Δομές Δεδομένων και Αλγόριθμοι

Χρήστος Γκόγκος

ΤΕΙ Ηπείρου

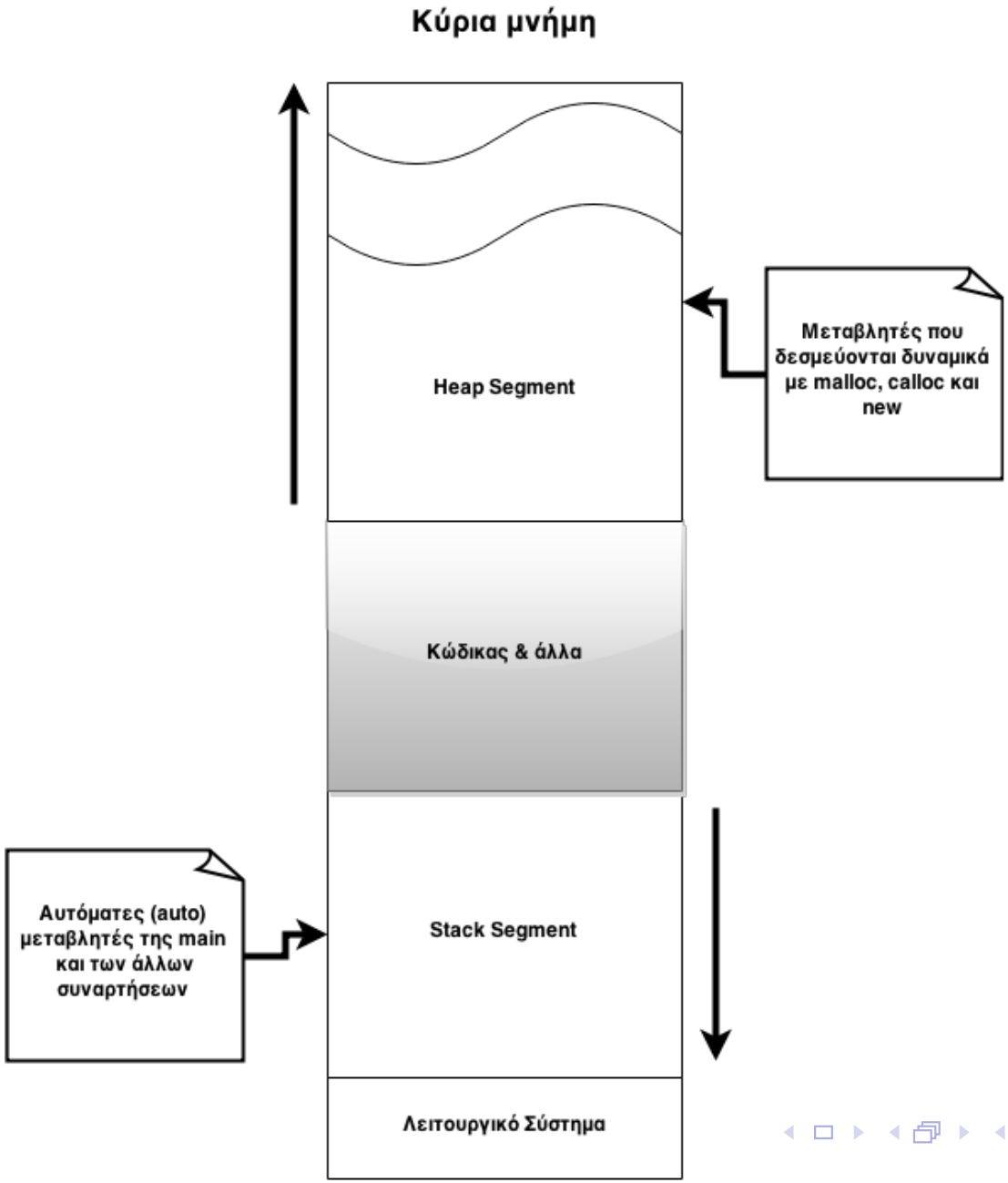
Χειμερινό Εξάμηνο 2014-2015
Παρουσίαση 3. Δείκτες - Pointers

Δείκτες

Οι δείκτες είναι ειδικού τύπου μεταβλητές που περιέχουν διευθύνσεις άλλων θέσεων μνήμης.

```
int  y = 30;
int  *py;
py = &y;
 printf (”y=%d *py=%d py=%p &y=%p\n”, y, *py, py, &y);
 //  output
 //  y=30 *py=30 py=000000000022fd3c &y=000000000022fd3c
```

Δείκτες και μνήμη RAM



Δείκτες προς διάφορους τύπους δεδομένων

```
char c = 'a';
char* pc = &c;

int  i = 5;
int  *pi = &i;

double d = 10.5;
double *pd = &d;

printf ("char pointer to address=%p pointer size=%d
    dereferenced value=%c\n",
      pc, sizeof(pc), *pc);
printf (
      "integer pointer to address=%p pointer size=%d
          dereferenced value=%d\n",
      pi, sizeof(pi), *pi);
printf (
      "double pointer to address=%p pointer size=%d
          dereferenced value=%f\n",
      pd, sizeof(pd), *pd);

// char pointer to address=000000000022FDD7 pointer size=8
    dereferenced value=a
// integer pointer to address=000000000022FDD0 pointer
    size=8 dereferenced value=5
// double pointer to address=000000000022FDC8 pointer
    size=8 dereferenced value=10.500000
```

Δήλωση δείκτη και δυναμική δέσμευση μνήμης στη C

```
int  *px;
px = (int *) malloc(sizeof(int));
*px = 10;
printf (
        "Dynamic memory allocation pointer to an integer in the
             heap address=%p value=%d\n",
        px, *px);
free(px);
// Output:
// Dynamic memory allocation pointer to an integer in the heap
   address=0000000000587E20 value=10
```

Δήλωση δείκτη και δυναμική δέσμευση μνήμης στη C++

```
int  *px;
px = new int;
*px = 10;
//  ή
//   int  *px = new int(10);
printf (
            ”Dynamic memory allocation pointer to an integer in  the
                 heap address=%p value=%d\n”,
            px,  *px);
delete  px;
//  output Dynamic memory allocation pointer to an integer in  the
     heap address=000000000327ef0 value=10
```

Πίνακες και δείκτες

Το όνομα του πίνακα είναι η διεύθυνση της πρώτης θέσης του πίνακα.

```
int z[]={10, 20, 30, 40, 50};
int *p = z;
p++;
*p = 21;
for(int i=0;i<5;i++)
    cout << z[i] << " ";
cout << endl;
// output
// 10 21 30 40 50
```

Εγγραφές και δείκτες

```
struct date {
    int day;
    int month;
    int year;
};
struct date d1;
d1.day = 9;
d1.month = 10;
d1.year = 2014;

struct date d2 = { 10, 10, 2014 };
struct date *pd2 = &d2;

printf ("Date 1: %d-%d-%d size=%d\n", d1.day, d1.month,
    d1.year, sizeof(d1));
printf ("Date 2: %d-%d-%d size=%d\n", d2.day, d2.month,
    d2.year, sizeof(d2));
printf ("Date 2: %d-%d-%d size=%d\n", pd2->day,
    pd2->month, pd2->year, sizeof(*pd2));
// output
// Date 1: 9-10-2014 size=12
// Date 2: 10-10-2014 size=12
// Date 2: 10-10-2014 size=12
```

Δείκτες πολλαπλών επιπέδων

Ένας pointer μπορεί να περιέχει τη διεύθυνση ενός άλλου pointer.

```
int  x = 10;
int  *px = &x;
int  **ppx = &px;
printf ("[3 alternatives to get value of x] x=%d *px=%d
     **ppx=%d\n", x, *px,
       **ppx);
printf ("[3 alternatives to get address of x] x=%p *px=%p
     **ppx=%p\n", &x,
       px, *ppx);
// [3 alternatives to get value of x] x=10 *px=10 **ppx=10
// [3 alternatives to get address of x] x=000000000022FDE4
     *px=000000000022FDE4 **ppx=000000000022FDE4
```