ld -e \_main -Ttext 0x1000 -o kernel.o main.o video.o ports.o

بعد از اجرای این اسکریپت یک فایل تولید می شود که هنگام اجرا تابع \_mainشروع به اجرا شدن می شود که می توان آن را تغییر داد که بعد از پارامتر -e نام تابع را می آوریم. پارامتر –Ttext موجب می شود که فایل در آدرس 1000 بارگذار شود.

در رابطه با GDTR و LDTR این نکته را باید متذکر شد وقتی که ستراکچر GDT را تشکیل و آن را با دستور LGDT بار گذاری کردیم برای ایجاد LDT اولاً باید جای خالی 8 بایتی را در استراکچر LGDT در نظر گرفت.حال این سوال پیش می آید که این همان GDT می باشد و هیچ تفاوتی نمی کند !!! دقیقاً صحیح است هیچتفاوتی نمی کند و فقط در selector لوکال یا گلوبال بودنش را تعیین می کنیم.. یعنی می توان هر Descriptor را به عنوان Global یا Local فراخوانی کنیم.

نکته دیگری که باید به آن اشاره کرد این است که برای تعریف چندین DT نمی توان LLDT استفاده کرد یعنی ابتدا LGDT آن را فراخوانی می کنیم سپس LLDT به یک جدول(Table) در GDTاشاره می کنیم

نکته:در حالت PM یک نوع آدرس دهی داریم که به آن آدرس دهی خطی می گویند مانند کد زیر:

mov bp,ds

shl ebp,4

mov eax,ebp

lea eax,[ebp + \_idt]

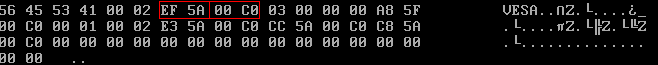
mov [\_idt\_ptr + 2],eax

دستور زیر آدرس موثر را بدست می آورد که می توان با دو دستور mov وadd آن را انجام داد:

lea eax,[esp + 76]

آدرس های Far در حافظه پشت سرهم ذخیره می شوند به عنوان مثال 0xc000:0x5AEF که می توان همین را به این فرم

0xc0000+0x5AEF نوشت که در حافظه به صورت زیر ذخیره می شود



درتابع های MK\_FP(seg,off) وFP\_SEG(fp) وFP\_OFF(fp) وFP\_TO\_LINEAR(seg, off) کاربرد دارد

:IDTوظیفهIDT(interrupt Description Table) اینکه 255مدخل در حافظه ایجاد می کند که هر 8بایت مشخصات یک روال می باشد که این روالها بر 2 گونه هستند یکی برای IRQ و دیگری برای EXCEPTION می باشد. در برنامه برای آدرس دادن از تابع idt\_set\_gate استفاده می کنیم لذا برای EXCEPTION از

idt\_set\_gate(0, (unsigned)isr0, 0x08, 0x8E);

تا

idt\_set\_gate(31, (unsigned)isr31, 0x08, 0x8E);

و برای IRQاز

idt\_set\_gate(32, (unsigned)irq0, 0x08, 0x8E);

تا

idt\_set\_gate(47, (unsigned)irq15, 0x08, 0x8E);

برای دادن آدرس روالی مناسب می باشد.