

Εξέταση 102 – Μάθημα 8

107.2 Αυτοματοποίηση καθηκόντων
διαχείρισης συστήματος με
προγραμματισμό εργασιών



Το σύστημα *cron*

- Το σύστημα **cron** χρησιμοποιείται για τον προγραμματισμό εντολών περιοδικά, αναλόγως της ώρας, ημερομηνίας, ημέρας της εβδομάδας ή του μήνα
- Αποτελείται από το δαίμονα **crond** (**cron** σε Debian), την εντολή **crontab** και διάφορα αρχεία ρυθμίσεων



Τα αρχεία ρυθμίσεων

- **/etc/crontab** # είναι το βασικό αρχείο ρυθμίσεων όπου περιέχονται εντολές για επαναλαμβανόμενη εκτέλεση κάθε ώρα, μέρα, εβδομάδα μήνα. Συνήθως αποφεύγουμε να εκτελούμε άλλες εργασίες μέσω αυτού του αρχείου και χρησιμοποιούμε κάποιο από τα υπόλοιπα αρχεία ρυθμίσεων
- **/etc/cron.d/** # είναι ένας κατάλογος όπου μπορούμε να δημιουργήσουμε αρχεία με την ίδια μορφή όπως το **/etc/crontab**. Συστήνεται να υπάρχει ένα ξεχωριστό αρχείο για κάθε εργασία
- **/etc/cron.hourly/**, **/etc/cron.daily**, **/etc/cron.weekly**, **/etc/cron.monthly** # είναι κατάλογοι όπου φιλοξενούνται διάφορα αρχεία σεναρίων (scripts) τα οποία εκτελούνται κάθε ώρα, κάθε μέρα, κάθε εβδομάδα ή κάθε μήνα αντίστοιχα. Οι ακριβείς ώρες εκτέλεσης των σεναρίων καθορίζονται στο **/etc/crontab**

Μορφή των αρχείων `/etc/crontab` και `/etc/cron.d/*`

- `# minute hour dayofmonth month dayofweek user command`

`45 7 3 * * root myscript`

minute: λεπτά της ώρας. Παίρνει τιμές από 0 – 59

hour: ώρα της ημέρας. Παίρνει τιμές από 0 – 23

dayofmonth: ημέρα του μήνα. Παίρνει τιμές από 1 – 31

month: μήνας. Παίρνει τιμές από 1-12 ή jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec

dayofweek: ημέρα της εβδομάδας. Παίρνει τιμές από 0 – 7 ή sun - sat όπου 0 ή 7 είναι η Κυριακή (sun), 1 η Δευτέρα (mon), 2 η Τρίτη (tue), 3 η Τετάρτη (wed), 4 η Πέμπτη (thu), 5 η Παρασκευή (fri) και 6 το Σάββατο (sat)

- Στο παράδειγμα πιο πάνω η εντολή `myscript` θα εκτελείται κάθε τρεις του μήνα στις 7:45πμ από τον χρήστη `root`

Παράδειγμα των περιεχομένων του `/etc/crontab`

- Στο αρχείο `/etc/crontab` καθώς και στα αρχεία `/etc/cron.d/*` είναι καλή ιδέα να δηλώνουμε το κέλυφος (`$SHELL`) και την τιμή της `$PATH`. Επίσης μπορεί να καθορίσουμε ποιος θα ειδοποιείται με `$MAILTO`

```
# cat /etc/crontab
```

```
# /etc/crontab: system-wide crontab
```

```
SHELL=/bin/sh
```

```
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

```
MAILTO=root
```

```
# m h dom mon dow user command
```

```
17 * * * * root run-parts /etc/cron.hourly
```

```
25 6 * * * root run-parts /etc/cron.daily
```

```
47 6 * * 7 root run-parts /etc/cron.weekly
```

```
52 6 1 * * root run-parts /etc/cron.monthly
```




Άλλα παραδείγματα

```
# month      hour dayofmonth  month dayofweek user  command
# εκτέλεση κάθε τέταρτο των ωρών 6:00 πμ και μμ κάθε 5 μέρες
0,15,30,45  6,18 5,10,15,20,25,30 *      *      root  myscript
# πανομοιότυπη ρύθμιση με την πιο πάνω
*/15        6,18 5-30/5 *      *      apache apscript
# Εκτέλεση backup κάθε Παρασκευή
0           4     *      *      5      root  backup.sh
# Προσοχή! Αυτό δεν σημαίνει κάθε τελευταία Παρασκευή του
# μήνα αλλά κάθε μέρα μεταξύ 24-31 και κάθε Παρασκευή του μήνα!
0           4     24-31 *      5      root  mbackup.sh
# Εκτέλεση κάθε Παρασκευή του Αυγούστου στις 6:30 το πρωί
30          6     *      aug    fri    root  reminder.sh
# Πανομοιότυπη ρύθμιση με την πιο πάνω
30          6     *      8      5      root  reminder.sh
# Εκτέλεση εντολής κάθε λεπτό!
*           *     *      *      *      root  script.sh
```



Προγραμματισμός εντολών για όλους τους χρήστες με *crontab*

- Η εντολή **crontab** δεν πρέπει να συγχέεται με το αρχείο ρυθμίσεων **/etc/crontab**.
 - Χρησιμοποιείται από του χρήστες για τον προγραμματισμό εργασιών του κάθε χρήστη ξεχωριστά
 - Χρησιμοποιεί τον προκαθορισμένο κειμενογράφο του συστήματος, συνήθως τον vi, (nano σε Debian) διαφορετικά χρησιμοποιεί τον κειμενογράφο που καθορίζεται από την μεταβλητή **\$EDITOR**
 - Η μορφή του αρχείου είναι παρόμοια με την μορφή του **/etc/crontab** με την διαφορά ότι δεν υπάρχει στήλη για τον χρήστη αφού καθορίζεται από το ποιος εκτελεί την εντολή
 - Τα επιμέρους αρχεία crontab για κάθε χρήστη αποθηκεύονται κάτω από τον κατάλογο **/var/spool/cron/***
- 

Προγραμματισμός εντολών για όλους τους χρήστες με *crontab*

- `$ crontab -e` # επεξεργασία του προσωπικού αρχείου crontab κάθε χρήστη
- | # | min | hour | dom | mon | dow | command |
|---|-----|------|-----|-----|-----|--|
| | 30 | 8 | 3 | 11 | * | reminder_wifes_birthday.sh |
| | 0 | 18 | * | * | 5 | echo "yooipi\!" mail -s "It's Friday" user@example.com |

Επιλογές:

- `-l` # προβολή όλων των προγραμματισμένων εργασιών του τρέχοντος χρήστη
- `-r` # διαγραφή του προσωπικού αρχείου crontab του τρέχοντος χρήστη με όλες της εργασίες που περιέχει
- `-u user` # μόνο ο χρήστης root έχει το δικαίωμα να επεξεργάζεται, προβάλλει και διαγράφει τις εργασίες cron άλλων χρηστών

Έλεγχος δικαιωμάτων χρήσης cron με `/etc/cron.allow` και `/etc/cron.deny`

- Τα αρχεία `/etc/cron.allow` και `/etc/cron.deny` ελέγχουν ποιοι χρήστες μπορούν να χρησιμοποιούν την υπηρεσία `cron`
- Αν δεν υπάρχει κανένα αρχείο από τα δύο τότε όλοι οι χρήστες έχουν δικαίωμα χρήσης του `cron` με την εντολή `crontab`
- Αν υπάρχει το αρχείο `cron.deny` τότε όσοι υπάρχουν σε αυτό, δεν θα έχουν δικαίωμα χρήσης του `cron`
- Αν υπάρχει το αρχείο `cron.allow` τότε μόνο όσοι υπάρχουν σε αυτό, θα έχουν δικαίωμα χρήσης του `cron`
- Αν υπάρχει και τα δύο τότε το `cron.deny` θα αγνοηθεί εντελώς και θα εφαρμοστούν οι κανόνες του `cron.allow`



Το σύστημα *at*

- Το σύστημα *at* αποτελείται από τον δαίμονα *atd* και τις εντολές *at*, *atq*, *atrm* και *batch*.
- Το σύστημα αυτό επιτρέπει την εκτέλεση εντολών στο μέλλον μία μόνο φορά (και όχι περιοδικά όπως το *cron*)



Προγραμματισμός εντολών με *at*

- Η εντολή **at** χρησιμοποιείται για τον προγραμματισμό μελλοντικών εντολών
- **# echo "shutdown -r now" | at 0400** # επανεκκίνηση στις 04:00 το πρωί
- **# echo 'mail -s Logs root < /var/log/messages' | at 00:00 feb 28**
αποστολή των αρχείων καταγραφής σε email τα μεσάνυκτα της 28ης Φεβρουαρίου
- **# at 12:00 dec 21 2012** # Κλείσιμο του συστήματος στις 21/12/2012
shutdown -h now "This is the end"
^D # (πατήστε Ctrl-D για να τερματίσετε το κείμενο)
- **# at 8pm + 3 days <<EOF** # Εκτέλεση στις 8μμ σε τρεις μέρες
backup.sh
EOF

Προγραμματισμός εντολών με *at*

- `# at -f commands.txt tomorrow` # εκτέλεση εντολών από την λίστα `commands.txt` αύριο τέτοια ώρα
- `# at -f commands.txt now + 2 days` # εκτέλεση εντολών από την λίστα `commands.txt` σε δύο μέρες τέτοια ώρα

Επιλογές:

- `-f file` # εκτέλεση εντολών από αρχείο `file`
- `-l` # προβολή λίστας εργασιών. Ο χρήστης `root` θα δει όλες τις εργασίες από όλους τους χρήστες
- `-d job1 job2 job3 ...` # διαγραφή εργασιών



Προβολή εργασιών με *atq*

- Η εντολή **atq** είναι ισοδύναμη με **at -l** και χρησιμοποιείται για την προβολή των μελλοντικών εργασιών
- **# atq # = at -l**. Προβολή λίστας εργασιών του συστήματος **at**. Οι κανονικοί χρήστες βλέπουν μόνο τις δικές τους εργασίες ενώ ο χρήστης **root** βλέπει τις εργασίες όλων των χρηστών



Διαγραφή εργασιών με *atrm*

- Η εντολή **atrm** είναι ισοδύναμη με **at -d** και χρησιμοποιείται για την διαγραφή μελλοντικών εργασιών
- **# atrm 3 5 6 # = at -d**. Διαγραφή εργασιών 3, 5 και 6. Οι κανονικοί χρήστες μπορούν να διαγράψουν μόνο τις δικές τους εργασίες ενώ ο root μπορεί να διαγράψει τις εργασίες όλων




Έλεγχος δικαιωμάτων χρήσης `at` με `/etc/at.allow` και `/etc/at.deny`

- Τα αρχεία `/etc/at.allow` και `/etc/at.deny` ελέγχουν ποιοι χρήστες μπορούν να χρησιμοποιούν την υπηρεσία `atd`
- Αν δεν υπάρχει κανένα αρχείο από τα δύο τότε κανένας χρήστης δεν έχει το δικαίωμα χρήσης του `at`
- Αν υπάρχει το αρχείο `at.deny` τότε όσοι υπάρχουν σε αυτό, δεν θα έχουν δικαίωμα χρήσης του `at`
- Αν υπάρχει το αρχείο `at.allow` τότε μόνο όσοι υπάρχουν σε αυτό, θα έχουν δικαίωμα χρήσης του `at`
- Αν υπάρχει και τα δύο τότε το `at.deny` θα αγνοηθεί εντελώς και θα εφαρμοστούν οι κανόνες του `at.allow`



Εργαστήριο 8

Ξεκινήστε και τις δύο εικονικές μηχανές και συνδεθείτε σαν "root"

- # ps aux | grep cron
 - # ps aux | grep atd
 - # view /etc/crontab
 - # ls -la /etc/cron.d
 - # cd /etc/cron.d
 - # for FILE in *
do ls -l \$FILE
cat \$FILE
echo ""
done
 - # ls -la /etc/cron.*
 - # vi /etc/cron.d/test
*/2 * * * * root echo date is `date` > /dev/tty1
:wq
- Σε άλλο τερματικό:
- # tail -f /var/log/cron # (για RedHat)
 - # tail -f /var/log/syslog # (για Debian)
 - # ls -la `which crontab`
 - # vi /etc/cron/deny
user
:wq
 - # su - user
 - \$ export EDITOR=vi
 - \$ crontab -e
- 

Εργαστήριο 8

- `$ exit`
 - `# sed -i '/^user$/d' /etc/cron.deny`
 - `# cat /etc/cron.deny`
 - `# su – user`
 - `$ export EDITOR=vi`
 - `$ crontab -e`
 - `*/5 * * * * touch /tmp/file.tmp`
 - `:wq`
 - `$ ls -la /tmp`
 - `$ exit`
 - `# ls -la /var/spool/cron`
 - `# echo 'echo Testing at > /dev/tty1' | at \`
 - `now + 5 min`
 - `# at 14:45 jan 12`
 - `true`
 - `^D`
 - `# at 0000 dec 21 2015 << EOF`
 - `true`
 - `EOF`
 - `# su – user`
 - `$ at tomorrow`
 - `echo Hello`
 - `^D`
 - `$ atq`
 - `# exit`
 - `# atq`
 - `# find /var/spool/cron`
- 

Εργαστήριο 8

- `# find /var/spool/at # σε RedHat`
- `# atrm 1 2 3 4`
- `# atq`
- `# find /var/spool/cron`
- `# find /var/spool/at # σε RedHat`
- `# su – user`
- `$ crontab -r`
- `$ exit`
- `# find /var/spool/cron`

