

# Λειτουργικά Συστήματα Ι

## Ασκήσεις Πράξης

Τμήμα Μηχανικών Πληροφορικής & Υπολογιστών  
Σχολή Μηχανικών



# Χρονοδρομολόγηση Εργασιών

## Αρχειοθέτηση, Συμπίεση

# cron

Η **cron** (από το *chronos*) είναι μια υπηρεσία συστήματος που τρέχει στο παρασκήνιο και εκτελεί επιλεγμένες εργασίες σε προκαθορισμένες χρονικές στιγμές. Οι εργασίες είναι καταχωρημένες στα ειδικά αρχεία **crontab**. Πιο συγκεκριμένα, η υπηρεσία ψάχνει για αρχεία **crontab** στον κατάλογο **/var/spool/cron/crontab**, όπου καταχωρούνται τα αρχεία των χρηστών καθώς και το αρχείο **/etc/crontab** που αποτελεί το αρχείο διαμόρφωσης του συστήματος. Σε πολλές διανομές τα περιεχόμενα του αρχείου **/etc/crontab** παραπέμπουν σε εντολές ή σενάρια εντολών που βρίσκονται στους καταλόγους **/etc/cron.{hourly,daily,weekly,monthly}/**.

# cron

Ένα αρχείο **crontab** (**cron table**) είναι ένα αρχείο κειμένου που περιέχει γραμμές με πεδία που προσδιορίζουν την επιθυμητή εργασία και την αντίστοιχη χρονική στιγμή εκτέλεσης. Κάθε γραμμή ενός αρχείου crontab έχει μορφή:

**minute hour day month dayofweek command**

όπου

- **minute** – ένας ακέραιος από 0 έως 59
- **hour** – ένας ακέραιος από 0 έως 23
- **day** – ένας ακέραιος από 1 έως 31
- **month** – ένας ακέραιος από 1 έως 12 ή συντομογραφία ενός μήνα (π.χ. jan, feb)

# cron

- **dayofweek** – ένας ακέραιος από 0 έως 6, όπου το 0 αντιστοιχεί στην Κυριακή ή συντομογραφία μιας ημέρας (π.χ. sun, mon)
- **command** – η εντολή που είναι προς εκτέλεση

Για όλα τα παραπάνω πεδία μπορεί να χρησιμοποιηθεί ως τιμή ο χαρακτήρας αστερίσκος “\*” που αντιστοιχεί σε “όλες” τις τιμές. Η παύλα “-” μπορεί να χρησιμοποιηθεί για τον καθορισμό ενός εύρους τιμών. Μία λίστα τιμών ορίζεται από τιμές χωρισμένες με κόμμα. Η δεξιά κάθετος “/” χρησιμεύει για τον καθορισμό βήματος.

# /etc/crontab

```
Terminal
$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
$
```

Το αρχείο crontab του συστήματος περιέχει επιπλέον το πεδίο **user** που προσδιορίζει τον χρήστη που εκτελεί την εργασία.

# crontab

Οι χρήστες μπορούν να δημιουργήσουν το δικό τους αρχείο crontab με την ομώνυμη εντολή και την επιλογή **-e**:

**\$ crontab -e**

Εκτελείται ο προκαθορισμένος συντάκτης κειμένου, όπου καταχωρούμε την cron εντολή σύμφωνα με τη μορφή που προαναφέραμε. Αν για παράδειγμα θέλουμε να καταγράφουμε κάθε 5 λεπτά της ώρας σε ένα αρχείο την τρέχουσα ημερομηνία και τους συνδεδεμένους χρήστες καταχωρούμε στο αρχείο crontab τη γραμμή:

```
*/5 * * * * (date ; who | cut -d" " -f1) >> testfile
```

# crontab

Στην περίπτωση που αντί για μία εντολή καταχωρήσουμε ένα σενάριο εντολών, δεν πρέπει να παραλείψουμε να εκχωρήσουμε το δικαίωμα εκτέλεσης σ' αυτό. Η επόμενη cron εντολή εκτελεί ένα σενάριο εντολών με το όνομα backup.sh κάθε 1<sup>η</sup> και 15<sup>η</sup> ημέρα του μήνα και κάθε Παρασκευή στις 22:30:

```
30 22 1,15 * Fri ./backup.sh 2> errors.log
```

Η προηγούμενη εντολή θα εκτελείται κάθε 1<sup>η</sup> και 15<sup>η</sup> του μήνα αλλά και κάθε Παρασκευή επειδή έχουμε ορίσει συγκεκριμένη τιμή και στα δύο πεδία. Αν δεν έχουμε δηλώσει μονοπάτια αναζήτησης με τη μεταβλητή περιβάλλοντος PATH θα πρέπει να χρησιμοποιούμε απόλυτες διαδρομές για τα εκτελέσιμα αρχεία.



# crontab

- Αν δεν γίνει ανακατεύθυνση εξόδου της εντολής σε αρχείο, η έξοδος θα αποσταλεί με ηλεκτρονικό ταχυδρομείο. Άρα θα πρέπει να έχουμε εγκαταστήσει και ρυθμίσει ένα πρόγραμμα ηλεκτρονικής αλληλογραφίας όπως είναι το **sendmail**.
- Αν στο αρχείο crontab έχει δηλωθεί η μεταβλητή MAILTO και έχει ως τιμή διευθύνσεις ηλεκτρονικού ταχυδρομείου, το μήνυμα θα αποσταλεί στους αντίστοιχους χρήστες. Αν έχει δηλωθεί η μεταβλητή αλλά δεν έχει περιεχόμενο (MAILTO="") δε θα αποσταλεί μήνυμα. Σε διαφορετική περίπτωση η αποστολή του μηνύματος θα γίνει στον ιδιοκτήτη του αρχείου crontab.

# crontab

Ο διαχειριστής του συστήματος μπορεί να εμφανίσει το αρχείο crontab ενός άλλου χρήστη με συνδυασμό των επιλογών **-u** και **-l**:

```
# crontab -u nemo -l    # display another users crontab file
```

...

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

```
30 22 1,15 * Fri /home/nemo/backup.sh 2> errors.log
```

```
00 15 * * * echo "Would you like to join me for a cup of  
coffee?" | mail -s "Hi Alice" alice@uniwa.gr
```

# crontab

Η επεξεργασία του αρχείου από τον διαχειριστή μπορεί να γίνει με συνδυασμό των επιλογών **-u** και **-e**:

**# crontab -u nemo -e**      # edit another user's crontab file

Τέλος, η διαγραφή ενός αρχείου crontab γίνεται με την επιλογή **-r** ή με τη **-i** για να ζητηθεί επιβεβαίωση διαγραφής.

# at

Η εντολή **at** χρησιμοποιείται για την εκτέλεση μιας εντολής σε μια συγκεκριμένη χρονική στιγμή (όχι επαναλαμβανόμενα). Η εντολή που θα εκτελεσθεί μπορεί να είναι ένα απλό μήνυμα υπενθύμισης ή κάτι πιο σύνθετο, για παράδειγμα ένα σενάριο εντολών το οποίο κάνει αντίγραφα ασφάλειας και στη συνέχεια τερματίζει τη λειτουργία του συστήματος.

Η σύνταξή της είναι:

**at** [OPTION]... TIME

όπου TIME η επιθυμητή χρονική στιγμή δρομολόγησης της εργασίας.

# at

Οι πιο σημαντικές επιλογές της εντολής at είναι:

- **-m** αποστολή μηνύματος ηλεκτρονικής αλληλογραφίας στον χρήστη μετά την ολοκλήρωση της προγραμματισμένης εργασίας εφόσον υπάρχει έξοδος στην οθόνη
- **-M** αποτροπή αποστολής email
- **-f FILE** εκτέλεση εργασιών από το αρχείο FILE
- **-l** εμφάνιση λίστας εργασιών
- **-r, -d** διαγραφή εργασίας
- **-t time** ορισμός χρόνου εκτέλεσης στη μορφή `[[CC]YY]MMDDhhmm[.ss]`

# at

Η προκαθορισμένη είσοδος της εντολής `at` είναι το πληκτρολόγιο. Αν δεν προσδιορίσουμε κάποιο αρχείο εντολών με την επιλογή `-f` ή με ανακατεύθυνση εισόδου εμφανίζεται το σήμα προτροπής **at>** της εντολής, όπου σε κάθε γραμμή πληκτρολογούμε από μία εντολή και πατάμε το πλήκτρο [Enter].

Τερματίζουμε την είσοδο κατά τα γνωστά με τον συνδυασμό πλήκτρων [Ctrl]+[D].

---

at

Το πρώτο μας παράδειγμα εκτελεί την εντολή echo στις 13:00 την ερχόμενη Κυριακή:

```
$ at 13:00 Sun                                # next Sunday at 13:00
at> echo 'Dinner with Alice'
at> <EOT>                                     # Ctrl + D
job 1 at Sun Dec  2 13:00:00 2018
```

Η at εμφανίζει τον αριθμό εργασίας (job control) και τη χρονική στιγμή για την οποία δρομολογήθηκε η εργασία.

---

at

Η επόμενη εντολή εκτελεί τις εντολές του αρχείου backup.sh στις 31 Δεκεμβρίου και ώρα 2345:

```
$ at -f backup.sh 2345 Dec 31                                # 23:40, 31 Dec  
job 2 at Mon Dec 31 23:45:00 2018
```

Η ώρα δρομολόγησης καθορίζεται από ένα ή δύο ψηφία (0-23) και τα λεπτά της ώρας από δύο ψηφία (0-59) που χωρίζονται (προαιρετικά) με άνω και κάτω τελεία. Αν χρησιμοποιήσουμε ένα ψηφίο για την ώρα πρέπει να προσδιορίσουμε με **am** ή **pm** αν πρόκειται για προ ή μετά μεσημβρίας.



---

at

Έχουμε επίσης τη δυνατότητα να χρησιμοποιήσουμε τους προσδιορισμούς **now**, **noon**, **teatime** και **midnight**. Η δε ημερομηνία δρομολόγησης της εργασίας αποτελείται συνήθως από την πλήρη ημερομηνία στη μορφή **month day** (και προαιρετικά) **year**, π.χ. July 20 2024. Μπορούμε επίσης να χρησιμοποιήσουμε ως προσδιορισμούς χρόνου τους όρους **today** και **tomorrow**. Εναλλακτικά μπορούμε να ορίσουμε κάποια χρονική στιγμή σε σχέση την τρέχουσα ώρα. Αυτό γίνεται με τη μορφή **+ count time-units**, όπου count ένας ακέραιος αριθμός και time-units η μονάδα χρόνου που ορίζεται από τους όρους **minutes**, **hours**, **days**, **weeks**, **months** και **years**.

---

---

at

**\$ at now +3 days < commands**

# after 3 days

job 3 at Sun Dec 2 21:12:00 2018

Οι εντολές του αρχείου commands θα εκτελεσθούν μετά από 3 ημέρες.

**\$ at -f backup.sh tomorrow +3 days**

# after 4 days

job 4 at Mon Dec 3 21:33:00 2018

Οι εντολές του αρχείου backup.sh θα εκτελεστούν 3 ημέρες από αύριο.

---

at

**\$ at -f backup.sh tuesday +2 hours**

job 5 at Tue Dec 4 23:33:00 2018

Οι εντολές του αρχείου backup.sh θα εκτελεστούν την  
ερχόμενη Τρίτη και 2 ώρες μετά.

**\$ at -f backup.sh 9am February 28**

# February 28, 9am

job 6 at Thu Feb 28 09:00:00 2019

Οι εντολές του αρχείου backup.sh θα εκτελεστούν στις 28  
Φεβρουαρίου και ώρα 9 π.μ.

## at

Σε περίπτωση που δεν δώσουμε σωστή ώρα δρομολόγησης η εντολή `at` θα εμφανίσει το μήνυμα λάθους **Garbled time**.

Οι λίστα των δρομολογημένων εργασιών εμφανίζεται με την επιλογή `-l`:

```
$ at -l # an alias for atq
```

```
1      Mon Dec 2 13:00:00 2018 a nemo
```

```
...
```

```
6      Thu Feb 28 09:00:00 2018 a nemo
```

Η αφαίρεση μιας εργασίας από τη λίστα γίνεται με την επιλογή `-r` και τον αριθμό εργασίας:

```
$ at -r 2
```

# tar

Η εντολή **tar** (**t**ape **a**rchive) κάνει αρχειοθέτηση αρχείων, τοποθετεί δηλαδή πολλά αρχεία μαζί σε ένα ενιαίο αρχείο για την εύκολη μεταφορά και αποθήκευσή τους αλλά και το αντίστροφο, την εξαγωγή δηλαδή των αρχείων από το αρχειοθετήριο (tarball). Το όνομα της εντολής προέρχεται από την πρώιμη εποχή του UNIX στην οποία η δημιουργία αντιγράφων ασφαλείας γινόταν σε μαγνητικές ταινίες. Όλες οι νεότερες εκδόσεις της εντολής υποστηρίζουν επιπλέον τη συμπίεση και αποσυμπίεση των αρχείων.

# tar

Η σύνταξη της εντολής είναι:

**tar** [OPERATION[OPTION]]... TARFILE [FILE]...

όπου OPERATION καθορίζει τη βασική λειτουργία της εντολής. Οι λειτουργίες αυτές είναι:

- A, --concatenate** προσθήκη ενός αρχειοθετηρίου στο τέλος ενός άλλου.
- c, --create** δημιουργία ενός νέου αρχειοθετηρίου
- d, --compare** σύγκριση αρχειοθετηρίου και συστήματος αρχείων
- delete** διαγραφή αρχείου από το αρχειοθετήριο

# tar

- r, --append** προσθήκη αρχείων στο τέλος του αρχειοθετηρίου
- t, --list** εμφάνιση περιεχομένων αρχειοθετηρίου
- u, --update** ενημέρωση αρχειοθετηρίου με τη προσθήκη αρχείων που είναι νεότερα από τα αντίστοιχα
- x, --extract** εξαγωγή αρχείων από το αρχειοθετήριο
- , --help** εμφάνιση σύνοψης επιλογών

# tar

Οι σημαντικότερες επιλογές της εντολής tar είναι:

**-a, --auto-compress**

αυτόματη αναγνώριση προγράμματος συμπίεσης από την επέκταση του αρχειοθετηρίου

**-C, --directory=DIR**

εξαγωγή των αρχείων στον κατάλογο DIR και όχι στον τρέχοντα κατάλογο (σε συνδυασμό με τη λειτουργία -x).

**-f, --file=ARCHIVE**

χρήση του αρχείου ARCHIVE ως αρχειοθετήριο.

**-H, --format=FORMAT**

χρήση αρχειοθετηρίου της μορφής FORMAT με τιμές gnu, oldgnu, posix, ustar και v7.



# tar

**-I, --use-compress-program=PROG**

χρήση του PROG ως πρόγραμμα συμπίεσης.

**-O, --to-stdout**

εξαγωγή αρχείων στην προκαθορισμένη έξοδο.

**-v, --verbose**

εμφάνιση των ονομάτων των αρχείων που επεξεργάζονται.

**-z, --gzip, --gunzip**

συμπίεση με χρήση του gzip.

# tar

```
$ tar -cf myscripts.tar ./scripts      # create new archive
```

```
$ ls -l myscripts.tar
```

```
-rw-r--r-- 1 nemo nemo 71680 Dec 15 16:12 myscripts.tar
```

Χωρίς την επιλογή **-v** η εντολή tar δεν εμφανίζει τα αρχεία που αρχειοθετεί. Για να εμφανίσουμε τη λίστα των αρχείων χρησιμοποιούμε την επιλογή **-t**:

```
$ tar -tf myscripts.tar                # list contents of archive
```

```
./scripts/
```

```
./scripts/myls.sh
```

```
...
```

# tar

Αν θέλουμε να κάνουμε εξαγωγή των αρχείων στην προκαθορισμένη έξοδο (συνήθως για περαιτέρω επεξεργασία σε μία σωλήνωση) χρησιμοποιούμε τον συνδυασμό επιλογών **-x** και **-O**, όπως δείχνει η εντολή που ακολουθεί:

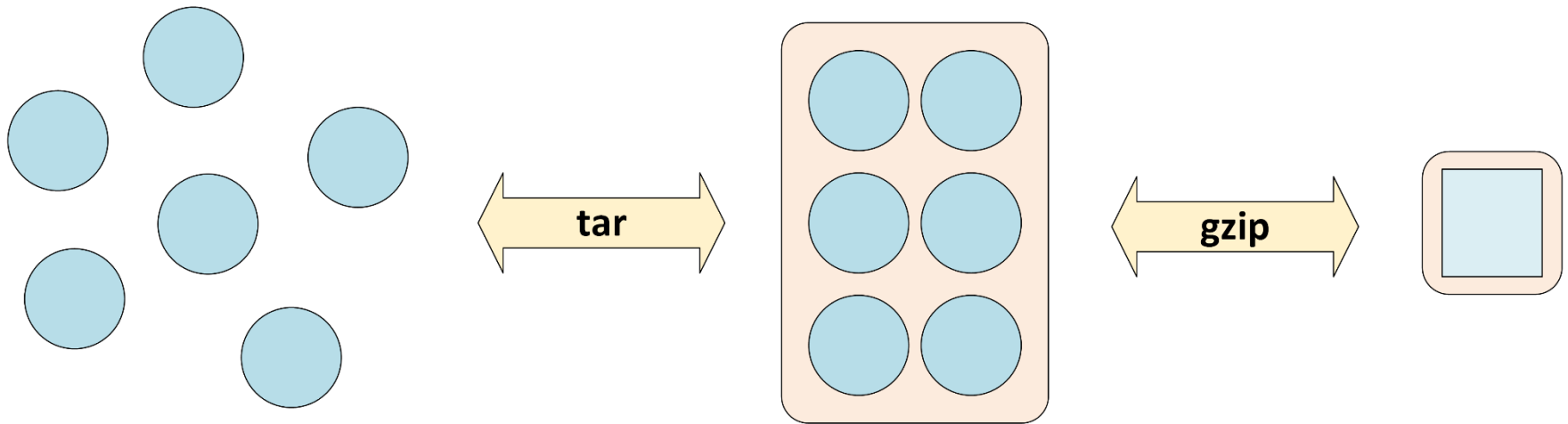
```
$ tar -xOf myscripts.tar          # extract archive to stdout  
#!/bin/bash
```

...

Η εξαγωγή σε κάποιο άλλον κατάλογο και όχι στον τρέχοντα γίνεται με την επιλογή **-C**. Είναι προφανές πως ο κατάλογος πρέπει να υπάρχει.

```
$ tar -xf myscripts.tar -C /tmp/bestscripts    # otherwere
```

# tar - gzip



Η αρχειοθέτηση χρησιμοποιείται συνήθως σε συνδυασμό με κάποια μέθοδο συμπίεσης, όπως η **gzip**, για τη δημιουργία συμπιεσμένου αρχειοθετηρίου.

# tar - gzip

Ο αλγόριθμος **Gzip** είναι ο πιο δημοφιλής αλγόριθμος συμπίεσης αρχείων. Τα αρχειοθετήρια που είναι συμπιεσμένα με εφαρμογή gzip έχουν επέκταση **.tar.gz** ή **.tgz**.

```
$ tar -czf myscripts.tar.gz ./scripts # gzip
```

Αν συγκρίνουμε τώρα τα δύο αρχειοθετήρια (συμπιεσμένο και μη) θα δούμε πως υπάρχει μεγάλη διαφορά στο μέγεθός τους, κάτι που είναι απολύτως φυσιολογικό καθώς πρόκειται για σενάρια φλοιού δηλαδή αρχεία κειμένου.

```
$ ls -l myscripts.tar*
```

```
-rw-r--r-- 1 nemo nemo 71680 Dec 15 16:12 myscripts.tar  
-rw-r--r-- 1 nemo nemo 5685 Dec 16 21:59 myscripts.tar.gz
```

# tar - bzip2

Αν χρησιμοποιήσουμε τον αλγόριθμο **Bzip2** για συμπίεση θα κάνουμε χρήση της επιλογής **-j** με την εντολή **tar**. Τα αρχειοθετήρια αυτά έχουν επεκτάσεις **.tar.bz2**, **.tar.bz** ή **.tbz**.

```
$ tar -cjf mp3.tar.bz ./Music/mp3/RHCP ./Music/mp3/ACDC
```

Η εξαγωγή των αρχείων και η αποσυμπίεσή τους σε κάποιον άλλον κατάλογο μπορεί να γίνει με την εντολή:

```
$ tar -xvjf mp3.tar.bz -C /tmp/songs # unzip and extract
```

```
./Music/mp3/RHCP/
```

```
./Music/mp3/RHCP/08 Porcelain.mp3
```

```
...
```

```
./Music/mp3/ACDC/04. Given The Dog A Bone.mp3
```

# bzip2

Η εντολή **bzip2** βασίζεται στον αλγόριθμο ταξινόμησης μπλοκ κειμένου **Burrows-Wheeler** και συνήθως παρέχει καλύτερους λόγους συμπίεσης από την εντολή `gzip`. Είναι διαθέσιμη στις περισσότερες διανομές GNU/Linux, και αν όχι, μπορούμε εύκολα να την εγκαταστήσουμε με τη βοήθεια του αντίστοιχου διαχειριστή πακέτων:

```
$ sudo apt install bzip2
```

```
# Debian/Ubuntu
```

Η σύνταξη της εντολής είναι:

```
bzip2 [OPTION]... [FILE]...
```

όπου `FILE` είναι το προς συμπίεση ή αποσυμπίεση αρχείο.

# bzip2

Χωρίς κάποια επιλογή, η εντολή συμπιέζει το πρωτότυπο αρχείο και το αποθηκεύει προσαρτώντας μετά το όνομά του την επέκταση **.bz2**.

- **-d, --decompress** αποσυμπίεση
- **-z, --compress** συμπίεση
- **-f, --force** εγγραφή πάνω από προϋπάρχοντα αρχεία
- **-k, --keep** δεν γίνεται διαγραφή του αρχείου εισόδου κατά τη συμπίεση ή την αποσυμπίεση.
- **-1 έως -9** καθορίζει το μέγεθος του μπλοκ που χρησιμοποιείται κατά τη συμπίεση για μεγάλα αρχεία (100K έως 900K).



# bzip2

```
$ ls -l                                # check file size
-rw-r--r-- 1 nemo nemo 138432 Dec 17 20:25 errors.log
$ bzip2 errors.log                     # compress file
$ ls -l                                # check compressed file size
-rw-r--r-- 1 nemo nemo 893 Dec 17 20:25 errors.log.bz2
$ bzip2 -d errors.log.bz2             # decompress
$ ls -l
-rw-r--r-- 1 nemo nemo 138432 Dec 17 20:25 errors.log
```

# bzip2

Αν θέλουμε να κρατήσουμε το αρχικό μας αρχείο εισόδου και να δημιουργήσουμε ένα συμπιεσμένο αντίγραφο θα πρέπει να χρησιμοποιήσουμε την επιλογή **-k**:

```
$ bzip2 -k errors.log           # compress but keep input file
```

```
$ ls -l
```

```
total 140
```

```
-rw-r--r-- 1 nemo nemo 138432 Dec 17 20:25 errors.log
```

```
-rw-r--r-- 1 nemo nemo      893 Dec 17 20:25 errors.log.bz2
```

```
$ bzip2 -v errors.log           # verbose
```

```
errors.log: 155.019:1, 0.052 bits/byte, 99.35% saved,  
138432 in, 893 out.
```

# bzip2recover

Η εντολή **bzip2recover** μας παρέχει τη δυνατότητα να προσπαθήσουμε να επαναφέρουμε ένα κατεστραμμένο συμπιεσμένο αρχείο:

```
$ bzip2recover damaged-errors.log.bz2
```

```
bzip2recover 1.0.6: extracts blocks from damaged .bz2 files.
```

```
bzip2recover: searching for block boundaries ...
```

```
    block 1 runs from 80 to 7212
```

```
bzip2recover: splitting into blocks
```

```
    writing block 1 to `rec00001damaged-errors.log.bz2' ...
```

```
bzip2recover: finished
```

# Ερωτήσεις

