



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ

Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

— ΙΔΡΥΘΕΝ ΤΟ 1837 —



Τμήμα Πληροφορικής και Τηλεπικοινωνιών • Εισαγωγή στον Προγραμματισμό

Περιεχόμενα

Κεφάλαιο 10: Ο προεπεξεργαστής της C	1
Ο προεπεξεργαστής της C	1

Κεφάλαιο 10: Ο προεπεξεργαστής της C

Ο προεπεξεργαστής της C

- Ο προεπεξεργαστής της C είναι ένα αυτόνομο υποσύστημα του λογισμικού μεταγλώττισης, το οποίο καλείται αυτόματα πριν από την πραγματική διαδικασία της μεταγλώττισης. Ο σκοπός του προεπεξεργαστή είναι να μετασχηματίσει το πηγαίο αρχείο C που του δόθηκε σ' ένα άλλο πηγαίο αρχείο, σύμφωνα με μία σειρά από οδηγίες που απευθύνονται σ' αυτόν. Το αποτέλεσμα του προεπεξεργαστή είναι αυτό το οποίο θα υποστεί τη διαδικασία της μεταγλώττισης.
- Οι γραμμές του πηγαίου αρχείου που αρχίζουν από # είναι οδηγίες προς τον προεπεξεργαστή.
- Ακολουθεί μία συνοπτική περιγραφή των πιο συχνά χρησιμοποιούμενων οδηγιών του προεπεξεργαστή. Σχετικά πιο αναλυτική αναφορά γίνεται στην παράγραφο §4.11 του [KR] (σελ. 128–133).

#include

- Χρησιμοποιείται για την εισαγωγή των περιεχομένων αρχείων επικεφαλίδας στη θέση που βρίσκεται η οδηγία. Ένα αρχείο επικεφαλίδας περιέχει συνήθως δηλώσεις πρωτοτύπων συναρτήσεων, ορισμούς συμβολικών σταθερών και μακροεντολών (μέσω #define), συμπεριλήψεις άλλων αρχείων επικεφαλίδας, δηλώσεις μορφής τύπων δεδομένων (μέσω struct, enum, union, typedef), κλπ. Σε αρχεία επικεφαλίδας, *δεν* συνηθίζεται να έχουμε κώδικα προγράμματος. Επίσης, *δεν* κάνουμε #include πηγαία αρχεία κώδικα. Η οδηγία αυτή εμφανίζεται με δύο εκδοχές.

- Πρώτη εκδοχή: `#include <sysfile.h>`
Ο προεπεξεργαστής θα ψάξει να βρει το αρχείο `sysfile.h` σε καταλόγους που έχει δηλώσει με κάποιο τρόπο ο χρήστης, για παράδειγμα με την επιλογή `-I` κατά την κλήση του `gcc`, ή στους καταλόγους που βρίσκονται τα προκαθορισμένα αρχεία επικεφαλίδας της γλώσσας, οι οποίοι μπορεί να διαφέρουν από εγκατάσταση σε εγκατάσταση.
- Δεύτερη εκδοχή: `#include "myfile.h"`
Ο προεπεξεργαστής θα ψάξει κατ' αρχήν να βρει το αρχείο `myfile.h` στον κατάλογο που βρίσκεται το πηγαίο αρχείο, και αν δεν βρεθεί εκεί, θα το αναζητήσει σε καταλόγους όπως θα γινόταν αν η οδηγία είχε δοθεί με την πρώτη της εκδοχή.
- Φυσικά, είναι επιτρεπτές και δηλώσεις:

```
#include <sys/types.h>
#include "mydir/anotherdir/myfile.h"
```

`#define`

- Χρησιμοποιείται για τον ορισμό συμβολικών σταθερών ή μακροεντολών. Όπου βρει ο προεπεξεργαστής σαν λεκτικό σύμβολο (όχι μέσα σε " ή σε σχόλια) τη συμβολική σταθερά ή την παραμετρική μακροεντολή που ορίζεται με την οδηγία, την αντικαθιστά με το κείμενο που της έχει αντιστοιχηθεί.
- Παράδειγμα ορισμού συμβολικής σταθεράς:

```
#define YES 1
```

Ο προεπεξεργαστής θα αντικαταστήσει το `YES`, όπου το βρει σαν λεκτικό σύμβολο μέσα στο πηγαίο αρχείο με το `1` (όχι όμως στο `printf("YES\n")`, ούτε και στο `x = YESMAN++`).

- Παράδειγμα ορισμού μακροεντολής:

```
#define max(A, B) ((A) > (B) ? (A) : (B))
```

Αν βρει ο προεπεξεργαστής στη συνέχεια στο αρχείο κάποια εντολή της μορφής

```
x = max(p+q, r+s);
```

θα την αντικαταστήσει με την

```
x = ((p+q) > (r+s) ? (p+q) : (r+s));
```

Παρατηρήστε, στο παράδειγμα αυτό, ότι τα `A` και `B` θα υπολογισθούν τελικά δύο φορές, αφού τόσες βρίσκονται στο κείμενο αντικατάστασης, πράγμα που μπορεί στο συγκεκριμένο παράδειγμα να μην ενοχλεί, αλλά σε άλλες περιπτώσεις χρειάζεται προσοχή.¹

¹Πιστεύετε ότι η έκφραση `max(i++, j++)` θα αντικατασταθεί σωστά με βάση τι είχε στο μυαλό του μάλλον ο προγραμματιστής που την έγραψε;

- Οι παρενθέσεις γύρω από τις παραμέτρους στο κείμενο αντικατάστασης μίας μακροεντολής ενδέχεται, σε κάποιες περιπτώσεις, να είναι κρίσιμες. Παράδειγμα:

```
#define square(X)      X * X
```

Πώς θα αντικατασταθεί το `square(a+1)`; ² Αυτό ήταν που θέλαμε; ³ Πώς θα έπρεπε να δηλωθεί η μακροεντολή; ⁴

- Αρκετές από τις συναρτήσεις της γλώσσας που χρησιμοποιούμε στα προγράμματά μας είναι ορισμένες σαν μακροεντολές μέσα σε αρχεία επικεφαλίδας και όχι με κώδικα στη βιβλιοθήκη της γλώσσας. Για παράδειγμα, οι γνωστές μας συναρτήσεις `getchar` και `putchar` ορίζονται μέσα στο αρχείο `stdio.h` ως εξής:

```
#define getchar()      getc(stdin)
#define putchar(x)     putc(x, stdout)
```

`#if`, `#else`, `#elif`, `#endif`, `#ifdef`, `#ifndef`

- Στην C, υπάρχει η δυνατότητα μεταγλώττισης υπό συνθήκη. Ό,τι περιλαμβάνεται μεταξύ ενός `#if` και του αμέσως επόμενου `#endif`, που δεν έχει αντιστοιχηθεί με κάποιο άλλο `#if`, θα μεταγλωττιστεί μόνο αν η ακέραιο παράσταση που βρίσκεται μετά το `#if` αποτιμηθεί σε μη μηδενική τιμή.
- Συνήθως, οι ακέραιοι παραστάσεις μετά το `#if` συγκρίνουν τιμές συμβολικών σταθερών, που έχουν αντιστοιχηθεί σε ακέραιοις, μεταξύ τους ή με ακέραιοις σταθερές, μέσω των τελεστών σύγκρισης της C (`==`, `!=`, `>`, κλπ.).
- Με την οδηγία `#else`, αρχίζει το τμήμα μίας δομής `#if/#endif` που θα μεταγλωττιστεί αν η παράσταση μετά το `#if` είναι ψευδής (ίση με μηδέν).
- Η οδηγία `#elif` μπορεί να χρησιμοποιηθεί για πιο σύντομη γραφή μίας `#else` της οποίας το σώμα αποτελείται από μία `#if`, αποφεύγοντας έτσι και το επιπλέον `#endif`.
- Παράδειγμα:

```
#if SYSTEM == SYSV
    #define HDR "sysv.h"
#elif SYSTEM == BSD
    #define HDR "bsd.h"
#elif SYSTEM == MSDOS
    #define HDR "msdos.h"
#else
    #define HDR "default.h"
#endif
#include HDR
```

²`a+1 * a+1`

³Όχι, βέβαια.

⁴`#define square(X) ((X) * (X))`

- Κάποια ειδική παράσταση που μπορεί να υπάρχει μετά από ένα `#if` είναι η `defined(NAME)`, που αποτιμάται σε 1 αν το NAME έχει οριστεί με `#define`, αλλιώς σε 0. Η οδηγία

```
#if defined(NAME)
```

είναι ισοδύναμη με την

```
#ifdef NAME
```

- Στις παραστάσεις μετά από ένα `#if` μπορούμε να έχουμε και άρνηση (!). Για παράδειγμα, αν θέλουμε να εξασφαλίσουμε ότι τα περιεχόμενα του αρχείου `hdr.h` θα συμπεριληφθούν μία μόνο φορά σε μία πολύπλοκη εφαρμογή με πολλές συμπεριλήψεις, θα μπορούσαμε να το δομήσουμε ως εξής:

```
#if !defined(HDR)
#define HDR
/* The real contents of hdr.h go here */
#endif
```

Το `#if !defined(HDR)` θα μπορούσε να γραφεί ισοδύναμα και σαν `#ifndef HDR`

- Δομές `#if` (`#ifdef`, `#ifndef`)/`#endif` μπορεί να είναι εμφωλευμένες. Με τη βοήθειά τους μπορούμε να προσομοιώσουμε εμφωλευμένα σχόλια, κάτι που με τον συνηθισμένο τρόπο γραφής σχολίων (`/* ... */`) δεν είναι δυνατόν. Πώς;