



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά
μαθήματα ΠΠ

Τεχνολογία και Προγραμματισμός Υπολογιστών

Ενότητα 9: Αρχεία - Λίστες

Διδάσκων

Χρήστος Μακρής

Τμήμα Μηχανικών Η/Υ & Πληροφορικής

Πανεπιστήμιο Πατρών

Αρχεία – Λίστες

(οι διαφάνειες είναι βασισμένες στο βιβλίο Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C KERNIGHAN W. BRIAN, RITCHIE M. DENNIS, εκδόσεις Κλειδάριθμος δεύτερη έκδοση, 2011)

Ενώσεις (Unions) -1

Μία ένωση (union) μπορεί να περιέχει αντικείμενα διαφορετικών τύπων και μεγεθών

```
π.χ. union triple_to_one {  
    int inter_i;  
    float inter_f;  
    char inter_char;  
} var;
```

Ενώσεις (Unions) -2

```
if (type == INT)
```

```
    printf(“%d\n”, var. inter_i);
```

```
else if (utype==FLOAT)
```

```
    printf(“%f\n”, var. inter_f);
```

```
else if (utype==STRING)
```

```
    printf(“%s\n”, var. inter_char);
```

Στις ενώσεις επιτρέπεται απόδοση τιμής ή αντιγραφή σαν ενότητα, λήψη διεύθυνσης, προσπέλαση μέλους.

Η αρχική τιμή πρέπει να έχει τύπο ίδιο με αυτό του πρώτου της μέλους.

Ορίσματα Γραμμής Διαταγών

```
#include <stdio.h>

main(int argc, char * argv[])
{
    int i;
    for (i=1; i<argc; i++) printf("%s ", argv[i]);
    printf("\n");
    return 0;
}
```

Δείκτες σε Συναρτήσεις

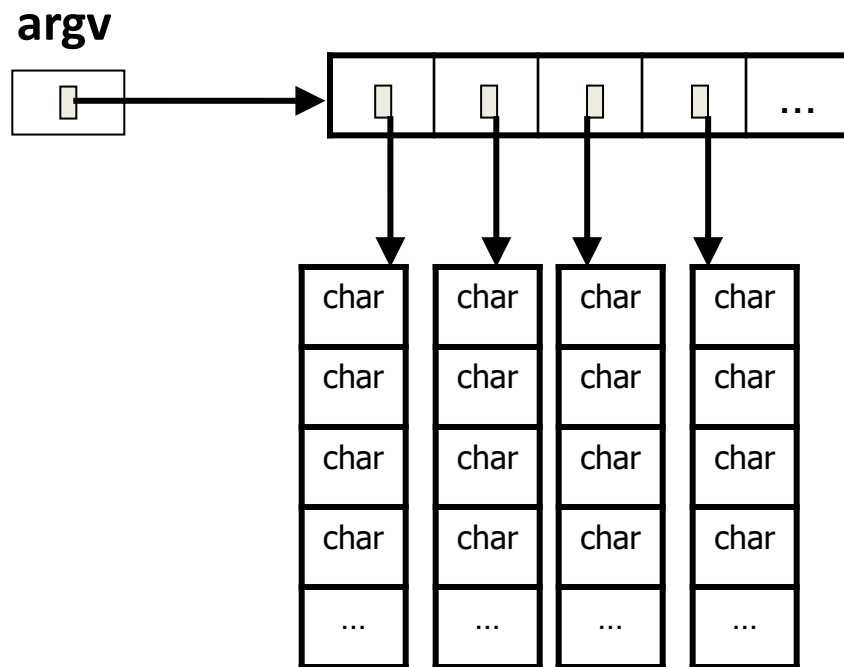
Στην C η συνάρτηση δεν είναι μεταβλητή. Μπορούμε να ορίσουμε δείκτες σε συναρτήσεις, που μπορούν να αποδίδονται σαν τιμές, να τοποθετούνται σε πίνακες, να μεταβιβάζονται σε συναρτήσεις και να επιστρέφονται από αυτές.

Παραδείγματα Δηλώσεων

<code>char **argv</code>	δείκτης σε δείκτη σε char
<code>int (*daytab)[13]</code>	δείκτης σε πίνακα[13] int
<code>int *daytab[13]</code>	πίνακας[13] με δείκτες σε int
<code>void *comp()</code>	συνάρτηση που επιστρέφει δείκτη σε void
<code>void (*comp)()</code>	δείκτης σε συνάρτηση που επιστρέφει void
<code>char ((*x())[5])()</code>	συνάρτηση που επιστρέφει δείκτη σε πίνακα δεικτών σε συνάρτηση που επιστρέφει char
<code>char ((*x[3])())[5]</code>	πίνακας [3] με δείκτες σε συνάρτηση που επιστρέφει δείκτη σε πίνακα [] char

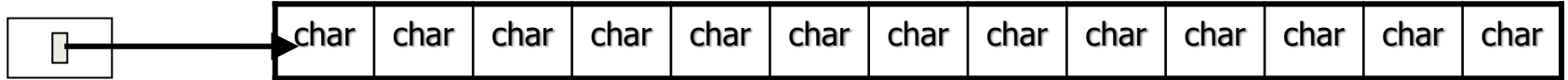
Από: Η ΓΛΩΣΣΑ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ C
KERNIGHAN W. BRIAN, RITCHIE M. DENNIS, εκδόσεις Κλειδάριθμος, δεύτερη έκδοση. 1990

char **argv

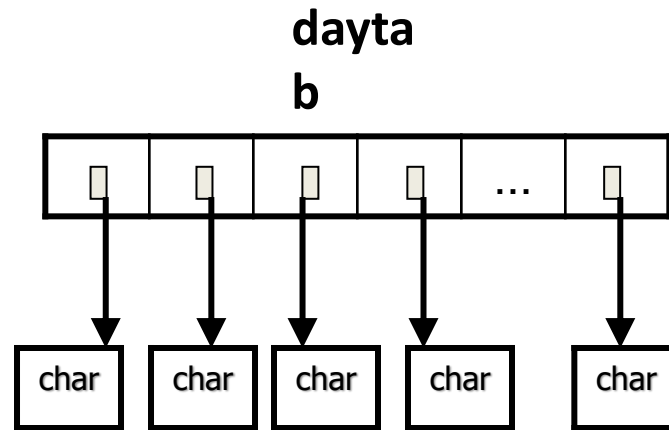


char (*daytab)[13]

daytab

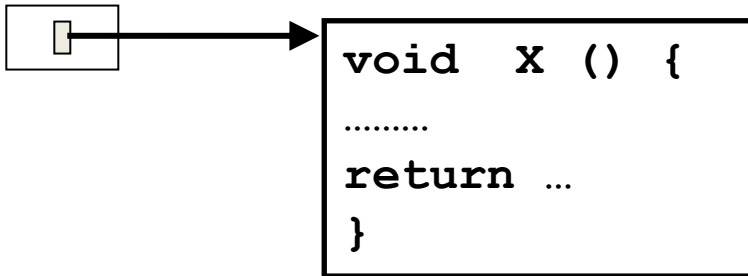


char *daytab[13]

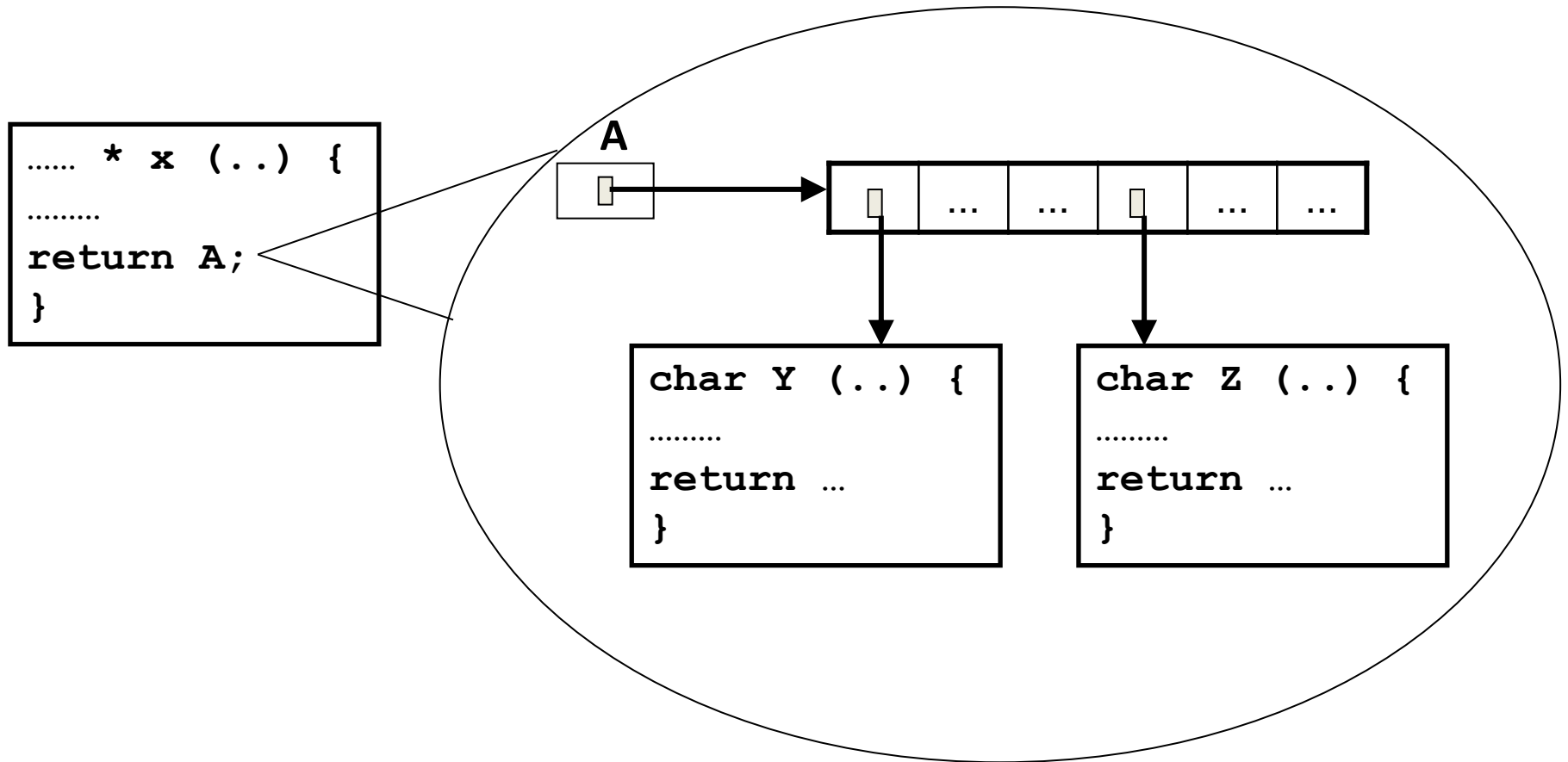


void (*comp)()

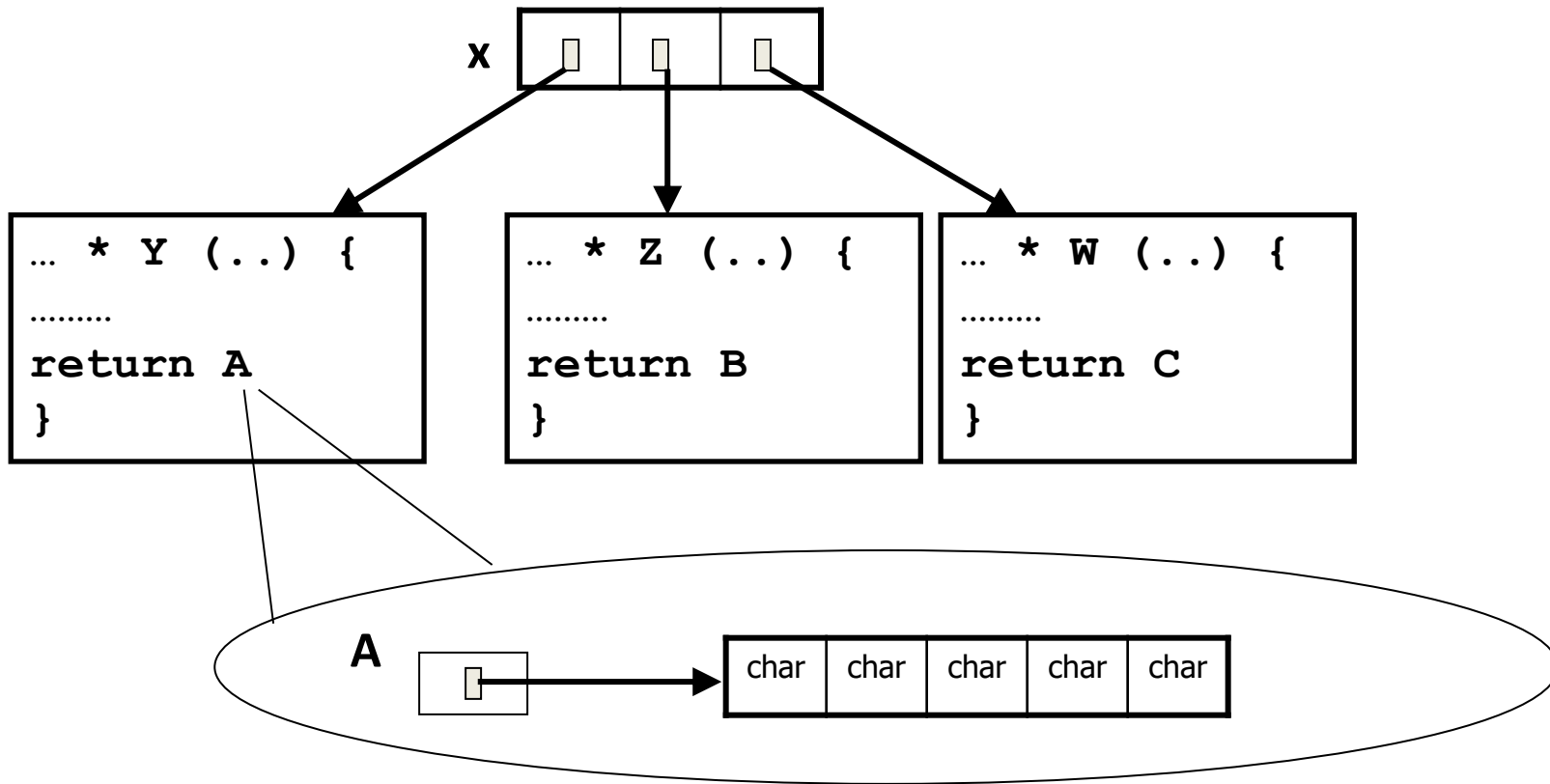
comp



char (*(*x())[])()



char (*(*x[3]))[5]



Επεξεργασία Αρχείων στη C

- Με τη χρήση `FILE *` αναπαριστάνουμε δείκτη σε ένα αρχείο.
- `fopen` χρησιμοποιείται για να ανοίγει ένα αρχείο. Επιστρέφει την ειδική τιμή `NULL` για να δείξει ότι δεν μπορεί να ανοίξει ένα αρχείο.

```
FILE *fptr;
char filename[] = "file2.dat";
fptr = fopen (filename, "w");
if (fptr == NULL) {
    fprintf (stderr, "ERROR");
    /* DO SOMETHING */
}
```

Τρόποι Ανοίγματος Αρχείου

- `FILE * fopen(const char * filename, const char * mode)`
- Το δεύτερο όρισμα της `fopen` καθορίζει τον τρόπο με τον οποίο ανοίγουμε ένα αρχείο.
Υπάρχουν οι εξής επιλογές:
- "r" ανοίγει ένα αρχείο για διάβασμα
- "w" ανοίγει ένα αρχείο για εγγραφή
- "a" ανοίγει ένα αρχείο για προσθήκη / εγγραφή
- "b" δυαδικά δεδομένα (binary)
- `rb, wb, ab, r+, w+, a+, rb+,wb+,ab+`

Διαχείριση Αρχείων

- Κλείσιμο αρχείων

```
int fclose (FILE * fp);
```

- Είσοδος – έξοδος χαρακτήρων

```
int putc(int c, FILE * fp);
```

(ισοδύναμα fputs())

```
int getc(FILE * fp);
```

(ισοδύναμα fgets())

- Είσοδος – έξοδος συμβολοσειρών

```
int fputs (const char * s, FILE * fp);
```

```
char * fgets (char * s, int n, FILE * fp);
```


- `int feof(FILE *fp)`
Not 0: end of file
- `void rewind (FILE *fp);`
- `int fflush(FILE *fp)`
0: success, EOF: error
- `int fseek(FILE *fp, long int apostasi, int thesi)`
thesi=SEEK_SET (0), SEEK_CUR (1), SEEK_END (2)

`int ftell(FILE *fp)`
- `int fread(void *buffer, int arbytes, int fores, FILE *fp)`
Διάβασε fores στοιχεία, μεγέθους arbytes το καθένα, στο buffer array
- `int fwrite(void *buffer, int arbytes, int fores, FILE *fp)`
Γράψε fores στοιχεία, μεγέθους arbytes το καθένα, από το buffer array

Γράφοντας σε Αρχείο με *fprintf*

- Η `fprintf` δουλεύει όπως η `printf` και η `sprintf` με την διαφορά ότι το πρώτο όρισμα είναι ένας δείκτης σε αρχείο.

```
FILE *fptr;  
fptr= fopen ("file.dat", "w");  
/* Check it's open */  
fprintf (fptr, "Hello World!\n");
```

- Θα μπορούσαμε επίσης να διαβάσουμε αριθμούς από ένα αρχείο χρησιμοποιώντας `fscanf` – αλλά υπάρχει και καλύτερος τρόπος.

Διαβάζοντας από Αρχείο με *fgets*

- *fgets* είναι ένας καλύτερος τρόπος ανάγνωσης από αρχείο

- Μπορούμε να διαβάσουμε από αρχείο με *fgets*

```
FILE *fptr;  
char line [1000];  
/* Open file and check it is open */  
while (fgets(line,1000,fptr) != NULL) {  
    printf ("Read line %s",line);  
}
```

fgets έχει 3 ορίσματα, μία συμβολοσειρά, ένα μέγιστο αριθμό χαρακτήρων προς ανάγνωση και ένα δείκτη αρχείου. It returns NULL if there is an error (such as EOF)

fgets(char*s, int n, FILE *fstream); (διαβάζει το πολύ n-1 χαρακτήρες σταματώντας αν συναντήσει χαρακτήρα νέας γραμμής)

Κλείνοντας ένα Αρχείο

- Μπορούμε να κλείσουμε ένα αρχείο απλώς χρησιμοποιώντας `fclose` και το δείκτη αρχείου. Ακολουθεί παράδειγμα "hello files".

```
FILE *fptr;
char filename[] = "myfile.dat";
fptr = fopen (filename, "w");
if (fptr == NULL) {
    printf ("Cannot open file to write!\n");
    exit(-1);
}
fprintf (fptr, "Hello World of filing!\n");
fclose (fptr);
```

Παράδειγμα (3)

Να υλοποιηθούν και να χρησιμοποιηθούν οι ακόλουθες συναρτήσεις:

1. **int createRandFile(char *fileName,int N):** Δημιουργεί ένα αρχείο με όνομα `fileName` στο οποίο γράφει `N` τυχαίους αριθμούς. Επιστρέφει `1` σε επιτυχία και `-1` σε αποτυχία
2. **int *loadData(char *fileName,int N):** Διαβάζει από το αρχείο `fileName` `N` τυχαίους αριθμούς και τους επιστρέφει σε ένα δείκτη ακεραίων.
3. **int writeArray2File(char *fileName,int *array,int N):** Γράφει τους αριθμούς του διανύσματος `array` στο αρχείο `fileName`. Επιστρέφει `1` σε επιτυχία και `-1` σε αποτυχία.

```
int createRandFile(char *fileName,int N)
{
    FILE *fp;
    int i;

    fp = fopen(fileName,"w");
    if (fp)
    {
        srand((unsigned)time(NULL));
        for (i=0;i<N;i++)
        {
            fprintf(fp,"%d\n",(MAX*rand())/RAND_MAX);
        }
        fclose(fp);
        return 1;
    }
    else
        return -1;
}
```

```
int *loadData(char *fileName,int N)
{
    int *array = (int*)malloc(sizeof(int)*N);
    FILE *fp;
    int i;

    fp = fopen(fileName,"r");
    if (fp)
    {
        for (i=0;i<N;i++)
        {
            fscanf(fp,"%d",&array[i]);
        }
        fclose(fp);
        return array;
    }
    else
        return NULL;
}
```

```
int writeArray2File(char *fileName,int *array,int N)
{
    FILE *fp;
    int i;

    fp = fopen(fileName,"w");
    if (fp)
    {
        for (i=0;i<N;i++)
        {
            fprintf(fp,"%d\n",array[i]);
        }
        fclose(fp);
        return 1;
    }
    else
    return -1;
}
```


Τέλος Ενότητας

Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Πατρών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Σημειώματα

Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση **1.0**.



Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.