

Fundamentele programării

Curs 9

Şef lucr.dr.ing. GENGE Béla

Universitatea “Petru Maior”, Departamentul de Informatică
Tîrgu Mureş, România
bela.genge@ing.upm.ro

- Pointerii permit utilizarea variabilelor dinamice
- Variabilele dinamice sunt create și eliminate la cererea programatorului
- Dimensiunea memoriei alocate este decisă în timpul execuției și nu la compilare
- Zona de memorie: *heap*

Funcții uzuale C standard

- `void* malloc(size_t dim)`: alocă o zonă de memorie de dim. `dim` octeți. Întoarce un pointer la zona alocată sau `NULL`.
- `void free(void* bloc)`: eliberează zona de memorie alocată cu una din funcțiile: `malloc`, `calloc`, sau `realloc`
- `void* calloc(size_t nr, size_t dim)`: alocă memorie pentru `nr` elemente, fiecare de dimensiunea `dim`. Fiecare element este inițializat cu 0.
- `void* realloc(void* bloc, size_t dim)`: realocă o zonă de memorie heap. Zona anterioară este eliberată.

Exemple

Exemplu1

```
char* str = NULL;
str = (char*)malloc(100*sizeof(char));
if ( NULL == str ) {
    exit(1);
}
free(str);
```

Exemplu2

```
int* v = NULL;
v = (int*)malloc(100*sizeof(int));
if ( NULL == v ) {
    exit(1);
}
free(v);
```

- Să se aloce și dealoce memorie pentru 1000 de întregi fără semn. Să se inițializeze toate elementele cu valoarea 0.
- Să se citească N numere întregi de la tastatură într-un vector alocat dinamic.
- Să se implementeze citirea unui șir de caractere de la tastatură alocând exact numărul de octeți necesari.

- Să se aloce și dealoce memorie pentru 1000 de întregi fără semn. Să se inițializeze toate elementele cu valoarea 0.
- Să se citească N numere întregi de la tastatură într-un vector alocat dinamic.
- Să se implementeze citirea unui șir de caractere de la tastatură alocând exact numărul de octeți necesari.

- Să se aloce și dealoce memorie pentru 1000 de întregi fără semn. Să se inițializeze toate elementele cu valoarea 0.
- Să se citească N numere întregi de la tastatură într-un vector alocat dinamic.
- Să se implementeze citirea unui șir de caractere de la tastatură alocând exact numărul de octeți necesari.

Funcții C++ pentru alocarea dinamică

- `new`: alocă o zonă de memorie.
- `delete []` și `delete`: șterge o zonă de memorie.

Exemple

Exemplu1

```
char* str = NULL;
str = new char[100];
if ( NULL == str ) {
    exit(1);
}
delete[] str;
```

Exemplu2

```
int* v = NULL;
v = new int[100];
if ( NULL == v ) {
    exit(1);
}
delete[] v;
```

Pointeri la pointeri

- Pointeri care adresează alți pointeri

Exemplu

```
int n = 10;
int* p1 = NULL;
int** p2 = &p1;
*p2 = &n;
**p2 = 0;
cout << &p2;
cout << p2;
cout << *p2;
cout << **p2;
cout << n;
```

- Să se aloce și dealoce dinamic memorie pentru o matrice de $N \times M$ întregi. Matricea este folosită pentru citiri de la tastatură.
- Să se citească N șiruri de caractere de la tastatură. Memoria va fi alocată dinamic printr-un pointer către un tablou de pointeri. Să se elibereze memoria ocupată în final.
- Să se elimine (dealocare memorie) pentru ultimul șir de caractere/ultima linie dintr-o matrice $n \times m$.