά-κεφαλαία. Έτσι το `Science` είναι διαφορετικό από το `science`.

Για να μην γίνεται διαχωρισμός μικρών-κεφαλαίων, χρησιμοποιήστε την επιλογή `-i`, δηλ. πληκτρολογήστε:

```
% grep -i science science.txt
```

Για να ψάξετε για μια φράση ή ένα `pattern`, θα πρέπει να το περικλείσετε σε μονά εισαγωγικά. Για παράδειγμα, για να ψάξετε για τη φράση `spinning top`, πληκτρολογήστε:

```
% grep -i 'spinning top' science.txt
```

Κάποιες ακόμα επιλογές της `grep` είναι:

`-v` δείχνει τις γραμμές που ΔΕΝ ταιριάζουν

`-n` δείχνει τον αριθμό γραμμής μπροστά από κάθε γραμμή

`-c` τυπώνει μόνο τον αριθμό των γραμμών που ταίριαξαν

Δοκιμάστε κάποιες από αυτές τις επιλογές και δείτε τα διαφορετικά αποτελέσματα. Μην ξεχνάτε ότι μπορείτε να χρησιμοποιήσετε περισσότερες από μια επιλογές κάθε φορά. Για παράδειγμα, βρείτε τον αριθμό των γραμμών που δεν περιέχουν τις λέξεις `science` ή `Science`.

```
% grep -ivc science science.txt
```

wc (word count)

Ένα ιδιαίτερα βοηθητικό μικρό πρόγραμμα είναι το `wc` το οποίο μετράει λέξεις. Για να μετρήσετε τις λέξεις στο αρχείο `science.txt`, πληκτρολογήστε:

```
% wc -w science.txt
```

Για να βρείτε πόσες γραμμές έχει το αρχείο, πληκτρολογήστε:

```
% wc -l science.txt
```

Εντολές

Εντολή	Σημασία
<code>cp file1 file2</code>	Αντέγραψε το <code>file1</code> και ονόμασε το αντίγραφο <code>file2</code>
<code>mv file1 file2</code>	Μετακίνησε ή μετονόμασε το <code>file1</code> σε <code>file2</code>
<code>rm file</code>	Διέγραψε ένα αρχείο

rmkdir directory	Διέγραψε ένα κατάλογο
cat file	Εμφάνισε ένα αρχείο
less file	Εμφάνισε ένα αρχείο μια σελίδα κάθε φορά
head file	Εμφάνισε τις πρώτες γραμμές ενός αρχείου
tail file	Εμφάνισε τις τελευταίες γραμμές ενός αρχείου
grep 'keyword' file	Ψάξε ένα αρχείο για μια συγκεκριμένη λέξη
wc file	Μέτρηση γραμμών/λέξεων/χαρακτήρων ενός αρχείου

Ανακατεύθυνση / Φίλτρα / Σωλήνωση

Οι περισσότερες διεργασίες που ξεκινούν μέσω εντολών του UNIX γράφουν στην **τυποποιημένη έξοδο** (δηλ. στην οθόνη ή παράθυρο του τερματικού). Πολλές από αυτές τις εντολές παίρνουν την είσοδο τους από την **τυποποιημένη είσοδο** (δηλ. διαβάζουν από το πληκτρολόγιο). Εκτός από την τυποποιημένη είσοδο και έξοδο υπάρχει επίσης και η **τυποποιημένη έξοδος σφάλματος** όπου οι διεργασίες εμφανίζουν τα μηνύματά λάθους τους. Από προεπιλογή η έξοδος σφάλματος κατευθύνεται στην οθόνη ή παράθυρο του τερματικού.

Στο UNIX έχουμε τη δυνατότητα να ανακατευθύνουμε τόσο την είσοδο όσο και την έξοδο των εντολών.

Ανακατεύθυνση της Εξόδου

Χρησιμοποιούμε το σύμβολο `>` για να **ανακατευθύνουμε την έξοδο** μιας εντολής. Για παράδειγμα, για να δημιουργήσουμε ένα αρχείο με όνομα `list1` το οποίο να περιέχει μια λίστα από φρούτα, πληκτρολογούμε:

```
% cat > list1
```

Γράψτε έπειτα τα ονόματα μερικών φρούτων. Πιέστε [Return] μετά από κάθε όνομα.

```
pear  
banana  
apple  
^D {Αυτό σημαίνει [Ctrl] και [d]}
```

Αυτό που συμβαίνει, είναι ότι η εντολή `cat` διαβάζει την τυποποιημένη είσοδο (το πληκτρολόγιο) και το `>` ανακατευθύνει την έξοδο - που φυσιολογικά είναι η οθόνη - σε ένα αρχείο με το όνομα `list1`. Με τον συνδυασμό `^D` λέμε στο φλοιό ότι τελειώσαμε με το αρχείο.

Για να διαβάσετε τα περιεχόμενα του αρχείου, πληκτρολογήστε:

```
% cat list1
```

Μόνοι σας

Χρησιμοποιώντας την παραπάνω μέθοδο, δημιουργήστε ένα αρχείο με το όνομα `list2` που να περιέχει τα παρακάτω φρούτα: πορτοκάλι, μανταρίνι, μάνγκο, γκρέιπφρουτ. Χρησιμοποιήστε την κατάλληλη εντολή για να διαβάσετε τα περιεχόμενα του `list2`

Προσθήκη Περιεχομένων σε Ένα Αρχείο

Χρησιμοποιώντας τη μορφή `>>` μπορείτε να προσθέσετε την τυποποιημένη έξοδο στο τέλος ενός ήδη υπάρχοντος αρχείου, χωρίς να διαγράψετε τα περιεχόμενα που έχει ήδη. Έτσι, για να προσθέσετε περισσότερες καταχωρίσεις στο αρχείο `list1`, πληκτρολογήστε:

```
% cat >> list1
```

Και έπειτα πληκτρολογήστε τα ονόματα περισσότερων φρούτων :)

```
peach
grape
orange
^D
```

Για να διαβάσετε τα περιεχόμενα του αρχείου, πληκτρολογήστε:

```
% cat list1
```

Θα πρέπει να έχετε τώρα δύο αρχεία. Το ένα περιέχει έξι φρούτα, το άλλο περιέχει τέσσερα φρούτα.

Θα χρησιμοποιήσουμε τώρα την εντολή `cat` για να ενώσουμε (concatenate) τα δύο αρχεία, το `list1` και το `list2` σε ένα νέο αρχείο με το όνομα `biglist`. Πληκτρολογήστε:

```
% cat list1 list2 > biglist
```

Η εντολή αυτή διαβάζει τα περιεχόμενα των αρχείων `list1` και `list2` με τη σειρά που φαίνονται και γράφει την έξοδο στο αρχείο `biglist`

Για να διαβάσετε τα περιεχόμενα του νέου αρχείου, πληκτρολογήστε:

```
% cat biglist
```

Ανακατεύθυνση της Είσοδου

Θα χρησιμοποιήσουμε το σύμβολο `<` για να **ανακατευθύνουμε την είσοδο** της εντολής.

Η εντολή `sort` ταξινομεί αλφαβητικά ή αριθμητικά μια λίστα. Πληκτρολογήστε:

```
% sort
```

Γράψτε έπειτα τα ονόματα μερικών ζώων. Πιέστε [Return] μετά από κάθε όνομα.

```
dog
cat
bird
ape
^D (control d για να σταματήσετε)
```

Η έξοδος θα μοιάζει με την παρακάτω:

```
ape
bird
cat
dog
```

Μπορείτε να χρησιμοποιήσετε το `<` για να ανακατευθύνετε την είσοδο ώστε να προέρχεται από ένα αρχείο αντί για το πληκτρολόγιο. Για παράδειγμα, για να ταξινομήσετε τη λίστα με τα φρούτα, πληκτρολογήστε:

```
% sort < biglist
```

και θα δείτε την ταξινομημένη λίστα στην οθόνη σας.

Για να γράψετε την ταξινομημένη λίστα σε ένα αρχείο, πληκτρολογήστε:


```
% sort < biglist > slist
```

Χρησιμοποιήστε την εντολή `cat` για να διαβάσετε τα περιεχόμενα του αρχείου `slist`

Φίλτρα

Η εντολή `sort` επεξεργάζεται την είσοδό της (είτε είναι τυπικό αρχείο είτε δίνεται από το πληκτρολόγιο) και παράγει ένα νέο αρχείο ως αποτέλεσμα της επεξεργασίας αυτής. Τέτοιου είδους εντολές / προγράμματα ονομάζονται φίλτρα. **Φίλτρο** (filter) είναι ένα μικρό εξειδικευμένο πρόγραμμα το οποίο παίρνει είσοδο από την τυποποιημένη είσοδο ή ένα αρχείο, κάνει μία συγκεκριμένη επεξεργασία (πχ ταξινόμηση ή εύρεση λέξης ή διαγραφή γραμμών ή κάτι άλλο) και στέλνει τα αποτελέσματα στην τυποποιημένη έξοδο ή σε κάποιο αρχείο.

Τα φίλτρα χρησιμοποιούνται ως τμήματα σωληνώσεων, όπως θα δούμε παρακάτω.

Σωληνώσεις

Για να δείτε ποιος άλλος είναι συνδεδεμένος στο σύστημα μαζί με εσάς, πληκτρολογήστε:

```
% who
```

Μια μέθοδος για να δείτε αυτή τη λίστα ονομάτων ταξινομημένη, μπορείτε να χρησιμοποιήσετε το φίλτρο `sort`:

```
% who > names.txt  
% sort < names.txt
```

Αυτό είναι κάπως αργό, και θα πρέπει επίσης να θυμηθείτε να διαγράψετε το προσωρινό αρχείο `names.txt` όταν τελειώσετε. Αυτό που στην πραγματικότητα θέλετε να κάνετε, είναι να **ενώσετε με κάποιο τρόπο την έξοδο** της εντολής `who` **με την είσοδο**



της εντολής `sort`. Αυτό ακριβώς κάνουν οι σωληνώσεις (ή διοχετεύσεις ή αγωγοί). Το σύμβολο της σωλήνωσης είναι η κάθετη μπάρα `|` (συνήθως βρίσκεται μαζί με το `\`, πάνω από το `Enter`). Τι λέμε κάθε φορά που γράφουμε το σύμβολο της σωλήνωσης: η εντολή στα αριστερά θα εκτελεστεί και το παραγόμενο της δεν θα σταλεί στην τυποποιημένη έξοδο (οθόνη) αλλά θα χρησιμεύσει σαν τυποποιημένη είσοδος στην εντολή που βρίσκεται στα δεξιά του συμβόλου.

Για παράδειγμα, πληκτρολογήστε:

```
% who | sort
```

και θα πάρετε το ίδιο αποτέλεσμα με το παραπάνω, αλλά πιο γρήγορα και καθαρά. Η τυποποιημένη έξοδος της `who` έγινε τυποποιημένη είσοδος για τη `sort`.

Για να βρείτε πόσοι χρήστες βρίσκονται αυτή τη στιγμή στο σύστημα, πληκτρολογήστε:

```
% who | wc -l
```

Ποιά η διαφορά μεταξύ `|` και `>`, λοιπόν; Δεν κάνουν την ίδια δουλειά; Μάλλον κάνουν παρόμοια ;-) δουλειά αφού η ανακατεύθυνση στέλνει τα αποτελέσματα σε αρχείο ενώ η σωλήνωση σε άλλη εντολή. Και μία δεύτερη διαφορά είναι ότι μπορούμε να έχουμε μία ανακατεύθυνση αλλά απεριόριστες σωληνώσεις.

```
ls /sbin | grep mk | sort
```

Τι πιστεύετε ότι κάνει η παραπάνω εντολή; Μπορείτε να την ξαναγράψετε με ανακατευθύνσεις;

Μόνοι σας

Χρησιμοποιώντας σωληνώσεις, δείξτε όλες τις γραμμές των αρχείων list1 και list2 που περιέχουν το γράμμα 'p' και ταξινομήστε το αποτέλεσμα.

Εντολές

Εντολή	Σημασία
command > file	Ανακατεύθυνση τυποποιημένης εξόδου σε αρχείο
command >> file	Προσθήκη τυποποιημένης εξόδου στο τέλος υπάρχοντος αρχείου
command < file	Ανακατεύθυνση τυποποιημένης εισόδου από αρχείου
command1 command2	Σωλήνωση της εξόδου της command1 στην είσοδο της command2
cat file1 file2 > file0	Συνένωση των αρχείων file1 και file2 στο file0
sort	Ταξινόμηση δεδομένων
who	Λίστα χρηστών που είναι συνδεδεμένοι στο σύστημα τη δεδομένη στιγμή

Χαρακτήρες Μπαλαντέρ - Βοήθεια

Χαρακτήρες Μπαλαντέρ

Οι χαρακτήρες μπαλαντέρ μπορούν να χρησιμοποιηθούν για να υποκαταστήσουν έναν ή περισσότερους χαρακτήρες σε αναζητήσεις.

0 Χαρακτήρας *

Ο χαρακτήρας ταιριάζει με οποιοδήποτε πλήθος χαρακτήρων (από μηδέν έως όσους νά 'ναι) σε ένα όνομα αρχείου ή καταλόγου. Για παράδειγμα, στον κατάλογο unixstuff πληκτρολογήστε:

```
% ls list*
```

Αυτό θα δείξει όλα τα αρχεία του τρέχοντος καταλόγου που ξεκινάνε με τα γράμματα list (list1, list_old, list.f κλπ)

Δοκιμάστε να πληκτρολογήσετε:

```
% ls *list
```

Αυτό θα δείξει όλα τα αρχεία του τρέχοντος καταλόγου που τελειώνουν με τα γράμματα list (thelist, mylist, list, thisisalongnamelist, κλπ)

0 Χαρακτήρας ?

Ο χαρακτήρας ? ταιριάζει με ένα ακριβώς χαρακτήρα. Έτσι το ?ouse θα ταιριάζει με αρχεία όπως το house και το mouse, αλλά όχι το grouse. Δοκιμάστε να πληκτρολογήσετε:

```
% ls ?list
```

Συμβάσεις στα Ονόματα Αρχείων

Όταν ονομάζουμε αρχεία, πρέπει να αποφεύγουμε χαρακτήρες που έχουν ειδική σημασία όπως / * & %. Επίσης καλό είναι να αποφεύγετε τα κενά διαστήματα στα ονόματα. Ο ασφαλέστερος τρόπος να ονομάζετε αρχεία, είναι με γράμματα και αριθμούς του λατινικού αλφάβητου, καθώς και με τα σύμβολα _ (κάτω παύλα) και . (τελεία)³.

Καλά ονόματα	Κακά ονόματα
project.txt	project
my_big_program.c	my big program.c
fred_dave.doc	fred & dave.doc

Κατά παράδοση, τα ονόματα αρχείων ξεκινούν με ένα μικρό γράμμα και μπορεί να τελειώνουν με μια τελεία που ακολουθείται από μια ομάδα γραμμάτων που χαρακτηρίζουν τα περιεχόμενα του αρχείου. Για παράδειγμα, όλα τα αρχεία που περιέχουν κώδικα σε γλώσσα C μπορεί να τελειώνουν με την κατάληξη .c, π.χ. prog1.c. Έπειτα, για να δείτε μια λίστα όλων των αρχείων που περιέχουν κώδικα σε C στον κατάλογο σας,

³ Στα μοντέρνα Unix και Unix-based λειτουργικά, υποστηρίζονται πλήρως όλοι οι χαρακτήρες, οποιουδήποτε αλφάβητου, αφού υποστηρίζεται εγγενώς το πρότυπο Unicode.

χρειάζεται απλώς να γράψετε `ls*.c` όταν βρίσκεστε σε αυτό τον κατάλογο.

Θα πρέπει να σημειώσουμε εδώ ότι ένας κατάλογος είναι απλώς ένας ειδικός τύπος αρχείου. Έτσι οι συμβάσεις που ισχύουν για την ονοματολογία των αρχείων, ισχύουν και για τους καταλόγους.

Αναζήτηση στη Βοήθεια

Ενσωματωμένες Σελίδες Βοήθειας

Το σύστημά σας περιέχει ενσωματωμένες **σελίδες βοήθειας** που μπορούν να σας δώσουν πληροφορίες για τις περισσότερες εντολές. Οι σελίδες βοήθειας θα σας πληροφορήσουν σχετικά με τις επιλογές που έχει κάθε εντολή και πως αυτές μεταβάλλουν τη συμπεριφορά της. Πληκτρολογήστε `man command` για να δείτε τη σελίδα βοήθειας (manual) μιας συγκεκριμένης εντολής.

Για παράδειγμα, για να μάθετε λεπτομέρειες για την εντολή `wc` (word count), πληκτρολογήστε:

```
% man wc
```

Εναλλακτικά, μπορείτε να γράψετε:

```
% whatis wc
```

και θα πάρετε μια σύντομη περιγραφή (σε μια γραμμή) της εντολής, χωρίς τις λεπτομέρειες σχετικά με τις επιλογές κ.λ.π.

Αpropos

Αν δεν γνωρίζετε το ακριβές όνομα μιας εντολής, δοκιμάστε:

```
% apropos keyword
```

Η εντολή αυτή⁴ θα σας δώσει τις εντολές που περιέχουν τη λέξη κλειδί (keyword) στην επικεφαλίδα της σελίδας βοήθειας τους. Για παράδειγμα:

```
% apropos copy
```

Εντολές

Εντολή	Σημασία
*	Ταίριασμα με οποιοδήποτε πλήθος χαρακτήρων
?	Ταίριασμα με ένα μόνο χαρακτήρα
man command	Σελίδα βοήθειας για μια εντολή
whatis command	Σύντομη περιγραφή μιας εντολής
apropos keyword	Εύρεση των εντολών που περιέχουν το keyword στην περίληψή τους

⁴ Η λέξη apropos σημαίνει "σχετικά με, αναφορικά με".

Ασφάλεια

Χρήστες και ομάδες

Για να μπορέσει κάποιος να εργαστεί με το Unix θα πρέπει να έχει δηλωθεί σαν **χρήστης**. Κάθε χρήστης ανήκει σε μία ή παραπάνω **ομάδες** χρηστών (group). Για να δείτε σε ποιές ομάδες ανήκετε, δώστε

```
% groups
```

θα εμφανιστεί μία λίστα με τις ομάδες

```
user1 dialout cdrom plugdev netdev lpadmin sambashare developers
```

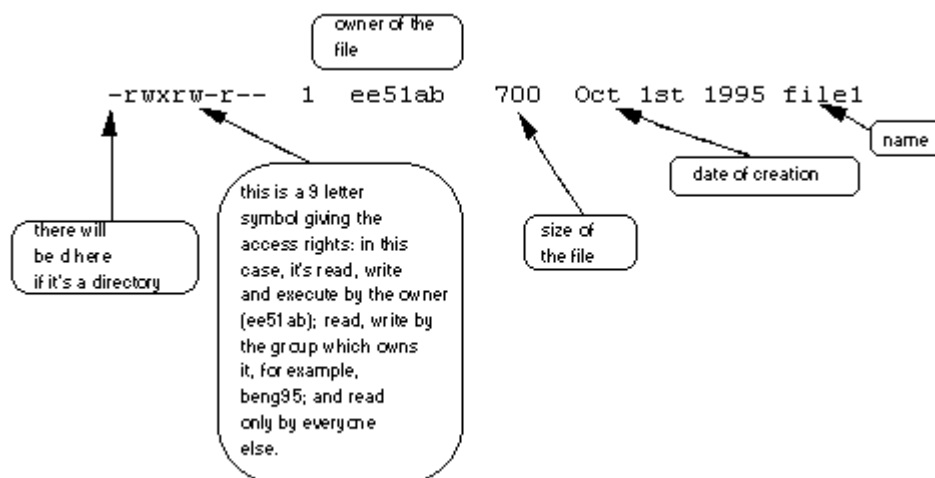
Για να δείτε ποιοί χρήστες είναι δηλωμένοι στον υπολογιστή σας, μπορείτε να δείτε το αρχείο /etc/passw. Αν θέλετε να δείτε τις ομάδες, μπορείτε να δείτε το αρχείο /etc/groups.

Ασφάλεια Συστήματος Αρχείων (Δικαιώματα Πρόσβασης)

Με τρέχοντα κατάλογο του unixstuff, πληκτρολογήστε:

```
% ls -l
```

Θα παρατηρήσετε ότι βλέπετε τώρα πολλές περισσότερες λεπτομέρειες σχετικά με τα περιεχόμενα του καταλόγου σας. Θα μοιάζουν με το παράδειγμα που βλέπετε παρακάτω:



Κάθε αρχείο (και κατάλογος) έχει κάποια δικαιώματα πρόσβασης που του αναλογούν και τα οποία μπορείτε να δείτε γράφοντας `ls -l`. Επίσης, η εντολή `ls -lg` δίνει πρόσθετες πληροφορίες, όπως για παράδειγμα το όνομα της ομάδας στην οποία ανήκει το αρχείο (στο παράδειγμα μας, `beng95`):

```
-rwxrw-r-- 1 ee51lab beng95 2450 Sept29 11:52 file1
```

Στην πιο αριστερή στήλη, υπάρχει μια σειρά από 10 σύμβολα στα οποία εμφανίζονται τα γράμματα `d`, `r`, `w`, `x`, `-` και περιστασιακά το `s` ή το `S`. Αν υπάρχει το `d`, θα είναι εντελώς αριστερά και συμβολίζει ότι το όνομα που βλέπουμε είναι στην πραγματικότητα κατάλογος. Αν είναι αρχείο, στη θέση του `d` θα υπάρχει μια παύλα (-).

Τα 9 σύμβολα που απομένουν, δείχνουν τις άδειες ή δικαιώματα πρόσβασης και ομαδοποιούνται σε τρεις ομάδες με τρία σύμβολα η κάθε μία

- Η αριστερή τριάδα συμβόλων δείχνει τα δικαιώματα για τον χρήστη ο οποίος είναι κάτοχος (owner) του αρχείου ή του καταλόγου (ee51ab στο παραπάνω παράδειγμα)
- Η μεσαία τριάδα δείχνει τα δικαιώματα για την ομάδα (group) στην οποία ανήκει το αρχείο ή κατάλογος (eebeng95 στο παραπάνω παράδειγμα)
- Η δεξιά τριάδα δείχνει τα δικαιώματα για όλους τους υπόλοιπους χρήστες (others).

Τα σύμβολα r, w, κ.λ.π. έχουν ελαφρά διαφορετικό νόημα ανάλογα με το αν αναφέρονται σε ένα απλό αρχείο ή σε ένα κατάλογο.

Δικαιώματα Πρόσβασης σε Αρχεία

- [read] Το r (ή -), δείχνει δικαίωμα ανάγνωσης (ή όχι). Με λίγα λόγια, αν υπάρχει μπορούμε να διαβάσουμε και να αντιγράψουμε το αρχείο.
- [write] Το w (ή -), δείχνει δικαίωμα εγγραφής (ή όχι). Με λίγα λόγια αν μπορούμε να γράψουμε ή με κάποιο τρόπο να αλλάξουμε το αρχείο.
- [execute] Το x (ή -), δείχνει δικαίωμα εκτέλεσης (ή όχι). Με λίγα λόγια αν μπορούμε να εκτελέσουμε αυτό το αρχείο (Αν αυτό φυσικά έχει νόημα, π.χ. αν το αρχείο είναι κάποιο εκτελέσιμο πρόγραμμα).

Δικαιώματα Πρόσβασης σε Κατάλογους

- Το r επιτρέπει σε χρήστες να βλέπουν τη λίστα των αρχείων ενός καταλόγου.
- Το w επιτρέπει σε χρήστες να διαγράφουν αρχεία από τον κατάλογο ή να μετακινούν αρχεία μέσα σε αυτόν.
- Το x επιτρέπει την πρόσβαση σε αρχεία του καταλόγου. Αυτό υπονοεί ότι έχετε τη δυνατότητα να διαβάζετε αρχεία στον κατάλογο, με την προϋπόθεση ότι έχει δικαιώματα στα συγκεκριμένα αρχεία.

Έτσι, για να μπορείτε να διαβάσετε ένα αρχείο, θα πρέπει να έχετε δικαιώματα εκτέλεσης στον κατάλογο στον οποίο βρίσκεται και άρα και στον κατάλογο που τον περιέχει κ.ο.κ.

Μερικά Παραδείγματα

-rwxrwxrwx	Ένα αρχείο το οποίο μπορεί καθένας να διαβάσει, να αλλάξει και να διαγράψει.
-rw-----	Ένα αρχείο το οποίο μπορεί μόνο ο ιδιοκτήτης να γράψει και να διαβάσει, και κανείς άλλος δεν έχει οποιοδήποτε δικαίωμα. Κανείς (ούτε ο ιδιοκτήτης) έχει δικαίωμα εκτέλεσης. Για παράδειγμα, μπορεί να είναι το αρχείο με τα email σας.

Αλλαγή Δικαιωμάτων Πρόσβασης

chmod (αλλαγή κατάστασης ενός αρχείου)

Μόνο ο ιδιοκτήτης ενός αρχείου μπορεί να χρησιμοποιήσει την chmod (change mode) για να αλλάξει τα δικαιώματα ενός αρχείου. Οι επιλογές της εντολής chmod είναι οι παρακάτω:

Σύμβολο	Ποιός	Σύμβολο	Δικαίωμα	Σύμβολο	Ναι / Όχι
u	χρήστης	r	ανάγνωση	+	προσθήκη δι-

Σύμβολο	Ποιός	Σύμβολο	Δικαίωμα	Σύμβολο	Ναι / Όχι
					καιώματος
g	ομάδα	w	εγγραφή (και διαγραφή)	-	διαγραφή δικαιώματος
o	υπόλοιποι	x	εκτέλεση (και πρόσβαση στον κατάλογο)		
a	όλοι				

Για παράδειγμα, για να αφαιρέσετε τα δικαιώματα εγγραφής και εκτέλεσης στο αρχείο `biglist` για την ομάδα και τους υπόλοιπους χρήστες:

```
% chmod go-rwx biglist
```

Η παραπάνω εντολή δεν θα πειράξει τα υπόλοιπα δικαιώματα (π.χ. του ιδιοκτήτη).

Για να δώσετε δικαιώματα ανάγνωσης και εγγραφής στο αρχείο `biglist` σε όλους:

```
% chmod a+rw biglist
```

Μόνοι σας

Δοκιμάστε να αλλάξετε δικαιώματα στο αρχείο `science.txt` και στον κατάλογο `backups`.

Χρησιμοποιήστε την εντολή `ls -l` για να ελέγξετε ότι τα δικαιώματα έχουν πραγματικά αλλάξει.

Διεργασίες

Διεργασίες και Εργασίες

Μια **διεργασία (process)** είναι ένα πρόγραμμα που εκτελείται και αναγνωρίζεται από ένα μοναδικό **αναγνωριστικό διεργασίας**, PID. Για να δείτε πληροφορίες για τις διεργασίες σας και το αναγνωριστικό τους, γράψτε:

```
% ps
```

Μια διεργασία μπορεί να βρίσκεται στο **προσκήνιο**, στο **παρασκήνιο**, ή σε **παύση**. Σε γενικές γραμμές το κέλυφος δεν επιστρέφει την προτροπή εντολής μέχρι να τερματιστεί η τρέχουσα διεργασία.

Μερικές διεργασίες χρειάζονται πολύ χρόνο για να εκτελεστούν και στο μεταξύ εμποδίζουν τη χρήση του τερματικού. Αν μεταφέρετε μια τέτοια διεργασία στο παρασκήνιο, θα μπορέσετε να έχετε άμεσα διαθέσιμη την προτροπή εντολών. Μπορείτε έτσι να συνεχίσετε να εκτελείτε άλλες εντολές καθώς η αρχική διεργασία συνεχίζει να εκτελείται.

Εκτέλεση Διεργασιών στο Παρασκήνιο

Για να ξεκινήσετε μια διεργασία στο παρασκήνιο, προσθέστε ένα & στο τέλος της γραμμής εντολής. Για παράδειγμα, η εντολή `sleep` εισάγει μια αναμονή, για όσα δευτερόλεπτα της δώσουμε. Πληκτρολογήστε:

```
% sleep 10
```

Η παραπάνω εντολή περιμένει 10 δευτερόλεπτα πριν επιστρέψει η προτροπή του κελύφους. Μέχρι να δείτε ξανά την προτροπή δεν μπορείτε να κάνετε τίποτα εκτός από το να περιμένετε.

Για να εκτελέσετε την `sleep` στο παρασκήνιο, γράψτε:

```
% sleep 10 &
```

```
[1] 6259
```

Με το & η εντολής εκτελείται στο παρασκήνιο και η προτροπή επιστρέφει αμέσως. Μπορείτε να εκτελέσετε άλλα προγράμματα καθώς περιμένετε αυτή την εντολή να τελειώσει.

Η πρώτη γραμμή στο παραπάνω παράδειγμα, έχει πληκτρολογηθεί από τον χρήστη. Η επόμενη γραμμή που αναφέρει το αριθμό της διεργασίας και της εργασίας, προέρχεται από το μηχάνημα. Ο χρήστης ειδοποιείται για τον αριθμό της εργασίας (ξεκινάει από τον αριθμό 1) ο οποίος περικλείεται σε αγκύλες και για τον αριθμό της διεργασίας. Επίσης ειδοποιείται όταν τελειώσει η εκτέλεση μιας διεργασίας που εκτελείται στο παρασκήνιο. Η εκτέλεση στο παρασκήνιο είναι χρήσιμη για διεργασίες που κάνουν αρκετό χρόνο να ολοκληρωθούν.

Αποστολή στο Παρασκήνιο μιας Διεργασίας που εκτελείται στο Προσκήνιο

Στην προτροπή, πληκτρολογήστε:

```
% sleep 1000
```

Μπορείτε να βάλετε την διεργασία που εκτελείται στο προσκήνιο σε αναμονή, πληκτρολογώντας `^Z`, δηλ. κρατώντας πιεσμένο το πλήκτρο [Ctrl] και πιέζοντας το [z]. Για να θέσετε κατόπιν τη διεργασία στο παρασκήνιο, πληκτρολογήστε:

```
% bg
```

Σημείωση: Μη στέλνετε στο παρασκήνιο διεργασίες που χρειάζονται είσοδο από το χρήστη, π.χ. κάποιον επεξεργαστή κειμένου όπως το vi

Λίστα Διεργασιών που Έχουν Διακοπεί ή Εκτελούνται στο Παρασκήνιο

Όταν μια διεργασία εκτελείται στο παρασκήνιο ή έχει διακοπεί, βρίσκεται σε μια λίστα με ένα αντίστοιχο αριθμό εργασίας. Για να δείτε αυτή τη λίστα, γράψτε:

```
% jobs
```

Θα δείτε μια έξοδο όπως την παρακάτω:

```
[1] Suspended sleep 1000
[2] Running netscape
[3] Running matlab
```

Για να ξεκινήσετε ξανά (στο προσκήνιο) μια διεργασία που έχει διακοπεί, πληκτρολογήστε:

```
% fg %αριθμός_εργασίας
```

Για παράδειγμα, για να ξεκινήσετε ξανά την εντολή `sleep 1000`, πληκτρολογήστε:

```
% fg %1
```

Αν γράψετε `fg` χωρίς να δώσετε αριθμό εργασίας, θα επαναφέρετε στο προσκήνιο την τελευταία διεργασία που διακόψατε.

Τερματίζοντας μια Διεργασία

kill (τερματισμός ή αποστολή σήματος σε μια διεργασία)

Μερικές φορές είναι απαραίτητο να τερματίσουμε μια εργασία (όταν για παράδειγμα εκτελεί ένα ατέρμονο βρόχο).

Για να τερματίσετε μια διεργασία που εκτελείται στο προσκήνιο, πιέστε `^C` (control c). Για παράδειγμα, εκτελέστε:

```
% sleep 100
^C
```

Για να τερματίσετε μια εργασία που βρίσκεται στο παρασκήνιο ή έχει διακοπεί, γράψτε (!!! προσέξτε το σύμβολο %):

```
% kill %αριθμός_εργασίας
```

Για παράδειγμα, εκτελέστε:

```
% sleep 100 &
```

```
% jobs
```

Αν για παράδειγμα ο αριθμός της εργασίας είναι 4, γράψτε:

```
% kill %4
```

Για να ελέγξετε αν το παραπάνω λειτούργησε, ελέγξτε ξανά τη λίστα των εργασιών για να δείτε αν έχει αφαιρεθεί η εργασία.

Στο επόμενο τμήμα θα δείτε πως μπορείτε να σταματήσετε διεργασίες.

ps (κατάσταση διεργασίας)

Με την εντολή `ps` (process status) μπορείτε να δείτε τη λίστα με τις διεργασίες.

```
% sleep 1000 &
% ps

PID TT S TIME COMMAND
20077 pts/5 S 0:05 sleep 1000
21563 pts/5 T 0:00 netscape
21873 pts/5 S 0:25 nedit
```

Εναλλακτικά, μπορείτε να τερματίσετε μια διεργασία χρησιμοποιώντας τον αναγνωριστικό αριθμό της (PID) και την εντολή `kill PID_number` (προσέξτε ότι λείπει το σύμβολο %). Έτσι, για να τερματίσετε τη διεργασία `sleep 1000`, πληκτρολογήστε:

```
% kill 20077
```

και έπειτα πληκτρολογήστε `ps` ξανά για να δείτε αν έχει διαγραφεί από την λίστα.

Αν μια διεργασία αρνείται να τερματίσει, χρησιμοποιήστε την επιλογή `-9`:

```
% kill -9 20077
```

|| Σημείωση: Δεν μπορείτε να τερματίσετε τις διεργασίες άλλων χρηστών !!!

Εντολές

Εντολή	Σημασία
<code>ls -lag</code>	Δείχνει τα δικαιώματα πρόσβασης για όλα τα αρχεία
<code>chmod [options] file</code>	Αλλάζει τα δικαιώματα πρόσβασης του συγκεκριμένου αρχείου
<code>command &</code>	Εκτελεί την εντολή στο παρασκήνιο
<code>^C</code>	Τερματίζει την διεργασία που εκτελείται στο προσκήνιο
<code>^Z</code>	Διακόπτει τη διεργασία που εκτελείται στο προσκήνιο
<code>bg</code>	Θέτει στο παρασκήνιο την διεργασία που έχει διακοπεί

jobs	Δείχνει λίστα με τα τρέχοντα jobs
fg %1	Επαναφέρει στο προσκήνιο το job με αριθμό 1
kill %1	Τερματίζει το job με αριθμό 1
ps	Λίστα των τρεχόντων διεργασιών
kill 26152	Τερματίζει την διεργασία με αριθμό 26152

Διάφορα (και όχι αδιάφορα)

Άλλες Χρήσιμες Εντολές του UNIX

quota

Σε ένα περιβάλλον σχολικού εργαστηρίου, συνήθως οι σπουδαστές έχουν διαθέσιμο περιορισμένο χώρο στο δίσκο για τα προσωπικά τους αρχεία. Το μέγεθος ποικίλει, αλλά μια τυπική ελάχιστη τιμή είναι τα 100 Mb. Αν ξεπεράσετε τον προκαθορισμένο αυτό χώρο (που ονομάζεται quota) έχετε διαθέσιμες 7 μέρες για να σβήσετε τα περιττά αρχεία.

Για να ελέγξετε το μέγιστο διαθέσιμο σας χώρο καθώς και πόσο έχετε καταναλώσει, πληκτρολογήστε:

```
% quota -v
```

df

Η εντολή df τον ελεύθερο χώρο που έχει απομείνει στο σύστημα αρχείων. Για παράδειγμα, για να δείτε πόσος διαθέσιμος χώρος υπάρχει στο μηχάνημα σας γράψτε:

```
% df -h
```

du

Η εντολή du δείχνει σε kilobytes το χώρο που καταλαμβάνει κάθε υποκατάλογος. Είναι χρήσιμη αν έχετε ξεπεράσει το διαθέσιμο χώρο που έχει καθοριστεί για το λογαριασμό σας και θέλετε να δείτε ποιος κατάλογος είναι αυτός που περιέχει τα περισσότερα αρχεία. Στο προσωπικό σας κατάλογο, πληκτρολογήστε:

```
% du -s *
```

Η επιλογή -s θα δείξει μόνο την περίληψη (το συνολικό μέγεθος) ενώ το * σημαίνει ότι θα ληφθούν υπόψιν όλα τα αρχεία και οι κατάλογοι.

gzip

Το πρόγραμμα αυτό συμπιέζει αρχεία, μειώνοντας έτσι το χώρο που καταλαμβάνουν στο δίσκο, και μας εξοικονομεί χώρο. Για παράδειγμα, πληκτρολογήστε:

```
% ls -l science.txt
```

και σημειώστε το μέγεθος του αρχείου όπως αναφέρεται από την εντολή ls -l. Για να συμπιέσετε έπειτα το science.txt, πληκτρολογήστε:

```
% gzip science.txt
```

Αυτό θα το συμπιέσει και θα το τοποθετήσει σε ένα αρχείο με το όνομα science.txt.gz

Για να δείτε την αλλαγή στο μέγεθος, χρησιμοποιήστε ξανά την εντολή ls -l.

Για να αποσυμπιέσετε το αρχείο, χρησιμοποιήστε την εντολή gunzip:


```
% gunzip science.txt.gz
```

zcat

Η εντολή `zcat` απεικονίζει αρχεία συμπιεσμένα με το `gzip`, χωρίς να χρειάζεται να τα αποσυμπιέσει πρώτα.

```
% zcat science.txt.gz
```

Αν το κείμενο προχωράει πολύ γρήγορα για σας, περάστε την έξοδο μέσω της εντολής `less`:

```
% zcat science.txt.gz | less
```

file

Η εντολή `file` αναγνωρίζει το είδος του αρχείου ανάλογα με το περιεχόμενό του, για παράδειγμα κείμενο (ASCII), φωτογραφία, συμπιεσμένα δεδομένα κ.α. Για να πάρετε μια αναφορά για όλα τα είδη αρχείων που υπάρχουν στον κατάλογο σας, πληκτρολογήστε:

```
% file *
```

diff

Η εντολή αυτή συγκρίνει τα περιεχόμενα δύο αρχείων και δείχνει τις διαφορές μεταξύ τους. Υποθέστε ότι για παράδειγμα έχετε το αρχείο `file1`, αλλάζετε κάποιο τμήμα του, και το αποθηκεύετε ως `file2`. Για να δείτε τις διαφορές μεταξύ τους, γράψτε:

```
% diff file1 file2
```

Οι γραμμές που ξεκινάνε με `<` αναφέρονται στο `file1`, ενώ αυτές που ξεκινάνε με `>` δείχνουν το `file2`.

find

Η εντολή αυτή ψάχνει στους καταλόγους, για αρχεία και καταλόγους που ταιριάζουν με ένα όνομα, ημερομηνία, μέγεθος ή οποιαδήποτε άλλη παράμετρο έχετε δώσει. Πρόκειται για απλή εντολή, αλλά έχει πάρα πολλές επιλογές - μπορείτε να τις δείτε στην αντίστοιχη σελίδα βοήθειας με την εντολή [man find](#).

Για να ψάξετε για όλα τα αρχεία με την κατάληξη `.txt`, ξεκινώντας από τον τρέχοντα κατάλογο (`.`) και συνεχίζοντας σε όλους τους υποκαταλόγους, τυπώνοντας το όνομα του αρχείου στην οθόνη, πληκτρολογήστε:

```
% find . -name "*.txt" -print
```

Για να βρείτε όλα τα αρχεία με μέγεθος μεγαλύτερο από 1 Mb και να δείτε τα αποτελέσματα ως λίστα με λεπτομερείς πληροφορίες, πληκτρολογήστε:

```
% find . -size +1M -ls
```

history

Το κέλυφος κρατάει μια λίστα (χρονικά ταξινομημένη) με τις εντολές που έχετε ήδη

εκτελέσει. Κάθε εντολή παίρνει ένα αριθμό ανάλογα με τη σειρά εκτέλεσης της:

```
% history (λίστα ιστορικού εντολών)
```

Αν χρησιμοποιείται το κέλυφος C (csh), μπορείτε να χρησιμοποιήσετε το θαυμαστικό (!) για να ανακαλέσετε τις εντολές με εύκολο τρόπο.

```
% !! (ανάκληση της τελευταίας εντολής)
```

```
% !-3 (ανάκληση της τρίτης πιο πρόσφατης εντολής)
```

```
% !5 (ανάκληση της πέμπτης εντολής από τη λίστα)
```

```
% !grep (ανάκληση της τελευταίας εντολής που ξεκινάει με grep)
```

Μπορείτε να αυξήσετε το μέγεθος της προσωρινής αποθήκευσης του ιστορικού, πληκτρολογώντας:

```
% set history=200
```

Εκτύπωση (lp)

Για να εκτυπώσετε ένα αρχείο, δώστε

```
% lp filename
```

Παροχέτευση (spooling)

Είναι γνωστό ότι οι εκτυπωτές είναι αργά περιφερειακά. Σε σύγκριση, οι δίσκοι και οι επεξεργαστές είναι χιλιάδες φορές ταχύτεροι ταχύτεροι. Ισχύει επίσης το εξής: ένα πρόγραμμα είναι τόσο αργό όσο το πιο αργό κομμάτι του (τμήμα του). Ένα πρόγραμμα το οποίο είναι πανγρήγορο αλλά περιμένει σταματημένο προκειμένου να εκτυπώσει πάυει να είναι πανγρήγορο και είναι απλά σταματημένο. Πάρτε για παράδειγμα μία εφαρμογή που ετοιμάζει τα εκκαθαριστικά της εφορίας. Οι υπολογισμοί του κάθε εκκαθαριστικού παίρνουν κάποιο κλάσμα του δευτερολέπτου - η εκτύπωση παίρνει 1 λεπτό. Έτσι, αν πρέπει να ετοιμαστούν 1000 εκκαθαριστικά, θα έπρεπε ο υπολογιστής να απασχολείται 1000 λεπτά (16+ ώρες) ενώ στην ουσία δεν θα έκανε τίποτα παραπάνω από τον περιμένει τον αργό εκτυπωτή. Ή, φανταστείτε να στέλνατε για εκτύπωση την εργασία σας και μέχρι να τελειώσει η εκτύπωση να μην μπορούσατε να χρησιμοποιήσετε τον επεξεργαστή κειμένου. Ξέρετε όμως ότι δε γίνεται έτσι: στέλνετε την εκτύπωσή σας και ελάχιστα δευτερόλεπτα μετά μπορείτε να συνεχίσετε να γράφετε ενώ ο εκτυπωτής λειτουργεί παράλληλα με εσάς. Η τεχνική που χρησιμοποιείται ώστε ένα πρόγραμμα να μην περιορίζεται ένα πρόγραμμα από τα αργά τμήματά του⁵ ονομάζεται παροχέτευση (spooling) και όπως καταλάβατε, το κλειδί είναι οι δουλειές να γίνονται ασύγχρονα κι όχι σειριακά. Χωρίς παροχέτευση, τα περισσότερα προγράμματα θα έκαναν κύκλους ταχύτατης επεξεργασίας και μακριών αναμονών, κάτι ιδιαίτερα αναποτελεσματικό.

Πως λειτουργεί

Η παροχέτευση της εκτύπωσης έχει δύο σκέλη:

1. ενσωματωμένη δυνατότητα στο ΛΣ να παρακρατεί τα δεδομένα που πάνε στον εκτυπωτή και να τα αποθηκεύει σε προσωρινό χώρο (buffer) και
2. ένα απλό πρόγραμμα το οποίο θα περνάει τα δεδομένα από το buffer στον εκτυ-

⁵ Ένα τμήμα είναι τόσο πιο αργό όσο περισσότερα μηχανικά κομμάτια έχει: ο εκτυπωτής είναι αργότερος από το δίσκο, ο δίσκος είναι αργότερος από τον επεξεργαστή.

πωτή.

Χωρίς παροχέτευση

Η εφαρμογή προσπαθεί να περάσει γραμμές ή σελίδες στον (αργό) εκτυπωτή. Το ΛΣ παίρνει αυτές τις αιτήσεις, τις περνάει μία μία στον εκτυπωτή και όλοι μαζί περιμένουν πότε ο εκτυπωτής θα τελειώσει με τη γραμμή ή τη σελίδα για να του δοθεί η επόμενη και η επόμενη και η επόμενη.

Με παροχέτευση

Αντί να δίνονται οι γραμμές / σελίδες στον εκτυπωτή, δίνονται στο μηχανισμό παροχέτευσης. Αυτός, αντί να τα στείλει στον εκτυπωτή τα γράφει σε buffer στο σκληρό δίσκο, ο οποίος είναι πάρα πολύ γρηγορότερος. Έτσι η εφαρμογή απελευθερώνεται να συνεχίσει. Ενώ η εφαρμογή συνεχίζει, ήσυχη ότι έχει τελειώσει με την εκτύπωση, ο μηχανισμός ξεκινάει το άλλο κομμάτι της παροχέτευσης, την πραγματική εκτύπωση. Αρχίζει και ταΐζει τον εκτυπωτή με τα δεδομένα που έχει αποθηκεύσει στο buffer. Έτσι, τώρα, οι δύο εργασίες (εφαρμογή και εκτύπωση) γίνονται παράλληλα (άσε που η εφαρμογή μπορεί και να έχει κλείσει μέχρι να τελειώσει η εκτύπωση).

Τα οφέλη

Η παροχέτευση βελτιώνει τις δυνατότητες των συστημάτων για πολυδιεργασία (multitasking). Οι περισσότερες εφαρμογές θέλουν είσοδο και παράγουν έξοδο. Χωρίς παροχέτευση, το πλήθος των εφαρμογών που θα μπορούσαν να τρέξουν ταυτόχρονα θα περιοριζόταν από το πλήθος και τη διαθεσιμότητα των περιφερειακών συσκευών. ΜΕ την παροχέτευση, μία διαδικασία δεν χρειάζεται να περιμένει να έχει διαθέσιμη για την πραγματική συσκευή

Εξαιτίας της τεράστιας διαφοράς ταχύτητας δίσκου - εκτυπωτή, το να περνάνε αρχικά τα δεδομένα στο δίσκο και από κει να περνάνε στον εκτυπωτή με τους ρυθμούς του εκτυπωτή, βελτιώνει την ρυθμαπόδοση (throughput)⁶ του συστήματος.

⁶ Ρυθμαπόδοση είναι ο ρυθμός με τον οποίο ολοκληρώνονται εργασίες στη μονάδα του χρόνου. Πρακτικά, όσο λιγότερο χρόνο μένει ανενεργό κάποιο τμήμα και όσο λιγότερα τμήματα του συστήματος μένουν ανενεργά, τόσο περισσότερες εργασίες ολοκληρώνονται στο λεπτό.

Προγραμματισμός κελύφους

Μεταβλητές στο UNIX

Οι **μεταβλητές** είναι ένας τρόπος να περάσουμε πληροφορίες από το κέλυφος στα προγράμματα όταν τα εκτελούμε. Τα προγράμματα κοιτάζουν "το περιβάλλον" για συγκεκριμένες μεταβλητές και αν τις βρουν, χρησιμοποιούν τις τιμές τους. Μερικές από τις μεταβλητές ορίζονται από το σύστημα, άλλες από εσάς και άλλες από το κέλυφος ή οποιοδήποτε πρόγραμμα φορτώνει ένα άλλο πρόγραμμα.

Οι **τυποποιημένες μεταβλητές** του UNIX χωρίζονται σε δύο κατηγορίες, τις **μεταβλητές περιβάλλοντος** και τις **μεταβλητές κελύφους**. Σε γενικές γραμμές, οι μεταβλητές κελύφους εφαρμόζονται μόνο στο κέλυφος που εκτελείται τη δεδομένη στιγμή και χρησιμοποιούνται για να θέσουν συνθήκες εργασίες για κάποιο μικρό χρονικό διάστημα. Οι μεταβλητές περιβάλλοντος έχουν μεγαλύτερη επίδραση, και αυτές που τίθενται κατά την είσοδο σας στο σύστημα είναι έγκυρες για όσο διάστημα παραμένετε συνδεδεμένοι. Κατά σύμβαση, οι μεταβλητές περιβάλλοντος έχουν ονόματα με ΚΕΦΑΛΑΙΑ ΓΡΑΜΜΑΤΑ ενώ οι μεταβλητές κελύφους με μικρά.

Μεταβλητές Περιβάλλοντος

Ένα παράδειγμα μεταβλητής περιβάλλοντος είναι η μεταβλητή `OSTYPE`. Η τιμή της είναι το λειτουργικό σύστημα που χρησιμοποιείτε. Πληκτρολογήστε:

```
% echo $OSTYPE
```

Μερικά ακόμα παραδείγματα μεταβλητών περιβάλλοντος είναι:

- `USER` (το όνομα χρήστη σας)
- `HOME` (η πλήρης διαδρομή για τον προσωπικό σας κατάλογο)
- `HOST` (το όνομα του υπολογιστή που χρησιμοποιείτε)
- `ARCH` (η αρχιτεκτονική επεξεργαστή του υπολογιστή σας)
- `DISPLAY` (το όνομα της οθόνης που χρησιμοποιείται από τα X Windows)
- `PRINTER` (ο εκτυπωτής που δέχεται εργασίες εκτύπωσης από προεπιλογή)
- `PATH` (οι κατάλογοι που ψάχνει το κέλυφος για να εντοπίσει μια εντολή)

Εύρεση των Τρεχόντων Τιμών των Μεταβλητών

Οι μεταβλητές τύπου `ENVIRONMENT` (περιβάλλοντος) μπορούν να τεθούν με την `setenv` και να απεικονιστούν με τη χρήση της `printenv` ή `env`. Μπορούν να διαγραφούν με την χρήση της `unsetenv`.

Για να δείτε τις τιμές όλων αυτών των μεταβλητών, πληκτρολογήστε:

```
% printenv | less
```

Μεταβλητές Κελύφους

Ένα παράδειγμα μεταβλητής του κελύφους, είναι η μεταβλητή που σχετίζεται με την τήρηση ιστορικού. Η τιμή αυτής της μεταβλητής αναφέρεται στο πόσες εντολές θα θυμάται το κέλυφος, επιτρέποντας έτσι στο χρήστη να ανατρέξει στις εντολές που έχει ήδη εκτελέσει. Πληκτρολογήστε:

```
% echo $history
```

Περισσότερα παραδείγματα μεταβλητών κελύφους είναι:

- `cwd` (ο τρέχον κατάλογος σας)
- `home` (η διαδρομή για τον προσωπικό σας κατάλογο)
- `path` (οι κατάλογοι που θα ψάξει το κέλυφος για να βρει μια εντολή)
- `prompt` (το κείμενο που δείχνει ως προτροπή το κέλυφος σας για να εισάγετε μια εντολή)
- `shell` (το κέλυφος με το οποίο κάνατε login στο σύστημα)

Εύρεση Τρεχόντων Τιμών Αυτών των Μεταβλητών

Οι μεταβλητές του κελύφους μπορούν να τεθούν και να απεικονιστούν χρησιμοποιώντας την εντολή `set`. Μπορούν επίσης να διαγραφούν με την εντολή `unset command`.

Για να δείτε τις τιμές όλων αυτών των μεταβλητών, πληκτρολογήστε:

```
% set | less
```

Τελικά ποια είναι η διαφορά μεταξύ `path` και `PATH`;

Σε γενικές γραμμές, οι μεταβλητές περιβάλλοντος και κελύφους που έχουν το ίδιο όνομα (εκτός από το γεγονός ότι οι μεν είναι κεφαλαία και οι δε μικρά) είναι χωριστές και ανεξάρτητες εκτός από το γεγονός ότι πιθανόν έχουν τις ίδιες αρχικές τιμές. Ωστόσο υπάρχουν και εξαιρέσεις.

Κάθε φορά που αλλάζουν οι μεταβλητές κελύφους `home`, `user` και `term` ανανεώνονται και οι αντίστοιχες μεταβλητές περιβάλλοντος `HOME`, `USER` και `TERM`. Ωστόσο αν ανανεωθούν πρώτα οι μεταβλητές περιβάλλοντος, οι αντίστοιχες του κελύφους δεν αλλάζουν τιμή.

Οι μεταβλητές `PATH` και `path` καθορίζουν τους καταλόγους που ανιχνεύονται για την εύρεση εντολών και προγραμμάτων. Και οι δύο μεταβλητές πάντοτε αντιπροσωπεύουν την ίδια λίστα καταλόγων - όποια από τις δύο και αν αλλάξει, η άλλη ανανεώνεται αυτόματα.

Πως χρησιμοποιούμε και πως θέτουμε Μεταβλητές

Κάθε φορά που εισέρχεστε στο UNIX σύστημα σας, το σύστημα ψάχνει στον προσωπικό σας κατάλογο για αρχεία αρχικοποίησης. Οι πληροφορίες σε αυτά τα αρχεία χρησιμοποιούνται για να ρυθμίσουν τις παραμέτρους του προσωπικού σας περιβάλλοντος. Τα κέλυφη `C` και `TC` χρησιμοποιούν δύο αρχεία, τα `.login` και `.cshrc` (προσέξτε ότι και τα δύο αρχεία ξεκινούν με τελεία).

Κατά την είσοδο σας, το κέλυφος `C` διαβάζει πρώτα το `.cshrc` και μετά το `.login`

Το `.login` χρησιμοποιείται για να θέσει ρυθμίσεις που ισχύουν για όλη την σύνοδο (`session`) και για να εκτελέσει λειτουργίες που απαιτούνται μόνο στη διαδικασία εισόδου.

Το `.cshrc` χρησιμοποιείται για να θέσει ρυθμίσεις και να εκτελέσει λειτουργίες που σχετίζονται με το κέλυφος και πρέπει να γίνονται κάθε φορά που ξεκινά - άσχετα αν είναι το κέλυφος εισόδου ή όχι).

Σε γενικές γραμμές, είναι σωστότερο να ρυθμίζονται οι μεταβλητές `ENVIRONMENT` (περιβάλλοντος) στο αρχείο `.login` και οι μεταβλητές `SHELL` (κέλυφος) στο `.cshrc`.

ΠΡΟΣΟΧΗ: ΠΟΤΕ μη βάλετε εντολές που ενεργοποιούν το γραφικό περιβάλλον στα αρχεία `.cshrc` ή `.login`.

Μεταβλητές Περιβάλλοντος στο Αρχείο `.cshrc`

Αν για παράδειγμα θέλετε να αλλάξετε τον αριθμό των εντολών κελύφους που αποθηκεύονται στη λίστα ιστορικού, θα πρέπει να θέσετε μια νέα τιμή στη μεταβλητή του κελύφους που ονομάζεται `history`. Η προεπιλεγμένη τιμή είναι 100, αλλά μπορείτε να την αυξήσετε αν θέλετε.

```
% set history = 200
```

Ελέγξτε ότι η παραπάνω εντολή λειτούργησε:

```
% echo $history
```

Το παραπάνω ωστόσο θέτει την μεταβλητή μόνο για όση ώρα εκτελείται το τρέχον κέλυφος. Αν ανοίξετε ένα νέο παράθυρο `xterm` (για παράδειγμα), θα έχει την παλιά τιμή της μεταβλητής. Για να αλλάξετε αυτή την τιμή ΜΟΝΙΜΑ θα πρέπει να προσθέσετε την εντολή `set` μέσα στο αρχείο `.cshrc`.

Ανοίξτε το αρχείο `.cshrc` σε ένα συντάκτη κειμένου. Μπορείτε να χρησιμοποιήσετε κάποιο συντάκτη που σας παρέχει το γραφικό περιβάλλον σας ή ένα εύκολο συντάκτη για την κονσόλα (π.χ. τον `ee`).

```
% ee ~/.cshrc
```

Προσθέστε την παρακάτω γραμμή `META` τη λίστα των άλλων εντολών.

```
set history = 200
```

Αποθηκεύστε το αρχείο και αναγκάστε το κέλυφος να ξαναδιαβάσει το αρχείο `.cshrc` χρησιμοποιώντας την εντολή του κελύφους `source`.

```
% source .cshrc
```

Ελέγξτε τη σωστή λειτουργία, πληκτρολογώντας:

```
% echo $history
```

Ρύθμιση του `path`

Όταν πληκτρολογείτε μια εντολή, η μεταβλητή `path` (ή `PATH`) ορίζει σε ποιους καταλόγους θα ψάξει το κέλυφος για να την βρει. Αν το σύστημα επιστρέψει ένα μήνυμα του τύπου "command: Command not found", σημαίνει είτε ότι η εντολή δεν υπάρχει καθόλου στο σύστημα, ή απλά ότι δεν υπάρχει στο `path` σας.

Για παράδειγμα, για να εκτελέσετε το πρόγραμμα `units` πρέπει είτε να ορίσετε απευθείας τη πλήρη διαδρομή προς αυτό (`~/units174/bin/units`) ή να έχετε προσθέσει στο `path` σας τον κατάλογο `~/units174/bin`.

Μπορείτε να τον προσθέσετε στο τέλος του υπάρχοντος `path` (το οποίο αντιπροσωπεύεται από το `$path`) χρησιμοποιώντας την παρακάτω εντολή:

```
% set path = ($path ~/units174/bin)
```

Ελέγξτε τη λειτουργία του παραπάνω, προσπαθώντας να εκτελέσετε το `units` ενώ βρίσκεστε σε οποιοδήποτε κατάλογο.

```
% cd  
% units
```

Για να κάνετε αυτή την αλλαγή στο path MONIMH, προσθέστε την παρακάτω γραμμή στο αρχείο .cshrc ΜΕΤΑ τη λίστα των υπόλοιπων εντολών.

```
set path = ($path ~/units174/bin)
```


Σύνδεσμοι

Επεξεργασία για σκληροπυρηνικούς - vi

Υπάρχουν πολλοί τρόποι για να πειράξετε αρχεία κειμένου στο Unix. Ο περισσότερο παράξενος, για όσους δεν τον έχουν δει, είναι ο vi. Ο vi είναι από τους λεγόμενους **κειμενογράφους** οθόνης (screen editor): δεν έχει καθόλου γραφικά, μενού κλπ και οι εντολές δίνονται με άλλο τρόπο (θα δείτε!)

Πρόκειται για τον εξ ορισμού κειμενογράφο σε περιβάλλοντα Unix γιατί

- υπάρχει σε όλες τις εκδόσεις
- η υλοποίησή του είναι ολόιδια σε όλες τις εκδόσεις
- απαιτεί ελάχιστους υπολογιστικούς πόρους
- είναι περισσότερο φιλικός από άλλους κειμενογράφους (οι οποίοι ανακαλύφθηκαν σε σκοτεινές εποχές και τους οποίους δε θα θέλατε να μάθετε)



Πάμε να τον χρησιμοποιήσουμε.

Μπορεί στην έκδοσή σας να έχετε εγκατεστημένο το VIM (VI iMproved) και ορισμένες εντολές να φαίνονται ελαφρά διαφορετικές (με την καλή έννοια, αφού μιλάμε για βελτιωμένη έκδοση, ε?)

Φορτώστε

Για να ξεκινήσετε με ένα νέο αρχείο, απλά δώστε

```
$ vi
```

Αν θέλετε να ξεκινήσετε φορτώνοντας ένα υπάρχον αρχείο, δώστε [vi \[filename\]](#), πχ:

```
$ vi myfile.txt
```

Το αρχείο θα φορτωθεί και ο κέρσορας θα τοποθετηθεί στην αρχή της πρώτης γραμμής. Προσέξτε ότι οι αχρησιμοποίητες γραμμές φαίνονται σαν tilde (~). Αν μία γραμμή δεν έχει τίποτα τότε έχει κάτι ;-) κενό, enter, tab, κάτι έχει.

Κατάσταση εντολών και Κατάσταση εισαγωγής

Στο vi έχουμε δύο καταστάσεις λειτουργίας: εντολών (command mode) και εισαγωγής (insert mode). Αυτό είναι το πλέον μπερδευτικό στοιχείο για τους νέους χρήστες αλλά τελικά δεν είναι και τόσο.

- Στην **κατάσταση εισαγωγής** γράφουμε, όπως ξέρουμε.
- Στην **κατάσταση εντολών** τα πλήκτρα που πατάμε μεταφράζονται σε εντολές διαχείρισης (πχ αποθήκευση ή εκτύπωση). Σε κατάσταση εντολών κάνουμε όλα τα άλλα εκτός από το να γράφουμε.

Εκκινώντας το vi είμαστε σε κατάσταση εντολών. Για να πάμε σε κατάσταση εισαγωγής πατάμε i. Τίποτα δεν αλλάζει αλλά μπορείτε να γράψετε, όπως θα κάνατε με οποιονδήποτε άλλο κειμενογράφο ή επεξεργαστή κειμένου. Γράψτε μερικές γραμμές κειμένου.

Αφού τελειώσατε το γράψιμο μπορείτε να γυρίσετε σε κατάσταση εντολών. Για να το κάνετε, πατήστε το πλήκτρο Esc. Τώρα, οτι πληκτρολογείτε δεν γράφεται αλλά μεταφράζεται σε εντολές.

Ξεκινάμε

Πάμε με μία ασκησούλα:

1. Φορτώστε το vi δίνοντας:

```
$ vi
```

Η οθόνη θα γεμίσει αχρησιμοποίητες γραμμές που σηματοδοτούνται με tilde (~)

```
~
~
~
~
~
```

Βρισκόμαστε σε κατάσταση εντολών.

2. Πατήστε το i για να πάτε σε κατάσταση εισαγωγής. Δώστε δύο τρεις γραμμές κειμένου

```
Hello. I am a vi startet.¶
This is the second line of text. It is a long long long line.¶
This is the third line.¶
~
~
~
```

3. Όταν τελειώσετε, πατήστε 'Esc' για να επιστρέψετε σε κατάσταση εντολών.

4. Πάμε για την πρώτη χρήσιμη εντολή: αποθήκευση. Η εντολή είναι ':w' (προσέξτε την άνω κάτω τελεία πριν το w και εννοείται ότι δε γράφετε τα εισαγωγικά). Μετά το w αφήστε ένα διάστημα και γράψτε το όνομα του αρχείου, πχ:

```
:w hello
```

Με το που ξεκινάτε να γράφετε την εντολή, αυτά που γράφετε εμφανίζονται στο κάτω αριστερό μέρος της οθόνης. Όταν γράψετε όλη την εντολή πατήστε Enter. Θα δείτε μήνυμα επιβεβαίωσης για την αποθήκευση του αρχείου και μερικές πληροφορίες.

Προσέξτε ότι και οι εντολές του vi διαχωρίζουν πεζά γράμματα από κεφαλαία. Για παράδειγμα, σε κατάσταση εντολών αν πατήσουμε i θα ξεκινήσουμε να γράφουμε εκεί που είμαστε ενώ αν πατήσουμε I θα μεταφερθούμε στην αρχή της γραμμής για να γράψουμε.

5. Για να κλείσουμε αυτή τη μικρή εισαγωγή, χρειαζόμαστε ακόμα μία βασικότερη εντολή: το κλείσιμο. Η εντολή είναι η :q. Μην ξεχνάτε ότι για να τη δώσετε πρέπει να είστε σε κατάσταση εντολών.

```
:q
```

Τώρα θα πρέπει να επιστρέψατε στην προτροπή.

Ερώτηση: Μία σύνηθης συμβουλή για αυτούς που δεν είναι σίγουροι σε τι κατάσταση βρίσκονται, είναι να πατήσουν το Esc. Γιατί;

Έξω τώρα!

Ακολουθεί χρήσιμη εντολή, ειδικά για αρχάριους. Γιατί ειδικά για αυτούς; Γιατί στην αρχή είναι σύνηθες να δίνουν εντολές και να μην έχουν συνειδητοποιήσει ότι βρίσκονται ακόμα σε κατάσταση εισαγωγής (ή το ανάποδο) και να κάνουν το αρχείο χάλια.

Για να κλείσουμε το vi χωρίς να αποθηκευθούν οι αλλαγές, δίνουμε μία πειραγμένη :q, βάζουμε θαυμαστικό (!) αμέσως μετά.

```
:q!
```

Αυτό μας γυρίζει στην προτροπή χωρίς αποθηκεύσεις και χωρίς ερωτήσεις. Άρα θα πρέπει να χρησιμοποιείται με σύνεση, ε;

Μετακίνηση

Τα πλέον συνηθισμένα πλήκτρα είναι τα βελάκια μετακίνησης του κέρσορα. Για να μπορέσετε να τα χρησιμοποιήσετε, θα πρέπει να είστε σε κατάσταση εντολών (δείτε μόνοι σας τι γίνεται όταν τα χρησιμοποιείτε σε κατάσταση εισαγωγής!). Στο νί δεν είναι απαραίτητο να χρησιμοποιήσετε τα βελάκια για να μετακινηθείτε. Για λόγους ιστορικούς (ή μάλλον αρχαιολογικούς, τότε που τα πληκτρολόγια δεν είχαν κουμπιά βελάκια), στο νί μπορούμε να μετακινηθούμε και με τη χρήση των παρακάτω πλήκτρων:

- h - αριστερά
- j - κάτω
- k - πάνω
- l - δεξιά

Ασκησούλα

1. Φορτώστε το νί.
2. Στο αρχείο βάλτε μερικές γραμμές κειμένου.
3. Μετακινηθείτε με τα βελάκια.
4. Τώρα δοκιμάστε να μετακινηθείτε χωρίς τα βελάκια αλλά με τα γράμματα.
5. Γράψτε ακόμα μερικές γραμμές.
6. Μετακινηθείτε με τα γράμματα.
7. Φύγετε χωρίς να αποθηκεύσετε (θυμηθείτε την :q!)

Τα πλήκτρα

Βλέπετε ότι ανάλογα με το αν είστε σε κατάσταση εντολών ή εισαγωγής, τα πλήκτρα h, j, k, l έχουν διαφορετική λειτουργικότητα. Αυτό ισχύει για όλα τα πλήκτρα Ένα πολύ συνηθισμένο λάθος για τους νιούφιδες του νί είναι να πληκτρολογούν ενώ είναι σε κατάσταση εντολών - το βάθος της καταστροφής εξαρτάται αποκλειστικά από τα πλήκτρα που τυχαίνει να πατηθούν!!!

Μετακίνηση

Θυμηθείτε ότι πρέπει να είστε σε κατάσταση εντολών.

Εντολή	Τι κάνει
h	Αριστερά μία θέση
l	Δεξιά μία θέση
j	Κάτω μία θέση
k	Πάνω μία θέση
nG	Μεταφέρει το χρήστη στη n-οστή γραμμή.
G	Σκέτο το G θα σας πάει στην τελευταία γραμμή
H	Κορυφή της οθόνης
L	Τέρμα της οθόνης
^U	Κύλιση σελίδας προς τα πάνω (page up)

^D	Κύλιση σελίδας προς τα κάτω (page down)
----	---

Μαζική εκτέλεση εντολής

Σε εάν κείμενο κι ενώ είστε σε κατάσταση εντολών και ο κέρσορας είναι προ το τέλος μιάς γραμμής, δώστε 3h. Θα δείτε ότι ο κέρσορας μετακινείται 3 θέσεις αριστερά. Αν δώσετε 3l θα σας πάει 3 θέσεις δεξιά. Τι κάνατε; Αντί να πατήσετε 3 φορές το h, διατάξατε το νι να εκτελέσει 3 φορές την εντολή.

Αυτό συμβαίνει με πολλές εντολές του νι: γράφοντας έναν αριθμό ν πριν από μία εντολή, διατάζουμε το νι να εκτελέσει την εντολή ν φορές (κι έτσι γλυτώνουμε το να γράψουμε εμείς την εντολή ν φορές).

Διαγραφή κειμένου

Θυμηθείτε ότι πρέπει να είστε σε κατάσταση εντολών.

Εντολή	Τι κάνει
x	Διαγράφει το χαρακτήρα κάτω από τον κέρσορα (όπως το πλήκτρο Del)
dw	Διαγράφει μία λέξη
dd	Διαγράφει την τρέχουσα γραμμή
d\$	Διαγράφει από τον κέρσορα μέχρι το τέλος της γραμμής

Αναζήτηση

Θυμηθείτε ότι πρέπει να είστε σε κατάσταση εντολών.

Εντολή	Τι κάνει
/λέξη	Αναζητά τη λέξη
n	Αναζητά την επόμενη εμφάνιση της λέξης
N	Αναζητά την προηγούμενη εμφάνιση της λέξης

Αντικατάσταση

Η γενική μορφή της εντολής αντικατάστασης είναι

:αρχή,τέλοςs/τι_ψάχνω/με_τι_αλλάζει

Όπου

- αρχή από που ξεκινάει το ψάξιμο
- τέλος που τελειώνει
- τι_ψάχνω το κείμενο που θέλω να αντικατασταθεί
- με_τι_αλλάζει το νέο κείμενο το οποίο θα μπει στη θέση του παλιού

Ειδικές τιμές για τις θέσεις:

- . (τελεία) η θέση που βρίσκεται ο κέρσορας

- \$ τελευταία γραμμή του αρχείου

Παραδείγματα

:2,6s/mad/sane

Ψάχνει από τη γραμμή 2 μέχρι τη γραμμή κι όπου βρίσκει mad το αντικαθιστά με sane

:2,\$s/mad/sane

Ψάχνει από τη γραμμή 4 μέχρι το τέλος του αρχείου κι όπου βρίσκει mad το αντικαθιστά με sane

::\$s/mad/sane

Ψάχνει από την τρέχουσα γραμμή (τη γραμμή που βρίσκεται ο κέρσορας) μέχρι το τέλος του αρχείου κι όπου βρίσκει mad το αντικαθιστά με sane

Αντιγραφή και επικόλληση

Θυμηθείτε ότι πρέπει να είστε σε κατάσταση εντολών.

Εντολή	Τι κάνει
yy	Αντιγράφει την τρέχουσα γραμμή
p	Επικολλά το αντιγραμμένο κείμενο μετά τον κέρσορα.
P	Επικολλά το αντιγραμμένο κείμενο πριν τον κέρσορα.

Γραφικό περιβάλλον

Κανονισμός Κατάρτισης Ειδικότητας

Η κατάρτιση που θα πρέπει να κάνετε, σε σχέση με τα Λειτουργικά Συστήματα (γενικότερα) και το Unix (ειδικότερα), όπως καθορίζεται από τον Κανονισμό Κατάρτισης, ορίζεται ως εξής (τα παρακάτω αποσπάσματα είναι κατά λέξη αντιγραφή από τον Κανονισμό):

Ανάλυση Επαγγελματικών Δραστηριοτήτων

Ο διπλωματούχος της ειδικότητας "ΤΕΧΝΙΚΟΣ ΕΦΑΡΜΟΓΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΠΟΛΥΜΕΣΑ (MULTIMEDIA)" θα πρέπει να:

....

1. γνωρίζει τη βασική ορολογία πληροφορικής τόσο στον τομέα του υλικού (Hardware), όσο και στον τομέα του λογισμικού (Software) που αντιστοιχεί στην ειδικότητά του.

....

3. έχει την ικανότητα διαχείρισης του λειτουργικού συστήματος UNIX,

.....

Τελικές Εξετάσεις Πιστοποίησης

Θεωρητικό μέρος

.....

Για την πιστοποίηση της επαγγελματικής ικανότητας, κατά το Θεωρητικό Μέρος, οι υποψήφιοι της ειδικότητας Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα (Multimedia) εξετάζονται σε γενικά θέματα επαγγελματικών γνώσεων και ικανοτήτων και επίσης σε ειδικές επαγγελματικές γνώσεις και ικανότητες, που περιλαμβάνονται αποκλειστικά στη στοχοθεσία του Θεωρητικού Μέρους της ειδικότητας.

A. Γενικές γνώσεις:

.....

Λειτουργικό Σύστημα Unix

B. Ειδικές γνώσεις:

.....

Πρακτικό μέρος

.....

Για την πιστοποίηση της επαγγελματικής ικανότητας, κατά το Πρακτικό Μέρος, οι υποψήφιοι της ειδικότητας Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα (Multimedia), εξετάζονται σε γενικά θέματα επαγγελματικών γνώσεων και ικανοτήτων και επίσης σε ειδικές επαγγελματικές γνώσεις και ικανότητες, που περιλαμβάνονται αποκλειστικά στη στοχοθεσία του πρακτικού μέρους της ειδικότητας.

.....

2. ... γνωριμία με το λειτουργικό σύστημα UNIX (LINUX), έτσι ώστε να έχουν δυνατότητα να χρησιμοποιούν το ανάλογο περιβάλλον (ΛΕΙΤΟΥΡΓΙΚΟ ΣΥΣΤΗΜΑ UNIX),

.....

Κατάλογος ερωτήσεων

A. ΟΜΑΔΑ ΕΡΩΤΗΣΕΩΝ ΓΕΝΙΚΩΝ ΓΝΩΣΕΩΝ

...

- Ποια είναι τα δομικά μέρη του UNIX;
- Ποιες είναι οι λειτουργίες του πυρήνα (kernel) του UNIX;
- Τι είναι οι «χαρακτήρες μπαλαντέρ» και ποια η χρήση τους;
- Σε ποιο δομικό μέρος του UNIX ερμηνεύονται οι εντολές;
- Τι κάνουμε προκειμένου να αποφύγουμε το output μιας επεξεργασίας στο παρασκήνιο να διακόψει την επεξεργασία στο προσκήνιο;
- Σε ποιο αρχείο ορίζεται ο τύπος του τερματικού, ώστε η έξοδος να παρουσιάζεται ανάλογα με το τερματικό που χρησιμοποιείται στο Unix;

.....

Η μεγάλη εικόνα

Τα παραπάνω αναφέρονται στον Κανονισμό Κατάρτισης της Ειδικότητας “Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα (Multimedia)”, 2η Περίοδος 2009 του ΟΕΕΚ. Μπορείτε να τον βρείτε στο site του ΙΕΚ Χαϊδαρίου, iek-chai.att.sch.gr, και είναι ο οδηγός ο οποίος περιέχει τα πάντα για την ειδικότητά σας.

Ερωτήσεις πιστοποίησης 2008

Αυτές οι ερωτήσεις βρέθηκαν στο διαδίκτυο, με αυτόν τον τίτλο. Μπορεί να είναι στην τράπεζα θεμάτων, μπορεί και να μην είναι - θα μπορούσαν να είναι ;-))

ΟΜΑΔΑ Δ. UNIX

1. Το UNIX είναι multiuser ή multitasking Λ. Σ.; Τι σημαίνει ο κάθε όρος;
2. Τι σημαίνει παροχέτευση (spooling) και ποια τα πλεονεκτήματα αυτής της μεθόδου;
3. Σε ποιο αρχείο ορίζεται ο τύπος του τερματικού, ώστε η έξοδος να παρουσιάζεται ανάλογα με το τερματικό που χρησιμοποιείται;
4. Να δώσετε ένα παράδειγμα στο οποίο η χρήση ψευδωνύμου θα ήταν χρήσιμη.
5. Τι είναι οι χαρακτήρες μπαλαντέρ και ποια η χρήση τους;
6. Ποια είναι τα δομικά μέρη του UNIX;
7. Ποιες είναι οι λειτουργίες του πυρήνα (kernel);
8. Ο επεξεργαστής κειμένου νι περιλαμβάνεται
 - α. Στον πυρήνα
 - β. Στα ενσωματωμένα βοηθητικά προγράμματα
 - γ. Στο κέλυφος (φλοιό)

9. Σε ποιο δομικό μέρος του UNIX ερμηνεύονται οι διαταγές;
10. Τι κάνει και πότε δημιουργείται το αρχείο «.profile» κάποιου χρήστη;
11. Τι είναι η μεταβλητή history;
12. Σε ποιο δομικό μέρος του UNIX ανήκει η history;
13. Τι κάνουμε προκειμένου να αποφύγουμε το output μιας επεξεργασίας στο παρασκήνιο να διακόψει την επεξεργασία στο προσκήνιο;
14. Αν ως απλός χρήστης ξεχάσατε το συνθηματικό σας προκειμένου να μείτε στο σύστημα, υπάρχει δυνατότητα να το βρείτε κάπου γραμμένο; Αν όχι εσείς, μπορεί να το βρει κάποιος άλλος;
15. Μια εκτεταμένη λίστα πληροφοριών αρχείων - καταλόγων θα άρχιζε από drwxrwxrwx κλπ με πιθανότητα παύλας σε μερικούς από τους παραπάνω χαρακτήρες. Να αναλύσετε τι σημαίνουν.
16. Να εξηγήσετε τι είναι η διεργασία (process). Έχει ο χρήστης τη δυνατότητα να φτιάξει μια διεργασία; Να δώσετε παράδειγμα.
17. Η δυνατότητα δημιουργίας διεργασίας είναι ένδειξης multiuser ή multitasking Λ.Σ. ;
18. Η δημιουργία πολλών παρασκηνιακών διεργασιών έχει επιπτώσεις στη λειτουργία του συστήματος;
19. Είναι δυνατή η προσωρινή χρήση εντολών κελύφους (φλοιού), ενώ βρισκόμαστε στον νι; Αν ναι, από ποια κατάσταση και πώς;
20. Πού περιλαμβάνονται οι ειδικά διαμορφωμένες διαταγές κελύφους (φλοιού) του υπερχρήστη;

Εκδόσεις - Διάθεση

Διάθεση και Δικαιώματα

Οι σημειώσεις αυτές είναι βασισμένες στα

- Μαθήματα Unix για αρχάριους” του Μ.Κιαγιά (<http://www.freebsdworld.gr/diktia/UNIX/index.html>), τα οποία βασίστηκαν στο
- “Unix tutorial for beginners” του Μ. Stonebank, <http://info.ee.surrey.ac.uk/Teaching/Unix/unixintro.html>

Προσθήκες σε κείμενο έγιναν από

- wikipedia (en.wikipedia.org)
- βικιπέδια (el.wikipedia.org)
- www.adimos.gr (επιπλέον ερωτήσεις πιστοποίησης)

Προσθήκες σε εικόνες έγιναν από

- wikimedia commons (<http://commons.wikimedia.org/>)
- <http://www.unix.org/>



Ο πίνακας του εξώφυλλου είναι ανατολίτικη ελαιογραφία του Jean Leon Gerome με τίτλο "Πυρρίχιος χορός" (εναλλακτικά "Χορός των μαχαιριών εμπρός στα αιγυπτιακά χαλάσματα").

Όλο το υλικό που αναφέρεται παραπάνω, αλλά κι αυτές οι σημειώσεις, είναι CC BY-NC-SA 2.0. Για να δείτε την άδεια Creative Commons που χρησιμοποιείται μπορείτε να πάτε στο <http://creativecommons.org/licenses/by-nc-sa/2.0/>.



Εκδόσεις

1.0

[2/2011] Αρχικό κείμενο ώστε να είναι στα πλαίσια του αντίστοιχου μαθήματος της ειδικότητας Τεχνικός Εφαρμογών Πληροφορικής με Πολυμέσα (Multimedia) των ΙΕΚ. Καλύπτονται και τα δύο εξάμηνα (Γ και Δ).

1. Δεν υπάρχει κεφάλαιο για τους συνδέσμους.
2. Δεν υπάρχει κεφάλαιο για το γραφικό περιβάλλον.

