

Τ_EXνικές: Αναδρομικές μακροεντολές — μέρος 1ο

Απόστολος Συρόπουλος

28ης Οκτωβρίου 366

671 00 Ξάνθη

E-mail: apostolo@obelix.ee.duth.gr

Το Τ_EX εκτός από άριστο σύστημα στοιχειοθεσίας είναι και ένας άψογος επεξεργαστής μακροεντολών. Μια μακροεντολή είναι πολύ απλά ένα συμβολικό όνομα που δίνουμε σε μία συλλογή εντολών. Για παράδειγμα αν σ' ένα έγγραφο χρησιμοποιούμε πολλές φορές την έκφραση «Κυρίες και Κύριοι», τότε μπορούμε να ορίσουμε τη μακροεντολή:

```
\def\lag{Κυρίες και Κύριοι,}
```

Τώρα κάθε φορά που θα γράφουμε στο αρχείο Τ_EX/Λ^AΤ_EX/ την εντολή `\lag`, αυτή θα παράγει αυτόματα την έκφραση «Κυρίες και Κύριοι». Όπως φαίνεται από το παράδειγμα μια μακροεντολή ορίζεται με το γράφουμε την εντολή `\def`, το όνομα της νέας εντολής (στην περίπτωση μας `\lag`) και, τέλος, σ' άγκιστρα το κείμενο ή και τις εντολές που θα εμφανίζονται ή θα εκτελούνται κάθε φορά που εκτελούμε την συγκεκριμένη νέα μακροεντολή. Από τη στιγμή που ορίσουμε μια νέα μακροεντολή, μπορούμε να τη χρησιμοποιήσουμε στον ορισμό μιας άλλης νέας μακροεν-

τολής. Αν και εξαιρετικά χρήσιμες, αυτού του είδους οι μακροεντολές δεν επιτρέπουν στον προγραμματιστή να κάνει πολλά πράγματα. Αυτό που κάνει το Τ_EX ένα εξαιρετικό εργαλείο είναι το γεγονός ότι επιτρέπει αναδρομικές (recursive) μακροεντολές. Όμως τι είναι η αναδρομή;

Ένας ορισμός καλείται αναδρομικός αν γίνεται με βάση τον εαυτό του. Για παράδειγμα το παραγοντικό ενός ακεραίου n ορίζεται ως εξής:

$$n! = \begin{cases} 1, & \text{αν } n = 0 \\ n \cdot (n - 1)!, & \text{αλλιώς} \end{cases}$$

Η πρώτη περίπτωση ονομάζεται *τερματική* συνθήκη και η δεύτερη *συνθηκη αναδρομής*. Ένα βασικό χαρακτηριστικό όλων των αναδρομικών ορισμών αποτελεί το γεγονός ότι μπορούμε να τους υλοποιήσουμε με καθαρά επαναληπτικές μεθόδους και με τη χρήση μιας ειδικής δομής δεδομένων που ονομάζεται *στοίβα*. Σκοπός μας βέβαια δεν είναι να κάνουμε μια σύντομη εισαγωγή στην επιστήμη των Η/Υ, αλλά απλά να δείξουμε κάποιες ιδιαίτερες δυνατότητες

```

\newcount\nfac
\newcount\n
\newcount\rn
\nfac=1
\immediate\writel6 {*****}
\immediate\writel6 {Δώσε ένα θετικό ακέραιο μικρότερο από 13.}
\read-1to\num
\n\number\num
\rn\n
\def\fac {\ifnum\n>1\dofac\fi}
\def\dofac {\multiply\nfac\n
           \advance\n-1\fac}
\fac
\immediate\writel6 {Το παραγοντικό του \the\rn\space είναι %
                   \the\nfac.}
\immediate\writel6 {*****}
\end

```

Σχήμα 1: Ο κώδικας αρχείου T_EX που υπολογίζει το παραγοντικό ενός ακεραίου.

```

\documentclass{article}
\begin{document}
\makeatletter
\newcounter{nfac} % Το LaTeX ορίζει αυτομάτως τους εσωτερικούς
\newcounter{n} % μετρητές: \c@nfac, \c@n και \c@rn. Αυτούς
\newcounter{rn} % τους μετρητές τούς χρησιμοποιούμε παρακάτω.
\setcounter{nfac}{1}
\typeout{*****}
\typein[\num]{Δώσε ένα θετικό ακέραιο μικρότερο από 13.}
\setcounter{n}{\number\num}
\setcounter{rn}{\value{n}}
\newcommand{\fac}{\ifnum\value{n}>1\dofac\fi}
\newcommand{\dofac}{\multiply\c@nfac\c@n
                   \addtocounter{n}{-1}\fac}
\fac
\typeout{Το παραγοντικό του \thern\space είναι \thenfac.}
\typeout{*****}
\makeatother
\end{document}

```

Σχήμα 2: Ο κώδικας αρχείου L^AT_EX που υπολογίζει το παραγοντικό ενός ακεραίου.

του ΤΕΧ. Έτσι, δεν θα επεκταθούμε περισσότερο στον τρόπο υλοποίησης της αναδρομής. Ας υποθέσουμε τώρα ότι θέλουμε να φτιάξουμε ένα απλό αρχείο ΤΕΧ το οποίο θα διαβάσει ένα θετικό ακέραιο¹ από το πληκτρολόγιο και θα εμφανίζει το παραγοντικό του αριθμού αυτού στην οθόνη του υπολογιστή μας. Αν αυτό φαντάζει αδύνατο, έχουμε καλά νέα για σας που θέλετε να κάνετε ενδιαφέροντα πράγματα με το ΤΕΧ: οι μακροεντολές του ΤΕΧ μπορούν να είναι αναδρομικές! Στο Σχήμα 1 δίνεται ο κώδικας ενός αρχείου ΤΕΧ το οποίο κάνει ακριβώς αυτό που ζητάμε. Ας προσπαθήσουμε να εξηγήσουμε τι ακριβώς κάνει ο κώδικας του Σχήματος 1. (Ο κώδικας του Σχήματος 2, που δίνεται χωρίς λεπτομερείς εξηγήσεις, κάνει ακριβώς τα ίδια πράγματα χρησιμοποιώντας τις αντίστοιχες «καθαρές» εντολές του \LaTeX .)

Με την εντολή `\newcount` ορίζουμε ένα νέο μετρητή και με την εντολή `\nfac=1` αναθέτουμε τον αριθμό 1 στο μετρητή αυτό. Σημειώστε ότι το σύμβολο `=` είναι προαιρετικό. Οι εντολές `\read-1` και `\immediate\write16` διαβάζουν κείμενο από το πληκτρολόγιο και εμφανίζουν κείμενο στην οθόνη του υπολογιστή αντίστοιχα. Η εντολή `\n\number\num` αναθέτει στο μετρητή `\n` τον αριθμό που προκύπτει από τη μετατροπή της λέξης που περιέχεται στη μακροεντολή `\num`. Αν για τον οποιοδήποτε λόγο η μακροεντολή `\num` δεν αντιστοιχεί σε μια ακολουθία ψηφίων το ΤΕΧ σταματά με ανάλογο μήνυμα λά-

θους. Σημειώστε ότι αγνοεί παντελώς ότι κενά πληκτρολογήσουμε.

Περνάμε τώρα στον ορισμό της μακροεντολής `\fac`. Εδώ βλέπουμε χρήση μιας πρωτόγονης εντολής του ΤΕΧ: της εντολής `\if`. Η εντολή `\if` έχει διάφορες μορφές και μία από αυτές, η `\ifnum`, επιτρέπει την σύγκριση αριθμών ή/και μετρητών. Οι δυνατές συγκρίσεις είναι τρεις: μεγαλύτερο (`>`), μικρότερο (`<`) και ίσο (`=`). Αν η σύγκριση είναι αληθής, εκτελείται ότι ακολουθεί τη σύγκριση μέχρι το ΤΕΧ να συναντήσει τη λέξη `\fi`, οπότε και τερματίζεται. Αν συναντήσει τη λέξη `\else` πριν από τη λέξη `\fi`, τότε απλά εκτελεί ότι ακολουθεί αυτή τη λέξη μέχρι να βρει τη λέξη `\fi` αν και μόνο αν η σύγκριση είναι ψευδής. Μετά από αυτή την αναγκαστική παρένθεση, συνεχίζουμε με τον ορισμό της μακροεντολής `\fac`: η μακροεντολή αυτή ελέγχει αν ο μετρητής `\n` είναι μεγαλύτερος από το 1 και αν είναι εκτελεί τη μακροεντολή `\dofac`. Η μακροεντολή αυτή με την σειρά της πολλαπλασιάζει το μετρητή `\nfac` με την τρέχουσα τιμή του μετρητή `\n`, μειώνει τη τιμή του μετρητή `\n` κατά ένα και εκτελεί τη μακροεντολή `\fac`. Απλό δεν είναι; Προσέξτε ότι δημιουργούμε δύο μακροεντολές: η μια περιέχει την τερματική συνθήκη και μία τη συνθήκη αναδρομής. Ως άσκηση μπορείτε να μετατρέψετε τον παραπάνω κώδικα ώστε να υπολογίζει το άθροισμα των αριθμών από το 1 ως την τιμή του μετρητή `\nfac`; (Υπόδειξη: το άθροισμα μπορεί

¹ Η μέγιστη τιμή που μπορεί να λάβει ένας μετρητής του ΤΕΧ/Ω μπορεί να είναι μικρότερη ή ίση του $2^{30} = 1073741824$.

να ορισθεί ως εξής:

$$s(n) = \begin{cases} 0, & \text{αν } n = 0 \\ n + s(n - 1), & \text{αλλιώς} \end{cases}$$

όπου s μια συνάρτηση που υπολογίζει το άθροισμα των αριθμών από το 1 ως το n . Η απάντηση βρίσκεται στο τέλος του άρθρου.)

Αν και το αρχείο μας παράγει τα αναμενόμενα αποτελέσματα, εντούτοις ένας καλός ΤΕΧνίτης θα δήλωνε την μακροεντολή `\fac` ως εξής:

```
\def\fac{\ifnum\n>1%
\expandafter\do\fac\fi}
```

Δηλαδή θα χρησιμοποιούσε την πρωτόγονη εντολή `\expandafter` ακριβώς πριν από τη μακροεντολή `\do\fac`. Ασφαλώς θα αναρωτιέστε γιατί θα πρέπει να γίνει αυτό και επιπλέον ποιά η σημασία αυτής της εντολής. Ας πάρουμε τα πράγματα με την σειρά. Όταν το ΤΕΧ εκτελεί μία εντολή αυτό που κάνει είναι ουσιαστικά να την αναγάγει (`expand`) σε πρωτόγονες εντολές

τις οποίες φυσικά μπορεί να εκτελέσει. Υπάρχουν συγκεκριμένοι κανόνες που διέπουν την αναγωγή μακροεντολών, αλλά υπάρχουν και εντολές που αλλάζουν αυτή την συμπεριφορά. Μια τέτοια εντολή είναι η `\expandafter`. Πιο συγκεκριμένα αν έχουμε τη σειρά εντολών `\expandafter\la\lb`, όπου τα `\la` `\lb` είναι είτε μακροεντολές είτε κάποιο ειδικό σύμβολο, π.χ., το σύμβολο `{`, το ΤΕΧ πρώτα αναγάγει την `\lb` και μετά βάζει μπροστά την `\la` και αναγάγει συνολικά το αποτέλεσμα. Στην περίπτωση μας, αυτό απλά σημαίνει ότι έχουμε λιγότερες απαιτήσεις από το ΤΕΧ και για του λόγου το αληθές συγκρίνεται τα στατιστικά στο τέλος του αρχείου `.log` που παράγεται μετά την εκτέλεση του αρχείου `ΛΤΕΧ`.

Είδαμε τον τρόπο με τον οποίο κανείς μπορεί να δηλώσει απλές αναδρομικές μακροεντολές. Στο επόμενο τεύχος του περιοδικού θα συζητήσουμε τον τρόπο με τον οποίο κανείς μπορεί να δηλώσει *παραμετρικές μακροεντολές*, δηλ. νέες εντολές που έχουν παραμέτρους.

Σχήμα 3: Η απάντηση στην άσκηση του άρθρου σε κώδικα ΤΕΧ.

```
\newcount\sum
\newcount\n
\newcount\rn
\sum=1
\immediate\writel6{*****}
\immediate\writel6{Δώστε ένα θετικό ακέραιο}
\read-1to\num
\n\number\num
\rn\n
\def\Sum{\ifnum\n>1\expandafter\do\Sum\fi}
\def\doSum{\advance\sum by\n}
```

```

\advance\n-1\Sum}
\ifnum\rn>0
  \Sum
  \immediate\write16{Το άθροισμα των πρώτων %
    \the\rn\space ακεραίων είναι \the\sum.}
\else
  \immediate\write16{0 αριθμός που δώσατε δεν είναι θετικός!}
\fi
\immediate\write16{*****}
\end

```

Σχήμα 4: Η απάντηση στην άσκηση του άρθρου σε κώδικα L^AT_EX.

```

\documentclass{article}
\begin{document}
\newcounter{sum} % Στην περίπτωση ετούτη, δεν χρειάζεται
\newcounter{n} % να χρησιμοποιήσουμε τους εσωτερικούς
\newcounter{rn} % μετρητές του LaTeX (βλ. Σχήμα 2).
\setcounter{sum}{0}
\typeout{*****}
\typein[\num] {Δώστε ένα θετικό ακέραιο}
\setcounter{n}{\number\num}
\setcounter{rn}{\value{n}}
\newcommand{\Sum}{\ifnum\value{n}>1\expandafter\doSum\fi}
\newcommand{\doSum}{\addtocounter{sum}{\value{n}}
  \addtocounter{n}{-1}\Sum}
\ifnum\value{rn}>0
  \Sum
  \typeout{Το άθροισμα των πρώτων %
    \thern\space ακεραίων είναι \thesum.}
\else
  \typeout{0 αριθμός που δώσατε δεν είναι θετικός!}
\fi
\typeout{*****}
\end{document}

```

