

# Linux με τη χρήση κονσόλας

## Κεφάλαιο 5

*Κείμενα Χρυσούλα Παπάζογλου*

*Πανεπιστήμιο Μακεδονίας, Βιβλιοθήκη και Κέντρο Πληροφόρησης*

*Υπηρεσία Ασύγχρονης Τηλεκπαίδευσης*

*Θεσσαλονίκη, Μάρτης 2009*

Το υλικό διατίθεται υπό την άδεια Creative Commons Αναφορά-Μη Εμπορική Χρήση-Παρόμοια διανομή 3.0 Unported <http://creativecommons.org/licenses/by-nc-sa/3.0/deed.el>



## Κατάλογος περιεχομένων

Εισαγωγή.....	4
Διαχείριση διεργασιών.....	4
Τι είναι οι διεργασίες.....	4
Τύποι διεργασιών.....	4
Διαλογικές διεργασίες.....	4
Δαίμονες.....	6
Εμφάνιση πληροφοριών διεργασίας.....	6
Δημιουργία και τερματισμός μιας διεργασίας.....	9
Δημιουργία διεργασίας.....	9
Τερματισμός διεργασιών.....	10
Σήματα.....	10
Εκκίνηση και τερματισμός συστήματος.....	11
Εισαγωγή.....	11
Η εκκίνηση.....	11
Τερματισμός.....	12
Παρακολούθηση συστήματος.....	12
Εισαγωγή.....	12
Έλεγχος διάρκειας εργασιών.....	13
Γραφικά εργαλεία.....	14
Διακοπή διεργασιών.....	14
Χρονοπρογραμματισμός διεργασιών.....	15
Εκμετάλλευση του αδρανούς χρόνου.....	15
Η εντολή sleep.....	15
Η εντολή at.....	16
Cron και crontab.....	16
Περίληψη.....	18
Ασκήσεις.....	19
Γενικά.....	20
Εκκίνηση, init κλπ.....	20
Χρονοπρογραμματισμός.....	21

## Εισαγωγή

Οι διεργασίες είναι ουσιαστικά προγράμματα που εκτελούνται στο σύστημά σας. Ο χρήστης ενός συστήματος θα πρέπει να μπορεί να εμφανίσει τη λίστα των διεργασιών που εκτελούνται, να εκκινήσει, να διακόψει, να επανεκκινήσει και να τερματίσει διεργασίες καθώς και να καθορίσει τις διεργασίες που ξεκινούν αυτόματα με την εκκίνηση του συστήματος.

Στο κεφάλαιο αυτό θα μάθετε:

- τί είναι οι διεργασίες,
- πώς εκκινείται και τερματίζεται μια διεργασία,
- πώς γίνεται παρακολούθηση του συστήματος,
- τί είναι ο χρονοπρογραμματισμός διεργασιών.

## Διαχείριση διεργασιών

### Τι είναι οι διεργασίες

Οι διεργασίες είναι ουσιαστικά προγράμματα που εκτελούνται στο σύστημά σας. Για παράδειγμα, για την εκτέλεση μιας εφαρμογής ή την υποστήριξη μιας υπηρεσίας μπορεί να απαιτείται η εκκίνηση μίας ή περισσότερων διεργασιών. Η διαχείριση και η εναλλαγή (δηλαδή η εκ περιτροπής χρήση της κεντρικής μονάδας επεξεργασίας) των διεργασιών που εκτελούνται είναι μία από τις βασικότερες αρμοδιότητες του λειτουργικού συστήματος. Ο χρήστης ενός συστήματος θα πρέπει να μπορεί να εμφανίσει τη λίστα των διεργασιών που εκτελούνται, να εκκινήσει, να διακόψει, να επανεκκινήσει και να τερματίσει διεργασίες καθώς και να καθορίσει τις διεργασίες που ξεκινούν αυτόματα με την εκκίνηση του συστήματος.

## Τύποι διεργασιών

### Διαλογικές διεργασίες

Οι διαλογικές διεργασίες αρχικοποιούνται και ελέγχονται από μια σύνοδο τερματικού. Με άλλα λόγια, πρέπει να υπάρχει κάποιος συνδεδεμένος στο σύστημα για να ξεκινήσει τέτοιες διεργασίες αφού δεν ξεκινούν αυτόματα ως μέρος των λειτουργιών του συστήματος. Αυτές οι διεργασίες μπορεί να εκτελούνται στο προσκήνιο, καταλαμβάνοντας το τερματικό που τις ξεκίνησε, με αποτέλεσμα να μην μπορείτε να ξεκινήσετε άλλες εφαρμογές όσο αυτές οι διεργασίες εκτελούνται. Εναλλακτικά μπορεί να εκτελούνται στο παρασκήνιο και το τερματικό που τις ξεκίνησε να μπορεί να δέχεται νέες εντολές, ενώ αυτές οι διεργασίες συνεχίζουν να εκτελούνται.

Οι εντολές που είδαμε μέχρι τώρα εκτελούνταν στο προσκήνιο (foreground), ωστόσο ο χρόνος εκτέλεσης ήταν πολύ μικρός για να γίνει αντιληπτός (δηλαδή το αποτέλεσμα της εντολής εμφανιζόταν άμεσα και είχατε τη δυνατότητα να πληκτρολογήσετε νέες εντολές). Σε κάποιες περιπτώσεις, η εντολή (για την ακρίβεια η διεργασία που ξεκινά με την εντολή) περιμένει κάποια ενέργεια από εσάς. Για παράδειγμα, κατά την προβολή ενός αρχείου με την εντολή more, η διεργασία περιμένει από εσάς να πατήσετε κάποιο πλήκτρο για να μεταβείτε σε επόμενη/ προηγούμενη σελίδα. Το πρόγραμμα που εκτελείται καταλαμβάνει τη γραμμή εντολών και το τερματικό πλέον δέχεται μόνο εντολές του προγράμματος. Όταν όμως μια διεργασία εκτελείται στο παρασκήνιο (background), ο χρήστης μπορεί να κάνει άλλα πράγματα στο τερματικό που ξεκίνησε τη διεργασία.

Το κέλυφος εντολών προσφέρει ένα χαρακτηριστικό που λέγεται έλεγχος εργασιών (job control) που επιτρέπει την εύκολη διαχείριση πολλαπλών διεργασιών. Αυτός ο μηχανισμός εναλλάσσει διεργασίες μεταξύ προσκηνίου και παρασκηνίου. Επίσης μπορεί να ξεκινήσει διεργασίες απευθείας στο παρασκήνιο.

Είναι αυτονόητο ότι η εκτέλεση διεργασιών στο παρασκήνιο είναι χρήσιμη μόνο όταν τα αντίστοιχα προγράμματα δεν απαιτούν επικοινωνία με το χρήστη (όπως στο παράδειγμα που αναφέρθηκε παραπάνω). Είναι συνήθης πρακτική να μεταφέρονται εργασίες στο παρασκήνιο όταν η διάρκειά τους αναμένεται να είναι μεγάλη. Για να ξεκινήσουμε μια διεργασία απευθείας στο παρασκήνιο προσθέτουμε το χαρακτήρα & μετά το τέλος της εντολής (μετά τα όρισματα και τις επιλογές της αν υπάρχουν). Για να εμφανίσουμε τη λίστα με τις διεργασίες που εκτελούνται στο παρασκήνιο χρησιμοποιούμε την εντολή jobs.

```
billy:~> more file1.txt &
[1] 26558

billy:~> jobs
[1]+  Stopped                  more client.c
```

Ο αριθμός ανάμεσα στις αγκύλες είναι το αναγνωριστικό της διεργασίας για την εναλλαγή της μεταξύ προσκηνίου και παρασκηνίου, ενώ ο άλλος αριθμός είναι το αναγνωριστικό της διεργασίας (process id) γενικότερα στο σύστημα.

Στο παράδειγμα χρησιμοποιούμε μια εντολή η οποία απαιτεί είσοδο από το χρήστη για να δείξουμε πως γίνεται η εναλλαγή μεταξύ προσκηνίου και παρασκηνίου. Εφόσον η διεργασία ξεκινά στο παρασκήνιο, θα εμφανίσει την πρώτη σελίδα του αρχείου και μετά θα περιμένει είσοδο από τον χρήστη. Για το λόγο αυτό όταν εκτελούμε την εντολή jobs εμφανίζεται στη λίστα σε κατάσταση stopped (ενώ μια διεργασία που συνέχιζε να εκτελείται θα εμφανιζόταν σε κατάσταση running). Για να μεταφέρουμε μια διεργασία από το παρασκήνιο στο προσκήνιο, χρησιμοποιούμε την εντολή fg ακολουθούμενη από το νούμερο μέσα στις αγκύλες (π.χ. fg 1 στο παραπάνω παράδειγμα). Εάν πληκτρολογήσετε την εντολή fg χωρίς κάποιο όρισμα, θα φέρει στο προσκήνιο την τελευταία διεργασία που εκκινήθηκε στο παρασκήνιο.

Για να διακόψετε προσωρινά την εκτέλεση μιας διεργασίας χρησιμοποιήστε το συνδυασμό πλήκτρων Ctrl+Z ενώ για να τερματίσετε τη διεργασία που εκτελείται στο προσκήνιο το συνδυασμό πλήκτρων Ctrl+C. Για να τερματίσετε μια διεργασία που εκτελείται στο παρασκήνιο μπορείτε να χρησιμοποιήσετε την εντολή kill με όρισμα το αναγνωριστικό της διεργασίας στο σύστημα (process id). Στον πίνακα που ακολουθεί συνοψίζονται οι βασικές εντολές για τον έλεγχο των διεργασιών.

#### Πίνακας 5-1. Έλεγχος διεργασιών

Εντολή	Αποτέλεσμα
command	Εκτέλεση εντολής command στο προσκήνιο.
command &	Εκτέλεση της εντολής command στο παρασκήνιο (απελευθέρωση τερματικού)
jobs	Εμφάνιση των εντολών που εκτελούνται στο παρασκήνιο.
Ctrl+Z	Αναστολή (παύση, αλλά όχι τερματισμός) μιας διεργασίας που εκτελείται στο προσκήνιο.
Ctrl+C	Τερματισμός (διακοπή) μιας διεργασίας που εκτελείται στο προσκήνιο.

Εντολή	Αποτέλεσμα
<b>bg</b>	Επανενεργοποίηση ενός προγράμματος που έχει ανασταλεί στο παρασκήνιο.
<b>fg</b>	Επαναφορά εργασίας στο προσκήνιο.
<b>kill</b>	Τερματισμός διεργασίας

## Δαίμονες

Οι δαίμονες (daemons) είναι διεργασίες διακομιστή (server processes) που εκτελούνται συνεχώς στο παρασκήνιο. Συνήθως ξεκινούν με την έναρξη του συστήματος και μετά περιμένουν στο παρασκήνιο μέχρι να ζητηθεί η συνδρομή τους. Ένα τυπικό παράδειγμα είναι ο δαίμονας της δικτύωσης, *xinetd*, που ξεκινά σχεδόν σε κάθε διαδικασία εκκίνησης. Μετά την εκκίνηση του συστήματος, ο δαίμονας αναμένει μέχρι ένα πρόγραμμα - πελάτης (client), για παράδειγμα ένας ftp client, χρειαστεί να συνδεθεί.

## Εμφάνιση πληροφοριών διεργασίας

Η εντολή **ps** είναι ένα βασικό εργαλείο για να εμφανίσουμε πληροφορίες σχετικά με τις διεργασίες που εκτελούνται. Η εντολή αυτή έχει πολλές επιλογές που μπορούν να συνδυαστούν για να εμφανίσουμε διάφορα χαρακτηριστικά διεργασιών.

Χωρίς επιλογές, η **ps** δίνει μόνο πληροφορίες για την τρέχουσα σύνοδο του κελύφους εντολών και τις αντίστοιχες διεργασίες:

```
theo:~> ps
  PID TTY          TIME CMD
 4245 pts/7        00:00:00 bash
 5314 pts/7        00:00:00 ps
```

Καθώς αυτό δε δίνει αρκετή πληροφορία -γενικά εκτελούνται πάνω από εκατό διεργασίες στο σύστημά μας- συνήθως επιλέγουμε συγκεκριμένες διεργασίες από τη λίστα όλων των διεργασιών, με χρήση της εντολής **grep** σε μια *σωλήνωση (pipe)*, όπως στη γραμμή που ακολουθεί, εμφανίζοντας όλες τις διεργασίες με ιδιοκτήτη κάποιο συγκεκριμένο χρήστη:

Για να εμφανιστεί η πλήρης λίστα των διεργασιών με όλες τις διαθέσιμες πληροφορίες χρησιμοποιήστε τις επιλογές **e** και **f**, αντίστοιχα.

### ps -ef

Επειδή η λίστα των εργασιών θα είναι αρκετά μεγάλη, μπορείτε να χρησιμοποιήσετε την εντολή **grep** για να εντοπίσετε κάτι που σας ενδιαφέρει στη λίστα των αποτελεσμάτων (π.χ. τις διεργασίες ενός συγκεκριμένου χρήστη). Στο παράδειγμα που ακολουθεί εντοπίζονται οι εγγραφές στη λίστα των διεργασιών που περιέχουν το αλφαριθμητικό **bash**. Το σύμβολο **|** (διοχέτευση - pipeline) ορίζει ότι η έξοδος της πρώτης εντολής θα χρησιμοποιηθεί ως είσοδος στην επόμενη, δηλαδή ότι η **grep** θα αναζητήσει στα αποτελέσματα της **ps**. Περισσότερα για τις διοχετεύσεις θα δούμε σε επόμενη ενότητα.

```
theo:> ps -ef | grep bash
brenda  31970  0.0  0.3  6080 1556 tty2    S   Feb23   0:00 -bash
root    32043  0.0  0.3  6112 1600 tty4    S   Feb23   0:00 -bash
theo    32581  0.0  0.3  6384 1864 pts/1   S   Feb23   0:00 bash
theo    32616  0.0  0.3  6396 1896 pts/2   S   Feb23   0:00 bash
theo    32629  0.0  0.3  6380 1856 pts/3   S   Feb23   0:00 bash
theo    2214   0.0  0.3  6412 1944 pts/5   S   16:18   0:02 bash
theo    4245   0.0  0.3  6392 1888 pts/7   S   17:26   0:00 bash
theo    5427   0.0  0.1  3720  548 pts/7   S   19:22   0:00 grep bash
```

Σημειώστε ότι η **ps** δίνει μόνο ένα στιγμιότυπο των ενεργών διεργασιών, δηλαδή η λίστα των αποτελεσμάτων δεν ανανεώνεται. Το πρόγραμμα **top** εμφανίζει μια ακριβέστερη εικόνα με συνεχή ανανέωση των αποτελεσμάτων της **ps** (με πολλές επιλογές) κάθε πέντε δευτερόλεπτα, κατατάσσοντας τις διεργασίες με βάση το φόρτο που επιφέρουν στο σύστημα και με πληροφορίες σχετικά με τη χρήση του χώρου εναλλαγής, τη κατάσταση της CPU, της μνήμης κλπ:

```
12:40pm up 9 days, 6:00, 4 users, load average: 0.21, 0.11, 0.03
89 processes: 86 sleeping, 3 running, 0 zombie, 0 stopped
CPU states:  2.5% user,  1.7% system,  0.0% nice, 95.6% idle
Mem:  255120K av, 239412K used, 15708K free, 756K shrd, 22620K buff
Swap: 1050176K av, 76428K used, 973748K free, 82756K cached
  PID USER  PRI  NI  SIZE  RSS SHARE STAT  %CPU %MEM TIME COMMAND
 5005 root   14   0 91572 15M 11580 R    1.9  6.0  7:53 X
19599 jeff   14   0  1024 1024   796 R    1.1  0.4  0:01 top
19100 jeff    9   0  5288 4948  3888 R    0.5  1.9  0:24 gnome-terminal
19328 jeff    9   0 37884 36M 14724 S    0.5 14.8  1:30 mozilla-bin
   1 root    8   0   516  472   464 S    0.0  0.1  0:06 init
   2 root    9   0     0     0     0 SW   0.0  0.0  0:02 keventd
   3 root    9   0     0     0     0 SW   0.0  0.0  0:00 kapm-idled
   4 root   19  19     0     0     0 SWN  0.0  0.0  0:00 ksoftirqd_CPU0
   5 root    9   0     0     0     0 SW   0.0  0.0  0:33 kswapd
   6 root    9   0     0     0     0 SW   0.0  0.0  0:00 kreclaimd
   7 root    9   0     0     0     0 SW   0.0  0.0  0:00 bdflush
   8 root    9   0     0     0     0 SW   0.0  0.0  0:05 kupdated
   9 root   -1-20     0     0     0 SW<  0.0  0.0  0:00 mdrecoveryd
  13 root    9   0     0     0     0 SW   0.0  0.0  0:01 kjournald
   89 root    9   0     0     0     0 SW   0.0  0.0  0:00 khubd
  219 root    9   0     0     0     0 SW   0.0  0.0  0:00 kjournald
  220 root    9   0     0     0     0 SW   0.0  0.0  0:00 kjournald
```

Η πρώτη σειρά της λίστας της **top** περιέχει κάποιες πληροφορίες που εμφανίζονται με την εντολή **uptime**:

```
jeff:~> uptime
3:30pm, up 12 days, 23:29, 6 users, load average: 0.01, 0.02, 0.00
```

Οι σχέσεις μεταξύ διεργασιών μπορούν να εμφανιστούν με την εντολή **pstree**:

```
sophie:~> pstree
init--+-amd
      |-apmd
      |-2*[artsd]
      |-atd
      |-crond
```

```

|-deskguide_apple
|-eth0
|-gdm---gdm--X
|   `--gnome-session--Gnome
|       |--ssh-agent
|       `--true
|-geyes_applet
|-gkb_applet
|-gnome-name-serv
|-gnome-smproxy
|-gnome-terminal--bash---vim
|   |--bash
|   |--bash---pstree
|   |--bash---ssh
|   |--bash---mozilla-bin---mozilla-bin---3*[mozilla-bin]
|   `--gnome-pty-helper
|-gpm
|-gweather
|-kapm-idled
|-3*[kdeinit]
|-keventd
|-khubd
|-5*[kjournald]
|-klogd
|-lockd---rpciod
|-lpd
|-mdrecoveryd
|-6*[mingetty]
|-8*[nfsd]
|-nscd---nscd---5*[nscd]
|-ntpd
|-3*[oafd]
|-panel
|-portmap
|-rhnsd
|-rpc.mountd
|-rpc.rquotad
|-rpc.statd
|-sawfish
|-screenshoter_a
|-sendmail
|-sshd---sshd---bash---su---bash
|-syslogd
|-tasklist_applet
|-vmnet-bridge
|-xfs
|-xinetd-ipv6

```

Οι επιλογές -u και -a δίνουν παραπάνω πληροφορίες.

# Δημιουργία και τερματισμός μιας διεργασίας

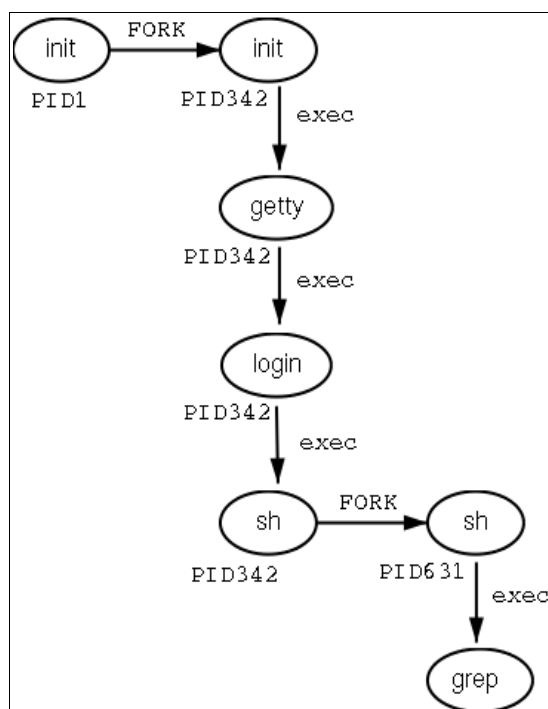
## Δημιουργία διεργασίας

Μια νέα διεργασία δημιουργείται επειδή μια υπάρχουσα διεργασία κάνει ένα ακριβές αντίγραφο του εαυτού της. Η νέα διεργασία-παιδί, έχει το ίδιο περιβάλλον εργασίας με τη διεργασία-γονέα και διαφέρουν μόνο στο αναγνωριστικό διεργασίας (process ID). Αυτή η διαδικασία λέγεται *διακλάδωση (forking)*.

Μετά τη διακλάδωση, ο χώρος διευθύνσεων της διεργασίας-παιδί ενημερώνεται με τα νέα δεδομένα της διεργασίας. Αυτό γίνεται με μια κλήση `exec` του συστήματος.

Ο μηχανισμός `fork-and-exec` μεταφέρει την εκτέλεση από την παλιά εντολή στη νέα, ενώ το νέο πρόγραμμα κληρονομεί το περιβάλλον εκτέλεσης το οποίο παραμένει ίδιο, συμπεριλαμβανομένων των συσκευών εισόδου και εξόδου, των μεταβλητών περιβάλλοντος και της προτεραιότητας. Αυτός ο μηχανισμός χρησιμοποιείται στο UNIX και στο Linux. Ακόμη και η πρώτη διεργασία η `init`, με αριθμό (process ID) 1, διακλαδώνεται κατά τη διαδικασία εκκίνησης σε μια διαδικασία που λέγεται *bootstrapping*.

Αυτό το σχήμα δείχνει τη διαδικασία του μηχανισμού `fork-and-exec`. Ο αριθμός της διεργασίας αλλάζει μετά τη διακλάδωση:



Εικόνα 5-1. Μηχανισμός `fork-and-exec`

Αραιά και που κάτι πάει στραβά, ακόμη και στα καλύτερα συστήματα. Σε αυτή την εξαιρετική περίπτωση, μια διεργασία μπορεί να τερματιστεί αλλά η γονική διεργασία να μην περιμένει το τερματισμό της (να έχει ήδη τερματιστεί και αυτή). Τέτοιες διεργασίες που δεν έχουν τερματιστεί κανονικά ονομάζονται διεργασίες *zombie*.

## Τερματισμός διεργασιών

Όταν μια διεργασία τερματιστεί κανονικά (δε σταματά με `kill` ή δε διακόπτεται απρόοπτα), επιστρέφει την κατάσταση εξόδου (*exit status*) στη γονική διεργασία. Η κατάσταση εξόδου είναι

ένας αριθμός που παρέχει κάποιες πληροφορίες σχετικές με την εκτέλεση της διεργασίας-παιδιού. Το σύστημα αυτό της επιστροφής πληροφοριών κατά την έξοδο προέρχεται από τη γλώσσα προγραμματισμού C στην οποία είναι γραμμένο το UNIX (και το Linux).

Οι κωδικοί επιστροφής (καταστάσεις εξόδου) μπορούν να ερμηνευτούν από τη διεργασία-γονέα ή από ειδικά σενάρια φλοιού. Οι τιμές των κωδικών εξαρτώνται από την εφαρμογή. Η σχετική πληροφορία μπορεί να βρεθεί στις σελίδες τεκμηρίωσης του συγκεκριμένου προγράμματος, για παράδειγμα η εντολή `grep` επιστέφει `-1` αν δεν υπάρξει καμιά ταύτιση, οπότε η διεργασία-γονέας θα μπορούσε να εμφανίσει ένα μήνυμα της μορφής "No files found". Ένα άλλο παράδειγμα είναι η ενσωματωμένη στο Bash, της εντολής `true`, που δεν κάνει τίποτε άλλο από το να επιστρέφει κατάσταση εξόδου 0, δηλαδή επιτυχία.

## Σήματα

Οι διεργασίες τερματίζουν επειδή λαμβάνουν ένα σήμα. Υπάρχουν πολλά σήματα που μπορείτε να στείλετε σε μια διεργασία. Η εντολή `kill` χρησιμοποιείται για την αποστολή σημάτων. Η επιλογή `kill -l` εμφανίσει μια λίστα από σήματα. Τα περισσότερα από αυτά είναι για εσωτερική χρήση του συστήματος ή για προγραμματιστές που γράφουν κώδικα συστήματος. Ως χρήστης θα χρειαστείτε τα παρακάτω σήματα:

Πίνακας 5-2. Συνηθισμένα σήματα

Όνομα σήματος	Αριθμός σήματος	Σημασία
SIGTERM	15	Κανονικός τερματισμός διεργασίας.
SIGINT	2	Διακοπή διεργασίας. Η διεργασία μπορεί να αγνοήσει αυτό το σήμα.
SIGKILL	9	Διακοπή διεργασίας. Η διεργασία δεν μπορεί να αγνοήσει αυτό το σήμα.
SIGHUP	1	Για δαίμονες: ξαναδιάβασε το αρχείο διαμόρφωσης..

Μπορείτε να διαβάσετε περισσότερα για τις διαδικασίες αποστολής και παραλαβής σήματος με τη `man 7 signal`.

## Εκκίνηση και τερματισμός συστήματος

### Εισαγωγή

Μια από τις πιο δυνατές πλευρές του Linux αφορά την ανοικτή μέθοδο εκκίνησης και τερματισμού του λειτουργικού συστήματος, όπου φορτώνει τα καθορισμένα προγράμματα με τις ρυθμίσεις τους. Ο χρήστης έχει τη δυνατότητα να παρέμβει τόσο στα προγράμματα όσο και στις ρυθμίσεις και με αυτό τον τρόπο να ελέγξει την εκκίνηση και τον τερματισμό, έτσι ώστε όλα να κυλούν ομαλά.

Πέρα από το θέμα του ελέγχου της εκκίνησης και του τερματισμού, η ανοικτή φύση του Linux διευκολύνει πολύ τον εντοπισμό των περισσότερων προβλημάτων εκκίνησης και τερματισμού. Μια βασική κατανόηση αυτών των διαδικασιών είναι ωφέλιμη για όλους όσοι χρησιμοποιούν Linux. Τη διαχείριση της εκκίνησης του συστήματος την αναλαμβάνει ένας `boot loader`.

Παλαιότερα, πολλά συστήματα χρησιμοποιούσαν τον Linux Loader (lilo). Οι σύγχρονες διανομές χρησιμοποιούν στην πλειοψηφία τους το GRUB.

## Η εκκίνηση

Η εκκίνηση ενός συστήματος ξεκινά από εκτέλεση του BIOS (Basic Input/Output System) το οποίο αποθηκεύεται στη μνήμη ROM. Το BIOS παρέχει διεπαφή χαμηλού επιπέδου για τις περιφερειακές συσκευές και ελέγχει το πρώτο βήμα της εκκίνησης.

Το BIOS ελέγχει το σύστημα, μετά τα περιφερειακά και στη συνέχεια ψάχνει για έναν οδηγό προκειμένου να φορτώσει το λειτουργικό σύστημα. Συνήθως ελέγχει για οδηγό δισκέτας (σε παλιότερα συστήματα), οδηγό CD-ROM (σε νεότερα) και για άλλα φορητά μέσα αποθήκευσης (DVD-ROM, USB θύρα, ΣτΜ), και φυσικά για οδηγούς σκληρού δίσκου. Η σειρά αναζήτησης οδηγών καθορίζεται στο BIOS και μπορεί να τροποποιηθεί από το χρήστη πριν την εκκίνηση. Αφού εγκατασταθεί το Linux σε ένα σκληρό δίσκο του συστήματος, το BIOS αναζητά την Κύρια Εγγραφή Εκκίνησης (Master Boot Record -MBR) που ξεκινά στο πρώτο τομέα του πρώτου δίσκου, φορτώνει τα περιεχόμενά της στη μνήμη και περνά τον έλεγχο σε αυτή.

Η MBR περιέχει εντολές για τον τρόπο φόρτωσης του boot loader και για το προεπιλεγμένο λειτουργικό σύστημα. Το GRUB είτε εμφανίζει κάποιο μενού με επιλογές ή μετά από μικρή αναμονή, συνεχίζει τη διαδικασία εκκίνησης. Με βάση είτε τις επιλογές του μενού ή τις προκαθορισμένες επιλογές το λειτουργικό σύστημα ξεκινά, είτε γραφικά είτε από τη γραμμή εντολών, με βάση τις ρυθμίσεις του.

## Τερματισμός

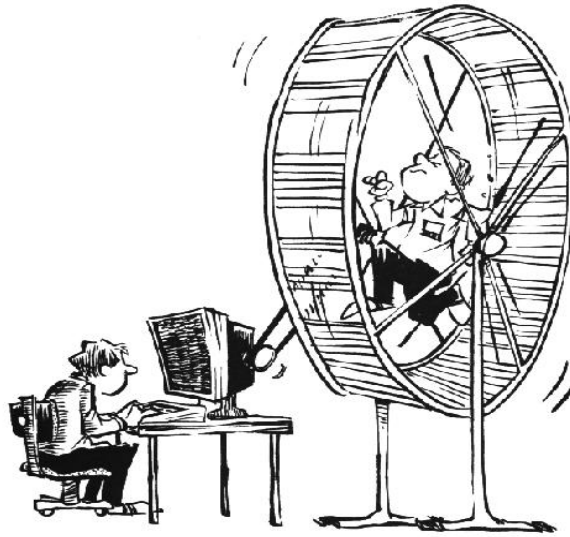
Για τον τερματισμό του συστήματος από τη γραμμή εντολών μπορείτε να χρησιμοποιήσετε την εντολή **shutdown**. Η επιλογή **-h** κλείνει το σύστημα ενώ η επιλογή **-r** το επανεκκινεί. Για παράδειγμα, για να τερματίσετε το σύστημά σας μπορείτε να χρησιμοποιήσετε την εντολή:

```
sudo shutdown -h now
```

## Παρακολούθηση συστήματος

### Εισαγωγή

Η σωστή διαχείριση και παρακολούθηση των πόρων του συστήματος μπορεί να μας βοηθήσει να βελτιώσουμε την απόδοσή του υλικού μας και συνεπώς να παρατείνουμε τα διαστήματα μεταξύ των αναβαθμίσεών του.



Εικόνα 5-2. Δεν μπορείς να πας πιο γρήγορα;

## Έλεγχος διάρκειας εργασιών

Το Bash παρέχει μια ενσωματωμένη εντολή `time` που εμφανίζει το χρόνο που χρειάστηκε μια εντολή για να εκτελεστεί. Η χρονομέτρηση είναι πολύ ακριβής και μπορεί να χρησιμοποιηθεί με κάθε εντολή. Στο παρακάτω παράδειγμα, απαιτείται περίπου ένα και μισό λεπτό για την προετοιμασία αυτού του βιβλίου:

```
tilly:~/xml/src> time make
Output written on abook.pdf (222 pages, 1619861 bytes).
Transcript written on abook.log.

real    1m41.056s
user    1m31.190s
sys     0m1.880s
```

Η GNU εντολή `time` στο `/usr/bin` (σε αντιδιαστολή με την ενσωματωμένη στο κέλυφος) εμφανίζει περισσότερη πληροφορία και μπορεί να μορφοποιηθεί με διαφορετικούς τρόπους. Επίσης δείχνει την κατάσταση εξόδου της εντολής και το συνολικό χρόνο που έχει δαπανηθεί. Το προηγούμενο παράδειγμα με την ανεξάρτητη εντολή `time` δίνει το αποτέλεσμα:

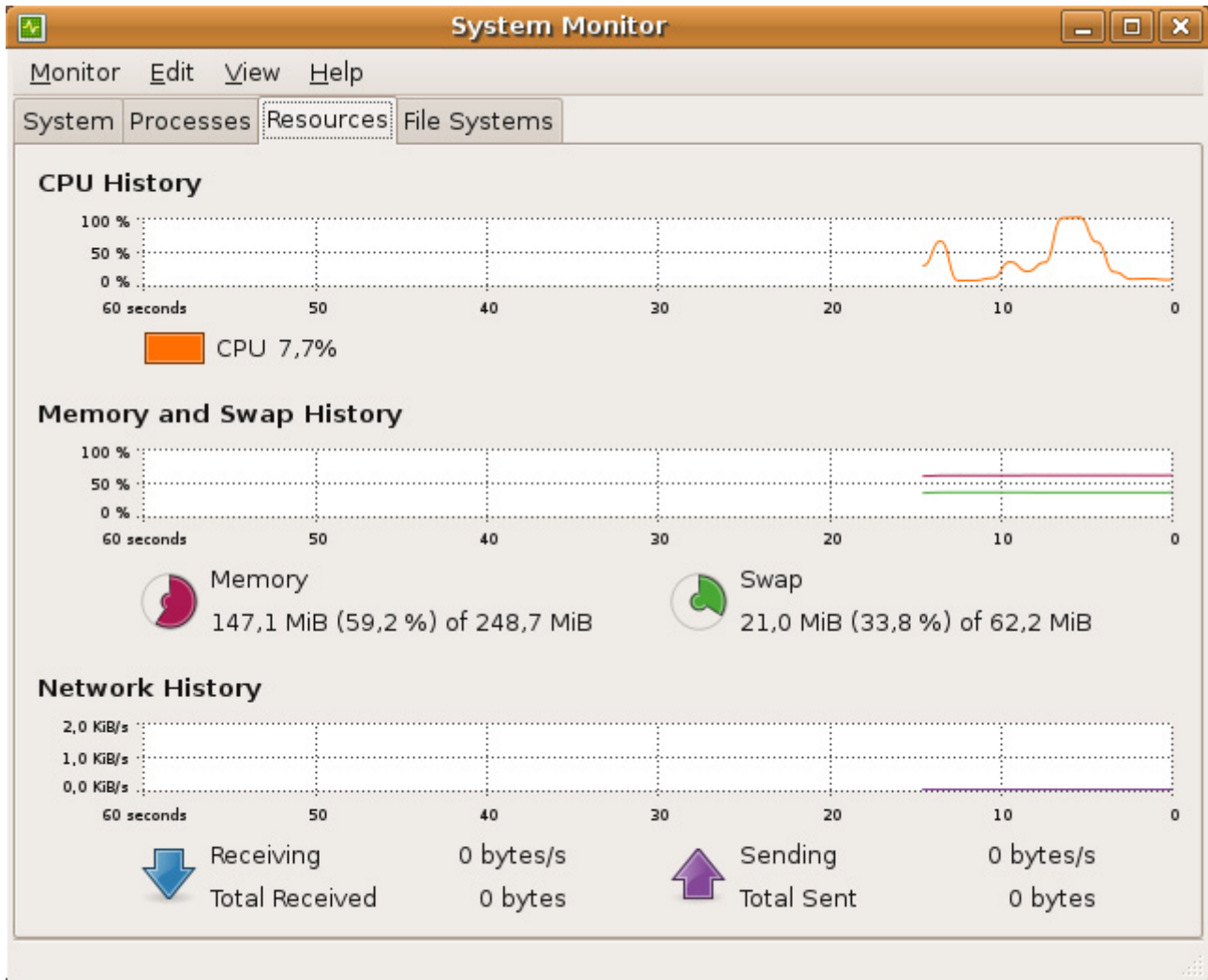
```
tilly:~/xml/src> /usr/bin/time make
Output written on abook.pdf (222 pages, 1595027 bytes).
Transcript written on abook.log.

Command exited with non-zero status 2
88.87user 1.74system 1:36.21elapsed 94%CPU
      (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (2192major+30002minor)pagefaults 0swaps
```

Μπορείτε να δοκιμάσετε και με απλές εντολές (π.χ. `ls`) στις οποίες οι χρόνοι εκτέλεσης είναι πολύ μικροί για να διαπιστώσετε την ακρίβεια των μετρήσεων.

## Γραφικά εργαλεία

Εκτός από τις πληροφορίες που μπορείτε να συλλέξετε από τη γραμμή εντολών, υπάρχουν και γραφικά εργαλεία παρακολούθησης συστήματος. Για παράδειγμα, στο Ubuntu Linux μπορείτε να επιλέξετε από το μενού Σύστημα -> Διαχείριση συστήματος -> Παρακολούθηση συστήματος:



Εικόνα 5-3. Παρακολούθηση συστήματος στο Ubuntu

## Διακοπή διεργασιών

Αν θέλετε να σταματήσετε μια διεργασία που δεν ανταποκρίνεται ή καταναλώνει υπερβολικούς πόρους του συστήματος μπορείτε να χρησιμοποιήσετε την εντολή `kill`. Αν έχετε την ευκαιρία, πρώτα τερματίστε τη διεργασία «ομαλά» στέλνοντας το σήμα `SIGTERM (15)`. Είναι μια εντολή για να τερματίσει ό,τι κάνει, σύμφωνα με τις διαδικασίες που προβλέπονται από το ίδιο το πρόγραμμα:

```
joe:~> ps -ef | grep mozilla
joe  25822  1  0 Mar11 ?    00:34:04 /usr/lib/mozilla-1.4.1/mozilla-
```

```
joe:~> kill -15 25822
```

Στο παραπάνω παράδειγμα, ο χρήστης *joe* σταμάτησε τον περιηγητή Mozilla γιατί "κρέμασε".

Αν θέλετε να διακόψετε άμεσα και οπωσδήποτε μια διεργασία χρησιμοποιήστε το ισχυρότερο σήμα, SIGKILL (9). Στο παρακάτω παράδειγμα, ο χρήστης *joe* σταματά μια διεργασία Mozilla που δεν αποκρίνεται:

```
joe:~> ps -ef | grep mozilla
joe    25915    1  0 Mar11 ?        00:15:06 /usr/lib/mozilla-1.4.1/mozilla-

joe:~> kill -9 25915

joe:~> ps -ef | grep 25915
joe    2634 32273  0 18:09 pts/4    00:00:00 grep 25915
```

Σε τέτοιες περιπτώσεις, μπορεί να θέλετε να ελέγξετε αν η διεργασία έχει όντως διακοπεί μπορείτε να επανεκτελέσετε το φίλτρο **grep** με το PID της διεργασίας. Αν δεν επιστραφεί τίποτε τότε είστε σίγουροι ότι την τερματίσατε.

Μεταξύ των διεργασιών που είναι δύσκολο να τερματιστούν είναι και το κέλυφος εντολών. Και αυτό είναι καλό: αν ήταν εύκολο να τερματιστεί, θα χάνατε το κέλυφος κάθε φορά που θα πατούσατε **Ctrl+C** έστω και κατά λάθος, αφού είναι ισοδύναμο με αποστολή SIGINT.

## Χρονοπρογραμματισμός διεργασιών

### Εκμετάλλευση του αδρανούς χρόνου

Συνήθως, ο φόρτος ενός συστήματος Linux (αναφερόμαστε σε συστήματα που χρησιμοποιούνται ως διακομιστές) είναι ιδιαίτερες αυξημένες κατά τις λεγόμενες ώρες γραφείου, ενώ μειώνεται σημαντικά τα βράδια και τα Σαββατοκύριακα. Αυτός ο χρόνος που το σύστημα είναι αδρανές ή λειτουργεί με χαμηλό φόρτο μπορεί να αξιοποιηθεί για την εκτέλεση χρονοβόρων διεργασιών.

Υπάρχουν τρεις τρόποι για να εκτελούμε τις διεργασίες με καθυστέρηση:

- Εκτέλεση μιας εντολής μετά από ένα προκαθορισμένο χρονικό διάστημα χρησιμοποιώντας την εντολή **sleep**. Η ώρα εκτέλεσης της διεργασίας εξαρτάται από την ώρα του συστήματος, τη στιγμή εισαγωγής της εντολής (προσδιορίζεται δηλαδή με σχετικό τρόπο).
- Εκτέλεση μιας εντολής σε προκαθορισμένη χρονική στιγμή χρησιμοποιώντας την εντολή **at**. Η εκτέλεση της εντολής αυτής εξαρτάται από την ώρα του συστήματος, (όχι από την ώρα εκτέλεσης της εντολής **at**), δηλαδή προσδιορίζεται με απόλυτο τρόπο.
- Χρήση της εντολής **cron** για εντολές που εκτελούνται περιοδικά, π.χ. μια φορά το μήνα, την εβδομάδα, την μέρα ή την ώρα.

Οι παραπάνω τρόποι αναλύονται στις παρακάτω ενότητες.

### Η εντολή **sleep**

Η εντολή **sleep** αναγκάζει το σύστημα να περιμένει πριν την εκτέλεση μιας εντολής. Ο χρόνος αναμονής προσδιορίζεται σε δευτερόλεπτα.

Μία σχετικά απλοϊκή χρήση της εντολής θα μπορούσε να είναι μια υπενθύμιση: αν για παράδειγμα θέλουμε να θυμηθούμε μια κλήση που πρέπει να πραγματοποιήσουμε σε μισή ώρα, μπορούμε να πληκτρολογήσουμε την εντολή:

```
(sleep 1800; echo "Call Jack") &
```

ώστε να εμφανιστεί το σχετικό μήνυμα μετά από 30 λεπτά. Το πιθανότερο είναι ότι θα θέλουμε οι εντολές να εκτελεστούν στο παρασκήνιο, ώστε εμείς να μπορούμε να δουλεύουμε κανονικά στο ενδιάμεσο διάστημα.

## Η εντολή at

Η εντολή **at** εκτελεί μια εντολή σε προκαθορισμένη χρονική στιγμή, χρησιμοποιώντας το προκαθορισμένο κέλυφος εντολών εκτός αν ο χρήστης προσδιορίσει κάτι άλλο.

Ο προγραμματισμός μιας διεργασίας με την εντολή **at** ξεκινάει με την πληκτρολόγηση της εντολής και της χρονικής στιγμής:

```
steven@home:~> at 14:37
warning: commands will be executed using (in order) a) $SHELL
                b) login shell c) /bin/sh
at> touch test2.php
at> <EOT>
job 1 at 2009-04-24 14:37
```

Η ολοκλήρωση της σύνταξης της εντολής δηλώνετε με τα πλήκτρα **Ctrl+D** οπότε και εμφανίζεται το μήνυμα "EOT".

Η επιλογή **-m** στην εντολή **at** στέλνει ένα mail στο χρήστη όταν η διεργασία ολοκληρωθεί ή εμφανίζει τους λόγους για τους οποίους η εντολή δεν μπορεί να εκτελεστεί. Η εντολή **atq** ταξινομεί τις διεργασίες και χρησιμοποιείται κυρίως για την αποφυγή ταυτόχρονης εκκίνησης. Με την εντολή **atrm** ο χρήστης μπορεί να ακυρώσει τις διεργασίες που έχει προηγουμένως προγραμματίσει να εκτελεστούν. Η εντολή **atrm** συνοδεύεται από το αναγνωριστικό της εργασίας όπως αυτό εμφανίζετε στη λίστα που προκύπτει από την εντολή **atq**.

Είναι καλή ιδέα να επιλέγετε "παράξενες" ώρες (δηλαδή όχι στρόγγυλες, π.χ. 01:00) για την εκτέλεση των διεργασιών σας γιατί συνήθως οι διεργασίες του συστήματος προγραμματίζονται τις "στρογγυλές" ώρες. Με αυτόν τον τρόπο αποφεύγετε την ταυτόχρονη εκτέλεση εντολών, η οποία επιβραδύνει το σύστημά σας. Για να αποφύγετε την εκτέλεση των διεργασιών την ίδια ώρα, μπορείτε να χρησιμοποιήσετε την εντολή **batch**, η οποία βάζει σε ουρά τις διεργασίες και τροφοδοτεί με αυτές ανά περιόδους το σύστημα με ένα ισορροπημένο τρόπο τέτοιο, ώστε να αποφεύγεται η υπερβολική κατανάλωση των πόρων του συστήματος.

## Cron και crontab

Το **cron** (η ονομασία προέκυψε από τη λέξη **chronograph**) είναι ένα πρόγραμμα, το οποίο επιτρέπει στους χρήστες και τους διαχειριστές ενός συστήματος να χρονοπρογραμματίζουν την εκτέλεση περιοδικών εργασιών. Οι πληροφορίες χρονοπρογραμματισμού συγκεντρώνονται σε αρχεία που ονομάζονται **crontabs** (η προεπιλεγμένη τοποθεσία του είναι στο **/etc**) το οποίο καλεί τα επιμέρους προγράμματα (ωριαία, ημερήσια, εβδομαδιαία, μηνιαία).

Μόνο ο διαχειριστής του συστήματος έχει πρόσβαση στα **crontabs** του συστήματος, ενώ κάθε άλλος χρήστης έχει πρόσβαση μόνο στα δικά του **crontabs**. Σε μερικά συστήματα (κάποιοι) χρήστες μπορεί να μην έχουν πρόσβαση στις λειτουργίες του **cron**.

Κάθε λεπτό, ο δαίμονας του **cron** ελέγχει αν υπάρχει κάποια προγραμματισμένη εργασία που πρέπει να εκτελεστεί. Κατά την εκτέλεση μιας διεργασίας το σύστημα εκτελεί τις κατάλληλες εντολές σαν να ήταν ο ιδιοκτήτης του αρχείου **crontab** και έπειτα στέλνει την έξοδο των εντολών στο χρήστη.

Ένα δημοφιλές συντακτικό του **cron** είναι το **Vixie cron**, το οποίο αναπτύχθηκε από τον Paul Vixie. Σε συστήματα που χρησιμοποιούν **Vixie cron**, οι εργασίες οι οποίες εκτελούνται ωριαία, ημερήσια, εβδομαδιαία και μηνιαία αποθηκεύονται σε ένα ξεχωριστό φάκελο στη διαδρομή

/etc, σε αντίθεση με την συνήθη πρακτική στο UNIX cron, όπου όλες οι εργασίες αποθηκεύονται σε ένα μεγάλο αρχείο.

Παράδειγμα από ένα αρχείο Vixie crontab:

```
[root@blob /etc]# more crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
# commands to execute every hour
01 * * * * root run-parts /etc/cron.hourly
# commands to execute every day
02 4 * * * root run-parts /etc/cron.daily
# commands to execute every week
22 4 * * 0 root run-parts /etc/cron.weekly
commands to execute every month
42 4 1 * * root run-parts /etc/cron.monthly
```

Μπορείτε να χρησιμοποιήσετε την εντολή `crontab -l` για να εμφανίσετε τα crontabs του τρέχοντος χρήστη.

Πριν ξεκινήσει ο χρονοπρογραμματισμός πρέπει πρώτα να πάρουν τιμή κάποιες μεταβλητές, σε κάθε γραμμή ορίζεται μια εργασία προς εκτέλεση και κάθε τέτοια γραμμή ξεκινά με 5 πεδία, τα οποία αντιπροσωπεύουν την ώρα και τη μέρα εκτέλεσης. Το πρώτο πεδίο περιέχει τα λεπτά (από 0 έως 59), το δεύτερο προσδιορίζει την ώρα εκτέλεσης (0-23), το τρίτο την μέρα του μήνα (1-31), το επόμενο τον αριθμό του μήνα (1-12) και το τελευταίο τη μέρα της εβδομάδας (0-7, το 0 και το 7 είναι η Κυριακή). Ο αστερίσκος σε καθένα από αυτά τα πεδία συμβολίζει οποιαδήποτε τιμή, σύμφωνα πάντα με το εύρος τιμών του πεδίου. Η εισαγωγή λίστας στοιχείων σε κάποιο πεδίο είναι επιτρεπτή, για παράδειγμα για να εκτελέσουμε μια εργασία από τη Δευτέρα ως την Παρασκευή εισάγουμε στο τελευταίο πεδίο 1-5, για να εκτελέσουμε μια εργασία τη Δευτέρα, Τετάρτη και Παρασκευή εισάγουμε 1,3,5.

Μετά από αυτά γράφουμε στην τελευταία στήλη το όνομα του χρήστη, για τον οποίο πρέπει να εκτελεστεί η προγραμματισμένη εργασία. Το παραπάνω παράδειγμα είναι από ένα Vixie cron, στο οποίο ο διαχειριστής εκτελεί το πρόγραμμα `run-parts` σε κανονικά διαστήματα, έχοντας ως όρισμα και τις κατάλληλες διαδρομές αρχείων. Στους καταλόγους αυτούς, οι εργασίες που πρέπει να εκτελεστούν σε προγραμματισμένο χρόνο είναι αποθηκευμένες ως σενάρια κελύφους, όπως για παράδειγμα το σενάριο αυτό που εκτελείται ημερήσια και ανανεώνει τη βάση δεδομένων χρησιμοποιώντας την εντολή `locate`:

```
billy@ahost cron.daily]$ cat slocate.cron
#!/bin/sh
renice +19 -p $$ >/dev/null 2>&1
/usr/bin/updatedb -f "nfs,smbfs,ncpfs,proc,devpts" -e \
"/tmp,/var/tmp, /usr/tmp,/afs,/net"
```

Οι χρήστες πρέπει να επεξεργάζονται τα crontabs με ασφαλή τρόπο χρησιμοποιώντας την εντολή `crontab -e`. Αυτό θα αποτρέψει ένα χρήστη να ανοίξει κατά λάθος παραπάνω από ένα αντίγραφο του ίδιου αρχείου crontab. Ο προεπιλεγμένος επεξεργαστής κειμένου που χρησιμοποιείται είναι ο `vi` (αλλά μπορεί να χρησιμοποιηθεί οποιοσδήποτε επεξεργαστής κειμένου).

Κατά την έξοδο από το σύστημα θα ειδοποιηθείτε ότι έχει ενεργοποιηθεί ένα καινούργιο crontab. Αυτή η καταχώρηση crontab υπενθυμίζει στο χρήστη *billy* να πάει στο γυμναστήριο κάθε Πέμπτη βράδυ:

```
billy:~> crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (/tmp/crontab.20264 installed on Sun Jul 20 22:35:14 2003)
# (Cron version -- $Id: chap4.xml,v 1.27 2006/10/26 15:37:52 tille Exp $)
38 16 * * 3 mail -s "sports evening" billy
```

Ύστερα από την προσθήκη μιας νέας προγραμματισμένης εργασίας, το σύστημα θα επιβεβαιώσει την εγκατάσταση του καινούργιου crontab. Για να ισχύουν οι αλλαγές που κάνατε θα πρέπει να επανεκκινήσετε το δαίμονα *cron*. Στο παράδειγμα, ο χρήστης *billy* προσθέτει μια νέα γραμμή, η οποία αναφέρεται σε ένα σενάριο με το όνομα *backup*:

```
billy:~> crontab -e
45 15 * * 3 mail -s "sports evening" billy
4 4 * * 4,7 /home/billy/bin/backup.sh

<--write and quit-->

crontab: installing new crontab

billy:~>
```

Το σενάριο *backup.sh* εκτελείται κάθε Πέμπτη και Κυριακή. Υπενθυμίζεται ότι η έξοδος των εντολών, αν υπάρχει, στέλνεται στο χρήστη στον οποίο ανήκει το crontab. Αν δεν έχει διαμορφωθεί κάποια υπηρεσία ταχυδρομείου, θα βρείτε την έξοδο των εντολών σε ένα αρχείο κειμένου στη διαδρομή */var/spool/mail/<your\_username>*.

Δε χρειάζεται να προσδιοριστεί ποιος χρήστης εκτελεί της εντολές. Εκτελούνται εξ ορισμού σύμφωνα με τα δικαιώματα του χρήστη στον οποίο ανήκουν.

## Περίληψη

Το Linux είναι ένα πολυ-χρηστικό και πολυ-εργασιακό λειτουργικό σύστημα, το οποίο χειρίζεται τις διεργασίες με τρόπο ίδιο με αυτό του UNIX. Η ταχύτητα εκτέλεσης των εντολών εξαρτάται από χιλιάδες πράγματα. Ανάμεσα στα άλλα, μάθαμε αρκετές νέες εντολές για να οπτικοποιούμε και να χειριζόμαστε διεργασίες. Να μία λίστα:

### Πίνακας 5-3. Νέες εντολές στο κεφάλαιο 5: Διεργασίες

Εντολή	Σημασία
<b>at</b>	Δημιουργεί μια ουρά από εργασίες που πρόκειται να εκτελεστούν.
<b>atq</b>	Εμφανίζει μια λίστα με τις εργασίες του χρήστη που βρίσκονται σε αναμονή.
<b>atrm</b>	Διαγράφει εργασίες, προσδιοριζόμενες από τον αριθμό τους.
<b>batch</b>	Εκτελεί εντολές όταν το επιτρέπει ο φόρτος του συστήματος..

Εντολή	Σημασία
<b>crontab</b>	Δημιουργεί αρχεία crontab διάφορους χρήστες.
<b>halt</b>	Σταματά την λειτουργία του συστήματος.
<b>init level</b> <i>run</i>	Αρχειοποίηση συστήματος και ελέγχου διεργασιών..
<b>jobs</b>	Εμφανίζει τις εργασίες που εκτελούνται.
<b>kill</b>	Τερματίζει μια διεργασία.
<b>mesg</b>	Έλεγχος του δικαιώματος εγγραφής στο τερματικό σας.
<b>netstat</b>	Εμφανίζει τις συνδέσεις δικτύου, τους δρομολογητές, στατιστικά διεπαφών, κεκαλυμμένες συνδέσεις και συμμετοχές σε πολυεκπομπές.
<b>nice</b>	Εκτελεί ένα πρόγραμμα με τροποποιημένη προτεραιότητα.
<b>pgrep</b>	Εμφανίζει τις διεργασίες που εκτελούνται.
<b>ps</b>	Αναφέρει την κατάσταση των διεργασιών.
<b>pstree</b>	Εμφανίζει τις διεργασίες σε μορφή δενδρογράμματος.
<b>reboot</b>	Σταματά την λειτουργία του συστήματος (επανεκκίνηση).
<b>renice</b>	Αλλάζει την προτεραιότητα των εκτελούμενων διεργασιών.
<b>shutdown</b>	Τερματίζει τη λειτουργία του συστήματος.
<b>sleep</b>	Δημιουργεί μια καθυστέρηση (περίοδο αδράνειας) συγκεκριμένου χρόνου.
<b>time</b>	Χρονομετράει μια εντολή ή εμφανίζει πληροφορίες για τη χρήση πόρων του συστήματος.
<b>top</b>	Εμφανίζει ποιες εργασίες χρησιμοποιούν μεγαλύτερο ποσοστό της CPU.
<b>uptime</b>	Εμφανίζει πόση ώρα είναι το σύστημα σε λειτουργία.
<b>vmstat</b>	Εμφανίζει στατιστικά στοιχεία για την εικονική μνήμη.
<b>w</b>	Δείχνει ποιοι χρήστες είναι συνδεδεμένοι στο σύστημα και τι διεργασίες εκτελούν.
<b>wall</b>	Στέλνει ένα μήνυμα σε όλα τα τερματικά.
<b>who</b>	Δείχνει ποιοι χρήστες είναι συνδεδεμένοι στο σύστημα.
<b>write</b>	Στέλνει ένα μήνυμα σε ένα συγκεκριμένο χρήστη.

## Ασκήσεις

Αυτές είναι μερικές ασκήσεις οι οποίες θα βοηθήσουν στην εξοικείωση με τις διεργασίες που εκτελούνται στο σύστημα σας.

## Γενικά

- Εκτέλεσε την εντολή **top** σε ένα τερματικό ενώ εργάζεστε σε κάποιο άλλο.
- Εκτέλεσε την εντολή **ps**.
- Διαβάστε τις σελίδες βοήθειας και βρείτε πως μπορείτε να εμφανίσετε όλες τις διεργασίες.
- Εκτέλεσε την εντολή **find /**. Πιο το αποτέλεσμα αυτής κατά την εκκίνηση του συστήματος; Σταμάτησε την εντολή αυτή.
- Σε γραφικό περιβάλλον, εκτέλεσε το πρόγραμμα **xclock** στο προσκήνιο. Έπειτα άφησε το να εκτελείται στο παρασκήνιο. Σταμάτησε το πρόγραμμα αυτό χρησιμοποιώντας την εντολή **kill**.
- Τρέξε το **xcals** κατευθείαν στο παρασκήνιο, έτσι ώστε η προτροπή του τερματικού να ελευθερωθεί.
- Τι κάνει η εντολή **kill -9 -1**;
- Ανοίξτε δυο τερματικά ή δύο παράθυρα τερματικών και χρησιμοποίησε την εντολή **write** για να στείλεις ένα μήνυμα από το ένα στο άλλο.
- Εκτέλεσε την εντολή **dmesg**. Τα μήνυμα επιστρέφει?
- Πόσο χρόνο χρειάζεται για να εκτελεστεί η εντολή **ls** στον τρέχοντα κατάλογο;
- Βασισμένοι στις καταχωρήσεις των διεργασιών στη διαδρομή **/proc**, οι οποίες το δικό σας UID, πως θα εργαστείτε για να βρείτε ποιες πραγματικά διεργασίες αναπαριστούν;
- Πόσο χρόνο δουλεύει το σύστημα σας;
- Ποιο είναι το τρέχον TTY σας;
- Αναφέρατε 3 εντολές οι οποίες δεν μπορούν να έχουν το **init** σαν γονέα..
- Αναφέρατε 3 εντολές οι οποίες χρησιμοποιούν SUID. Εξηγήστε γιατί.
- Αναφέρατε 3 εντολές οι οποίες είναι γενικά υπεύθυνες για το μεγάλο φόρτο του συστήματος.

## Εκκίνηση, init κλπ.

- Μπορείτε να επανεκκινήσετε το σύστημα σαν κανονικός χρήστης; Γιατί;
- Σύμφωνα με το τρέχον επίπεδο εκτέλεσης του συστήματος, ανάφερε τα βήματα που χρειάζονται για την διαδικασία τερματισμού της λειτουργίας του.
- Πώς μπορείτε να αλλάξετε το επίπεδο εκτέλεσης του συστήματος; Αλλάξτε το επίπεδο από το προεπιλεγμένο στο επίπεδο 1 και αντίστροφα.
- Δημιουργήστε μια λίστα με όλες τις υπηρεσίες και όλους του δαίμονες που ξεκινούν κατά την εκκίνηση του συστήματος.
- Ποιος πυρήνας φορτώνεται κατά την εκκίνηση;
- Υποτίθεται ότι πρέπει να εκκινήσετε μια εξωτική υπηρεσία διακομιστή κατά την εκκίνηση. Μέχρι τώρα, μετά την εκκίνηση του υπολογιστή και την σύνδεση σας στο σύστημα ξεκινούσατε χειροκίνητα την υπηρεσία διακομιστή αυτό χρησιμοποιώντας ένα σενάριο με το όνομα **deliver\_pizza** στον οικείο σας κατάλογο. Τι θα πρέπει να κάνετε έτσι ώστε η υπηρεσία αυτή να εκκινεί αυτόματα στο επίπεδο εκτέλεσης 4, το οποίο προσδιορίσατε γι' αυτό και μόνο το σκοπό;

## Χρονοπρογραμματισμός

Χρησιμοποιήστε την εντολή `sleep` για να δημιουργήσετε μια υπενθύμιση ότι τα μακαρόνια είναι έτοιμα σε 10 λεπτά.

- Προγραμματίστε με την εντολή `at` μια εργασία η οποία αντιγράφει όλα τα αρχεία από τον οικείο σας φάκελο στη διαδρομή `/var/tmp` μέσα σε μισή ώρα. Ίσως να χρειαστεί να δημιουργήσετε τον υποκατάλογο `/var/tmp`.
- Δημιουργήστε ένα cronjob το οποίο εκτελεί την παραπάνω εργασία από Δευτέρα μέχρι Παρασκευή κατά την ώρα του μεσημεριανού.
- Έλεγε αν λειτουργεί.
- Δημιουργήστε μια λάθος καταχώρηση `crontab` αυτό, για παράδειγμα να χρησιμοποιεί τη λάθος εντολή `corpy` αντί της `cp`. Τί συμβαίνει κατά την εκτέλεση του;