56. Обекти в паметта – особености при разполагането и чести програмни грешки

В С обектите в паметта се разполагат с командите:

 calloc()

 malloc()

 realloc()

Паметта се освобождава с функцията free()

В С++ за разполагане на обекти в паметта може да се използва и оператора new, а за освобождаване на паметта се използва оператора delete. В С++ може да се използват и командите от С.

malloc(size\_t size);

 \*Локализира битовете и връща указател към локализираната памет.

 \* Паметта не е изчистена.

Free(void \* p)

 \*освобождава паметта към която сочи p

 \*ако командата free(p) вече е била извикана може да се появи неопределено поведение.

 \* Ако р е NULL никаква операция не се извършва.

realloc (void \*p, size\_t size);

 \*Променя големината на блока с памет към който сочи р към размерни битове.

 \* Ново локализираната памет е неинициализирана.

 \* Ако р е NULL, командата е еквивалента на malloc(size)

 \*Ако size е 0, командата е еквивалентна на free(p).

calloc(size\_t nmemb, size\_t size);

 \*Локализира памет за масив от nmemb елементи с големина size и връща пойнтер към локализираната памет.

 \*Паметта се нулира.

Чести грешки: Повечето С програмисти използват malloc() за локализиране на блокове с памет и предполагат, че паметта е нулирана. Инициализирането на големи блокове с памет влиае на производителноста и не е винаги необходимо. По – добрия вариант за локализиране на памет е с memset() или извикването на calloc(), което нулира паметта.