作业01 分段分页与任务空间的硬件支持

本次作业完成后先不交，打印出来（手写也行）下次课带上。课上首先检查作业完成情况，然后在课上分组对比讨论。

1. 简要解释一下Intel在386中的逻辑地址、线程地址、物理地址，以及他们与分段分页的关系。

逻辑地址：由程序产生的与段相关的偏移地址部分。

线程地址：逻辑地址通过分段机制转换成的地址。若没开启分页机制，则线性地址就是物理地址。

物理地址：用于内存芯片级的单元寻址，与处理器和CPU连接的地址总线相对应。

逻辑地址通过分段机制转换位线性地址，线性地址通过分页机制转换位物理地址。

1. GDT，LDT，GDTR，LDTR，CR0，CR3的含义和作用。

GDT：全局描述符表。系统中所有任务共享的段由GDT来映射。

LDT：局部描述符表。LDT所映射的段可以在任务切换时被改变。

GDTR：全局描述符表寄存器。用于存放全局描述符表GDT的32位线性基地址和16位表长度值。

LDTR：局部描述符表寄存器。用于存放局部描述符表LDT的32位线性基地址、16位段限长和描述符属性值。

CR0：控制寄存器。含有控制处理器操作模式和状态的系统控制标志。

CR3：控制寄存器。含有页目录表物理内存基地址。

1. CPL、DPL、RPL的含义和作用。386的页表项里表示本页是内核页还是用户页的比特位中叫什么名字？作用是什么？

CPL：当前正在执行程序或任务的特权级。通常CPL等于当前代码段的特权级。当程序把控制转移到另一个具有不同特权级的代码段中时，处理器会改变CPL。

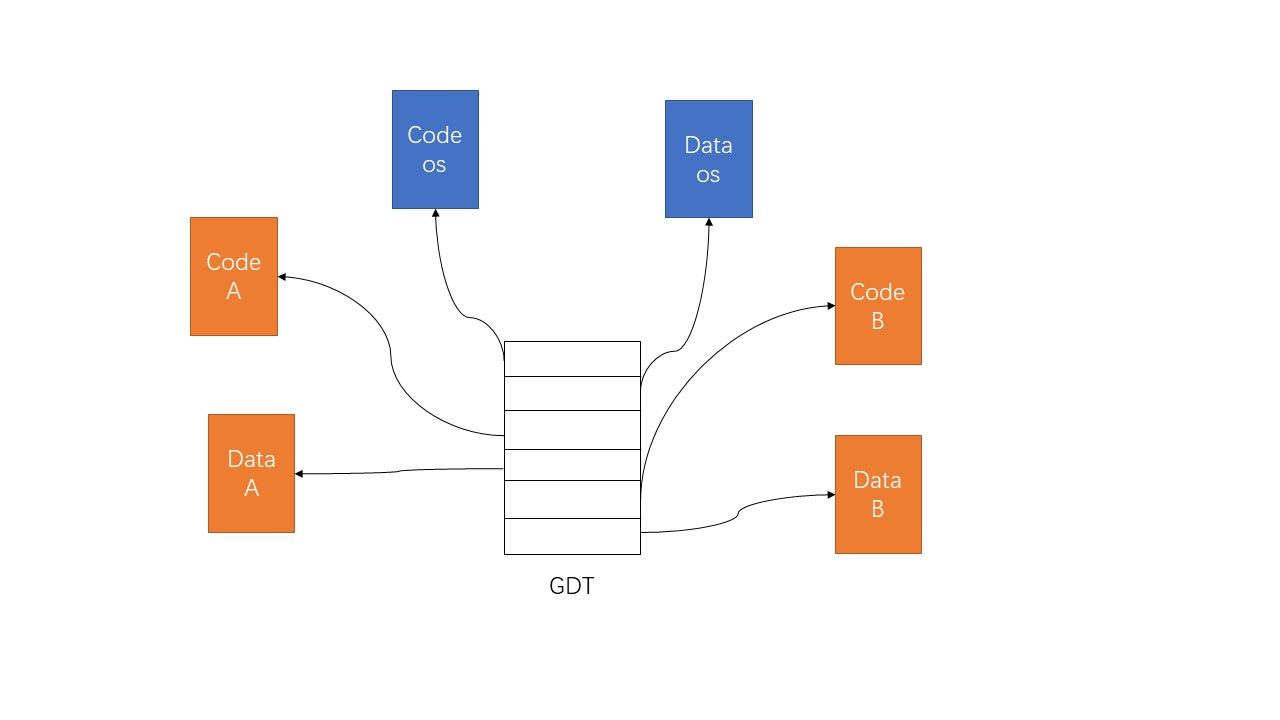
DPL：描述符特权级。DPL是一个段或门的特权级。它存放在段或门描述符的DPL字段中。在当前执行代码段试图访问一个段或门时，段或门的DPL会用来与CPL以及段或门选择符中的RPL作比较。

RPL：请求特权级。RPL是一种赋予段选择符的超越特权级，它存放在选择符的位0和位1中。处理器会同时检查RPL和CPL，以确定是否允许访问一个段。

叫U/S位。如果为0，那么页面只能被运行在超级用户特权级上的程序访问。页目录项中的U/S位对其所映射的所有页面起作用。

1. 综合以上，386的分段的作用就是把很多（逻辑地址的）段，映射（就是设计DT中的描述符）到线性空间。这些段可以是内核段，也可以是用户段。在理解选段符寄存器、段描述符格式的基础上，尝试自己设计一种能够把内核（代码段和数据段）和一个用户任务（代码段和数据段）映射到一个4GB的线性空间的方案。只用GDT能做到吗？可尝试一下。怎么隔离内核和用户程序呢？

可以做到



原来的段选择符中有表指示标志TI，用来指出包含指定段描述符的段描述符表是GDT还是LDT。现在可以将这一位定义为是内核段还是用户程序段。原来的方案中TI=0表示在GDT中，现在TI=0表示为内核段。原来方案中TI=1表示在LDT中，现在TI=1表示为用户程序段。

1. 在4题方案的基础上，把多个任务（代码段和数据段）和内核映射到一个线性空间的方案。只用GDT能做到吗？可尝试一下。怎么隔离内核和用户程序、用户程序和用户程序呢？

与上题一样的方案。

隔离内核和用户程序也是上体方案。

在原来的方案中，当进行任务切换时，处理器会把新任务LDT的段选择符和段描述符自动加载进LDTR中。

在现在方案中，在任务切换时，将新任务的段选择符和段描述符自动加载进LDTR也能直接实现用户程序和用户程序直接的隔离，即使用户程序的段描述符也在GDT中。

1. 在4题的基础上，如果有多个任务（代码段和数据段），能不能分别（一个一个来）与内核映射一个独立的线性空间？因为是一个一个来，能不能共享一个段表（DT）呢？怎么隔离内核和用户程序、用户程序和用户程序呢？

可以，通过使用分页机制，并且每个程序都使用自己的页目录就可以实现。

因为每个程序都有自己的页目录，所以可以共享一个段表。

段选择符中有表指示标志TI来隔离内核和用户程序。

通过CR3寄存器和LDTR来实现隔离用户程序和用户程序。

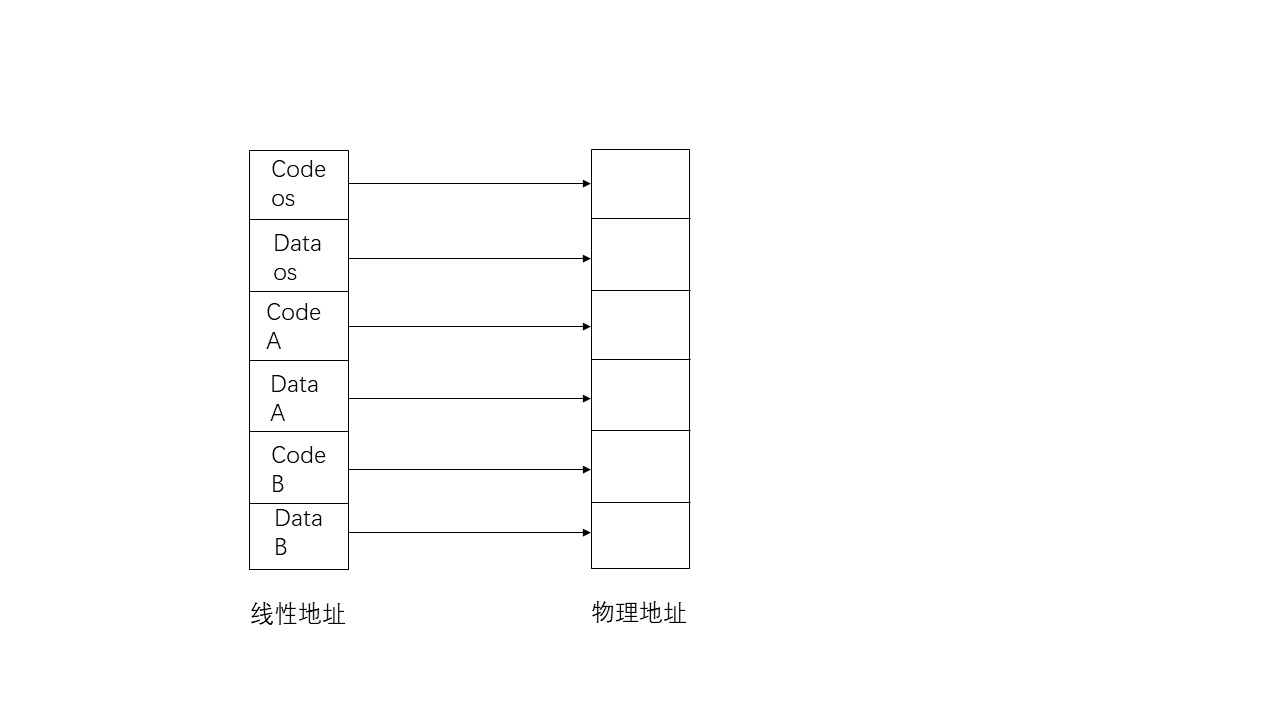
1. 以上4－6，线性空间怎么映射（页表）到物理空间呢？

4-5可以不启用分页机制，线性空间地址直接用于物理空间。

6启用了分页机制，需要通过分页机制去映射转换。

1. 想像一下，基于以上方案，2个任务的段和一个内核的段，加上分页后，在物理内存，可能是什么样子？

4-6



7

