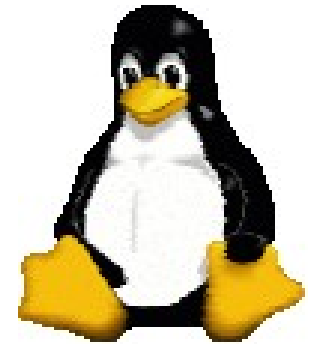




Προγραμματισμός Διαχείρισης Συστημάτων Ι

Μάθημα 5ο Φλοιούς Linux



Μιχαηλίδης Παναγιώτης

Φλοιός

- Ο φλοιός (shell) είναι ένα πρόγραμμα που παρέχει μια διεπαφή επικοινωνίας γραμμής εντολών ανάμεσα στον χρήστη και στον πυρήνα του Linux.
- Ο φλοιός είναι πρόγραμμα που ξεκινά αμέσως μετά την επιτυχή σύνδεση (login) στο σύστημα Linux.
- Είναι ένας διερμηνευτής εντολών που διαβάζει, ερμηνεύει τις εντολές που πληκτρολογεί ο χρήστης και τις στέλνει στον πυρήνα του συστήματος για εκτέλεση. Στην συνέχεια παρουσιάζει τα αποτελέσματα της εκτέλεσης στον χρήστη.

Προγράμματα φλοιών

- Σε κάθε σύστημα Linux υπάρχουν αρκετά διαθέσιμα και δημοφιλή προγράμματα φλοιού:
 - Φλοιός Bourne (sh)
 - Φλοιός C (csh)
 - Φλοιός Korn (ksh)
 - Φλοιός Bourne Again (bash) - Χρησιμοποιείται σχεδόν σε όλες τις διανομές Linux.
- Ο φλοιός που θα χρησιμοποιήσουμε είναι θέμα προτίμησης, δύναμης και διαθεσιμότητας.

Φλοιός bash

- Ο φλοιός bash είναι το πρότυπο και προεπιλεγμένο πρόγραμμα στις διανομές Linux.
- Είναι υπερσύνολο του φλοιού Bourne shell (sh).
- Περιλαμβάνει χαρακτηριστικά από άλλους φλοιούς όπως sh, csh, tcsh και ksh.
- Ο bash δημιουργήθηκε και διανέμεται από το έργο GNU.
- Λογισμικό και εγχειρίδιο αναφοράς για το bash υπάρχουν στις διευθύνσεις: <http://www.gnu.org/software/bash>
<http://www.gnu.org/manual/bash>.

Ευκολίες φλοιού bash

- Διερμηνεία εντολών
- Ενσωματωμένες (ή εσωτερικές) εντολές
- Ιστορικό εντολών
- Αυτόματη συμπλήρωση εντολών
- Μεταχαρακτήρες φλοιού
- Ανακατεύθυνση εισόδου/εξόδου
- Διασωλήνωση
- Αντικατάσταση εντολών
- Μεταβλητές

Ευκολίες φλοιού bash

- Προσαρμογή του περιβάλλοντος φλοιού στις απαιτήσεις του χρήστη
- Εκτέλεση παρασκηνιακών εργασιών
- Έλεγχος εργασιών
- Σενάρια φλοιού
- Προγραμματισμός φλοιού

Αναζήτηση φλοιού

- Για να δούμε ποιο πρόγραμμα φλοιού χρησιμοποιούμε στο σύστημα Linux μπορούμε:
 - Να εξετάσουμε την προτροπή του συστήματος (\$, %).
 - Να πληκτρολογήσουμε την εντολή `echo $SHELL`.

```
$ echo $SHELL  
/bin/bash  
$
```

- Τα προγράμματα φλοιών βρίσκονται κάτω από τον κατάλογο `/bin`.

Αλλαγή φλοιού

```
$ ls /bin/*sh*  
bash  ksh  tcsh  zsh-4.0.4  
$
```

- Για να αλλάξουμε το προεπιλεγμένο φλοιό ή να επιλέξουμε ένα φλοιό της προτίμησης μας χρησιμοποιούμε την εντολή `chsh` (change shell).

```
$ chsh  
Changing shell for panosm  
Password:  
New shell [/bin/sh]: /bin/bash  
Shell changed  
$
```


Εσωτερικές εντολές

- Ο φλοιός μπορεί να διερμηνεύσει δύο τύπους εντολών:
 - Εξωτερικά ή Εκτελέσιμα προγράμματα (*external programs*): είναι εντολές ή εκτελέσιμα αρχεία που έχουν φυσική υπόσταση στη δομή αρχείων του Linux. Έτσι, τα εκτελέσιμα αρχεία βρίσκονται σε καταλόγους όπως `/bin`, `/sbin` κλπ. Παράδειγμα, για την εντολή `ls` ο φλοιός εντοπίζει το εκτελέσιμο αρχείο που βρίσκεται στον κατάλογο `/bin` και το εκτελεί.
 - Εσωτερικές εντολές φλοιού (*shell internal commands*): είναι εντολές που δεν έχουν φυσική υπόσταση ως αρχεία στο δίσκο. Είναι εντολές τις οποίες αναγνωρίζει και εκτελεί εσωτερικά. Παραδείγματα εσωτερικών εντολών είναι οι `echo`, `cd`, `pwd`, `history`.

Ιστορικό εντολών

- Ο φλοιός bash υποστηρίζει ένα ιστορικό εντολών που καταγράφει τις εντολές που έχουμε πληκτρολογήσει. Μέσα από το ιστορικό μπορούμε να επανακτελέσουμε ορισμένες εντολές.
- Για να εμφανίσουμε το ιστορικό εντολών του φλοιού χρησιμοποιούμε την εντολή `history`.

```
$ history
1  dir
2  ls
3  startx
4  startkde
5  ld
6  ls
7  cd tmp
8  ls
```

Ιστορικό εντολών

- Για να επανακτελέσουμε μια εντολή από το ιστορικό εντολών μπορεί να γίνει με δύο τρόπους:
 - Μπορεί να γίνει με τα πλήκτρα επάνω και κάτω βέλους.
 - Να χρησιμοποιήσουμε τον τελεστή ! και δίπλα τον αριθμό εντολής που εμφανίζεται από την εντολή history. Δηλαδή, !αριθμός

Αυτόματη συμπλήρωση εντολών

- Ο φλοιός bash έχει την δυνατότητα να συμπληρώσει αυτόματα ένα όνομα αρχείου ή όνομα εντολής που έχουμε πληκτρολογήσει στην προτροπή μέσω του πλήκτρου Tab.
- Πληκτρολογούμε στην προτροπή ορισμένους χαρακτήρες ενός ονόματος και πατώντας το πλήκτρο Tab θα συμβεί ένα από τα ακόλουθα:
 - Αν υπάρχει ένα όνομα που ταιριάζει τότε θα συμπληρωθεί ολόκληρο το όνομα.
 - Αν υπάρχουν περισσότερα ονόματα που ταιριάζουν τότε θα εμφανιστεί μια λίστα ονομάτων που έχουν κοινούς τους χαρακτήρες.

Αρχεία Προτύπων

- Κάθε φλοιός (και πρόγραμμα) ανοίγουν αυτόματα τρία αρχεία κατά την εκκίνηση.
- Πρότυπη είσοδο - `stdin` (standard input)
 - Εξ ορισμού είναι το πληκτρολόγιο
- Πρότυπη έξοδο - `stdout` (standard output)
 - Εξ ορισμού είναι η οθόνη
- Πρότυπο σφάλματος - `stderr` (standard error)
 - Εξ ορισμού είναι η οθόνη

Ανακατεύθυνση εξόδου

- Ανακατεύθυνση εξόδου είναι όταν θέλουμε να στείλουμε την έξοδο μιας διεργασίας σε ένα αρχείο αντί στην πρότυπη έξοδο, δηλ. στην οθόνη. Για να ανακατευθύνουμε την έξοδο χρησιμοποιούμε το τελεστή `>` ή `>>`.
- Σύνταξη: `commandname > filename`
 - Στέλνει την έξοδο της εντολής `commandname` στο αρχείο με όνομα `filename`. Το αρχείο `filename` δημιουργείται αν δεν υπάρχει ή αντικαθιστά τα περιεχόμενα του αν υπάρχει.

Ανακατεύθυνση εξόδου

- Παράδειγμα:

```
$ ls -al > listfile  
$ cat listfile
```

- Σύνταξη: `commandname >> filename`

– Προσαρτά την έξοδο της εντολής `commandname` στο τέλος του υπάρχοντος αρχείου με όνομα `filename`. Το αρχείο `filename` δημιουργείται αν δεν υπάρχει.

```
$ ls /bin >> listfile  
$ cat listfile
```

Ανακατεύθυνση εισόδου

- Ανακατεύθυνση εισόδου είναι όταν θέλουμε η διεργασία να διαβάσει από ένα αρχείο αντί την πρότυπη είσοδο, δηλ. το πληκτρολόγιο. Για να ανακατευθύνουμε την είσοδο χρησιμοποιούμε το τελεστή `<` ή `<<`.
- Σύνταξη: `commandname < filename`
 - Η εντολή `commandname` διαβάσει τα περιεχόμενα του αρχείου με όνομα `filename`. Αν το αρχείο `filename` δεν υπάρχει εμφανίζεται σφάλμα.

Ανακατεύθυνση εισόδου

- Παράδειγμα:

```
$ cat < testfile.txt  
$
```

- Σύνταξη: `commandname << word`
 - Διαβάζει χαρακτήρες από το πληκτρολόγιο και σταματά μέχρι ο φλοιός να συναντήσει τη λέξη `word` και έπειτα εκτελεί την εντολή `commandname`.

```
$ cat << end
```

Διασωλήνωση

- Ο φλοιός μας επιτρέπει να συνδέσουμε την πρότυπη έξοδο μιας διεργασίας σαν πρότυπη είσοδο μιας άλλης διεργασίας, συνδέοντας τις διεργασίες μέσω του τελεστή διασωλήνωσης |.
- Σύνταξη: `commandname1 | commandname2`
 - Η έξοδος της εντολής `commandname1` θα περάσει ως είσοδος στην εντολή `commandname2`.
 - Μπορούμε να συνδέσουμε οποιοδήποτε αριθμό εντολών με διασωλήνωση.

Διασωλήνωση

- Παράδειγμα:

```
$ ls -l | wc -l  
$ man bash | grep "history"  
$ grep 'bash$' /etc/passwd | wc -l  
$ ls /usr | grep bin | more  
$ ls -l | grep '^d.....' | wc -l
```

- Υπάρχει μια χρήσιμη εντολή, η `tee` που μας επιτρέπει να αποθηκεύουμε την έξοδο μιας διασωλήνωσης σε αρχείο ενώ ταυτόχρονα ίδια έξοδος στέλνεται ως είσοδο στην διασωλήνωση.

```
$ who | tee users | wc -l  
$ cat users
```

Αντικατάσταση εντολών

- Μια εντολή που περικλείεται με βαρείες (` `) εκτελείται και η πρότυπη έξοδος της εισάγεται μέσα στην εντολή στη θέση της.
- Παράδειγμα:

```
$ ls `pwd`  
$ echo the date today is `date`  
$ wc `ls `pwd``
```

Συνδυασμός τελεστών <, >, |, ' '

- Παραδείγματα:

```
$ who | sort > users  
$ echo there are `who|wc -l` users  
$ ls `pwd` | grep '^.rw.....' | wc -l
```

Περιβάλλον φλοιού

- Οι προεπιλεγμένες ρυθμίσεις του περιβάλλοντος φλοιού είναι συνήθως επαρκείς για τους περισσότερους χρήστες.
- Ωστόσο, μερικοί χρήστες αλλάζουν τις ρυθμίσεις του περιβάλλοντος φλοιού που εργάζονται με βάση τις ανάγκες και προτιμήσεις τους.
- Με την εντολή `env` μπορούμε να δούμε τις ρυθμίσεις του περιβάλλοντος φλοιού που εργαζόμαστε.

Περιβάλλον φλοιού - Παράδειγμα

```
$ env
HOSTNAME=localhost
DM_CONTROL=/var/run/xdmctl
TERM=xterm
SHELL=/bin/bash
HISTSIZE=1000
XDM_MANAGED=/var/run/xdmctl/xdmctl-
:0,maysd,mayfn,sched,rsvd,method=classic
GTK2_RC_FILES=/usr/share/themes/Galaxy/gtk-
2.0/gtkrc:/etc/gtk-2.0/gtkrc:/home/panos/.gtkrc-
2.0:/home/panos/.kde/share/config/gtkrc
TMPDIR=/home/panos/tmp
KDE_FULL_SESSION=true
USER=panos
MAIL=/var/spool/mail/panos
DESKTOP_SESSION=01KDE
PATH=/usr/bin:/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin/:
/usr/games:/home/panos/bin
```

Στοιχεία ρυθμίσεων φλοιού

- Υπάρχουν τρεις βασικοί τύποι ρυθμίσεων για την προσαρμογή του περιβάλλοντος φλοιού:
 - Αρχεία εκκίνησης
 - Μεταβλητές περιβάλλοντος
 - Ψευδώνυμα

Αρχεία εκκίνησης

- Αρχείο εκκίνησης είναι ένα αρχείο που εκτελείται κατά την εκκίνηση του φλοιού και περιέχει ρυθμίσεις που προσαρμόζουν τη λειτουργικότητα του περιβάλλοντος εργασίας.
- Το κύριο αρχείο εκκίνησης στον bash είναι το αρχείο `/etc/profile` που ελέγχει το περιβάλλον bash για όλο το σύστημα και όλους τους χρήστες του.
- Στον προσωπικό κατάλογο κάθε χρήστη περιέχουν ένα ή πολλά αρχεία εκκίνησης bash που επηρεάζουν το προσωπικό περιβάλλον φλοιού.
 - `.bash_profile`, `.bashrc`, `.bash_logout`

Αρχεία εκκίνησης

- Όταν συνδεόμαστε στο σύστημα Linux, ο φλοιός bash εκτελεί ορισμένα από τα παρακάτω αρχεία εκκίνησης:
 - /etc/profile
 - .bash_profile
- Όταν ξεκινάμε ένα νέο φλοιό bash ή ανοίγουμε ένα παράθυρο τερματικού, τότε ο φλοιός εκτελεί το αρχείο εκκίνησης .bashrc.
- Όταν αποσυνδεόμαστε από το σύστημα Linux τότε ο φλοιός εκτελεί το αρχείο .bash_logout.

.bash_profile και .bashrc

- Όταν ξεκινάει ο φλοιός κατά την σύνδεση διαβάζει το αρχείο .bash_profile και εκτελούνται οι εντολές που περιέχει.

```
$ cat .bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/bin

export PATH
unset USERNAME
```

.bash_profile και .bashrc

- Στο αρχείο `.bash_profile` υπάρχουν αναφορές στο αρχείο `.bashrc`. Το αρχείο `.bash_profile` εκτελείται μόνο όταν συνδέεστε τον λογαριασμό σας ενώ το `.bashrc` εκτελείται κάθε φορά που ξεκινάμε ένα δευτερεύοντα φλοιό.

Ρυθμίσεις αρχείων εκκίνησης

- Η προσθήκη μιας εντολής που θα εκτελείται αυτόματα όταν κάθε χρήστης του συστήματος κάνει login πρέπει να γίνεται στο αρχείο `/etc/profile`. Ωστόσο χρειάζονται μόνο ο υπερχρήστης μπορεί να επεξεργαστεί αυτό το αρχείο.
- Η προσθήκη μιας εντολής που θα εκτελείται όταν ξεκινά ένας φλοιός σύνδεσης για το δικό μας λογαριασμό, πρέπει να γίνει στο αρχείο `.bash_profile`.
- Η προσθήκη μιας εντολής που εκτελείται όταν ξεκινάμε ένα φλοιό bash, γίνεται στο αρχείο `.bashrc`.

Μεταβλητές φλοιού

- Οι μεταβλητές φλοιού έχουν ένα όνομα και μια τιμή.
 - Οι επιτρεπτοί χαρακτήρες του ονόματος είναι τα γράμματα, ψηφία και η κάτω παύλα (_).
 - Το όνομα πρέπει να αρχίζει με ένα γράμμα ή με μια κάτω παύλα.
 - Σε μια μεταβλητή ανατίθεται μια τιμή.
- Ανάθεση μεταβλητών φλοιού
 - Η μεταβλητή δημιουργείται μόλις αναθέσει μια τιμή σε αυτή.
 - Η ανάθεση τιμής σε μια μεταβλητή γίνεται ως εξής:
varname = value.
 - PS1="bash > "
 - DATE=`date`

Μεταβλητές φλοιού

- Για να προσπελάσουμε τις τιμές των μεταβλητών, προσθέτουμε ένα \$ μπροστά στο όνομα της μεταβλητής.

```
$ firstname = Panos  
$ lastname = Michailidis  
$ echo My name is $firstname $lastname  
My name is Panos Michailidis
```

Μεταβλητές φλοιού

- Οι μεταβλητές φλοιού χωρίζονται σε δύο κατηγορίες:
 - Κανονικές μεταβλητές
 - Αρχικοποιούνται από τον φλοιό.
 - Δεν μεταβιβάζονται στις διεργασίες που εκτελούνται από τον τρέχοντα φλοιό.
 - Συνήθως τους αποδίδονται ονόματα με μικρά γράμματα.
 - Μεταβλητές περιβάλλοντος
 - Αρχικοποιούνται κατά την εκκίνηση (login).
 - Μεταβιβάζονται σε όλες τις διεργασίες που εκτελούνται από τον τρέχοντα φλοιό.
 - Συνήθως τους αποδίδονται ονόματα με κεφαλαία γράμματα.

Μεταβλητές περιβάλλοντος

- Οι προκαθορισμένες μεταβλητές περιβάλλοντος συνήθως περιέχονται στα αρχεία εκκίνησης.
- Οι μεταβλητές περιβάλλοντος τις αναγνωρίζει ο φλοιός και ρυθμίζουν διάφορες λειτουργίες του ίδιου ή και άλλων προγραμμάτων.

Μεταβλητές περιβάλλοντος

- Ο φλοιός αναγνωρίζει πολλές προκαθορισμένες μεταβλητές περιβάλλοντος και αναφέρουμε μερικές απο αυτές:
 - \$TERM: ο τύπος του τερματικού
 - \$PATH: η διαδρομή που γίνεται αναζήτηση για εντολές
 - \$USER: το όνομα του τρέχοντος χρήστη
 - \$HOME: το πλήρες όνομα διαδρομής του προσωπικού καταλόγου του χρήστη
 - \$MAIL: το πλήρες όνομα διαδρομής για εισερχόμενη αλληλογραφία

Μεταβλητές περιβάλλοντος

- \$EDITOR: ο προεπιλεγμένος επεξεργαστής κειμένου
- \$SHELL: το πλήρες όνομα διαδρομής του φλοιού
- \$PS1: ορισμός της ένδειξης προτροπής

```
$ echo HOME=$HOME, PATH=$PATH  
HOME=/home/panos, PATH=/bin:/usr/bin:/usr/sbin  
$ echo MAIL=$MAIL  
MAIL=/var/mail/panos
```

Εντολές για τις μεταβλητές

- Η εντολή `set` εμφανίζει μια λίστα όλων των μεταβλητών και των αντίστοιχων τιμών τους.
- Οι εντολές `env` και `printenv` εμφανίζει μόνο τις μεταβλητές περιβάλλοντος.
- Η εντολή `export` μετατρέπει από μια κανονική μεταβλητή σε μεταβλητή περιβάλλοντος και είναι διαθέσιμη σε όλες τις διεργασίες.

```
$ name=panos; number=10
$ echo $name $number
$ export name
$ sh
$ echo $name $number
```

Μεταβλητή περιβάλλοντος PATH

- Η μεταβλητή PATH περιέχει μια λίστα καταλόγων, διαχωρισμένων με τον χαρακτήρα : στους οποίους θα αναζητήσει ο φλοιός αυτόματα για εντολές ή εκτελέσιμα αρχεία.
- Όταν εκτελούμε μια εντολή ο φλοιός πραγματοποιεί αναζήτηση σε κάθε κατάλογο της μεταβλητής PATH για εκτελέσιμο αρχείο που να έχει όνομα ίδιο με το όνομα της εντολής. Αν βρεθεί το ταίριασμα, το αρχείο εκτελείται, διαφορετικά εμφανίζεται σφάλμα ότι δεν υπάρχει τέτοιο πρόγραμμα.
- Στην περίπτωση που υπάρξει σφάλμα για ένα πρόγραμμα πρέπει να ενσωματώσουμε την πλήρη διαδρομή του προγράμματος στην PATH.

Μεταβλητή περιβάλλοντος PATH

- Παράδειγμα: `$PATH=/bin:/usr/bin:/usr/sbin`
 - Για να εκτελέσουμε την εντολή `ls`, ο φλοιός θα αναζητήσει πρώτα στον κατάλογο `/bin` να βρεί το εκτελέσιμο αρχείο με όνομα `ls`.
 - Αν δεν βρεθεί το εκτελέσιμο αρχείο, τότε ο φλοιός θα αναζητήσει στον επόμενο κατάλογο `/usr/bin` το αρχείο `ls`.
 - Η διαδικασία επαναλαμβάνεται για κάθε ένα από τους υπόλοιπους καταλόγους στην `PATH`.
 - Στην περίπτωση που δεν βρεθεί το αρχείο και έχει εξαντλήσει όλους τους καταλόγους της `PATH` τότε ο φλοιός εμφανίζει ένα μήνυμα ότι δεν βρέθηκε η `ls`.

PATH - Παράδειγμα

- Έστω ο φλοιός δεν βρίσκει το πρόγραμμα traceroute. Πρώτα, βρίσκουμε την πλήρη διαδρομή της εντολής traceroute με την χρήση της εντολής whereis.

```
$ traceroute
bash: traceroute: command not found
$ whereis traceroute
traceroute: /usr/sbin/traceroute
```

- Για να εκτελεστεί η εντολή traceroute υπάρχουν τέσσερις τρόποι.

PATH - Παράδειγμα

- 1ος τρόπος: Να πληκτρολογήσουμε την πλήρη διαδρομή της εντολής traceroute.

```
$ /usr/sbin/traceroute
```

- 2ος τρόπος: Να προσθέσουμε την πλήρη διαδρομή της traceroute στην μεταβλητή PATH.

```
$ export PATH=$PATH:/usr/sbin  
$ echo $PATH  
$ traceroute
```

Με αυτό τον τρόπο θα εκτελεστεί η εντολή που θέλουμε αλλά θα είναι προσωρινό όσο είμαστε συνδεδεμένοι στο σύστημα.

PATH - Παράδειγμα

- 3ος τρόπος: Αν θέλουμε η `traceroute` να εκτελείται κάθε φορά που συνδεόμαστε στο σύστημα, πρέπει να προσθέσουμε την πλήρη διαδρομή της εντολής στο αρχείο `.bash_profile` (ή `.bashrc`) που βρίσκεται στον προσωπικό μας κατάλογο στην `PATH`.
- 4ος τρόπος: Αν θέλουμε η `traceroute` να εκτελείται για όλους τους χρήστες, πρέπει να προσθέσουμε την πλήρη διαδρομή της εντολής στην `PATH` στο αρχείο `/etc/profile`.

Μεταβλητή περιβάλλοντος PS1

- Για να αλλάξουμε την μορφή προτροπής φλοιού τροποποιούμε την τιμή της μεταβλητής περιβάλλοντος PS1.
- Η PS1 ορίζεται συνήθως με ειδικούς χαρακτήρες που αναγνωρίζονται από τον φλοιό.
- Οι πιο συνηθισμένοι ειδικοί χαρακτήρες είναι:
 - \d εκτύπωση της ημερομηνίας
 - \t εκτύπωση της τρέχουσας ώρας
 - \h εκτύπωση του ονόματος υπολογιστή
 - \s εκτύπωση του ονόματος φλοιού
 - \u εκτύπωση του ονόματος του χρήστη
 - \w εκτύπωση του ονόματος του τρέχοντος καταλόγου ερνασίας

Μεταβλητή περιβάλλοντος PS1

```
$ export PS1="\d \t"  
$ PS1="[ \t] [\u@\h: \w] \$"
```

- Η παραπάνω εντολή μπορεί να προστεθεί στο αρχείο `.bash_profile` του προσωπικού μας καταλόγου, αν θέλουμε η αλλαγή αυτή να ισχύει κάθε φορά που συνδεόμαστε στο σύστημα.
- Επίσης, μπορεί να προστεθεί στο αρχείο `/etc/profile` για να ισχύει η αλλαγή αυτή για όλους τους χρήστες του συστήματος.

Ψευδώνυμα (aliases)

- Ο bash παρέχει τρόπους συντόμευσης των εντολών και των ορισμάτων που πληκτρολογούμε μέσω των ψευδωνύμων.
- Τα ψευδώνυμα (aliases) είναι εικονικές εντολές που αντιστοιχούν σε μια παράσταση με εντολές Linux.
- Τα ψευδώνυμα δεν υφίστανται ως αρχεία αλλά ο φλοιός τις αναγνωρίζει και εκτελεί την παράσταση που έχουμε ορίσει.

Ψευδώνυμα (aliases)

- Για την δημιουργία ψευδωνύμων χρησιμοποιούμε την εντολή alias.
- Σύνταξη: alias [aliasname [= listcommands]]
 - Χωρίς παράμετρο επιστρέφει μια λίστα των υπαρχόντων ψευδωνύμων.
- Παράδειγμα

```
$ alias dir = "ls -al"  
$ alias move = "mv"  
$ dir  
  
$ alias
```

Ψευδώνυμα (aliases)

- Για να διαγράψουμε ένα ψευδώνυμο χρησιμοποιούμε την εντολή `unalias`.
- Σύνταξη: `unalias aliasname`
- Παράδειγμα

```
$ unalias dir  
$
```

Σενάρια φλοιού

- Σενάριο φλοιού (shell script) είναι ένα αρχείο που περιέχει εντολές φλοιού για να εκτελεστεί αργότερα.
- Οι εντολές φλοιού σε ένα σενάριο περιλαμβάνουν:
 - Εντολές Linux
 - Σχόλια (#)
 - Μεταβλητές φλοιού
 - Εντολές ελέγχου (if, case, for, while, κλπ)
- Για να εκτελεστεί το σενάριο φλοιού πληκτρολογούμε το όνομα του αρχείου. Δεν χρειάζεται μεταγλώττιση αλλά διερμηνεύεται από τον φλοιό.

Σενάρια φλοιού

- Τα σενάρια φλοιού είναι χρήσιμα για πολλούς λόγους:
 - Αποθηκεύει συχνά σύνολο χρησιμοποιούμενων εντολών που εκτελούν συγκεκριμένες λειτουργίες.
 - Αυτοματοποιεί ορισμένες επαναλαμβανόμενες διαχειριστικές εργασίες ιδιαίτερα χρήσιμο στους διαχειριστές συστημάτων.

Διερμηνεία σεναρίων φλοιών

- Όταν ο φλοιός παίρνει μια εντολή που δεν είναι εσωτερική, τότε καλεί τον πυρήνα του και του αναθέτει την εκτέλεση της εντολής. Αυτή η εντολή μπορεί να είναι είτε ένα εξωτερικό πρόγραμμα σε εκτελέσιμη μορφή είτε ένα σενάριο φλοιού. Εφόσον είναι σενάριο φλοιού, ο πυρήνας θα πρέπει να επιλέξει έναν φλοιό για να εκτελέσει το σενάριο.
- Ο τρόπος με τον οποίο θα γίνει η επιλογή φλοιού εξαρτάται από το σύστημα.

Δημιουργία σεναρίων φλοιών

- Για την δημιουργία σεναρίων φλοιών ακολουθούμε 4 βήματα:
 - Δημιουργούμε ένα αρχείο κειμένου είτε με την εντολή `cat` ή με έναν επεξεργαστή κειμένου (π.χ. `vi`, `emacs`).
 - Έπειτα γράφουμε στο αρχείο κειμένου εντολές για την εκτέλεση ορισμένων λειτουργιών.
 - Στη συνέχεια δίνουμε στο αρχείο την άδεια εκτέλεσης στον ιδιοκτήτη ώστε να είναι εκτελέσιμο αρχείο με την εντολή `chmod +x filename_shellscript`.
 - Τέλος, μπορούμε να εκτελέσουμε το σενάριο φλοιού πληκτρολογώντας το όνομα του.

Δημιουργία σεναρίων φλοιών

- Στην πρώτη γραμμή ενός σεναρίου φλοιού έχει τη μορφή `#!/shell`.
 - όπου `shell` είναι το πλήρες όνομα της διαδρομής ενός φλοιού που θα χρησιμοποιηθεί για την διερμηνεία του σεναρίου (ή εκτέλεση των εντολών που περιέχει το σενάριο).
 - Π.χ. `#!/bin/bash`
- Αν δεν ισχύει η παραπάνω σύμβαση, τότε το σενάριο διερμηνεύεται από ένα φλοιό Bourne.

Σενάρια φλοιού - Παράδειγμα 1

- Δημιουργούμε ένα αρχείο με το όνομα `dir` που περιέχει μια εντολή `ls`:

```
$ cat > dir  
ls -l  
Ctrl-c
```

- Μετατρέπουμε το αρχείο `dir` σε εκτελέσιμο με την εντολή `chmod`:

```
$ chmod +x dir
```

- Για να εκτελέσουμε το σενάριο `dir` πλητρολογούμε το όνομα του:

```
$ ./dir
```

Σενάρια φλοιού - Παράδειγμα 2

```
$ cat > commd  
cal  
date  
who  
Ctrl-c  
$ chmod +x commd  
$ ./commd
```

Σενάρια φλοιού - Παράδειγμα 3

```
$ cat > users
echo The date today is
date
echo The number of users:
who | wc -l
echo Personal info:
who am i
Ctrl-c
$ chmod +x users
$ ./users
```

Σενάρια φλοιού - Παράδειγμα 4

```
$ cat > statdir
echo The directory contains `ls -l | grep '^-.....' |
wc -l` files
echo The directory contains `ls -l | grep '^d.....' |
wc -l` directories
Ctrl-c
$ chmod +x statdir
$ ./statdir
```

Ορίσματα γραμμής εντολών

- Ο φλοιός διερμηνεύει μια εντολή συνδέοντας ονόματα μεταβλητών σε κάθε στοιχείο της γραμμής εντολών. Ως στοιχεία της γραμμής εντολών θεωρούνται ακολουθίες χαρακτήρων διαχωριζόμενες με κενά.
- Οι μεταβλητές που συνδέονται με τα στοιχεία της γραμμής εντολής είναι οι \$0 έως \$9. Αυτές οι 10 μεταβλητές αντιστοιχούν με τις θέσεις των στοιχείων στην γραμμή εντολών και λέγονται παράμετροι θέσης.
- \$0 είναι το όνομα του σενάριου.
- \$1 είναι το πρώτο όρισμα της γραμμής εντολών μετά το όνομα του σενάριου.
- \$9 είναι το τελευταίο όρισμα της γραμμής εντολών.

Σενάρια φλοιού - Παράδειγμα 5

```
$ cat > statdir
echo The directory $1 contains `ls -l $1 | grep '^-.....'
| wc -l` files
echo The directory $1 contains `ls -l $1 | grep '^d.....'
| wc -l` directories
Ctrl-c
$ chmod +x statdir
$ ./statdir /etc
```

Multiuser - Multitasking

- Στα συστήματα Linux περισσότεροι από ένας χρήστες μπορούν «να τρέχουν» προγράμματα την ίδια χρονική στιγμή και περισσότερες από μια διεργασίες μπορούν να εκτελούνται την ίδια χρονική στιγμή.
- Το Linux είναι ένα λειτουργικό σύστημα πολυχρηστικό (multiuser) και πολυδιεργασία (multitasking).

Προγράμματα - διεργασίες

- Πρόγραμμα

- Είναι μια συλλογή bytes που αντιπροσωπεύουν τον κώδικα αποθηκευμένο σε ένα αρχείο.
- Μετά το κλείσιμο του υπολογιστή τα προγράμματα παραμένουν στο δίσκο.

- Διεργασία

- Εκτελείται ένα στιγμιότυπο του προγράμματος στην RAM.
- Δημιουργείται και διαχειρίζεται από τον πυρήνα Linux.
- Όταν ένα πρόγραμμα εκτελείται δημιουργείται μια νέα διεργασία.
- Μπορού να υπάρχουν πολλές διεργασίες που εκτελούν το ίδιο πρόγραμμα.

Πολυδιεργασία

- Time slicing
 - Κάθε φορά μια διεργασία εκτελείται ανά CPU του συστήματος.
 - Ο πυρήνας διανέμει σε μια διεργασία μικρή ποσότητα χρόνου για να εκτελέσει.
 - Αυτός ο χρόνος ονομάζεται time slice.
 - Όταν ο χρόνος της διεργασίας εξαντληθεί τότε ο πυρήνας εναλλάσει την διεργασία.
 - Διακόπτει την εκτέλεση της παρούσας διεργασίας και εκτελείται μια άλλη διεργασία στην CPU.
 - Χρονοδρομολογητής
 - Τμήμα του πυρήνα που παίρνει αποφάσεις σχετικά με την ανάθεση της CPU στις διάφορες διεργασίες.

Διεργασίες προσκήνιου και παρασκήνιου

- Μια διεργασία εκτελείται είτε στο προσκήνιο (foreground) είτε στο παρασκήνιο (background).
- Διεργασία προσκήνιου
 - Στο προσκήνιο εκτελείται μια διεργασία κάθε στιγμή.
 - Έχει τον πλήρη έλεγχο του φλοιού και βρίσκεται σε κατάσταση αναμονής μέχρι να ολοκληρωθεί η εκτέλεση της εντολής.

```
$ ls -al
```

Διεργασίες προσκηνίου και παρασκηνίου

- Διεργασίες παρασκηνίου
 - Στο παρασκήνιο εκτελούνται μια ή περισσότερες διεργασίες την ίδια χρονική στιγμή.
 - Δεν έχουν τον έλεγχο του φλοιού και στο προσκήνιο μπορούμε να εκτελέσουμε μια άλλη εντολή.
 - Δεν απαιτούν είσοδο από το πληκτρολόγιο διαφορετικά δεν θα μπορούν να εκτελεστούν.
 - Για να εκτελέσουμε μια διεργασία στο παρασκήνιο τοποθετούμε τον τελεστή & στο τέλος της εντολής.

```
$ find / -name '*.c' -print > results.txt &
```

```
$ find / -name '*.c' -print > results.txt  
Ctrl-z  
$
```

Διεργασίες προσκηνίου και παρασκηνίου

- Η διεργασία παρασκηνίου είναι χρήσιμη για δύο λόγους:
 - Εξυπηρετεί την ταυτόχρονη εκτέλεση αρκετών διεργασιών στο παρασκήνιο.
 - Διεκολύνει τις εντολές που απαιτούν πολύ χρόνο για την εκτέλεση τους ώστε να αποφύγουμε τη δέσμευση του φλοιού για μεγάλο χρονικό διάστημα.

Διαχείριση διεργασιών

- Για να πάρουμε μια λίστα τρέχουσων διεργασιών στο σύστημα μας και να παρακολουθήσουμε τη συμπεριφορά τους, το Linux παρέχει αρκετά εργαλεία:
 - Εντολή ps
 - Εντολή top
 - Εντολή kill

Εντολή ps

- Η εντολή ps μας επιτρέπει να ελέγξουμε την κατάσταση διεργασιών. Παράγει μια λίστα πληροφοριών κατάστασης διεργασιών και τα χαρακτηριστικά τους όπως τα ονόματα τους, τους αριθμούς ταυτοτήτων διεργασίας, τους ιδιοκτήτες τους, το ποσοστό χρησιμοποίησης CPU των διεργασιών κλπ.
- Σύνταξη: ps [options]
 - -e θα περιλάβει όλες τις διεργασίες που εκτελούνται αυτή τη στιγμή.
 - -f θα παράγει μια πλήρη λίστα.
 - Η ps δεν είναι εντολή πραγματικού χρόνου και παίρνει αντίγραφο των διεργασιών που εκτελούνται εκείνη τη στιγμή.

Εντολή ps - Παράδειγμα

```
$ sleep 10 &  
[1] 21349  
$ ps -f  
UID          PID    PPID    C  STIME TTY          TIME CMD  
panosm      20199    170    0  11:30 pts/2        00:00:00 -bash  
panosm      21349  20199    0  11:34 pts/2        00:00:00 sleep 10  
panosm      15965  20199    0  11:34 pts/2        00:00:00 ps -f  
$
```

- **UID:** αριθμός ταυτότητας χρήστη της διεργασίας.
- **PID:** η ταυτότητα της διεργασίας.
- **PPID:** η ταυτότητα της πατρικής διεργασίας.
- **C:** το ποσοστό χρόνου CPU που χρησιμοποιήθηκε η διεργασία.
- **STIME:** ώρα δημιουργίας της διεργασίας.
- **TTY:** το τερματικό της διεργασίας
- **TIME:** ο χρόνος CPU που χρησιμοποιήθηκε ως τώρα.
- **CMD:** το όνομα της εντολής.

Εντολή top

- Η εντολή top παράγει μια λίστα με περιοδική ενημέρωση των διεργασιών που τρέχουν στο σύστημα.

```
$ top
12:42:04 up 34 days,  1:26,  2 users,  load average: 0.40, 0.31, 0.27
3 processes: 1 sleeping, 2 running, 0 zombie, 0 stopped
CPU states:  18.6% user,   2.9% system,   0.0% nice,  78.5% idle
Mem:   2069292K total,  1988292K used,    81000K free,    45756K
buffers
Swap:  2097136K total,    1916K used,  2095220K free,    751388K
cached
```

PID	USER	PRI	NI	SIZE	RSS	SHARE	STAT	%CPU	%MEM	TIME	COMMAND
1721	panosm	9	0	4128	4128	3912	R	0.0	0.1	0:00	sshd
11139	panosm	9	0	3076	3076	2752	S	0.0	0.1	0:00	bash
3108	panosm	9	0	1496	1496	1336	R	0.0	0.0	0:00	top

Σηματοδοσία διεργασιών

- Σήματα είναι μηνύματα που αποστέλονται στις διεργασίες.
 - Τα σήματα αναπαριστώνονται με αριθμούς και ονόματα.
 - Ο πυρήνας είναι υπεύθυνος για την προώθηση των σημάτων στις διεργασίες.
 - Η διεργασία θα αντιδράσει μόλις λάβει το σήμα.
 - Κάθε αριθμός σήματος έχει μια λειτουργία.
 - Μια διεργασία ίσως τερματιστεί ή αγνοήσει το σήμα.
- Η διεργασία μπορεί να συλλάβει ένα σήμα.
 - Εκτός το σήμα 9 (KILL) το οποίο δεν μπορεί να το αγνοήσει.
- Η εντολή kill χρησιμοποιείται για την αποστολή σημάτων στις διεργασίες.

Εντολή kill

- Σύνταξη: kill [-signalid] pid
 - Η εντολή kill στέλνει το σήμα με κωδικό signalid στην διεργασία pid.
 - Το signalid είναι ο αριθμός ή το όνομα ενός σήματος.
 - Η επιλογή -l εμφανίζει μια λίστα ονομάτων σημάτων.
 - Χωρίς επιλογές η εξ ορισμού τιμή σήματος είναι 15.
 - Για την αποστολή σήματος στις διεργασίες πρέπει να είναι ο ιδιοκτήτης της ή ο διαχειριστής root.
- Η kill -9 pid στέλνει ένα σήμα τερματισμού στην διεργασία pid, που το σύστημα δεν μπορεί να το αγνοήσει.
- Η εντολή kill 0 (το 0 σαν pid) τερματίζει όλες τις διεργασίες που σχετίζονται με το φλοιό.

Εντολή kill - Παράδειγμα

```
$ sleep 20 &
[1] 649
$ ps -f
UID          PID    PPID    C  STIME TTY          TIME CMD
panosm      4840  20108    0  12:36 pts/0        00:00:00 -bash
panosm       649   4840    0  12:36 pts/0        00:00:00 sleep 20
panosm     25452   4840    0  12:37 pts/0        00:00:00 ps -f
$ kill 649
bash: kill: (649) - No such process
[1]+  Done                  sleep 20
$ ps -f
UID          PID    PPID    C  STIME TTY          TIME CMD
panosm      4840  20108    0  12:36 pts/0        00:00:00 -bash
panosm     2210   4840    0  12:37 pts/0        00:00:00 ps -f
$
```