



Προγραμματισμός και Διαχείριση Συστημάτων I

Μάθημα 8ο
Διεργασίες
Πληροφορίες, έλεγχος, διαχείριση



Κ. Μαργαρίτης, Π. Μιχαηλίδης
Γ' Εξάμηνο 2007

Περιεχόμενα

Βασικές έννοιες διεργασιών

Πληροφορίες διεργασιών

Ταυτότητες διεργασιών

Έλεγχος διεργασιών

Απόδοση διεργασιών και συστήματος

Χρονοπρογραμματισμός διεργασιών

Πρόγραμμα - διεργασία

Πρόγραμμα

Συλλογή από bytes που αποτελούν τον εκτελέσιμο κώδικα αποθηκευμένο σε ένα αρχείο.

Μετά το κλείσιμο του υπολογιστή τα προγράμματα παραμένουν στο δίσκο.

Διεργασία

Ένα στιγμιότυπο του προγράμματος φορτωμένο στη κύρια μνήμη του υπολογιστή προς εκτέλεση.

Το δημιουργεί και το διαχειρίζεται ο πυρήνας.

Κάθε φορά που ένα πρόγραμμα εκτελείται δημιουργείται μια νέα διεργασία.

Μπορούν να υπάρχουν πολλές διεργασίες που εκτελούν (είναι στιγμιότυπα από..) το ίδιο πρόγραμμα.

Συστατικά μιας διεργασίας

Από την σκοπιά του πυρήνα, μια διεργασία είναι:

- Ταυτότητα διεργασίας (Process ID)

- Ταυτότητα γονικής διεργασίας (Parent Process ID)

- Ιδιοκτήτης, Ομάδα (UID, GID)

- Ένας χώρος διευθύνσεων (σύνολο σελίδων μνήμης με κώδικα, βιβλιοθήκες δεδομένα, χώρος εργασίας, stack, / heap)

- Τρέχουσα κατάσταση (π.χ. ποια εντολή εκτελείται αυτή τη στιγμή, τιμές καταχωρητών, context)

- Προτεραιότητα εκτέλεσης

- Χρήση πόρων (είσοδος / έξοδος, δίκτυο κλπ.)

- Μάσκα διακοπών (ποια σήματα έχουν ανασταλεί)

Διεργασία - νήμα

Το νήμα (thread) είναι:

Διεργασία χωρίς .. πολλή ιδιωτικότητα.

Κοινός χώρος διευθύνσεων πλην των ρητά τοπικών μεταβλητών.

Κοινό περιβάλλον εκτέλεσης (context).

Καταστάσεις διεργασίας

Μια διεργασία βρίσκεται σε μια από τις τέσσερις καταστάσεις

Εκτελούμενη – Εκτελείται τώρα

Εκτελέσιμη – Μπορεί να εκτελεστεί όταν έχει σειρά

Σε αναμονή – Περιμένει για κάποιους πόρους

Δεν παίρνει χρόνο CPU μέχρι ο πόρος να είναι διαθέσιμος

Zombie – προσπαθεί να τερματιστεί

Σε αναστολή – δεν επιτρέπεται να εκτελεστεί

Όπως στην αναμονή, δεν μπορεί να αφυπνιστεί μέχρι να λάβει κατάλληλο σήμα

Multuser - Multitasking

Το Linux είναι ένα λειτουργικό σύστημα πολυχρηστικό (multuser) και πολυεργασιακό (multitasking).

Κάθε χρήστης απαιτεί (τουλάχιστο) μία διεργασία, αυτή που εξυπηρετεί τη σύνδεσή του.

Εργασία (task) λέγεται το σύνολο των δημιουργούμενων διεργασιών που πρέπει να εκτελεστούν για να ολοκληρωθεί η δουλειά ενός χρήστη.

Στα συστήματα Linux περισσότεροι από ένας χρήστες μπορούν να εκτελούν την ίδια χρονική στιγμή περισσότερες από μια εργασίες ο καθένας. Η κάθε εργασία μπορεί να απαιτεί συντονισμό πολλών διεργασιών.

Πολυ-(δι)εργασία

Σε κάθε χρονική στιγμή στην (ή σε κάθε) CPU του συστήματος εκτελείται μόνο μια διεργασία. Ο πυρήνας εκχωρεί σε αυτή τη διεργασία μικρή ποσότητα χρόνου για να εκτελεστεί. Αυτός ο χρόνος ονομάζεται time slice και η όλη διαδικασία χρονομερισμός (time slicing/sharing).

Ο χρονοδρομολογητής (scheduler) του πυρήνα διατηρεί μια ουρά διεργασιών, με τα χαρακτηριστικά τους, τη κατάστασή τους και το βαθμό προτεραιότητάς τους. Όταν ο χρόνος της διεργασίας εξαντληθεί (ή αν αυτή τερματίσει ή ανασταλεί για κάποιο λόγο) τότε ο scheduler διακόπτει την εκτέλεση της παρούσας διεργασίας και την εναλλάσσει με κάποια άλλη από την ουρά των διεργασιών.

Διεργασίες προσκήνιου (διαλογικές)

Μια διεργασία εκτελείται είτε στο προσκήνιο (foreground) είτε στο παρασκήνιο (background).

Διεργασία προσκήνιου(Διαλογική διεργασία)

Στο προσκήνιο εκτελείται μια διεργασία κάθε στιγμή. Έχει τον πλήρη έλεγχο του φλοιού ο οποίος βρίσκεται σε κατάσταση αναμονής μέχρι να ολοκληρωθεί η εκτέλεση της εντολής.

Παράδειγμα:

```
$ ls -al
```

Διεργασίες παρασκηνίου (1)

Στο παρασκήνιο εκτελούνται μια ή περισσότερες διεργασίες την 'ίδια' χρονική στιγμή από το ίδιο τερματικό (φλοιό).

Οι διεργασίες παρασκηνίου 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 στο παρασκήνιο..).

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 παρασκηνίου (2)

Η διεργασία παρασκηνίου είναι χρήσιμη για δύο λόγους:

Εξυπηρετεί την ταυτόχρονη εκτέλεση αρκετών διεργασιών μέσω του ίδιου παραθύρου τερματικού (φλοιού).

Διευκολύνει τις εντολές που απαιτούν πολύ χρόνο για την εκτέλεση τους ώστε να αποφύγουμε τη δέσμευση του φλοιού για μεγάλο χρονικό διάστημα.

Πληροφορίες διεργασιών

Για να πάρουμε μια λίστα τρεχουσών διεργασιών στο σύστημα μας και να παρακολουθήσουμε τη συμπεριφορά τους, το Linux παρέχει αρκετά εργαλεία :

ps

pstree

top

Εντολή ps

Η εντολή ps μας επιτρέπει να ελέγξουμε την κατάσταση διεργασιών. Παράγει μια λίστα πληροφοριών κατάστασης διεργασιών και τα χαρακτηριστικά τους όπως τα όνομα, αριθμό διεργασίας, ιδιοκτήτη, ποσοστό χρησιμοποίησης CPU κλπ. Η ps δεν είναι εντολή πραγματικού χρόνου και παίρνει αντίγραφο των διεργασιών που εκτελούνται εκείνη τη στιγμή.

Σύνταξη: ps [options]

- e θα περιλάβει όλες τις διεργασίες που εκτελούνται αυτή τη στιγμή.
- f θα παράγει μια πλήρη λίστα.

Εντολή ps – Παράδειγμα (1)

```
$ 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: όνομα χρήστη ιδιοκτήτη της διεργασίας.

PID: η ταυτότητα της διεργασίας.

PPID: η ταυτότητα της γονικής 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 ps – Παράδειγμα (2)

```
kmarg@pdp1test1:~$ ps ps --help
***** simple selection *****
-A all processes
-N negate selection
-a all w/ tty except session leaders
-d all except session leaders
name
-e all processes
T all processes on this terminal
a all w/ tty, including other users
g OBSOLETE -- DO NOT USE
names)
r only running processes
x processes w/o controlling ttys
***** output format *****
-o,o user-defined -f full
-j,j job control s signal
-O,O preloaded -o v virtual memory
-l,l long u user-oriented
-F extra full X registers
***** misc options *****
-V,V show version L list format codes f ASCII art forest
-m,m,-L,-T,H threads S children in sum -y change -l format
-M,Z security data c true command name -c scheduling class
-w,w wide output n numeric WCHAN,UID -H process hierarchy

***** selection by list *****
-C by command name
-G by real group ID (supports names)
-U by real user ID (supports names)
-g by session OR by effective group
name
-p by process ID
-s processes in the sessions given
-t by tty
-u by effective user ID (supports
names)
U processes for specified users
t by tty
***** long options *****
--Group --User --pid --cols --ppid
--group --user --sid --rows --info
--cumulative --format --deselect
--sort --tty --forest --version
--heading --no-heading --context
```

Εντολή pstree

Η εντολή pstree παράγει ένα ιεραρχικό δένδρο των τρεχουσών διεργασιών.

```
kmarg@pdptest1:~$ pstree
init--acpid
    |--atd
    |--ccsd--{ccsd}
    |--cron
    |--cupsd
    |--dbus-daemon
    |--dd
    |--events/0
    |--gdm--gdm--Xorg
    |       |
    |       |--gdmgreeter
    |--6*[getty]
    |--hald--hald-runner--hald-addon-acpi
    |               |
    |               |--hald-addon-keyb
    |
    |...
    |--mdadm
    |--miniserv.pl
    |--python
    |--sdpd
    |--shpchpd_event
    |--sshd--sshd--sshd--bash--pstree
    |--syslogd
    |--udev
    |--watchdog/0
```


Εντολή top

Η εντολή top παράγει μια λίστα με περιοδική, πραγματικού χρόνου, ενημέρωση των διεργασιών που εκτελούνται στο σύστημα. Εμφανίζει πληροφορίες χρήσης πόρων του συστήματος, προτεραιότητας κλπ.

```
$ 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

Διαλογική λειτουργία εντολής top

Η εντολή top, εκτός από επιλογές γραμμής εντολής, παρέχει μεγάλο εύρος διαλογικών εντολών, οι οποίες μπορούν να εισαχθούν κατά τη λειτουργία της εντολής. Με 'h' λαμβάνουμε αυτή τη βοήθεια

```
Help for Interactive Commands - procps version 3.2.6
```

```
Window 2:Job: Cumulative mode Off.  System: Delay 3.0 secs; Secure mode Off.
```

```
Z,B      Global: 'Z' change color mappings; 'B' disable/enable bold
l,t,m    Toggle Summaries: 'l' load avg; 't' task/cpu stats; 'm' mem info
l,I      Toggle SMP view: 'l' single/separate states; 'I' Irix/Solaris mode
```

```
....
```

```
k,r      Manipulate tasks: 'k' kill; 'r' renice
d or s   Set update interval
W        Write configuration file
q        Quit
```

```
( commands shown with '.' require a visible task display window )
```

```
Press 'h' or '?' for help with Windows,
any other key to continue
```

Ταυτότητες διεργασίας

Ταυτότητα διεργασίας – Process ID (PID)

Μοναδική ταυτότητα, ανακυκλώνεται

Ταυτότητα γονικής διεργασίας – Parent PID (PPID)

Όταν δημιουργείται μια διεργασία, υπάρχει πάντα μια γονική και μια θυγατρική

Ταυτότητα (και Εκτελεστική ταυτότητα) χρήστη - UID και EUID

Η Εκτελεστική ταυτότητα (Effective UID) μπορεί να είναι διαφορετική από τη UID λόγω χρήσης setuid ή sudo

χρησιμοποιείται για να προσδιορίζει τι άδειες έχει η διεργασία πάντα καταγράφεται η αρχική EUID (δηλαδή την UID)

Ταυτότητα (και Εκτελεστική ταυτότητα) ομάδας - GID και EGID

SUID και SGID

Προσωρινή αλλαγή User ID ή Group ID για την εκτέλεση συγκεκριμένων εντολών.

```
kmarg@pdptest1:~$ which passwd
/usr/bin/passwd
kmarg@pdptest1:~$ ls -l /usr/bin/passwd
-rwsr-xr-x 1 root root 27132 2006-07-11 15:51 /usr/bin/passwd
kmarg@pdptest1:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1518 2007-05-16 19:52 /etc/passwd

kmarg@pdptest1:~$ which write
/usr/bin/write
...
kmarg@pdptest1:~$ ls -l /usr/bin/bsd-write
-rwxr-sr-x 1 root tty 7768 2005-10-25 05:13 /usr/bin/bsd-write

kmarg@pdptest1:~$ which wall
/usr/bin/wall
kmarg@pdptest1:~$ ls -l /usr/bin/wall
-rwxr-sr-x 1 root tty 10292 2007-10-22 07:49 /usr/bin/wall
```

Έλεγχος διεργασιών

command : Εκτέλεση εντολής **command** στο προσκήνιο.

command & : Εκτέλεση της εντολής **command** στο παρασκήνιο

jobs : Εμφάνιση των εντολών που εκτελούνται στο παρασκήνιο.

Ctrl+Z : Αναστολή διεργασίας που εκτελείται στο προσκήνιο.

Ctrl+C : Τερματισμός διεργασίας που εκτελείται στο προσκήνιο.

%n : Κάθε διεργασία που εκτελείται στο παρασκήνιο παίρνει έναν αριθμό. Με την έκφραση **%n** Μια εργασία αναφέρεται με τον αριθμό της, για παράδειγμα **fg %2**.

fg : Επαναφορά εργασίας παρασκηνίου στο προσκήνιο (κατάληψη φλοιού, γίνεται διαλογική).

bg : Τοποθέτηση διεργασίας στο παρασκήνιο (σαν να είχε ξεκινήσει με **&**).

kill : Αποστολή σήματος σε διεργασία.

Σήματα

Τα σήματα είναι αιτήσεις διακοπής (interrupt) σε επίπεδο διεργασιών

Χρήσεις

Δια-διεργασιακή επικοινωνία.

Ο χρήστης μπορεί να τερματίσει, να διακόψει ή να αναστείλει τις διεργασίες (π.χ. Ctrl-C, Ctrl-Z).

Σήματα μπορούν να αποσταλούν από τον διαχειριστή συστήματος (με την kill) για διάφορους σκοπούς.

Επίσης σήματα μπορούν να αποσταλούν από τον πυρήνα όταν μια διεργασία παραβιάζει ένα κανόνα π.χ. Διαίρεση με το μηδέν

Χειρισμός σημάτων

Η διεργασία μπορεί να προσδιορίσει μια ρουτίνα εξυπηρέτησης σήματος για κάθε συγκεκριμένο σήμα.

Αν δεν υπάρχει ρουτίνα, τότε ο πυρήνας εκτελεί μια προκαθορισμένη ενέργεια.

Όταν η ρουτίνα που έλαβε το σήμα τελειώσει, τότε συνεχίζεται η εκτέλεση της διεργασίας από το σημείο που είχε γίνει η αποστολή του σήματος.

Η διεργασία μπορεί να ζητήσει συγκεκριμένα σήματα να αγνοούνται ή να αναστέλλονται (signal mask).

Αν μια διεργασία λάβει ένα σήμα ενώ είναι σε αναστολή, ένα στιγμιότυπο του σήματος αποθηκεύεται μέχρι να ενεργοποιηθεί η διεργασία και να το χειριστεί.

Σημαντικά σήματα

#	Name	Description	Default	Catch?	Block?	Dump?
1	HUP	Hangup	Terminate	Yes	Yes	No
		<i>Reset request; clean up process on terminal (modem hangup)</i>				
		<i>*csh processes ignore HUP; bash users need nohup command</i>				
2	INT	Interrupt	Terminate	Yes	Yes	No
		<i>Control-C, can catch and clean up before quitting.</i>				
3	QUIT	Quit	Terminate	Yes	Yes	Yes
		<i>Similar to TERM, but generates a core dump</i>				
9	KILL	Kill	Terminate	No	No	No
		<i>Never received by process; OS terminates process.</i>				
*	BUS	Bus error	Terminate	Yes	Yes	Yes
		<i>Error signal. Typically memory alignment problem.</i>				
11	SEGV	Segmentation Fault	Terminate	Yes	Yes	Yes
		<i>Error signal. Typically memory access to protected space.</i>				

Περισσότερα σήματα

#	Name	Description	Default	Catch?	Block?	Dump?
15	TERM	Software termination	Terminate	Yes	Yes	No
		<i>Request to terminate execution. Process can clean up, exit.</i>				
*	STOP	Stop	Stop	No	No	No
		<i>OS suspends execution of process until CONT received.</i>				
*	TSTP	Keyboard stop	Stop	Yes	Yes	Yes
		<i>Keyboard Ctrl-Z request to stop. Catchable.</i>				
*	CONT	Continue after stop	Ignore	Yes	No	No
		<i>Continue after STOP.</i>				
*	WINCH	Window changed	Ignore	Yes	Yes	No
		<i>Sent by terminal emulator when config changes (resize)</i>				
*	USR1	User-defined	Terminate	Yes	Yes	No
		<i>User defined. Apache restarts gracefully.</i>				
*	USR2	User-defined	Terminate	Yes	Yes	No

Εντολή kill

Σύνταξη: kill [-signalid] pid

Η εντολή kill στέλνει το σήμα τερματισμού 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 (TERM).

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

```
$
```

Έλεγχος διεργασιών – παράδειγμα (1)

```
kmarg@kmarg-laptop:~$ gedit &
[1] 17737
kmarg@kmarg-laptop:~$ jobs
[1]+  Running                  gedit &
kmarg@kmarg-laptop:~$ xpdf &
[2] 17768
kmarg@kmarg-laptop:~$ jobs
[1]-  Running                  gedit &
[2]+  Running                  xpdf &
kmarg@kmarg-laptop:~$ kill %2
kmarg@kmarg-laptop:~$ jobs
[1]-  Running                  gedit &
[2]+  Terminated              xpdf
kmarg@kmarg-laptop:~$ kill %1
[1]+  Terminated              gedit
kmarg@kmarg-laptop:~$ jobs
kmarg@kmarg-laptop:~$
```

Έλεγχος διεργασιών – παράδειγμα (2)

```
kmarg@kmarg-laptop:~$ gedit &
[1] 19024
kmarg@kmarg-laptop:~$ jobs
[1]+  Running                  gedit &
kmarg@kmarg-laptop:~$ fg %1
gedit
.....(CTRL Z)

[1]+  Stopped                  gedit
kmarg@kmarg-laptop:~$ jobs
[1]+  Stopped                  gedit
kmarg@kmarg-laptop:~$ bg %1
[1]+ gedit &
kmarg@kmarg-laptop:~$ jobs
[1]+  Running                  gedit &
kmarg@kmarg-laptop:~$ kill %1
$
```

Απόδοση διεργασιών και συστήματος

uptime

Πληροφορίες για το χρόνο λειτουργίας και τη μέση απόδοση του συστήματος.

time

Πληροφορίες για το χρόνο εκτέλεσης μιας εντολής ή διεργασίας

```
kmarg@pdptest1:~$ time ls
afile      download  hello.c   kati      new       PoP       services
units174
Desktop    Examples  hier      MPWin     new-dir   ps_test   test.xyz

real       0m0.015s
user       0m0.008s
sys        0m0.004s
```

Προτεραιότητες (1)

top

Πολύ περισσότερες πληροφορίες για κάθε διεργασία.

nice

Πόσο 'καλή' είναι μια διεργασία; Όσο πιο 'καλή' τόσο χαμηλότερη η προτεραιότητά της. Εύρος τιμών από -20 (ψηλή προτεραιότητα) έως +19 (χαμηλή προτεραιότητα).

Μπορούμε να αλλάξουμε τη προτεραιότητα μέσω της top ή με τις εντολές nice, renice.

Προτεραιότητες (2)

Ο χρήστης ή μια διεργασία μπορούν να αυξήσουν αλλά όχι να ελαττώσουν το niceness.

Ο root μπορεί να δώσει χαμηλότερη προτεραιότητα.

Δεν πειράζουμε τη προτεραιότητα των διεργασιών εισόδου / εξόδου (πχ πληκτρολόγιο) ούτε των διαλογικών διεργασιών (προσκηνίου).

Παραδείγματα

```
nice +5 ~/bin/longtask
```

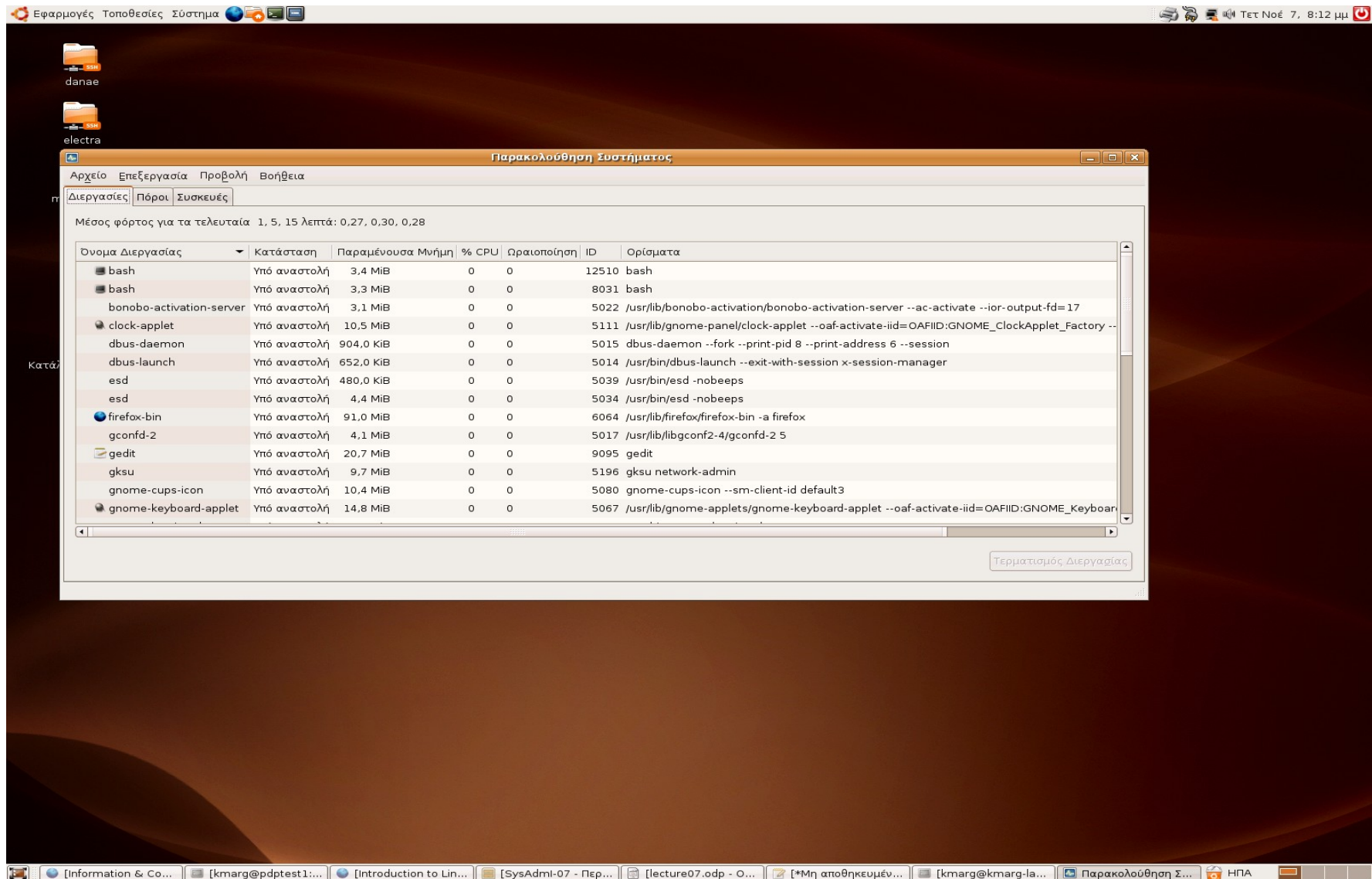
```
renice -5 8829
```

```
sudo renice 5 -u boggs
```


Προβληματικές διεργασίες

- Τι μπορούμε να κάνουμε σχετικά με τις διεργασίες που χρησιμοποιούν μια ασυνήθιστη ποσότητα πόρων (μνήμη, CPU, χώρος δίσκου);
- Αναγνωρίζουμε τις μεγάλες ποσότητες πόρων με τις εντολές `ps` και `top`.
- Ερχόμαστε σε επαφή με τον ιδιοκτήτη και το ρωτάμε σχετικά με την χρήση των πόρων.
- Αναστέλλουμε χρησιμοποιώντας το σήμα `STOP` (ίσως τερματίζουμε την εργασία).
- Ερχόμαστε σε επαφή με τον ιδιοκτήτη για επανεκκίνηση ή καθυστέρηση.
- Κάνουμε `renice` τις διεργασίες που χρησιμοποιούν πολύ CPU.

Gnome system monitor (1)



The screenshot shows the GNOME System Monitor application window titled "Παρακολούθηση Συστήματος". The "Διεργασίες" (Processes) tab is selected, displaying a list of running processes. The window includes a menu bar with "Αρχείο", "Επεξεργασία", "Προβολή", and "Βοήθεια". Below the menu bar are tabs for "Διεργασίες", "Πόροι", and "Συσκευές". A status bar at the bottom of the window shows "Μέσος φόρτος για τα τελευταία 1, 5, 15 λεπτά: 0,27, 0,30, 0,28".

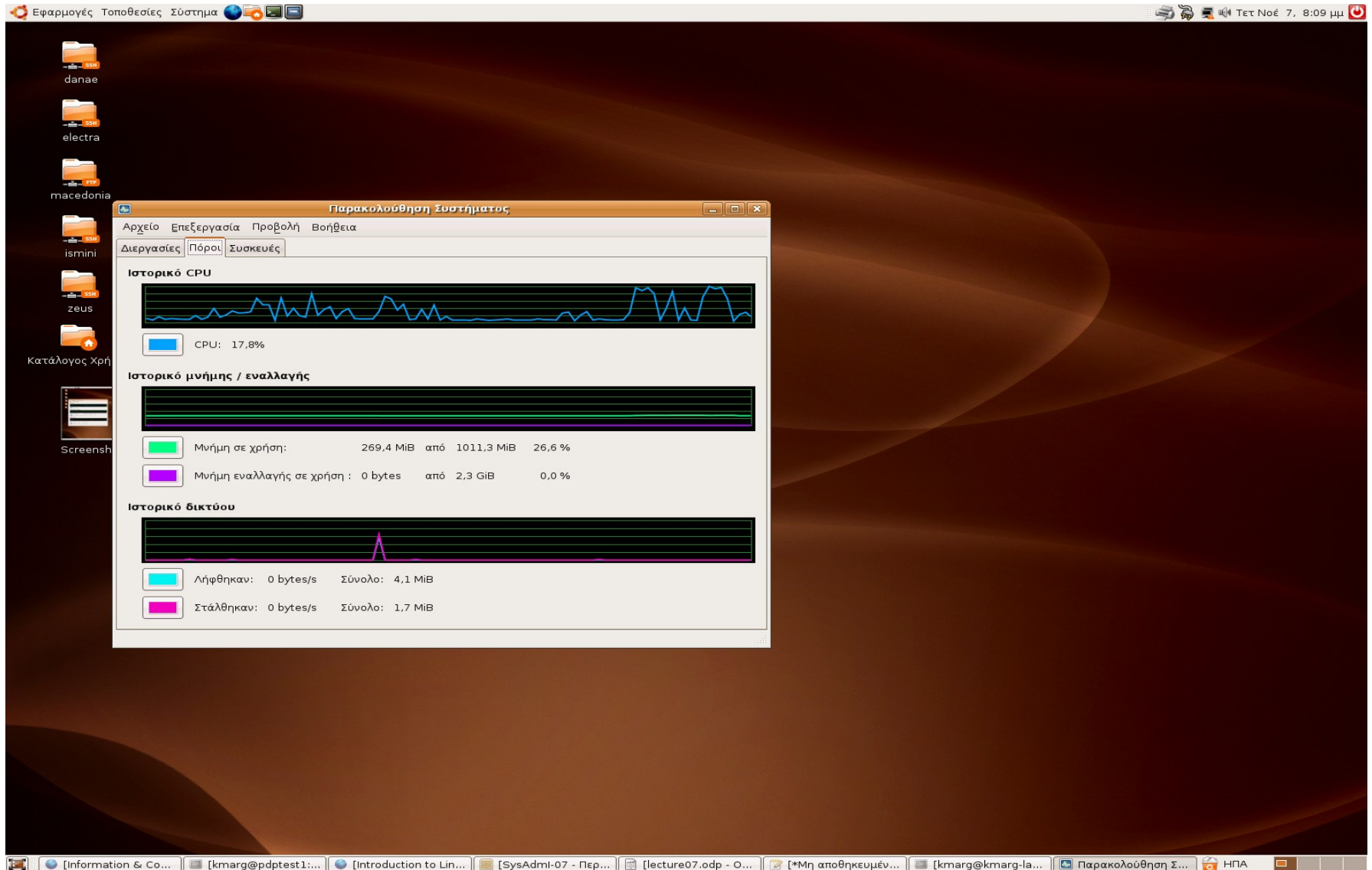
Όνομα Διεργασίας	Κατάσταση	Παραμένουσα Μνήμη	% CPU	Ωριοποίηση	ID	Ορίσματα
bash	Υπό αναστολή	3,4 MiB	0	0	12510	bash
bash	Υπό αναστολή	3,3 MiB	0	0	8031	bash
bonobo-activation-server	Υπό αναστολή	3,1 MiB	0	0	5022	/usr/lib/bonobo-activation/bonobo-activation-server --ac-activate --ior-output-fd=17
clock-applet	Υπό αναστολή	10,5 MiB	0	0	5111	/usr/lib/gnome-panel/clock-applet --oaf-activate-iid=OAFIID:GNOME_ClockApplet_Factory --
dbus-daemon	Υπό αναστολή	904,0 KiB	0	0	5015	dbus-daemon --fork --print-pid 8 --print-address 6 --session
dbus-launch	Υπό αναστολή	652,0 KiB	0	0	5014	/usr/bin/dbus-launch --exit-with-session x-session-manager
esd	Υπό αναστολή	480,0 KiB	0	0	5039	/usr/bin/esd -nobeeps
esd	Υπό αναστολή	4,4 MiB	0	0	5034	/usr/bin/esd -nobeeps
firefox-bin	Υπό αναστολή	91,0 MiB	0	0	6064	/usr/lib/firefox/firefox-bin -a firefox
gconfd-2	Υπό αναστολή	4,1 MiB	0	0	5017	/usr/lib/libgconf2-4/gconfd-2 5
gedit	Υπό αναστολή	20,7 MiB	0	0	9095	gedit
gksu	Υπό αναστολή	9,7 MiB	0	0	5196	gksu network-admin
gnome-cups-icon	Υπό αναστολή	10,4 MiB	0	0	5080	gnome-cups-icon --sm-client-id default3
gnome-keyboard-applet	Υπό αναστολή	14,8 MiB	0	0	5067	/usr/lib/gnome-applets/gnome-keyboard-applet --oaf-activate-iid=OAFIID:GNOME_Keyboar

Κατά

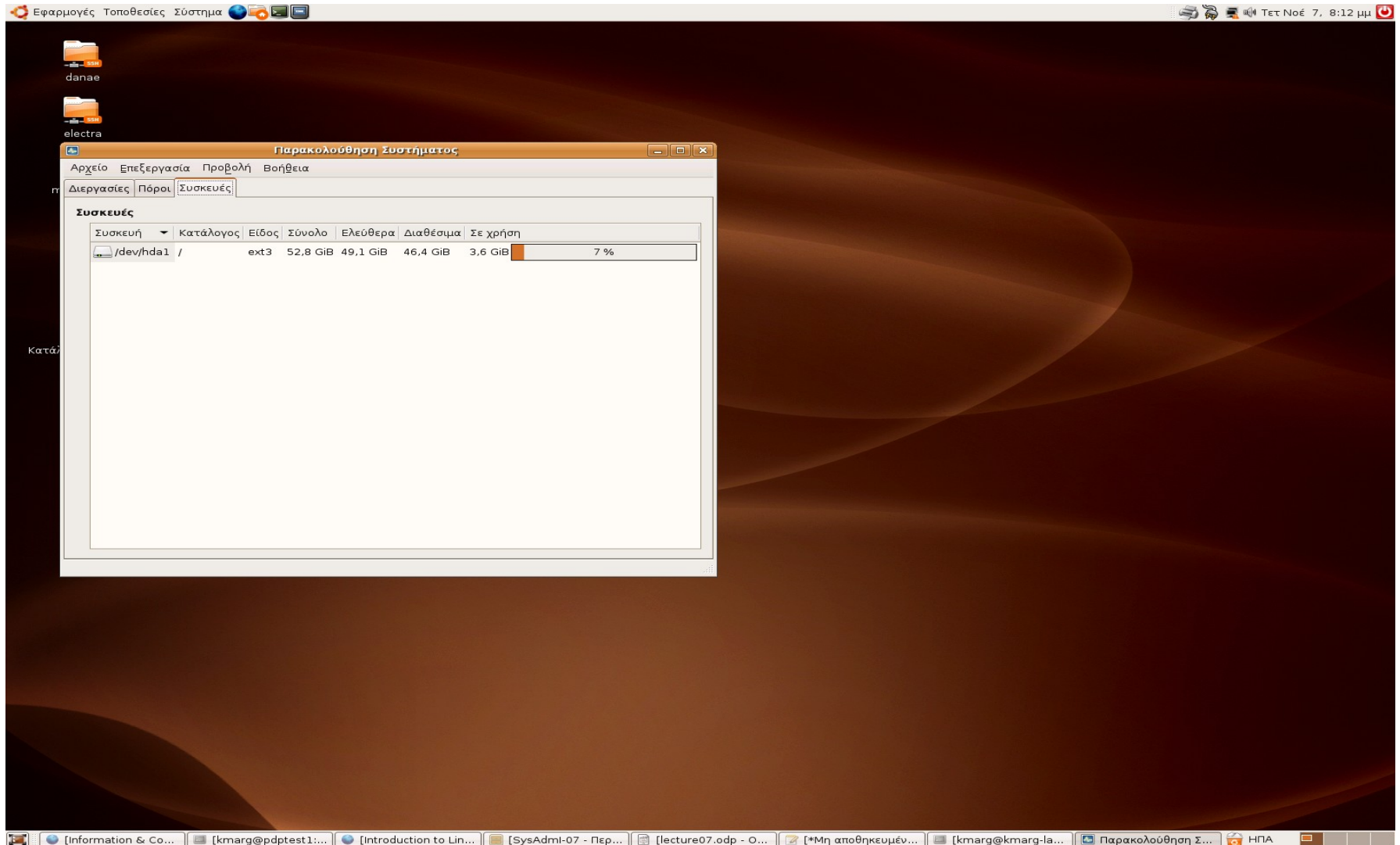
Τερματισμός Διεργασίας

Information & Co... [kmarg@pdp1:...] [Introduction to Lin... [SysAdmi-07 - Περ... [lecture07.odp - O... [*Μη αποθηκευμέν... [kmarg@kmarg-la... Παρακολούθηση Σ... ΗΠΑ

Gnome system monitor (2)



Gnome system monitor (3)



Χρονοπρογραμματισμός διεργασιών

sleep

Σταματά την εκτέλεση για N δευτερόλεπτα.

π.χ. (sleep 10000; gedit)&

batch

Εκτελεί διεργασία όταν το φορτίο του συστήματος είναι χαμηλότερο από ένα όριο.

π.χ. batch my_heavy_program

Εντολή At

at

Διαλογική εντολή που καθορίζουμε πότε θα εκτελεστεί μια ακολουθία εντολών (εργασία, task). Δέχεται αρκετά ευέλικτες μορφές καθορισμού του χρόνου όπως

at 10:30 tomorrow

at 4am + 5 days

at midnight 31 December 2007

atq

Δείχνει την ουρά εργασιών της at

atrm

Αφαιρεί εργασίες από την ουρά της at

Εντολή At : παράδειγμα

```
kmarg@pdptest1:~$ at 22:30
warning: commands will be executed using /bin/sh
at> touch some.file
at> <EOT> (CTRL-D)
job 1 at Wed Nov  7 22:30:00 2007
kmarg@pdptest1:~$ atq
1          Wed Nov  7 22:30:00 2007 a kmarg
kmarg@pdptest1:~$ atrm 1
Cannot find jobid 1
kmarg@pdptest1:~$ atq
kmarg@pdptest1:~$
```

Έλεγχος στο κατάλογο `/var/spool/cron/` πριν και μετά την εκτέλεση.

Όχι διαλογικές διεργασίες (παρασκηνίου ή batch).

Δημιουργία περιοδικών διεργασιών

Ο αυτοματισμός είναι το κλειδί της απόδοσης

Αντί να εκτελούμε εργασίες χειροκίνητα καθημερινά,
εβδομαδιαία ή μηνιαία μπορούμε να τα
χρονοδρομολογήσουμε με τι εντολές

cron

anacron

Περιλαμβάνονται εργασίες όπως: παρακολούθηση και
καταγραφή, καθαρισμός δίσκων και δημιουργία εφεδρικών
αντιγράφων, διανομή αρχείων κλπ.

cron

Ο δαίμονας cron εκτελεί εργασίες σε συγκεκριμένες χρονικές στιγμές

Εξετάζονται τα παρακάτω αρχεία crontab από τον cron για χρονοδρομολόγηση:

/etc/crontab

/etc/cron.d/*

/etc/cron.{hourly, daily, weekly, monthly}

/var/spool/cron/* (για τους χρήστες)

Ο cron αφυπνίζεται κάθε λεπτό και ελέγχει αν υπάρχει κάτι για να εκτελεστεί.

Ο cron είναι ευαίσθητος σε προβλήματα με την ώρα.

Ο anacron δουλεύει καθημερινά, καταγράφει πότε εκτελέστηκε τελευταία φορά η εργασία και αντισταθμίζει πιθανές προβλήματα ώρας.

Αρχεία crontab

Δομή:

λεπτά ώρα μέρα μήνας ημέρα_εβδομάδας
[όνομα_χρήστη] εντολή

Το όνομα χρήστη δεν αναφέρεται στα αρχεία /var/spool/cron
(χρησιμοποιείται το όνομα αρχείου)

Παραδείγματα εγγραφών crontab:

```
# run make at 2:30 each Monday morning
30 2 * * 1 (cd /home/joe4/project; make)
# remove files in /tmp not accessed in 3 days
20 1 * * * find /tmp -a atime +3 -exec rm -f {} ';'
# run system activity accounting tool every 10 minutes
*/10 * * * * root /usr/lib/sa/sa1 1 1
```

Διαχείριση crontab

Χρησιμοποιούμε την `crontab -e` για επεξεργασία

- Δημιουργεί ένα αντίγραφο

- Χρησιμοποιεί την μεταβλητή περιβάλλοντος `EDITOR`

- Δημιουργείται ή επανυποβάλλεται στο κατάλογο `/var/spool/cron` (υποκατάλογοι για κάθε χρήστη).

Το `crontab -l` εμφανίζει τα περιεχόμενα στην πρότυπη έξοδο (`stdout`), δηλαδή την οθόνη.

Τα αρχεία `/etc/cron.allow` , `/etc/cron.deny` μπορούν να ελέγξουν την πρόσβαση στις υπηρεσίες `cron`.

Εντολή crontab -e, -l : παράδειγμα

```
kmarg@pdptest1:~$ crontab -e
...
GNU nano 1.3.10      File: /tmp/crontab.TZ0I99/crontab      Modified
15 * 5 7 * touch some.file

^G Get Help      ^O WriteOut      ^R Read File      ^Y Prev Page      ^K Cut Text
^C Cur PoS
...

kmarg@pdptest1:~$ crontab -l
15 * 5 7 * touch some.file
```

Έλεγχος στο κατάλογο `/var/spool/crontabs/kmarg` πριν και μετά την εκτέλεση.

Όχι διαλογικές διεργασίες (παρασκηνίου ή batch).

Υπάρχει και το `gnome-schedule`.