Използване, живот и разположение на променливи.

Засенчване

**1.Използване на променливи**

Когато създаваме дадена променлива (int num;) единственото, което правим е да кажем на операционната система колко бита да маркира като заети. Това един блок памет да се маркира като зает значи никои друг, освен този, който го е маркирал, да не може да пише в него. В случая казваме на операционната система да задели 32 битове, които ние ще четем като цяло число.

**2. Живот и разположение на променливите**

За да можем да обсъждаме живота на една променлива първо трябва да дефинираме понятието обхват(scope). Обхватът е част от кода, която е отделена от останалия код, в смисъл, че нищо извън дадения обхват не може да променя нещата в него. В С обхвата се изразява в къдрави скоби{}. Както се досещате всички цикли, функции и структурни дефиниции имат собствен обхват в които се изпълняват( void func(){ …. }).

За да можем да си говорим за разположението на една променлива трябва първо да изясним къде може да бъде разположена тя. Една програма работи главно с 4 вида памет: code, data, stack, heap. В генералния случай тези 4-ри типа работят по следния начин:

* Code – в този сегмент се съдържат поредицата от инструкции, които процесора ще изпълнява.
* Data – в този сегмент се съдържа информация която трябва да е достъпна по всяко време на изпълнение на програмата.
* Stack – в този сегмент се запазват всички локални променливи в даден обхват, които не са достъпни по всяко време.
* Heap – той се използва за динамична памет, която ще бъде разгледана по-натам.

С тези неща изказани могат да се обособят три основни типа променливи:

* Локални;
* Глобални;
* Статични;

**2.1 Локални променливи**

Локалните променливи се създават по време на изпълнение на програмата. Те съществуват само в дадения обхват. При излизане/затваряне на обхвата паметта маркиран за тях се освобождава, в смисъл, че вече може да бъде използвана от други програми. Те се намират в stack частта от паметта на програмата понеже тяхното съществуване е временно.

**2.2 Глобални променливи**

Една глобална променлива съществува от самото пускане на програмата до нейния край. Една глобална променлива може да бъде достъпвана по всяко време независимо дали в момента сме отделени от обхват или не. Заради тези си характеристики глобалните променливи се намират в data сегмента на паметта. Те трябва да са активни през цялата програма

2.3 Статични променливи

Подобно на глобалните променливи, статичните променливи също съществуват по време на цялата програма. По отношение на живот и местоположение глобалните и статичните променливи са напълно идентични, тоест те също се намират в data сегмента и съществуват докато програмата съществува. Основната разлика при двата типа променливи ,е че статичните могат да биват достъпвани само във обхвата в който са дефинирани. Тоест една статична променлива ако е дефинирана във функция тя може да бива прочитана и променяна само в тялото на тази функция, но паметта за нея остава заделена дори след края на изпълнението на функцията. Това ни позволява да запазваме и пренасяме информация между отделните извиквания на една и съща функция, без това да пречи на останалата част от програмата.

**3. Засенчване на променливи**

В случаите, в които в един обхват е дефинирана дадена променлива(например int c;), а в друг по горен или по-долен обхват е дефинирана променлива със същото име, всички операции в сегашния обхват се обръщат към променливата дефинирана в този обхват(ако няма дефинирана променлива с такова има се обръщат към по-външния обхват, ако и там няма продължават така навън докато не се стигне глобалния обхват). Именно в това ползване на едно и също име за променлива във по близък обхват се нарича засенчваме на променливата.

То е резултат от това как работят функциите из основа, но силно се окуражава неговото **отбягване,** за да се улесни четенето и разбирането на кода.