Assembly - Memory Management

The **sys_brk()** system call is provided by the kernel, to allocate memory without the need of moving it later. This call allocates memory right behind the application image in the memory. This system function allows you to set the highest available address in the data section.

This system call takes one parameter, which is the highest memory address needed to be set. This value is stored in the EBX register.

In case of any error, sys_brk() returns -1 or returns the negative error code itself. The following example demonstrates dynamic memory allocation.

Example

The following program allocates 16kb of memory using the sys_brk() system call −

Live Demo

```
section .text
   global _start         ;must be declared for using gcc

_start:                  ;tell linker entry point

   mov  eax, 45          ;sys_brk
   xor  ebx, ebx
   int  80h

   add  eax, 16384       ;number of bytes to be reserved
   mov  ebx, eax
   mov  eax, 45          ;sys_brk
   int  80h

   cmp  eax, 0
   jl   exit    ;exit, if error
   mov  edi, eax         ;EDI = highest available address
   sub  edi, 4           ;pointing to the last DWORD
   mov  ecx, 4096        ;number of DWORDs allocated
   xor  eax, eax         ;clear eax
   std                   ;backward
   rep  stosd            ;repete for entire allocated area
   cld                   ;put DF flag to normal state

   mov  eax, 4
   mov  ebx, 1
   mov  ecx, msg
   mov  edx, len
   int  80h              ;print a message

exit:
   mov  eax, 1
   xor  ebx, ebx
   int  80h

section .data
msg     db      "Allocated 16 kb of memory!", 10
len     equ     $ - msg
```

When the above code is compiled and executed, it produces the following result −

```
Allocated 16 kb of memory!
```