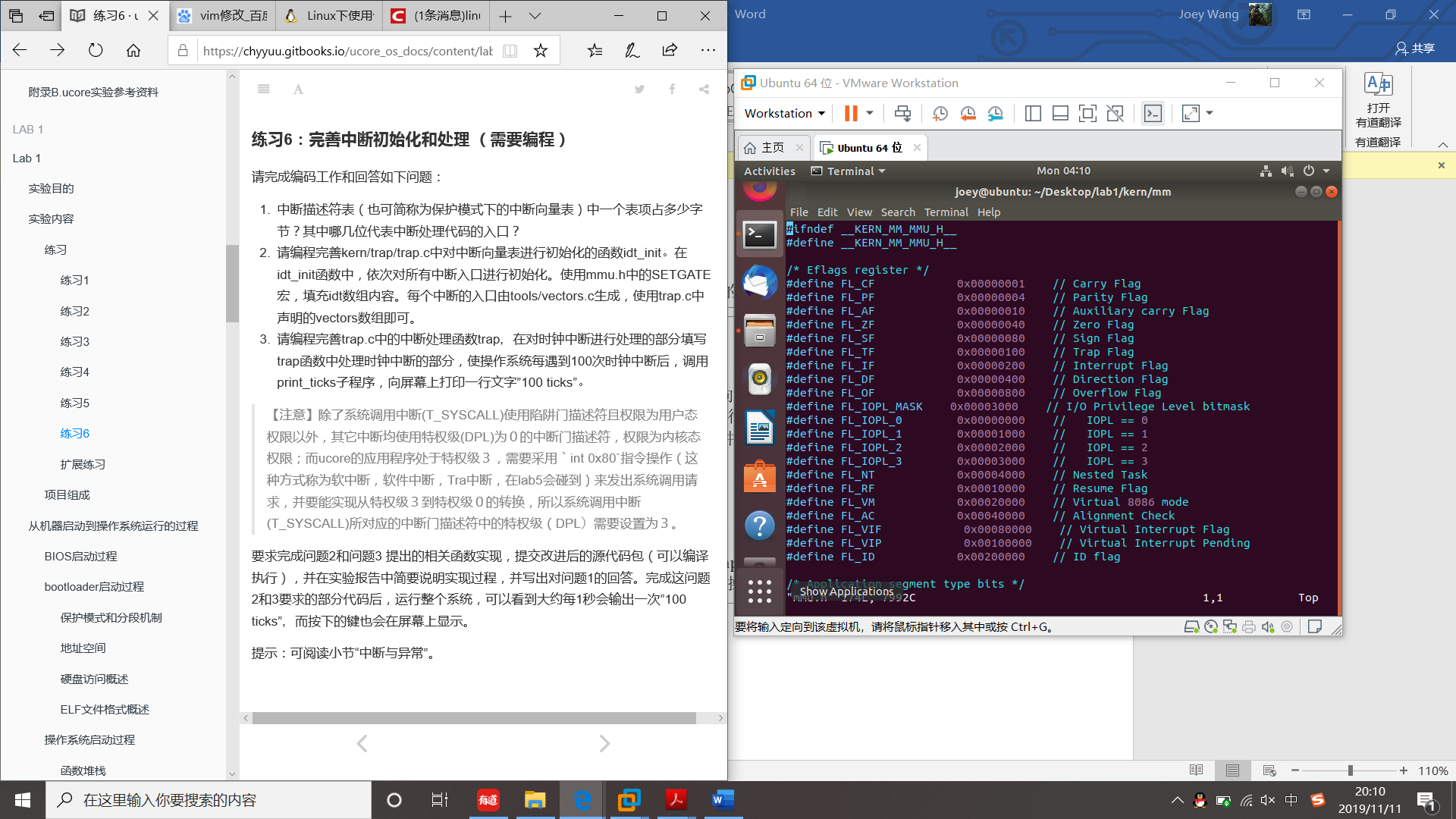
**练习6：完善中断初始化和处理 （需要编程）**

1. 中断描述符表（也可简称为保护模式下的中断向量表）中一个表项占多少字节？其中哪几位代表中断处理代码的入口？

打开mmu.h文件



得：

*struct gatedesc {*

*unsigned gd\_off\_15\_0 : 16; // low 16 bits of offset in segment*

*unsigned gd\_ss : 16; // segment selector*

*unsigned gd\_args : 5; // # args, 0 for interrupt/trap gates*

*unsigned gd\_rsv1 : 3; // reserved(should be zero I guess)*

*unsigned gd\_type : 4; // type(STS\_{TG,IG32,TG32})*

*unsigned gd\_s : 1; // must be 0 (system)*

*unsigned gd\_dpl : 2; // descriptor(meaning new) privilege level*

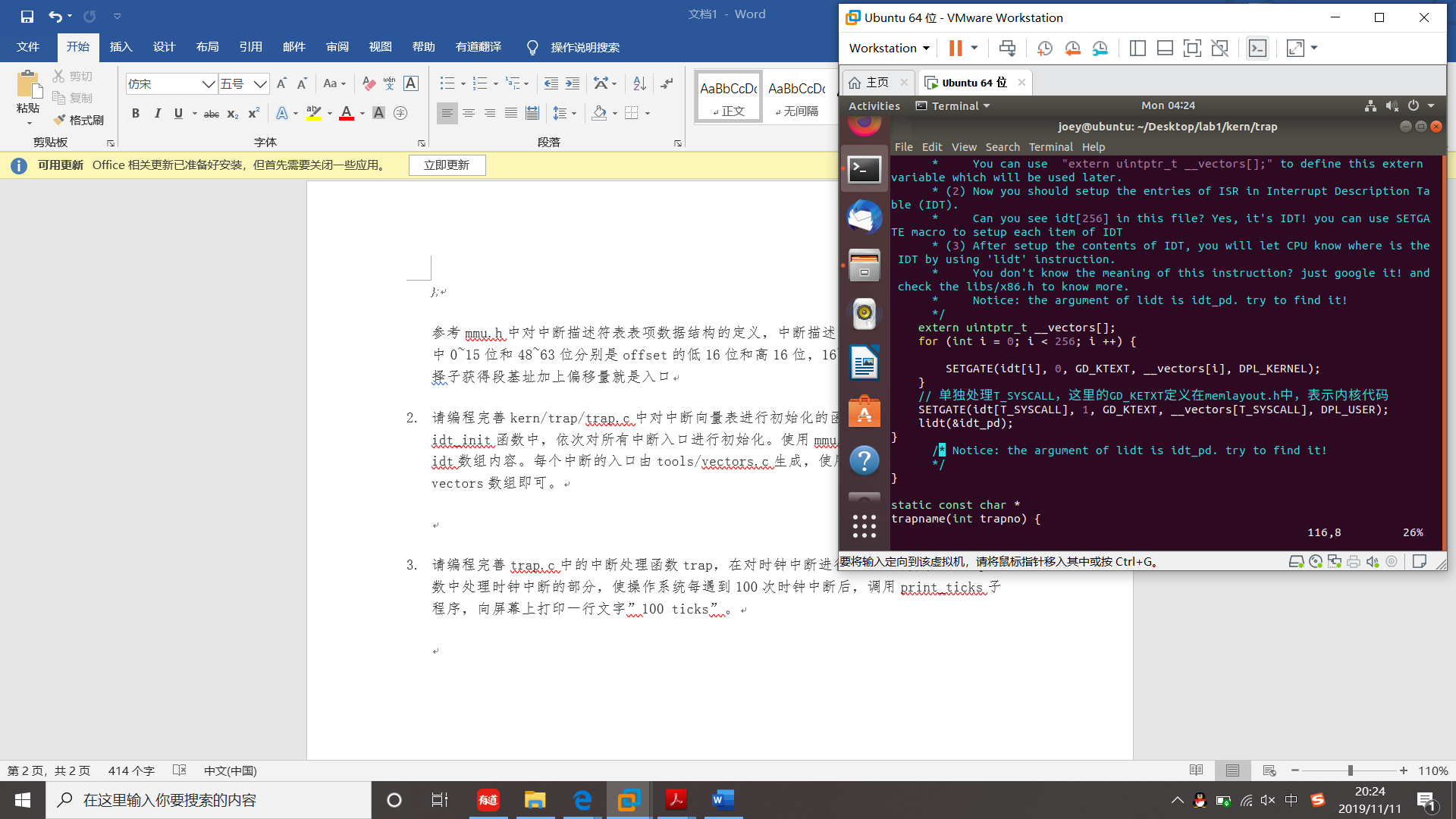
*unsigned gd\_p : 1; // Present*

*unsigned gd\_off\_31\_16 : 16; // high bits of offset in segment*

*};*

参考mmu.h中对中断描述符表表项数据结构的定义，中断描述符一个表项占8字节，其中0~15位和48~63位分别是offset的低16位和高16位，16~31位是段选择子，段选择子获得段基址加上偏移量就是入口

1. 请编程完善kern/trap/trap.c中对中断向量表进行初始化的函数idt\_init。在idt\_init函数中，依次对所有中断入口进行初始化。使用mmu.h中的SETGATE宏，填充idt数组内容。每个中断的入口由tools/vectors.c生成，使用trap.c中声明的vectors数组即可。



1. 请编程完善trap.c中的中断处理函数trap，在对时钟中断进行处理的部分填写trap函数中处理时钟中断的部分，使操作系统每遇到100次时钟中断后，调用print\_ticks子程序，向屏幕上打印一行文字”100 ticks”。

代码如下：

........

........

case IRQ\_OFFSET + IRQ\_TIMER:

ticks ++; //每一次时钟信号会使变量ticks加1

if (ticks==TICK\_NUM) {//TICK\_NUM已经被预定义成了100，每到100便调用print\_ticks()函数打印

ticks-=TICK\_NUM;

print\_ticks();

}

break;

.........

.........

但是编译运行失败。。。。

